



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

INSTITUTO TECNOLÓGICO DE DURANGO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



**“Cuantificación de horas-frío mediante modelos
de aprendizaje automático”**

TESIS

Que como parte de los requisitos para obtener el grado de

Maestría en Ingeniería

Presenta:

Ing. Ernesto Barboza Castañeda

Director de tesis:

M.C. José Antonio Martínez Rivera

Codirector:

Dr. Rubén Guerrero Rivera

Durango, Dgo. México, noviembre, 2024.



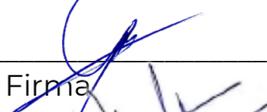


**Cuantificación de horas-frío mediante modelos
de aprendizaje automático**

Presenta:

Ing. Ernesto Barboza Castañeda

COMITÉ TUTORIAL

<p><u>MC José Antonio Martínez Rivera</u> Director</p>	<p>_____  Firma</p>
<p><u>Dr. Rubén Guerrero Rivera</u> Codirector</p>	<p>_____  Firma</p>
<p><u>MI. Eduardo Gamero Inda</u> Asesor</p>	<p>_____  Firma</p>
<p><u>M.C. Norma Alicia García Vidaña</u> Asesor</p>	<p>_____  Firma</p>

M.C. Norma Alicia García Vidaña

**Coordinadora del programa de la
maestría en ingeniería.**

Dr. Francisco Javier Godínez García

**Jefe de la División de Estudios de
Posgrado e Investigación**

Durango, Dgo. México

noviembre de 2024





TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Durango
División de Estudios de Posgrado e Investigación

Victoria de Durango, Dgo., a **25 / Noviembre / 2024.**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 570 / 2024.

ASUNTO: Autorización de Impresión de Tesis de Maestría.

C. ERNESTO BARBOZA CASTAÑEDA
No. DE CONTROL G14040355
PRESENTE.

De acuerdo al reglamento en vigor y tomando en cuenta el dictamen emitido por el jurado que le fue asignado para la revisión de su trabajo de tesis para obtener el **Grado de Maestro en Ingeniería**, esta División de Estudios de Posgrado e Investigación le autoriza la impresión del mismo, cuyo título es:

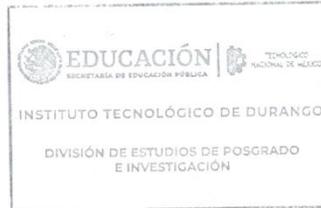
“Cuantificación de horas-frío mediante modelos de aprendizaje automático”

Sin otro particular de momento, quedo de Usted.

ATENTAMENTE.

Excelencia en Educación Tecnológica®
“La Técnica al Servicio de la Patria”

C. FRANCISCO JAVIER GODÍNEZ GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



FJGG'ammc.



Bvda. Felipe Pescador No. 1830 Ote., Durango, Dgo., C.P. 34080 Tels. 618-818-69-36
e-mail: depposgrado@itdurango.edu.mx tecna.mx | itdur



2024

Felipe Carrillo
PUERTO



TECNOLÓGICO
NACIONAL DE MÉXICO®



Instituto Tecnológico de Durango
División de Estudios de Posgrado e Investigación

Victoria de Durango, Dgo., a **25 / Noviembre / 2024.**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 569 / 2024.

ASUNTO: Autorización de Tema de Tesis de Maestría.

C. ERNESTO BARBOZA CASTAÑEDA
No. DE CONTROL G14040355
PRESENTE.

Con base en el Reglamento en vigor y teniendo en cuenta el dictamen emitido por el Jurado que le fue asignado, se le autoriza a desarrollar el tema de tesis para obtener el **Grado de Maestro en Ingeniería** cuyo título es:

“Cuantificación de horas-frío mediante modelos de aprendizaje automático”

CONTENIDO:

	RESUMEN
CAPÍTULO I	INTRODUCCIÓN
CAPÍTULO II	MARCO TEÓRICO
CAPÍTULO III	MATERIALES Y MÉTODOS
CAPÍTULO IV	RESULTADOS Y DISCUSIÓN
	REFERENCIAS
	ANEXOS

ATENTAMENTE.

Excelencia en Educación Tecnológica®
“La Técnica al Servicio de la Patria”


C. FRANCISCO JAVIER GODÍNEZ GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



FJGG'ammc.



Blvd. Felipe Pescador No. 1830 Ote., Durango, Dgo., C.P. 34080 Tels. 618-818-69-36
e-mail: deposgrado@itdurango.edu.mx tecnn.mx | itdur



2024

Felipe Carrillo
PUERTO

Agradecimientos

Agradezco principalmente a Dios por acompañarme en cada momento de mi desarrollo personal y académico, por la paciencia y la guía brindadas durante la evolución del proyecto, así mismo le agradezco por la oportunidad de observar desde un punto de vista científico solo un poco de la gran obra realizada por el más grande genio que pueda existir, literalmente sin él nada de esto sería posible.

Agradezco al CONAHCYT por su importante labor en el desarrollo científico y tecnológico brindado el apoyo económico para obtener soluciones a problemas de distinta índole que tienen un impacto en la calidad de la vida de los mexicanos, fortaleciendo el ámbito social, ambiental y económico.

Agradecimiento especial para mi madre la señora Isabel Castañeda Reyes quien con su amor incondicional y su lucha incansable nos permitió salir adelante durante los momentos de adversidad, cumpliendo cada día con su trabajo en tiempo y forma para llevar el pan a la mesa y brindarnos acceso a la educación.

A mi padre el señor Ernesto Barboza Sánchez quien me mostro que la corrección y la disciplina van acompañadas del cariño y quien fue parte fundamental para la educación desde el núcleo familiar mostrándome la importancia de la humildad y el servicio a los demás.

Mis más sinceras gracias a los maestros del programa por sus conocimientos y apoyo otorgados durante este periodo, especialmente a la doctora Norma quien fue parte fundamental para la creación de la función que permite la conversión de los datos cifrados de forma hexadecimal a flotante.

Un especial agradecimiento a los miembros de mi comité, con su ayuda y conocimientos fue posible realizar este proyecto.

Un agradecimiento especial al Ingeniero Antonio por su paciencia, compromiso y pasión por la enseñanza.

Resumen

La cantidad de horas-frío influye de manera directa en el desarrollo de los árboles caducifolios. En zonas donde ya se encuentran huertas establecidas el conocimiento de esta información permite al agricultor prepararse para la recolección y para las actividades previas a la misma, por otra parte, la cuantificación de horas frío brinda la posibilidad de tomar una decisión respecto a la viabilidad para la producción de un tipo de cultivo en una zona determinada. Existen modelos matemáticos para la estimación de horas-frío que presentan una alta variabilidad en el coeficiente de determinación.

En esta tesis se muestra el modelo de aplicación de redes neuronales para series de tiempo mediante regresión lineal multivariable para la estimación de la cantidad de horas-frío acumuladas en un huerto de manzano ubicado en el municipio de Canatlán, Durango. Se realizó el diseño e implementación de un sistema embebido que registra datos de temperatura y humedad relativa del aire cada 15 min, el sistema consiste en un nodo sensor que se comunica mediante protocolo LoRa a una puerta de enlace con conexión GSM para el envío de datos al servidor de Azure IoT Central[®] mediante peticiones tipo POST del protocolo HTTPS.

El entrenamiento supervisado en la plataforma Edge Impulse[®] para la predicción de horas-frío se realizó con los datos adquiridos del sistema de medición considerando los valores máximos y mínimos tanto de la temperatura como de la humedad relativa del aire, el modelo que considera las horas-frío como temperaturas inferiores a los 7.2°C obtuvo el mejor desempeño con 93.75% de precisión con un error cuadrático medio de 0.23 para una estimación diaria, con un error de 2.35 horas-frío por encima de las 116 horas-frío registradas por el sistema de medición.

Palabras clave. aprendizaje automático, cultivos caducifolios, horas-frío, protocolo LoRa, redes neuronales, series de tiempo, sistemas IoT, variables climatológicas

Abstract

The chill-hours quantity influences the development of deciduous trees in a direct way. In areas where there are already established orchards, knowledge of this information allows the farmer to prepare for the harvest and for the activities prior to it, on the other hand, the chill-hours quantification provides the possibility of planning regarding the viability to produce a type of crop in a determined area. Mathematical models for chill-hours estimation have a high variability in the coefficient of determination.

This thesis shows the application of an artificial neuronal network model for time series prediction through linear regression multivariable for chill-hours quantification in an apple orchard located in Canatlan, Durango. The design and the implementation of an embedded system that records data of temperature and relative humidity of the air every 15 minutes was realized during this project, this system consists of a sensor node that communicates through LoRa protocol with the gateway with GSM connection that allows the system to send the data to Azure IoT Central® server through POST requests of HTTPS protocol.

The supervised training into the Edge Impulse® platform for chill-hours prediction was made with a dataset obtained from the measurement system considering the maximum and the minimum values of temperature and relative humidity of the air, the model that considers chill hours as temperatures below 7.2°C obtained the best performance with 93.75% accuracy with a mean square error of 0.23 for a daily estimate, with an error of 2.35 chill hours above the 116 chill hours recorded by the measurement system.

Keywords. artificial neural network, chill hours, climatological variables, deciduous trees, embedded systems, IoT systems, LoRa protocol, machine learning, time series

Tabla De Contenido

Agradecimientos.....	VI
Resumen.....	VIII
Abstract.....	IX
Índice De Tablas.....	1
Capítulo 1 Introducción	2
1.1 Planteamiento Del Problema.....	7
1.2 Objetivos.....	7
1.2.1 <i>Objetivo General</i>	7
1.2.2 <i>Objetivos Específicos</i>	7
1.3 Hipótesis	8
1.4 Justificación	8
1.5 Delimitación Del Problema	10
1.6 Estado Del Arte.....	11
1.6.1 <i>Antecedentes</i>	11
Capítulo 2 Marco Teórico	15
2.1 Inteligencia Artificial	15
2.1.1 <i>Técnicas De La Inteligencia Artificial</i>	16
2.4 Series De Tiempo.....	22
2.4.1 <i>Métodos De Estimación Para Series Temporales</i>	23
2.5 Regresión Lineal	23
2.6 Estación Meteorológica.....	24
2.7 Diseño e Impresión 3D	24
2.8 Microcontroladores.....	25
2.8.1 <i>Microcontrolador ESP32</i>	25
2.8.2 <i>Función Dormido Profundo</i>	26
2.8.3 <i>Modulo SIM800L</i>	26
2.8.4 <i>Comandos AT</i>	26
2.9 Protocolos De Comunicación	26
2.9.1 <i>Protocolo LoRa</i>	27
2.9.2 <i>Protocolo HTTP</i>	27
2.9.3 <i>Protocolo HTTPS</i>	27
2.10 Plataformas Web Para Adquisición y Procesamiento De Datos.....	27

2.10.1 Azure IoT Central®	27
2.10.2 Edge Impulse®	28
2.11 Cultivos Caducifolios	28
2.12 Concepto De Horas-Frío	28
2.13 Cuantificación De Horas-Frío	29
2.14 Manzano	30
Capítulo 3 Materiales y Métodos	31
3.1 Construcción Del Sistema De Medición	31
3.1.1 Nodo Sensor LoRa	32
3.1.2 Puerta De Enlace Mediante ESP32 Con Módulo SIM800L Integrado	42
3.1.3 Desarrollo De La Aplicación Y Creación De Dispositivos En Azure Iot Central®	46
3.2 Adquisición y Depuración De Registros Recuperados	65
3.3 Cuantificación De Horas-Frío	68
3.4 Preprocesamiento y Exportación De Datos A La Plataforma Edge Impulse®	69
3.5 Creación Del Modelo De Red Neuronal Artificial Mediante Edge Impulse®	77
Capítulo 4 Resultados y Discusión	84
Conclusiones	91
Referencias	92
Apéndice A	94
Apéndice B	98

Índice de Figuras

Figura 1 Analogía red neuronal biológica-artificial.....	19
Figura 2 Función de activación escalón	21
Figura 3 Función de activación Sigmoide	21
Figura 4 Esquema general del sistema de medición.....	32
Figura 5 Diagrama de flujo del funcionamiento nodo sensor LoRa.....	33
Figura 6 Elementos del nodo LoRa protegidos dentro de caja IP65	34
Figura 7 Roseta de conexión RJ-11	35
Figura 8 Módulo LoRa e5.....	38
Figura 9 Arduino Nano Every	39
Figura 10 Diseño gratuito de pantalla de Stevenson para impresión 3D obtenido de Cults®.....	41
Figura 11 Configuración materia ASA	41
Figura 12 Pantalla de Stevenson	41
Figura 13 Microcontrolador ESP32 SIM800L T-Call V1.4.....	42
Figura 14 Diagrama de posicionamiento de los elementos del sistema de medición	44
Figura 15 Diagrama de operaciones generales realizadas por la puerta de enlace.....	45
Figura 16 Vista portal Azure®	46
Figura 17 Creación de aplicación IoT Central®	47
Figura 18 Selección de la suscripción para la creación de la aplicación	48
Figura 19 Aceptación de términos previa a la creación de la aplicación.....	48
Figura 20 Recurso horas-frío en el portal Azure	49
Figura 21 Acceso al recurso mediante la dirección URL de la aplicación	49
Figura 22 Vista general de la aplicación de Azure IoT Central®	50
Figura 23 Creación de un nuevo dispositivo	51
Figura 24 Dispositivo creado sin provisionamiento	51
Figura 25 Obtención de los parámetros para el aprovisionamiento y la conexión del dispositivo ..	52
Figura 26 Diagrama de seguimiento para la comunicación con la plataforma	52
Figura 27 Creación de token para la conexión segura al servicio DPS	53
Figura 28 Software POSTMAN para la obtención del id de operación mediante petición PUT.....	54
Figura 29 Cuerpo de la petición PUT para el registro del dispositivo	55
Figura 30 Encabezados petición GET para la obtención del Hub asignado.....	56
Figura 31 Cuerpo de la petición GET para la obtención del Hub asignado.....	56
Figura 32 Creación de token para la publicación de los datos mediante peticiones tipo POST	57
Figura 33 Test para el envío de datos a la plataforma mediante peticiones POST en formato JSON	58
Figura 34 Recepción exitosa de los datos en la plataforma Azure IoT Central sin estructurar	59
Figura 35 Creación de una plantilla de dispositivo	59
Figura 36 Selección del tipo de dispositivo	60
Figura 37 Asignación de nombre de plantilla	60
Figura 38 Selección de un modelo personalizado.....	61
Figura 39 Asignación de funcionalidades para la recepción de telemetría.....	61
Figura 40 Selección de vistas predeterminadas.....	62
Figura 41 Generación de los elementos visuales para la exhibición de las variables de telemetría	62

Figura 42	<i>Publicación de la plantilla creada</i>	63
Figura 43	<i>Asignación de plantilla al dispositivo</i>	63
Figura 44	<i>Vista de la plataforma para los datos sin procesar con la funcionalidad asignada</i>	64
Figura 45	<i>Vista gráfica de los datos tipo test recibidos</i>	64
Figura 46	<i>Adquisición de registros mediante explorador de datos</i>	65
Figura 47	<i>Imputación de registros faltantes</i>	66
Figura 48	<i>Valores máximos y mínimos diarios de la temperatura y humedad relativa del aire</i>	67
Figura 49	<i>Creación de un nuevo proyecto en Edge Impulse®</i>	71
Figura 50	<i>Construcción del dataset</i>	71
Figura 51	<i>Carga de información del dataset</i>	72
Figura 52	<i>Inicio del asistente para la importación de datos</i>	72
Figura 53	<i>Selección de archivo de información para la configuración</i>	73
Figura 54	<i>Selección del archivo almacenado en la PC</i>	73
Figura 55	<i>Vista preliminar del dataset cargado</i>	74
Figura 56	<i>Asignación de características del dataset</i>	74
Figura 57	<i>Asignación de la columna de marca de tiempo y frecuencia</i>	75
Figura 58	<i>Selección de variables predictoras y variable a predecir</i>	75
Figura 59	<i>Asignación del tamaño de muestra</i>	76
Figura 60	<i>Finalización de configuración del asistente para la importación del dataset</i>	76
Figura 61	<i>Creación del impulso</i>	79
Figura 62	<i>Selección de bloque de preprocesamiento de datos en forma lineal</i>	79
Figura 63	<i>Selección de bloque de regresión lineal</i>	80
Figura 64	<i>Selección de frecuencia de muestreo</i>	80
Figura 65	<i>Generación de características del dataset</i>	81
Figura 66	<i>Selección de los parámetros de entrenamiento</i>	81
Figura 67	<i>Visualización en modo experto para la selección de la función de activación</i>	82
Figura 68	<i>Selección de la estructura de la red neuronal artificial</i>	82
Figura 69	<i>Esquema de la estructura de la red neuronal artificial seleccionada</i>	83
Figura 70	<i>Sistema de adquisición y monitoreo de datos climatológicos en Azure IoT Central</i>	84
Figura 71	<i>Métricas de desempeño de la red neuronal de dos variables (validación)</i>	85
Figura 72	<i>Análisis visual del desempeño de la red de dos variables (validación)</i>	86
Figura 73	<i>Comparación de redes mediante las métricas de desempeño</i>	87

Índice De Tablas

Tabla 1	<i>Producción de manzana en México durante 2016-2024 y su estimación para 2030</i>	9
Tabla 2	<i>Distribución de conexión a la roseta RJ-11</i>	36
Tabla 3	<i>Configuración del regulador de carga para panel solar modelo W88-C</i>	36
Tabla 4	<i>Instrucciones de recarga a 20°C para batería sellada de ácido-plomo BR-1207</i>	37
Tabla 5	<i>Elementos para la energización del subsistema denominado nodo sensor LoRa</i>	37
Tabla 6	<i>Características del microcontrolador Arduino nano every ATmega4809</i>	38
Tabla 7	<i>Ficha técnica modulo DHT22</i>	40
Tabla 8	<i>Especificaciones técnicas del microcontrolador T-Call ESP32 con SIM800L integrado</i>	43
Tabla 9	<i>Estructura para los datos de entrenamiento</i>	69
Tabla 10	<i>Parámetros de diseño y configuración de red neuronal artificial</i>	77
Tabla 11	<i>Comparación del desempeño/configuración de redes neuronales de dos y cuatro variables</i>	88
Tabla 12	<i>Aproximaciones a la cantidad de horas-frío acumuladas bajo los diferentes criterios</i>	89
Tabla 13	<i>Comparación de los diferentes modelos con los modelos matemáticos de Da Mota, Sharp, y Crossa-Raynoud</i>	90

Capítulo 1 Introducción

En la actualidad, las técnicas usadas en la agricultura no han evolucionado lo suficiente y no difieren en gran medida de la agricultura precaria considerando su aparición durante la “*revolución neolítica*” hace unos 8000 años a.C., si bien, el uso de fertilizantes y pesticidas tras la implementación de la denominada “*revolución verde*” permitió un incremento en la producción de cultivos, (Ocampo et al, 2022), también tuvo un impacto negativo en el medio ambiente debido a que el uso de estos insumos genera la sobreexplotación de los suelos, aumenta la resistencia de plagas y la pérdida de la biodiversidad, (González, 2016).

Los países latinoamericanos esperan que el crecimiento en la producción de alimentos incremente a su vez las aportaciones económicas de las exportaciones agroalimentarias, permitiendo que esta actividad sustente su desarrollo económico y estabilidad bajo un enfoque capitalista que es incompatible con la sustentabilidad, (Segrelles, 2001).

La agricultura está íntimamente relacionada con la economía y el desarrollo del país, debido a que el crecimiento de la población implica un aumento en la demanda de producción alimentaria, en México, los alimentos producidos satisfacen parte de la demanda nacional y transnacional, los ingresos derivados de las exportaciones agroalimentarias tienen un impacto positivo en la economía del país, según los datos proporcionados por la Secretaría de Agricultura y Desarrollo Rural, para el año 2023 se obtuvo un ingreso de 51 mil 874 millones de dólares, (SADER, 2024).

México, ocupa el 12° lugar en la producción mundial de alimentos, (SADER, 2024), es el decimotercer productor de manzana a nivel mundial (USDA, 2017), Durango junto con los estados de Chihuahua y Puebla son los principales productores de manzana a nivel nacional según la información obtenida del Servicio de Información Agroalimentaria y Pesquera (SIAP, 2017).

De manera tradicional en México, la industria y la agricultura se ven como dos sectores separados, lo que genera una baja automatización e industrialización durante la etapa de producción a comparación con otros países del mundo que hacen uso de tecnologías que permiten la optimización de los procesos. Los agricultores no cuentan con la información suficiente ni con las herramientas tecnológicas debido a la falta de programas que permitan transmitir la información obtenida desde los centros de investigación y desarrollo al sector rural, por lo tanto, usan técnicas de producción transmitidas de generación en generación que no consideran las problemáticas a las que se enfrenta la agricultura actualmente, (Ramirez, 2016)

Según, Leiva (2003), el uso de técnicas generales que no consideran las necesidades propias del cultivo ni del sitio específico desemboca en la degradación ambiental y la ineficiencia en el uso de los recursos limitando la producción o impactando negativamente en la economía. La agricultura de precisión (AP) se presenta como un conjunto de técnicas que buscan satisfacer las necesidades de alta calidad y bajos precios exigidas por el mercado global, la AP busca minimizar los impactos negativos al medio ambiente mediante el uso de insumos en el lugar requerido, con la cantidad necesaria y en el tiempo oportuno.

La aplicación de la agricultura de precisión depende de las herramientas tecnológicas empleadas para obtener la información relevante que favorezca la toma de decisiones al productor, es decir, la información obtenida permite disponer de los insumos y aplicarlos de forma adecuada favoreciendo la optimización de los procesos de producción. Las tecnologías asociadas a la agricultura de precisión son los sistemas de posicionamiento global (GPS por sus siglas en inglés) y de información geográfica (GIS), sensores remotos, monitores de rendimiento/aplicación y maquinaria inteligente, (INCYTU, 2018).

Para la producción de los cultivos de hoja caduca de la cual forma parte el manzano (*malus domestica*), la cantidad de horas-frío acumuladas, es una variable de suma relevancia debido a su relación directa con la calidad del cultivo y los tiempos de producción. Debido principalmente a efectos del cambio climático la cantidad de horas-frío acumuladas anualmente ha sufrido una disminución respecto a las pasadas épocas y se espera que este comportamiento continúe durante los próximos años, (Maio, 2020)

Los cultivos de hoja caduca que se caracterizan por permitir la caída de sus hojas durante el periodo invernal como un mecanismo de protección en el cual la tasa de crecimiento y desarrollo sufre una disminución, es imprescindible que los cultivos permanezcan bajo condiciones que propicien el desarrollo fenológico, estas condiciones se ven afectadas principalmente por la temperatura y la humedad relativa del aire. Las bajas temperaturas propician el estado de dormancia o reposo, así mismo, la exposición a bajas temperaturas permite romper con el estado de dormancia para continuar con el ciclo de producción, (Flores, 2007)

La cantidad de tiempo de exposición medido en horas y la temperatura a la que se comienza a satisfacer las necesidades de frío se determina a partir del tipo de cultivo y su variedad, según Mendoza (2020), muchos investigadores coinciden que el valor promedio en el cual se comienza con la acumulación de tiempo bajo condiciones favorables es de 7.2 °C, bajo esta premisa, Agusti (2010), define las horas-frío como las horas transcurridas por debajo del umbral 7°C, sin embargo, se menciona que las temperaturas cercanas a los 0°C o por debajo no favorecen al desarrollo del cultivo.

El concepto de horas-frío se define como una hora ininterrumpida en la que el cultivo se encuentra expuesto a temperaturas comprendidas en el rango que va desde los 2.5°C a los 7.2 °C (Calderón, 1992). La conceptualización del término horas-frío padece de ambigüedad debido a que otros autores como Norvell & Moore (1982), consideran que las temperaturas por debajo de los 12°C contribuyen a la acumulación de frío, mientras que en otros lugares el rango de temperaturas óptimas para salir del estado de dormición está comprendido entre los 0°C y los 10°C, por otra parte, en base a los experimentos de (Nightingale & Blake, 1934) solo se consideran solo las temperaturas inferiores a 7°C.

Los métodos empleados para la cuantificación de horas-frío son el método directo mediante el uso de un termohigrógrafo que registra los valores de temperatura y humedad relativa del aire y el método indirecto que consiste en la aplicación de modelos matemáticos para una estimación mensual o diaria. El método directo es de gran precisión, sin embargo, su aplicación resulta tediosa y la obtención de uno de estos dispositivos no es sencilla, esto conlleva a que los agricultores recurran al uso de modelos matemáticos de fácil aplicación,

no obstante, los resultados obtenidos presentan variabilidad en sus estimaciones al compararlos con el método directo.

Para contrarrestar los efectos negativos del cambio climático en la acumulación de frío los agricultores recurren a la aplicación de compensadores de frío como nitratos de potasio y peróxido de hidrógeno, (Soto et al, 2020). Dadas las limitantes de los métodos para la estimación de horas-frío en la actualidad los agricultores usan indiscriminadamente estos insumos sin una perspectiva clara de las necesidades del cultivo en cuestión, esto implica un uso irracional que se traduce en costos elevados de producción o pérdida de producción por falta del conocimiento de las necesidades del cultivo.

La aplicación de los conceptos agricultura de precisión a los procesos de producción del manzano permitirían un mejor manejo de los insumos disponibles favoreciendo en el aspecto ambiental y económico. Este proyecto propone una metodología para cuantificación de horas-frío que favorezca la toma oportuna de decisiones del productor mediante el diseño y la implementación de una estación meteorológica que registre los valores de temperatura y humedad relativa del aire en periodos quinceminutales.

Además, se plantea un segundo método mediante el uso de técnicas de inteligencia artificial para la estimación de horas-frío que involucra el uso de los datos recolectados por el sistema de medición para un aprendizaje supervisado usando los valores máximos y mínimos diarios de la temperatura y la humedad relativa del aire como variables predictoras y la acumulación de horas-frío durante ese periodo como variable a predecir, el propósito de la investigación radica en obtener un método práctico y confiable.

1.1 Planteamiento Del Problema

La cantidad de horas-frío necesarias para los cultivos caducifolios varía respecto a cada especie frutícola, esta variable influye en la calidad y la producción de los cultivos, en el caso del manzano (*Malus domestica*) la cantidad mínima necesaria para cumplir con este requerimiento se encuentra en el rango de 200 a 800 horas frío y como máximo 1700 (Flores, 2007). Los métodos tradicionales para la estimación de horas-frío pueden ofrecer una precisión alta, pero son poco prácticos en su aplicación o bien una gran practicidad en su aplicación con una baja precisión en sus estimaciones. Esta tesis plantea la metodología para la obtención de un modelo derivado de la aplicación de máquinas de aprendizaje automático para la cuantificación de horas-frío, el cual proporcionara información relevante al agricultor para favorecer la toma de decisiones.

1.2 Objetivos

1.2.1 Objetivo General

Cuantificar y predecir la cantidad de horas-frío en función de la temperatura y la humedad relativa del aire mediante el uso de máquinas de aprendizaje automático.

1.2.2 Objetivos Específicos

- Implementar una estación meteorológica IoT que opere bajo los protocolos LoRa y HTTPS para la adquisición y publicación de las variables de proceso de temperatura y humedad relativa del aire en el servidor Azure IoT Central®.

- Diseñar el Dashboard de la plataforma Azure para el almacenamiento, procesamiento y exhibición de la información recopilada por los sensores de temperatura y humedad relativa del aire obtenidos remotamente mediante la puerta de enlace.
- Procesar la información almacenada en el servidor Azure IoT Central[®] para obtener los datos que permitan realizar el entrenamiento supervisado.
- Entrenar y validar el modelo de máquinas de aprendizaje para la predicción de la cantidad de horas-frío en función de la temperatura y la humedad relativa del aire mediante la plataforma Edge Impulse[®].

1.3 Hipótesis

H0: No es posible estimar la cantidad de horas-frío por medio del uso de máquinas de aprendizaje automático.

H1: Es posible estimar la cantidad de horas-frío por medio del uso de máquinas de aprendizaje automático.

1.4 Justificación

Según los datos proporcionados por la Secretaría de Agricultura y Desarrollo Rural (SADER, 2021), la manzana es considerada como el fruto de mayor consumo a nivel mundial, en México se produjeron 714 mil 203.20 toneladas durante el año 2020 aumentando su producción a una tasa media anual de 8.4 por ciento desde el 2012. En el estado de Durango se obtuvo una producción total de 15 mil 457 toneladas de manzanas para el año 2023, los municipios de Canatlán, Nuevo Ideal y Santiago Papasquiario destacaron como los principales productores de manzana.

Tabla 1

Producción de manzana en México durante 2016-2024 y su estimación para 2030

AÑO/PERIODO	Estimaciones			
	2016	2018	2024	2030
Producción potencial (miles de toneladas)	716.93	758.33	882.53	1006.73
Consumo nacional (miles de toneladas)	927.95	950.28	1020.52	1095.95

Nota: (SAGARPA: SUBSECRETARÍA DE AGRICULTURA, 2016)

En la tabla 1 se observa una tendencia creciente positiva de la demanda y la oferta de manzana, además, se aprecia que la producción es inferior al requerimiento del consumo nacional.

El manzano tiene un mayor requerimiento de frío que otros árboles caducifolios, cuando este requerimiento no se satisface con una cantidad suficiente de horas-frío, el árbol puede sufrir cambios fisiológicos, retraso en la producción, debilidad en brotes, una maduración irregular, así como una disminución en la producción y calidad de la fruta que se traducen de forma negativa a los productores y consumidores de manzana (Flores, 2007). Es indispensable para el agricultor contar con un sistema electrónico que le permita estimar la cantidad de horas-frío de forma efectiva para mejorar el control y la planeación de las actividades de producción.

1.5 Delimitación Del Problema

Esta tesis plantea el desarrollo de una metodología para la cuantificación y estimación de la cantidad de horas-frío acumuladas en una huerta de manzano dentro del municipio de Canatlán, Durango., el cual está ubicado entre los paralelos $24^{\circ}31'50.0''N$ $104^{\circ}47'27.4''W$, a una altura promedio de 2000 metros sobre el nivel medio del mar. El subsistema electrónico denominado como nodo sensor es energizado mediante la combinación de celdas solares, un regulador de voltaje y una batería de ácido-plomo los cuales no se encuentran exentos a fallas debido a eventos climáticos atípicos.

El subsistema electrónico denominado como puerta de enlace cuenta con la alimentación de la red eléctrica, lo que implica que la correcta comunicación con la nube depende del suministro proporcionado por el sistema eléctrico.

Los sensores utilizados son de grado comercial, es decir, están diseñados para trabajar bajo las siguientes condiciones climatológicas: rango de temperatura de $-40^{\circ}C$ a $+ 80^{\circ}C$, con una precisión de medición de $0.5^{\circ}C$ y la humedad relativa de 0-100% con precisión de 2%.

1.6 Estado Del Arte

1.6.1 Antecedentes

En (Brown y Luedeling, 2011), se realizó una comparación global de los métodos utilizados para la estimación del requerimiento de frío considerando los modelos para horas-frío, el modelo Utah y el modelo dinámico, el conjunto de datos de temperatura y precipitación utilizado en esta investigación fue obtenido del National Climatic Data Center (NCDC, 2008). Durante el periodo de 1973 al 2002, los resultados de la aplicación de los tres métodos mostraron variabilidad en la acumulación de frío alrededor de todo el mundo y de acuerdo con el modelo Utah y el modelo dinámico los climas marítimos cerca del océano norte en Europa y la costa del pacifico en Oregon presentaron una mayor acumulación de frío.

Existen factores físicos que afectan el desarrollo y el crecimiento de los cultivos caducifolios, de entre los cuales la temperatura destaca como una de las variables de mayor importancia, resaltando así la necesidad de un tiempo de exposición a temperaturas menores o iguales a 7.2°C denominadas comúnmente como horas-frío. Comparando los métodos Crossa-Raynaud, Da Mota, Gómez-Morales y Sharpe y contrastando las estimaciones de cada uno con los datos obtenidos por el observatorio Meteorológico del Colegio de Geografía, en la Ciudad Universitaria D.F., se concluyó que el método Gómez-Morales en su modalidad mensual y diaria lograron una mejor estimación alcanzando un porcentaje de confiabilidad del 77.6%, (Gomez y Morales, 2015).

En Konstantinos et al. (2018), se presenta una revisión de las aplicaciones de las máquinas de aprendizaje en la agricultura, donde se define el aprendizaje de máquina, su clasificación y algunos de los modelos, además se abordan los diferentes paradigmas de la inteligencia artificial y sus aplicaciones: control de enfermedades en plantas, detección de malezas, calidad y rendimiento de los cultivos, gestión de ganado y análisis de suelos, muestra información sobre el tipo de cultivo

de estudio, sus propiedades analizadas, el modelo utilizado y el porcentaje de precisión alcanzado. Concluye que los modelos más usados fueron los de redes neuronales y las máquinas de soporte vectorial.

La metodología para identificar y cuantificar la superficie geográfica que cumpla con los requerimientos de producción de cultivos caducifolios se muestra en Alvarez et al. (2019), donde se realizó el análisis del registro histórico de temperatura en un periodo de ocho años con la información de 38 estaciones meteorológicas ubicadas en el estado de Nayarit. Además, con ayuda del sistema de información geográfica Arcmap Versión 10.1 (ArcGIS ESRI, Redlands, California, Estados Unidos) se obtuvo una representación cartográfica de la distribución anual de las horas-frío en 20 municipios de Nayarit, determinando que el 72.6% del territorio estatal, acumula menos de 300 horas-frío y el resto acumula entre 300 y 700 horas-frío.

El aprendizaje de máquina y el aprendizaje profundo tienen diversas aplicaciones en el área de biotecnología, como en la agricultura, la bioingeniería, la genética poblacional y la biorremediación. (Edian y Ramos, 2019), menciona, como su nombre lo indica, que el objetivo es que las máquinas aprendan por medio de distintos métodos que involucren el uso de datos y su análisis estadístico, clasificando las técnicas de aprendizaje de máquina en tres grupos: supervisados, no supervisados y semi-supervisados; y las de aprendizaje profundo en dos grupos: supervisados y no supervisados. Se hace mención a las herramientas computacionales y los lenguajes de programación que se utilizan para el desarrollo de aplicaciones, como Python (Scikit-Learn), GNU R, Weka y Rapidminer. Las técnicas de aprendizaje de máquina y aprendizaje profundo son herramientas de suma importancia para el análisis en los estudios e investigaciones de diversas áreas de la biotecnología.

En Pertille et al. (2019), se desarrolló una librería basada en R para la estimación del requerimiento de frío mediante funciones que permiten la utilización de modelos para la cuantificación de horas-frío y unidades frío utilizando como parámetro de entrada las lecturas de temperatura diaria medidas en horas, el resultado de la aplicación de las funciones desarrolladas permitió obtener la cantidad de horas-frío estimadas mediante los distintos métodos.

En Fadón et al. (2020), se realizó una caracterización del requerimiento de frío y calor para los cultivos frutales de hueso mediante el uso de modelos basados en temperatura, estos modelos son: modelo de Weinenber, modelo Utah de Richardson, el modelo dinámico y el modelo Growing Degree Hours (GDH), el estudio determinó los requerimientos de temperatura para un total de 530 cultivos de hueso la mayoría de estos durazno y almendra.

Con el objetivo de encontrar el mejor método para la estimación de horas frío y días grado para la región de San Benito, se realizó un estudio donde se compararon diez modelos matemáticos con la información meteorológica recopilada de la Estación experimental de dicha ciudad, en un periodo de 58 años. Los métodos con mejor estimación de horas-frío fueron los de Damota y Crossa con un factor de corrección de 12. (Mendoza, 2020).

Siguiendo la metodología de Damario et al. (2008) se realizó la estimación de las horas-frío en San Pedro, Argentina con los valores observados por la estación experimental San Pedro INTA considerando la temperatura máxima y mínima mensual en el periodo de 1965 al 2018. Además, se realizaron dos estimaciones a futuro: de 2015 a 2039 y de 2075 a 2099. (Maio, 2020).

En Maulud y Abdulazeez (2020), se comparan diferentes algoritmos de regresión lineal y polinomial, considerando el mejor desempeño en términos de su predicción y precisión, se

concluyó que para la selección del modelo de regresión lineal se debe considerar las variables de estudio involucradas.

El almacenamiento en bases de datos de variables climatológicas como temperatura, humedad relativa y presión atmosférica permite la aplicación de modelos de aprendizaje para realizar predicciones en un tiempo determinado. Para la predicción de temperatura en un lapso de 24 horas en Egas y Roque (2020), se implementó una mini-estación meteorológica con sensores bajo el concepto de Internet de las Cosas (IoT), además, se integró una red neuronal recurrente entrenada con datos del Instituto Nacional de Meteorología e Hidrología. La información recopilada y la predicción del modelo de aprendizaje fueron expuestas en un sitio web para un posterior análisis estadístico. Se establece que por medio la aplicación de una red neuronal con datos de múltiples variables climáticas e independientes se disminuye el error de predicción en comparación con los modelos que solo consideran una variable.

El cambio climático es solo uno de muchos factores por los cuales surge la necesidad de predecir el comportamiento de la temperatura para la toma oportuna de decisiones, con el fin de satisfacer la necesidad antes mencionada, se emplean modelos de aprendizaje de máquina con una efectividad relativa. Para evaluar los modelos predictivos, Azari et al. (2022), compara el desempeño de seis modelos haciendo uso de información de la estación meteorológica del aeropuerto internacional de Memphis, Tennessee. Durante la fase de entrenamiento se consideraron los registros de 1980 a 2014 y en la fase de predicción los registros de 2015 a 2021, como resultado del análisis se determinó que el modelo de redes neuronales artificiales superó en ambas fases al resto de los modelos.

Capítulo 2 Marco Teórico

En este capítulo se presentan los conceptos sobre inteligencia artificial, aprendizaje automático, microcontroladores y protocolos de comunicación, además, se introducen los conceptos básicos de fruticultura relacionados con el requerimiento de frío en los cultivos caducifolios.

2.1 Inteligencia Artificial

La inteligencia es una característica inherente de los seres humanos y de algunas especies, pero no se limita solo a los seres vivos, debido a los avances tecnológicos las máquinas pueden simular un comportamiento inteligente similar o incluso mejor que el de los seres humanos, el concepto abstracto de inteligencia dentro de la mente asocia las habilidades de una persona o cosa para realizar una acción que se considera inteligente, por ejemplo, solucionar un problema de una forma eficiente en el menor tiempo posible, predecir con exactitud un fenómeno o suceso futuro, capacidad de memorizar y retener información, la capacidad de razonamiento lógico para la toma de decisiones, entre muchas otras.

Cada día es más frecuente escuchar el término inteligencia artificial, probablemente su frecuencia aumentará en tal medida que se convertirá en algo habitual para las futuras generaciones, todo esto gracias a los avances tecnológicos que permiten almacenar y procesar una gran cantidad de datos. La Inteligencia Artificial (IA) proporciona una solución práctica e innovadora a infinidad de problemas en diversas áreas con un gran desempeño. Aunque es posible mencionar que estamos viviendo el auge de la inteligencia artificial, un hito histórico trascendental comparable a la máquina de vapor, sus orígenes se remontan a mediados del siglo XIX.

Antes de definir una IA como tal es necesario considerar las habilidades intrínsecas relacionadas a la inteligencia misma, decir que alguien o en este caso algo tiene un comportamiento inteligente puede ser debido a que se tiene la capacidad de aprender de las experiencias, responder rápidamente a situaciones nuevas, aplicar y crear conocimiento, pensar o razonar. En términos generales la inteligencia puede definirse como la habilidad de que, en base a la experiencia, el conocimiento y el análisis de la situación permite encontrar una o múltiples soluciones para la resolución de problemas. Según la Artificial Intelligence for Europe (European Commission, 2018) La inteligencia artificial (IA) se refiere a sistemas que muestran un comportamiento inteligente: analizando su entorno pueden realizar diversas tareas con cierto grado de autonomía para alcanzar unos objetivos específicos.

2.1.1 Técnicas De La Inteligencia Artificial

La inteligencia artificial se basa en el comportamiento humano y a su vez este puede desenvolverse de manera efectiva en diversas áreas, la inteligencia artificial puede resolver un sinfín de problemas de diversos campos de investigación y desarrollo. Existen ramas de estudio y aplicación de la IA que permiten satisfacer necesidades específicas, estas subdivisiones son conocidas como técnicas de la inteligencia artificial y su propósito es la búsqueda e implementación de métodos para ofrecer soluciones particulares. A continuación, se mencionan algunas de las técnicas que componen la inteligencia artificial.

2.1.1.1 Sistemas Basados En Conocimiento.

También conocidos como sistemas expertos tiene como objetivo la resolución de problemas por medio del acceso al conocimiento humano almacenado en una computadora, este tipo de sistemas simulan el razonamiento de un experto en el tema.

2.1.1.2 Visión Computacional.

El uso de esta técnica de inteligencia artificial permite a la computadora la interpretación de imágenes mediante la extracción de propiedades como color, textura o profundidad. La visión artificial permite que la maquina obtenga información para comprender su entorno o características específicas del mismo. Una vez procesada la información se pueden realizar acciones acordes a las condiciones del objeto de estudio. Los sistemas de seguridad inteligentes, los vehículos autónomos, el diagnóstico de enfermedades y el reconocimiento facial son solo algunos ejemplos del uso de esta técnica.

2.1.1.3 Procesamiento De Voz Y Lenguaje Natural.

Consiste en métodos para comunicar a la maquina con las personas por medio del uso de lenguajes naturales como español o inglés, por mencionar algunos, la comunicación puede ser textual, oral o por medio de símbolos. La comunicación oral del humano a la maquina es procesada para ser convertida en texto y posteriormente interpretada por la inteligencia artificial. Alexa, el asistente de Google y Siri son algunos ejemplos.

2.1.1.4 Lógica Difusa.

El uso de esta técnica consiste en trabajar la información con un menor nivel de precisión haciendo uso de categorías que contienen un rango de valores, la relatividad en cuestión no se refiere a la lógica sino al objeto de estudio. La lógica difusa simula la percepción de las variables por un humano para la toma de decisiones. Por ejemplo, cuando una persona prepara el agua para tomar un baño la temperatura no es percibida exactamente en un valor si no en un rango de valores que pueden ser catalogados dentro de una clase como, fría, tibia, caliente o muy caliente, la decisión de tomar un baño depende de la categoría en la cual se encuentra la temperatura del agua.

2.1.1.5 Técnicas Heurísticas (Recocido Simulado).

Este tipo de técnicas permiten encontrar métodos para la solución de un problema en base al descubrimiento empírico de los algoritmos, es decir, a través de procedimientos experimentales de prueba y error. Con este método no se garantiza que la solución obtenida sea la mejor de las soluciones posibles.

2.1.1.6 Robótica.

Los límites de los robots programables para la realización de tareas específicas fueron superados por la implementación de la inteligencia artificial. Gracias al aprendizaje de máquina, los robots pueden ser entrenados y aprender de la experiencia, esto hace que los robots puedan actuar de una forma más independiente sin la necesidad de ser reprogramados constantemente.

2.1.1.7 Aprendizaje Automático (Machine Learning).

El *machine learning* o aprendizaje de máquina tiene como objetivo que las máquinas aprendan de la experiencia y las métricas usadas para medir el desempeño para realizar una tarea están basadas en métodos matemáticos y estadísticos, (Konstantinos, Busato, Moshou, Pearson, & Bochtis, 2018). En esta técnica es necesario que la máquina pase por una etapa de entrenamiento y validación, este entrenamiento puede ser supervisado, dando valores de entrada y la respuesta esperada, los algoritmos más utilizados con este tipo de entrenamiento son de clasificación y regresión lineal o logística., no supervisado, dando solo los datos de entrada para que los algoritmos busquen patrones de comportamiento que permitan obtener una salida con información relevante, algunos de estos algoritmos son *k-means clustering*, análisis de componentes principales (PCA) y reglas de asociación., y una combinación de ambos métodos de entrenamiento denominado aprendizaje

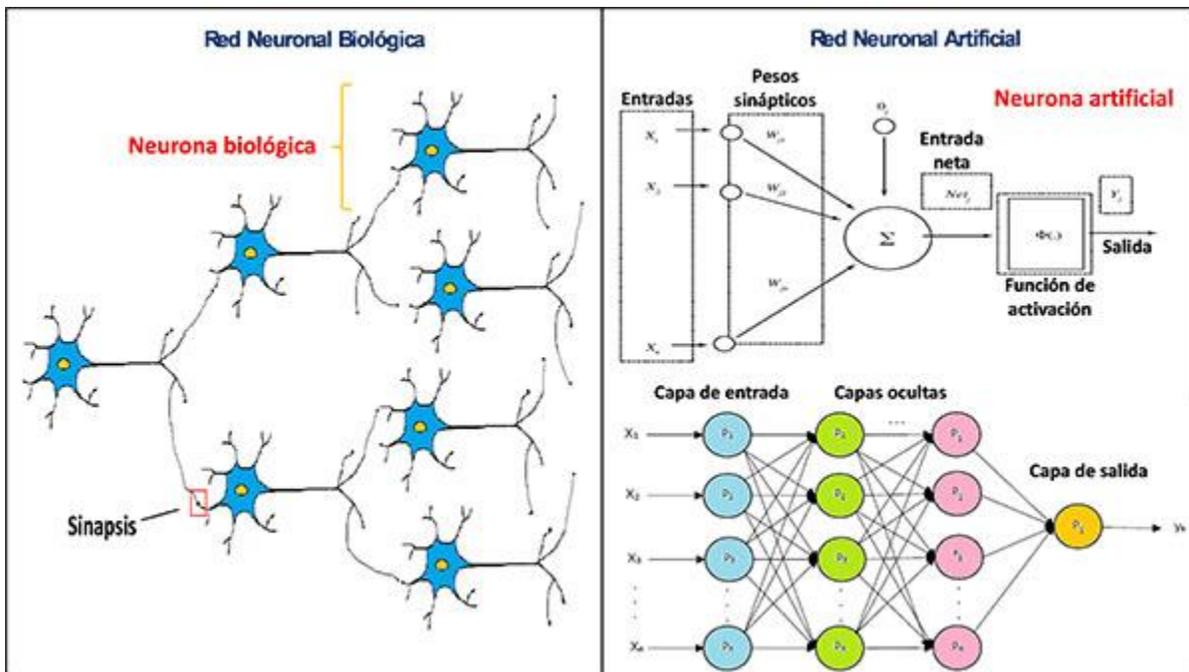
semi-supervisado, ejemplos de este tipo de algoritmos son *Deep Belief Networks (DBN)* y las denominadas *Restricted Boltzmann Machines (RBS)*.

2.1.1.8 Redes Neuronales.

Las redes neuronales artificiales imitan el comportamiento de las redes neuronales de los seres vivos, donde el sistema se compone de múltiples neuronas que trabajan de forma paralela realizando conexiones para obtener una respuesta en función de los estímulos eléctricos que reciben, figura 1. Las redes neuronales son capaces de aprender de la experiencia agilizando la velocidad para el procesamiento y la reacción de respuesta según su nivel de entrenamiento.

Figura 1

Analogía red neuronal biológica-artificial.



Nota. Recuperado de: <https://www.uaeh.edu.mx/divulgacion-ciencia/redes-neuronales/>

El entrenamiento para las redes neuronales artificiales se lleva a cabo mediante la modificación de los pesos sinápticos en las entradas y puede ser del tipo supervisado, proporcionando a la red los valores de entrada y los resultados esperados en la capa de salida o no supervisado, dando solo los valores de entrada y teniendo como objetivo que la red encuentre patrones de comportamiento de las variables para obtener información de la forma como se relacionan.

Los elementos de entrada pueden ser imágenes o datos organizados relacionadamente, las neuronas reciben estímulos de entrada con un peso w que afecta el resultado que se envía a la capa siguiente, el peso es una característica de suma importancia debido a que está directamente relacionado con el desempeño de la red neuronal, de hecho, uno de los principales objetivos durante el aprendizaje automático es ajustar los pesos de manera tal que la diferencia entre la predicción y el valor real sea el mínimo posible tanto para los datos del entrenamiento como para los valores nuevos.

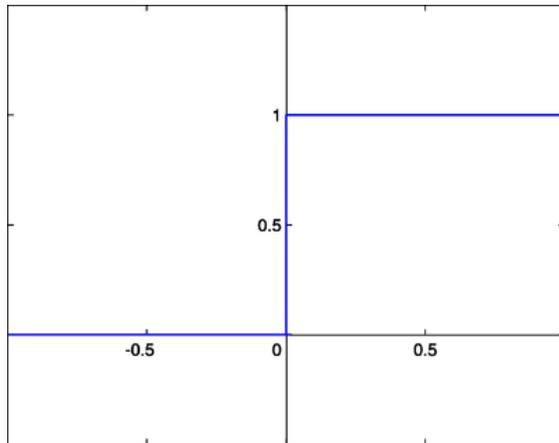
En una red neuronal artificial existen tres tipos de capas interconectadas, la capa de entrada donde se encuentran los valores iniciales del problema, la capa oculta que evalúa los valores que recibe de la capa anterior para enviar el resultado a la capa siguiente la cual puede ser a su vez otra capa oculta o la capa de salida donde se exhiben los resultados obtenidos por la red neuronal artificial.

Por lo general y exceptuando la capa de entrada la información que incide en las neuronas de las capas ocultas es multivariable, la función de entrada permite la interpretación de los datos de una forma global, comúnmente esto se logra por medio de una sumatoria, productoria o máximo. Es decir, sumando o multiplicando cada uno de los productos de las entradas por su peso correspondiente o en el caso del máximo tomando en cuenta el producto con mayor valor.

La función de activación usa un parámetro ajustable denominado umbral el cual debe ser superado por la función de entrada para que la neurona se active, las funciones más comunes son escalón, figura 2 y sigmoide, figura 3.

Figura 2

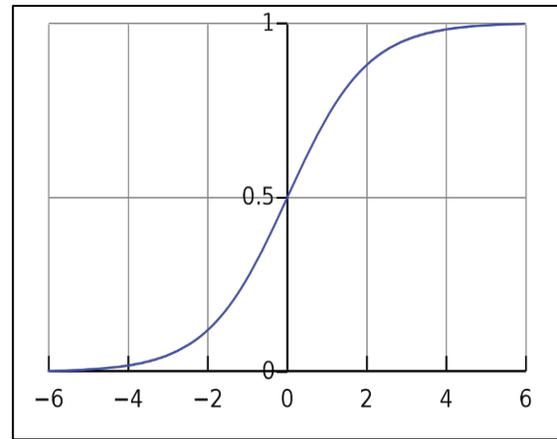
Función de activación escalón



$$1) y = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Figura 3

Función de activación Sigmoide



$$2) y = \frac{1}{1+e^{-x}}$$

la primera da como resultado un valor digital, es decir, el valor es cero cuando el resultado de la función de entrada es menor que el umbral, y es uno cuando es mayor o igual, en la función sigmoidea el resultado es un valor continuo entre 0 y 1 o -1 y 1 según sea la función sigmoidea seleccionada.

La estructura de una red neuronal artificial se puede clasificar según su número de capas, el tipo de conexión y el grado de conexión (Soria y Blanco, 2007, pág. 30, citado en Rivas et al., 2017, pág. 30).

Según su número de capas:

- Monocapa: las variables de entrada se conectan directamente a la capa de salida.
- Multicapa: la capa de entrada se conecta a la capa de salida mediante capas ocultas intermedias.

Según el tipo de conexión:

- Las redes neuronales no recurrentes no envían el resultado en la salida a capas que se encuentran en niveles anteriores ni en su mismo nivel.
- Las redes neuronales recurrentes envían la información a las capas anteriores y a neuronas en su misma capa para generar una retroalimentación.

Según el grado de conexión:

- La red se encuentra parcialmente conectada.
- La red se encuentra completamente conectada.
- Unidireccionales o de propagación hacia adelante.
- Redes de propagación hacia atrás.

Entre mayor sea el número de conexiones mayor es su complejidad y a su vez mayor es la similitud con el funcionamiento de un cerebro humano.

2.4 Series De Tiempo

Las series de tiempo son un conjunto de datos medidos a una frecuencia determinada con el fin de entender el comportamiento de una variable de interés a través del tiempo, esta información es utilizada en diversos campos debido a que al realizar un análisis sobre los datos es

posible encontrar patrones y características que permiten realizar predicciones futuras y así favorecer la toma de decisiones.

2.4.1 Métodos De Estimación Para Series Temporales

Los métodos estadísticos clásicos, que son usados para realizar el análisis de series de tiempo, son: regresión lineal y no lineal, suavización exponencial, promedios móviles y el modelo autorregresivo integrado de media móvil (ARIMA). En la literatura se encuentra un creciente interés por la aplicación de los modelos de redes neuronales en el análisis de las series de tiempo, esto debido principalmente a su buen desempeño y cualidades distintivas de métodos clásicos. [Cita]

2.5 Regresión Lineal

La regresión lineal es una herramienta que se usa frecuentemente en estadística y en modelos de máquinas de aprendizaje con el fin de encontrar patrones y relaciones de un conjunto de datos que permitan identificar tendencias y predecir el comportamiento de una variable en función de otra (s).

Se dice que un modelo de regresión es simple cuando se usa una sola variable independiente o predictor y múltiple al usar más de una variable. Existe a su vez un caso especial de la regresión lineal múltiple aplicable a funciones que presentan curvaturas y que se conoce como regresión polinomial de la forma

$$y = \beta_0 + \beta_1x + \beta_2x^2 + \dots + \beta_hx^h + \varepsilon$$

donde h es el grado mayor del polinomio.

El método de mínimos cuadrados fue desarrollado cerca del año 1800 por Gauss y Legendre, es ampliamente utilizado en casos donde existe un sistema de ecuaciones sobre definido, es decir, cuando en un sistema existen más ecuaciones que incógnitas.

Tras la experimentación o la medición de una variable en un tiempo determinado es común encontrar datos ordenados en pares (x,y) donde x e y son vectores con n elementos., cuando la dispersión de los datos presenta cierto grado de linealidad el uso de una función f(x) permite realizar estimaciones de las lecturas e incluso la extrapolación es semejante a una función lineal de la forma $y=mx+b$ donde b es la ordenada en el origen y m la pendiente, este método permite encontrar la recta que se ajusta en mayor medida al conjunto de valores que representan a la variable dependiente y, el método consiste en encontrar los parámetros de β_0 y β_1 que presenten el menor valor en la sumatoria de la diferencia del valor real y el estimado elevados al cuadrado como se muestra en la formula.

$$\sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2$$

2.6 Estación Meteorológica

Es un sistema que se encarga de recopilar información climatológica por medio de sensores que permiten obtener el valor de variables como temperatura, humedad, velocidad y dirección de viento, radiación solar entre otras, generalmente la estación está ubicada en un punto estratégico con el fin de analizar, exhibir y predecir variables de interés mediante el uso de la información recopilada.

2.7 Diseño e Impresión 3D

El diseño de un objeto tridimensional se lleva a cabo mediante el uso de un software especializado que permite realizar un conjunto de operaciones para la creación digital de dicho objeto, generalmente con el objetivo de llegar a la etapa de manufactura es decir, la creación física del objeto en cuestión mediante los procesos de fabricación, en el caso de la impresión 3D este

proceso comienza con la generación del código G y la especificación de los parámetros principalmente en función de la impresora y el filamento (material con el que se desea imprimir), algunos de los parámetros importantes son: temperatura de cama, temperatura y diámetro de extrusor, inclusión de soportes, habilitación del sistema de enfriamiento, tipo de capa de adhesión y material (PLA, PETG, ASA, ABS, etc.) los materiales tienen propiedades y características diferentes por lo tanto las condiciones para su correcta impresión varían., la selección del material se define según las condiciones ambientales y los esfuerzos a los que estará sujeta la pieza en cuestión.

2.8 Microcontroladores

Antes de definir un microcontrolador es necesario entender el concepto de un controlador el cual se define como aquel elemento o dispositivo diseñado para manejar uno o múltiples procesos. Debido principalmente a los avances tecnológicos los elementos que constituyen un controlador pueden ser compactados dentro de un dispositivo más pequeño con un chip integrado que tiene la capacidad de realizar operaciones lógicas y comunicarse con dispositivos periféricos por medio de pines de entrada y salida, este pequeño sistema o dispositivo es conocido como un microcontrolador.

2.8.1 Microcontrolador ESP32

Es un microcontrolador programable desarrollado por Espressif Systems considerado como el sucesor del ESP8266 que integra funciones que permiten la conectividad Bluetooth y Wifi sin la necesidad módulos externos, cuenta con soporte para establecer comunicación con otros dispositivos mediante protocolos como UART, SPI, MOSI, I2C y facilita su programación en lenguaje C mediante el software Arduino IDE.

2.8.2 Función Dormido Profundo

Es una función que permite el ahorro de energía debido principalmente a que desactiva el uso del CPU y las funciones de conectividad mediante Wifi y Bluetooth disminuyendo así el consumo de corriente del microcontrolador de forma radical y dejándolo en un estado de bajo consumo que puede ser desactivado mediante un interruptor externo, un temporizador o mediante el uso del coprocesador.

2.8.3 Modulo SIM800L

Es un módulo electrónico compatible con dispositivos como microcontroladores tipo Arduino, Esp32 o Raspberry cuya principal función es habilitar la conexión GSM de un dispositivo o un sistema para el uso de la red 2G obteniendo así acceso a internet y brindando la posibilidad de enviar y recibir mensajes y llamadas.

2.8.4 Comandos AT

Los comandos AT son instrucciones que permiten configurar y hacer uso de funciones de un dispositivo o modulo, la estructura general del uso de un comando AT es la siguiente AT + [Comando] = "parámetro" algunos comandos omiten la parte del parámetro debido a que su función es proporcionar información respecto a un comando, comúnmente el módulo envía una respuesta con la información de la ejecución del comando, si el comando se ejecutó correctamente responderá con "OK" en algunos casos provee información relevante respecto a su ejecución.

2.9 Protocolos De Comunicación

Un protocolo de comunicación es un conjunto de reglas o normas indispensables para la correcta comunicación entre los dispositivos electrónicos debido a que proporcionan la metodología para el proceso de transmisión y recepción de datos.

2.9.1 Protocolo LoRa

Es un protocolo de comunicación de baja frecuencia que permite el envío de paquetes de datos mediante ondas de radio a una distancia larga (hasta 10km en espacio abierto), con un bajo consumo de corriente y que se ha popularizado de la mano de las tecnologías IoT debido a que permite la comunicación inalámbrica entre microcontroladores, desde una comunicación punto a punto hasta redes avanzadas con múltiples nodos y puertas de enlace para la conexión de los dispositivos a internet.

2.9.2 Protocolo HTTP

Sus siglas hacen referencia a *Protocolo de Transferencia de Hipertexto* es un protocolo de comunicación basado en la relación cliente-servidor donde un dispositivo denominado cliente se comunica con el servidor para el envío o consulta de información mediante diferentes tipos de peticiones.

2.9.3 Protocolo HTTPS

Por sus siglas *protocolo de transferencia de hipertexto seguro (HTTPS)*, está basado y se puede considerar como la evolución del protocolo HTTP dado que se caracteriza por aumentar la seguridad en la transmisión de los datos verificando la veracidad de ambas partes (cliente/servidor) mediante el uso de credenciales y certificados.

2.10 Plataformas Web Para Adquisición y Procesamiento De Datos

2.10.1 Azure IoT Central®

Es una plataforma de Microsoft ® que permite el almacenamiento y la exhibición de datos obtenidos mediante dispositivos registrados que envían la información comúnmente mediante protocolo HTTPS o MQTT, la selección del protocolo se realiza según su aplicación debido a que

esta define el tipo de conexión que requiere el dispositivo, se recomienda el uso del protocolo MQTT para acciones de comando y control que requieren un tiempo de respuesta corto, este tipo de conectividad está asociado a un dispositivo cuyo tipo de conexión es persistente, por otra parte, las conexiones efímeras son ideales para aplicaciones donde el dispositivo envía datos de telemetría en periodos de tiempo, por ejemplo, un dispositivo que se conecta cada 10 min para enviar datos de telemetría, una vez enviada la información se desconecta y su conexión se realiza nuevamente en los próximos 10 min..

2.10.2 Edge Impulse®

Es una plataforma web que permite el diseño, la creación y la implementación de modelos de aprendizaje automático con un perfil agradable al usuario y fácil de entender para la creación de distintos tipos de modelos de redes neuronales con soporte para los problemas que van desde un modelo de regresión lineal hasta modelos de clasificación, la información utilizada para el entrenamiento pueden ser datos, audios, imágenes o videos.

2.11 Cultivos Caducifolios

Los cultivos caducifolios también conocidos como cultivos de hoja caduca se caracterizan por permitir la caída de sus hojas como un mecanismo de protección ante las condiciones climatológicas del frío invernal, a consecuencia de este fenómeno los cultivos entran en un estado de dormición, receso o latencia donde se reduce su actividad metabólica (Flores, 2007), la clave para entrar y salir de este proceso es la exposición a temperaturas bajas.

2.12 Concepto De Horas-Frío

Las horas-frío son uno de los factores climáticos más importantes en la adaptación y aclimatación de frutales, esta variable se encuentra estrechamente relacionada con la calidad y la

productividad del cultivo, el requerimiento de frío de una determinada especie se mide en unidades de tiempo en que ocurren las bajas temperaturas estimuladoras.

Para romper con el estado de latencia es necesario que el cultivo acumule una cantidad de tiempo de exhibición a bajas temperaturas, la cantidad de tiempo de exposición medido en horas y la temperatura a la que se comienza a satisfacer las necesidades de frío se determina a partir del tipo de cultivo y su variedad, según Mendoza (2020), el valor promedio en el cual se comienza con la acumulación de tiempo bajo condiciones favorables es con temperaturas inferiores a los 7.2 °C.

El concepto de horas-frío se define como una hora ininterrumpida en la que el cultivo se encuentra expuesto a temperaturas comprendidas en el rango que va desde los 2.5°C a los 7.2 °C (Calderón, 1992). Sin embargo, otros investigadores no consideran un límite inferior, lo que conlleva el concepto a la cantidad de tiempo donde el cultivo se encuentra expuesto a temperaturas inferiores a 7°C, en otras definiciones del concepto, el rango de temperaturas que propician el correcto desarrollo fenológico va de los 0°C a los 10°C, mientras que en otros trabajos las horas-frío se presentan como temperaturas inferiores a los 12°C, evidentemente esta ambigüedad en el concepto representa una relatividad en para su definición.

2.13 Cuantificación De Horas-Frío

Lo más conveniente para cuantificar las horas-frío es disponer de un termógrafo y sobre las gráficas de éste contar manualmente las horas de frío que se presentan cada día (Gómez et al. 2015). Aunque por medio de este método se obtiene una gran precisión sus principales desventajas son la dificultad con la que los agricultores pueden obtener un dispositivo y la falta de practicidad para realizar los cálculos.

En base a la problemática antes mencionada la cuantificación de horas-frío se realiza mediante el uso de modelos matemáticos cuya aplicación resulta muy sencilla, desafortunadamente el aumento en la facilidad de aplicación repercute en la precisión, la literatura muestra que los mejores modelos no alcanzan ni un 80% de confiabilidad al compararlos con los valores del método directo.

2.14 Manzano

El manzano (*Malus domestica*) forma parte de los cultivos caducifolios, también conocidos como cultivos de hoja caduca que se caracterizan por permitir la caída de sus hojas durante el periodo invernal como un mecanismo de protección en el cual la tasa de crecimiento y desarrollo sufre una disminución considerable, es imprescindible que los cultivos permanezcan bajo condiciones que propicien el desarrollo fenológico, estas condiciones se ven afectadas principalmente por la temperatura y la humedad relativa del aire. Las bajas temperaturas propician el estado de dormancia o reposo, así mismo, la exposición a bajas temperaturas permite romper con el estado de dormancia para continuar con el ciclo de producción.

Capítulo 3 Materiales y Métodos

El desarrollo de este proyecto consiste en el diseño de una red neuronal artificial que permita la aproximación de la cantidad de horas-frío mediante las variables de temperatura y humedad relativa del aire, se busca disminuir la brecha de desempeño que ofrecen los métodos tradicionales y proporcionar un método que permita el equilibrio entre practicidad y confiabilidad.

3.1 Construcción Del Sistema De Medición

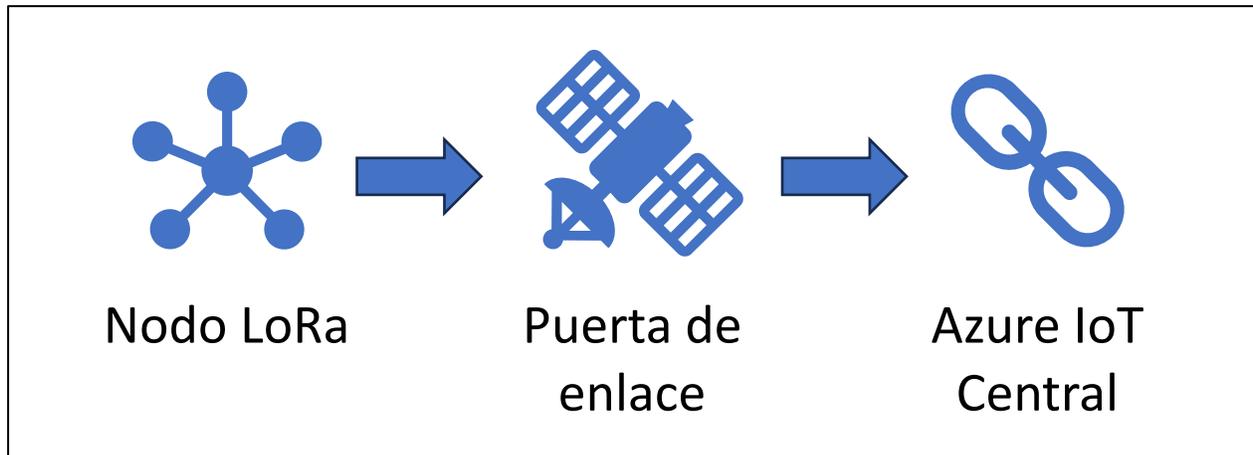
El concepto sobre el que se basa la agricultura de precisión es aplicar la cantidad correcta de insumos, en el momento adecuado y en el lugar exacto, (García y Flego, 2008). Para cumplir con los requisitos indispensables que definen la agricultura de precisión y que conllevan a la optimización de los procesos de producción, se emplean elementos de tecnología IoT para el monitoreo, la transmisión de datos y la ejecución de acciones de control. La falta de conectividad eléctrica e internet son las principales limitantes para la implementación y el correcto funcionamiento de los dispositivos IoT en el campo.

La construcción del sistema de medición se realizó bajo el esquema nodo sensor a puerta de enlace, esto debido a que el punto de ubicación geográfico donde se desean realizar las mediciones no cuenta con cobertura de red GSM, del mismo modo, la falta de acceso a la red eléctrica es una limitante que se consideró durante la etapa del diseño. El sistema fue diseñado para la adquisición de variables climatológicas de temperatura y humedad relativa del aire en periodos de 15 min.

Figura 4.

Figura 4

Esquema general del sistema de medición



El nodo sensor LoRa se encarga de registrar las variables de interés en el lugar donde se desea extraer la información para posteriormente realizar el envío mediante protocolo LoRa a la puerta de enlace, una vez que los datos son recibidos por este elemento ubicado en un punto estratégico con cobertura de red GSM comienzan las operaciones para la publicación de los datos recibidos mediante protocolo HTTPS en el servidor web Azure IoT Central[®], el cual permite el almacenamiento, procesamiento y el monitoreo de las variables climatológicas.

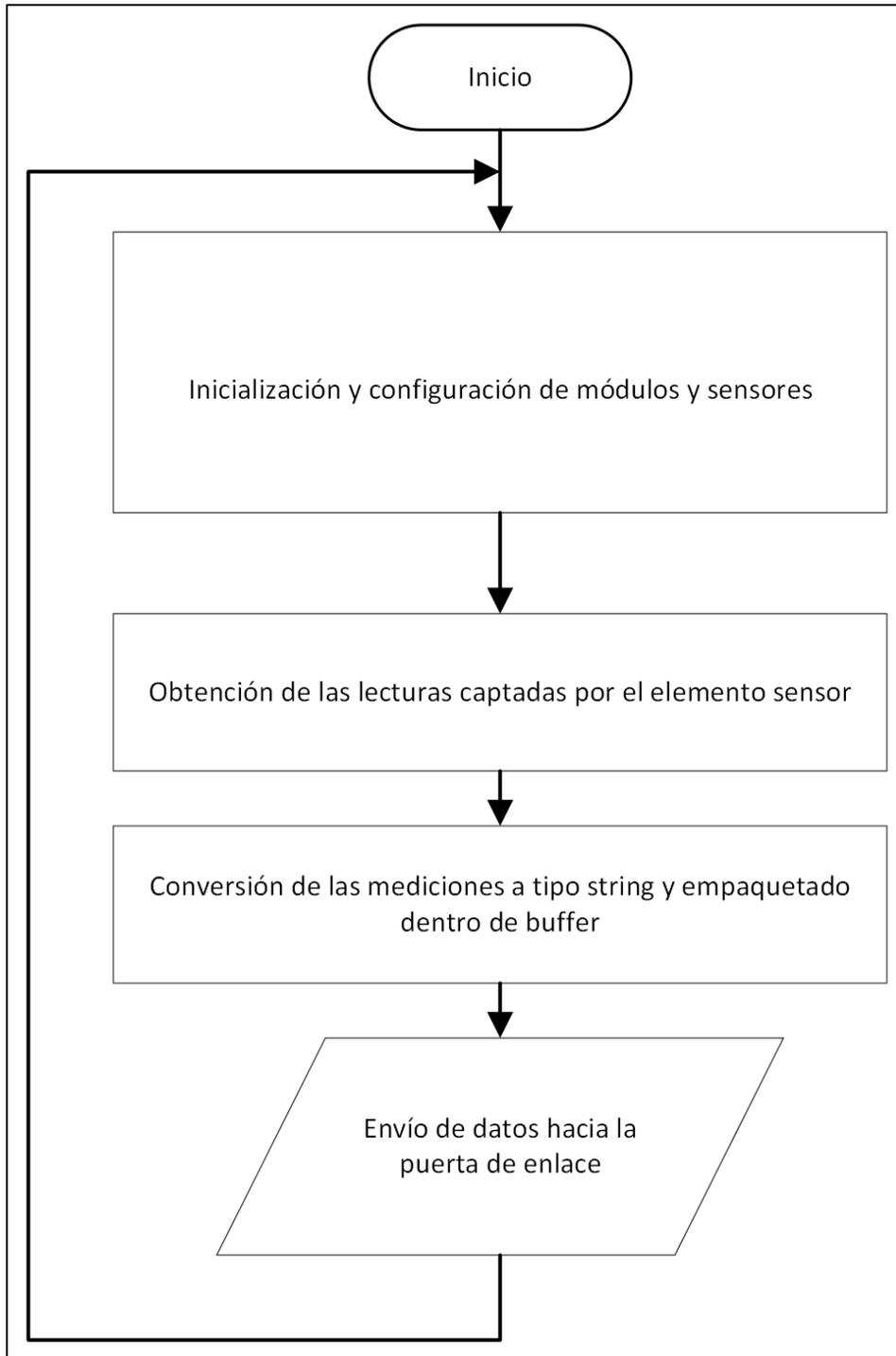
A continuación, se describen los elementos que integran las diferentes partes del sistema de medición y las funciones que realizan.

3.1.1 Nodo Sensor LoRa

El objetivo principal del nodo sensor LoRa es obtener y enviar las lecturas de temperatura y humedad relativa del aire hacia la puerta de enlace mediante protocolo LoRa, las operaciones realizadas por el subsistema se muestran en el diagrama de flujo de la figura 5.

Figura 5

Diagrama de flujo del funcionamiento nodo sensor LoRa

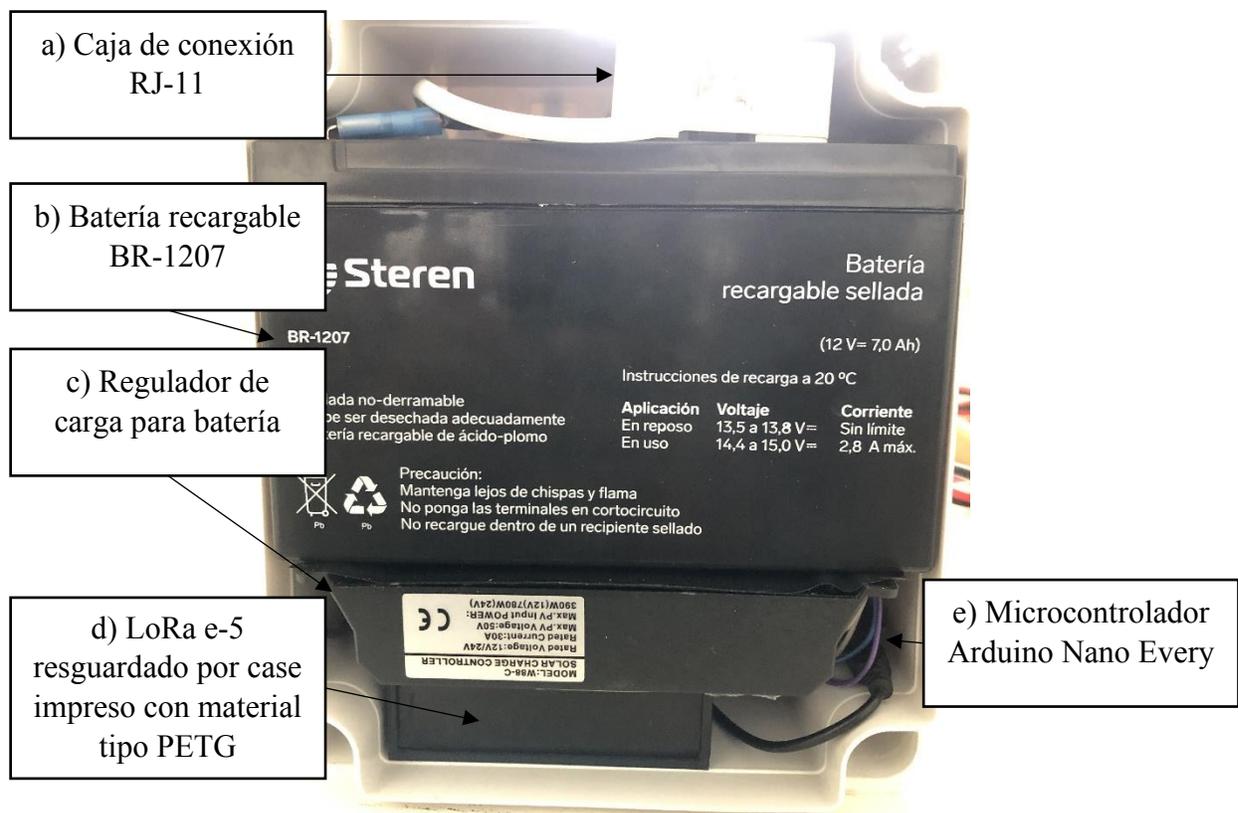


El código en lenguaje C del nodo sensor LoRa se muestra en el apéndice A.

Para el diseño del nodo sensor se consideraron las condiciones ambientales de la zona donde se implementó el subsistema, bajo esta previsión, se optó por el uso de una caja de tipo IP65 para la protección de los elementos electrónicos a una exposición directa al polvo, humedad y rayos UV.

Figura 6

Elementos del nodo LoRa protegidos dentro de caja IP65

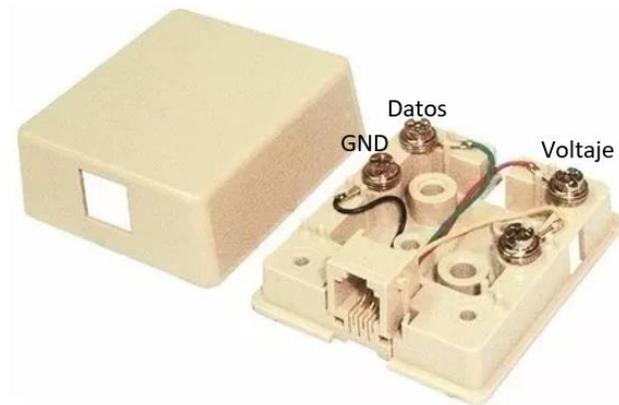


Los elementos resguardados dentro de la caja de protección IP65 y la función que desempeñan dentro del sistema de medición, figura 6, se describen con mayor detalle a continuación:

La caja de conexión RJ-11, figura 6 a), se implementó con el objetivo de facilitar la instalación del nodo de medición en el campo, está diseñada bajo el esquema de conectividad de 4 hilos, figura 7, sin embargo, debido a que la conexión del sensor con el microcontrolador involucra el uso de únicamente 3 hilos se optó por inhabilitar uno de ellos.

Figura 7

Roseta de conexión RJ-11



De manera convencional se definió el uso de los pines en base al color, dada la relación que se muestra en la tabla 2, las terminales destinadas para la alimentación y la comunicación del sensor digital DHT22 se soldaron con estaño a un extremo del cable de conexión RJ-11, teniendo en consideración el uso de únicamente 3 hilos se colocó el conector de cuatro vías al otro extremo del cable, la conexión de la roseta RJ-11 al microcontrolador Arduino Nano Every se realizó mediante cables de tipo hembra-macho, los cuales fueron capados en el extremo macho y sujetos a la caja de conexión RJ-11 ajustando los tornillos que se muestran en la figura 7.

Tabla 2*Distribución de conexión a la roseta RJ-11*

Función de conexión	Color
GND	Negro
Voltaje	Rojo
Datos	Verde

La batería, figura 6 b), en conjunto con el regulador de carga, figura 6 c), permiten suministrar la energía al microcontrolador y al módulo LoRa, el regulador de carga con la configuración de la tabla 3, permite recargar la batería de manera adecuada en base a las especificaciones de la tabla 4, además, proporciona el voltaje adecuado para el funcionamiento del microcontrolador a través del del conector USB integrado.

Tabla 3*Configuración del regulador de carga para panel solar modelo W88-C*

Parámetro	Valor
Nivel de carga máx.	13.8V
Nivel de carga min.	12.6V
Nivel de carga para desconexión	10.7V
Tipo de batería	Batería sellada (B01)

Tabla 4*Instrucciones de recarga a 20°C para batería sellada de ácido-plomo BR-1207*

Aplicación	Voltaje	Corriente
En reposo	13.5 a 13.8Vcc	Sin limite
En uso	14.4 a 15Vcc	2.8A máx.

Además de los elementos considerados en la subsección 3.1.1 la energización del subsistema involucra el uso de celdas fotovoltaicas como se explica en la tabla 5.

Tabla 5*Elementos para la energización del subsistema denominado nodo sensor LoRa*

Elemento	Función
Batería	Almacenar la energía eléctrica proporcionada por el panel solar para energizar el microcontrolador y los elementos periféricos conectados al mismo mediante el regulador de carga.
Panel solar	Suministrar voltaje hacia el regulador de carga según sea su exposición a la radiación solar para recargar la batería.
Regulador de carga	Controlar el voltaje suministrado por el panel solar para para recargar la batería bajo las condiciones adecuadas y proporcionar el voltaje para energizar el microcontrolador.

El módulo LoRa e5 figura 6 d) y figura 8, permite abrir un canal bidireccional de comunicación inalámbrica entre los microcontroladores empleados como nodo sensor y puerta de enlace. La configuración y el uso de funciones se establecen mediante la información de control enviada por el microcontrolador Arduino Nano Every que se comunica con el módulo a través del protocolo de comunicación UART.

Figura 8

Módulo LoRa e5



La selección del microcontrolador Arduino Nano Every, figura 6 e), se realizó en base a las especificaciones técnicas de la tabla 6, considerando su bajo consumo eléctrico como una ventaja estratégica para la optimización del funcionamiento del nodo sensor LoRa.

Tabla 6

Características del microcontrolador Arduino nano every ATmega4809

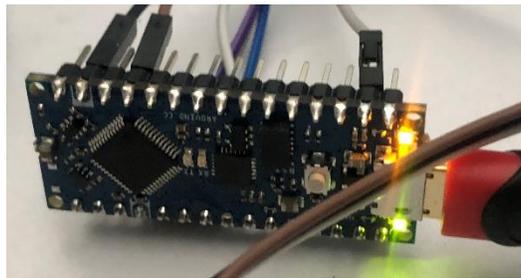
Voltaje de funcionamiento	5V
Voltaje de entrada	7-21V
Corriente DC por pin de E/S	20 mA
Velocidad de reloj	20 MHz
Memoria flash de la CPU	48 KB
SRAM	6 KB
EEPROM	256 bytes
Pines PWM	5 (D3, D5, D6, D9, D10).

Protocolos de comunicación	UART, SPI, I2C
Pines de entrada analógica	8 (ADC de 10 bits).
Pines de salida analógica	Solo a través de PWM.
Dimensiones	45 mm x 18 mm.

La función del microcontrolador Arduino Nano Every, figura 9, técnicamente define el propósito del nodo dentro del sistema de medición debido a que se encarga de organizar, empaquetar y enviar la información de temperatura y humedad relativa obtenida del sensor DHT22 hacia la puerta de enlace.

Figura 9

Arduino Nano Every



3.1.1.1 Construcción De La Carcasa De Protección Para El Sensor DHT22.

Para proteger el sensor de la exposición directa a las condiciones degradantes del ambiente de trabajo se usó una carcasa protectora conocida como pantalla de Stevenson cuyo objetivo principal es permitir el flujo de aire al elemento sensor protegiéndolo de la exposición directa a los rayos UV y a las precipitaciones evitando la alteración en las mediciones y los daños al módulo DHT22.

En la tabla 7, se exhiben los datos técnicos correspondientes al módulo del sensor digital DHT22.

Tabla 7

Ficha técnica modulo DHT22

Módulo DHT22 (AM2302) con PCB	
Modelo	DHT22
Voltaje de alimentación	3 a 5V
Corriente de consumo	Max 2.5mA durante la solicitud de datos
Lectura de humedad relativa	0 a 100% \pm 2-5%
Lectura de temperatura	-40 a 80 °C \pm 0.5°C
Muestreo	0.5Hz máx.
Dimensiones	27mm x 59mm x 13.5mm

La pantalla de Stevenson fue impresa con filamento ASA (material terpolímero amorfo termoplástico) color blanco, un material comúnmente comparado con ABS debido a sus propiedades termo-mecánicas, estas propiedades son especialmente útiles cuando las aplicaciones requieren una exposición al medio ambiente, es decir, el material estará en contacto con precipitaciones, polvo y rayos UV, el material ASA cuenta con un elastómero de acrílico que lo diferencia del ABS.

El diseño se obtuvo gratuitamente de la página Cults[®], figura 10. Mediante el programa *UltiMaker Cura v5.4.0* se generó el código G para la construcción de la carcasa de protección mediante una impresora 3D *Creality Ender-3 V2* con un extrusor de 2 mm, la configuración utilizada se muestra en la figura 11 y la pantalla de Stevenson impresa en 3D se muestra en la figura 12.

Figura 10

Diseño gratuito de pantalla de Stevenson para impresión 3D obtenido de Cults®



Figura 11

Configuración materia ASA

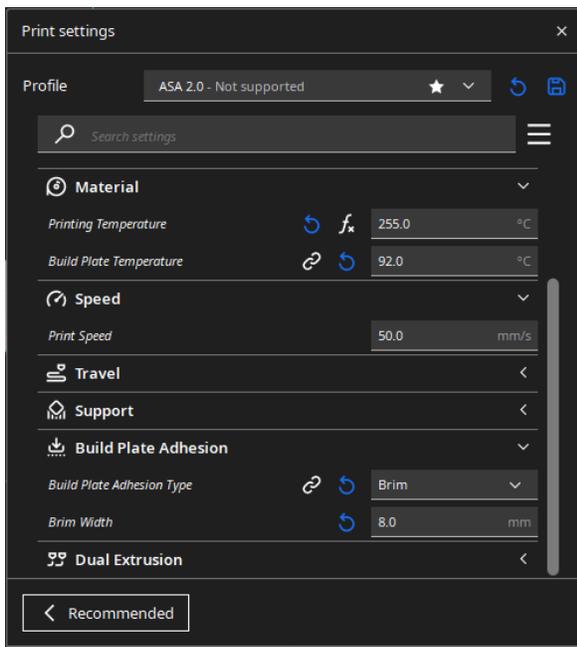


Figura 12

Pantalla de Stevenson

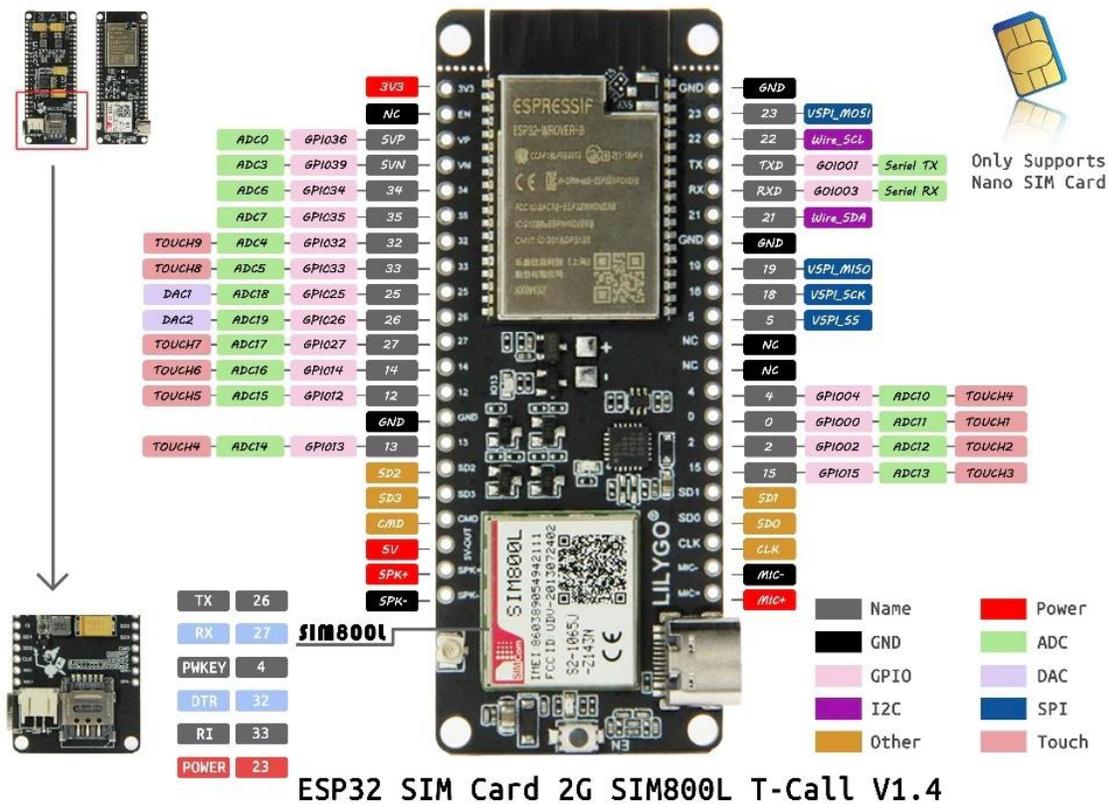


3.1.2 Puerta De Enlace Mediante ESP32 Con Módulo SIM800L Integrado

El microcontrolador TTGO T-CALL V1.4, figura 13, es una placa de desarrollo que además de integrar las funciones de conectividad (Wifi y Bluetooth) y de comunicación (UART, SPI, MOSI, I2C) propias del ESP32 añade un módulo interno SIM800L accesible mediante comunicación UART, a pesar de que el módulo está integrado en una misma placa la configuración y la transmisión de datos se realiza mediante los pines preestablecidos 26, 27 (Tx, Rx) y mediante el uso de comandos AT, las especificaciones técnicas se muestran en la tabla 8.

Figura 13

Microcontrolador ESP32 SIM800L T-Call V1.4



Nota. Fuente: Faomic, S (14 de febrero de 2024). Tigo Arduino Celular T-call V1.5 Esp32 Sim800l Gprs Gsm Iot. <https://www.faomic.com/product-page/tigo-arduino-celular-t-call-v1-5-esp32-sim800l-gprs-gsm-iot>

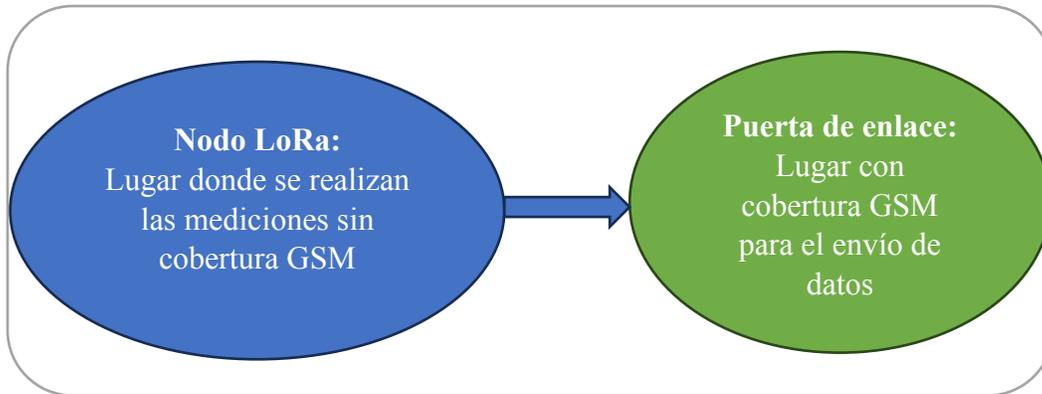
Tabla 8*Especificaciones técnicas del microcontrolador T-Call ESP32 con SIM800L integrado*

T-Call SIM800L	
MCU	ESP32
GSM/GPRS	SIM800L
frecuencia de reloj	240Mhz
Memoria Flash	4MB
RAM	8MB
Conectividad inalámbrica	Wi-Fi:802.11 b/g/n Bluetooth: V4.2 BR/EDR/BLE
Interfaces	I2C, SPI, UART, SDIO, I2S, CAN
Factor	LCC
GPRS	Clase 12/10
Dimensiones	15.8*17.8*2.4 mm

El dispositivo empleado como puerta de enlace que permite la conectividad a una red de telefonía móvil para la transmisión de los datos al servidor de almacenamiento en la nube fue colocado en un punto estratégico como se observa en la figura 14.

Figura 14

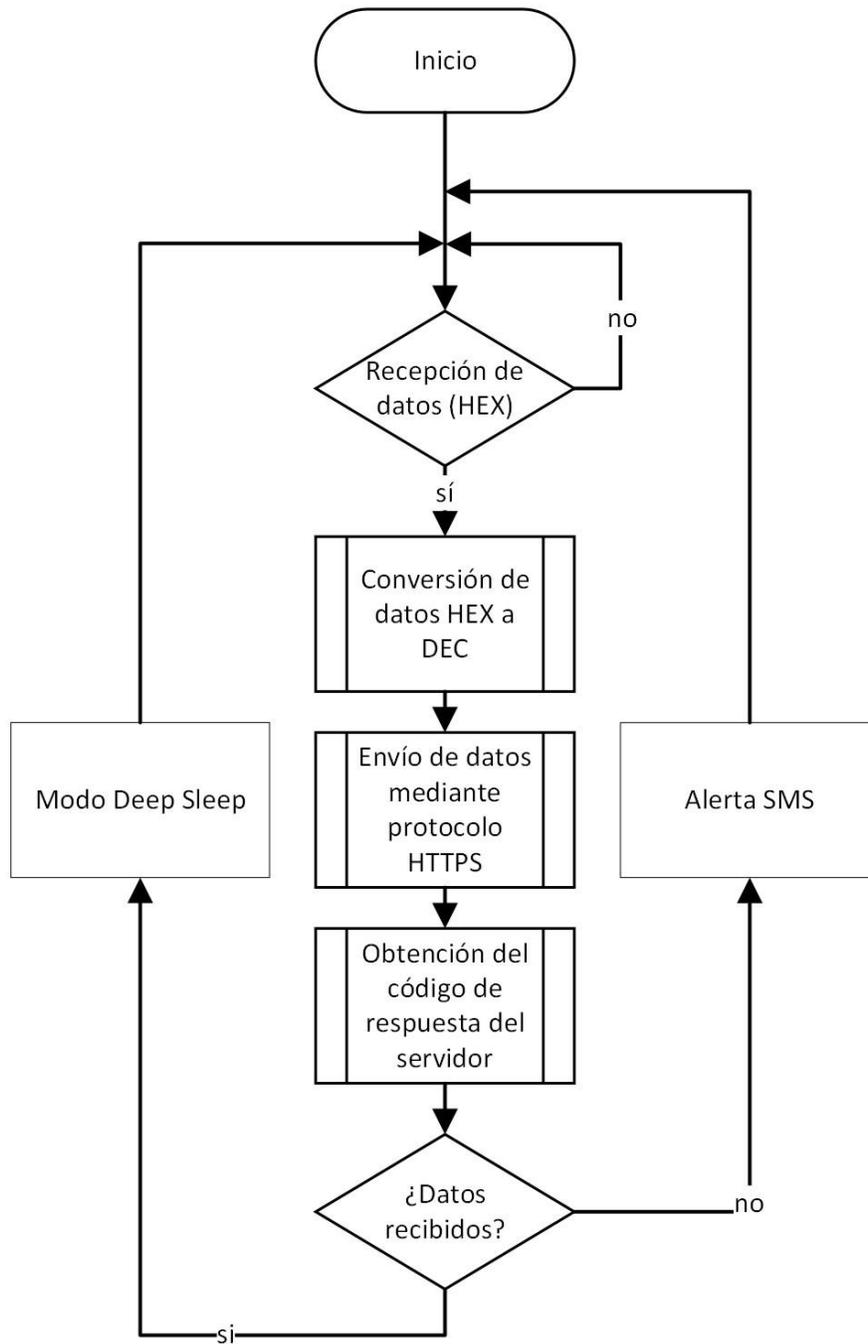
Diagrama de posicionamiento de los elementos del sistema de medición



El diagrama de flujo de la figura 15 explica de forma general las acciones que realiza el microcontrolador utilizado como puerta de enlace.

Figura 15

Diagrama de operaciones generales realizadas por la puerta de enlace



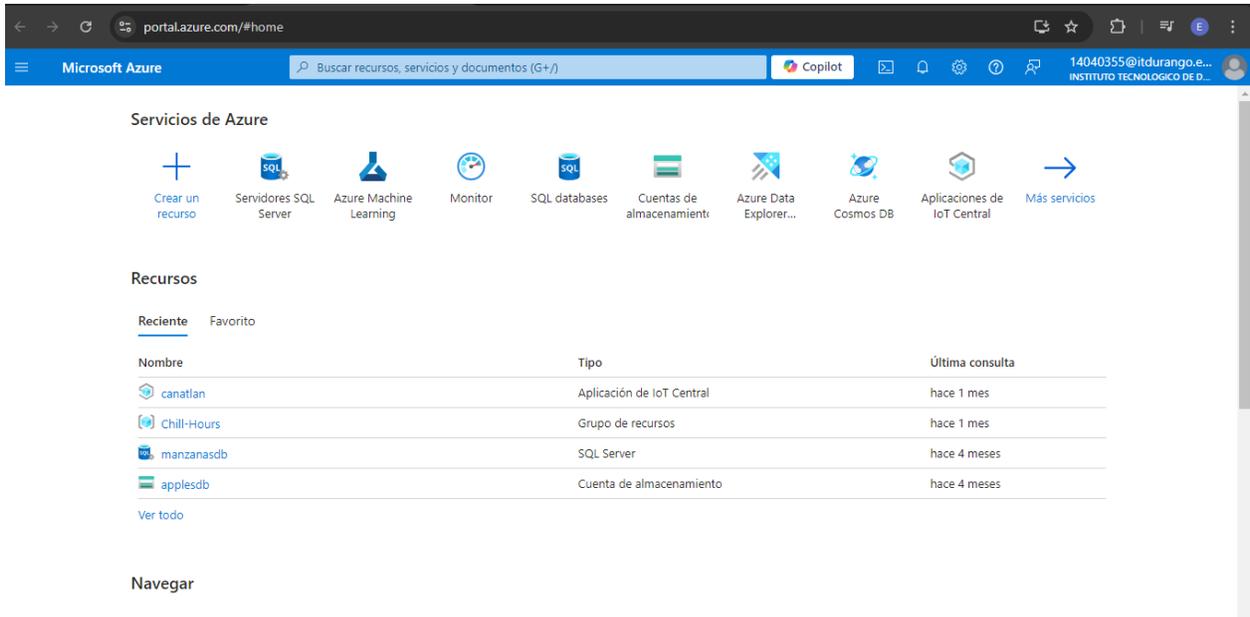
El código resultante en lenguaje C se muestra en el anexo A

3.1.3 Desarrollo De La Aplicación Y Creación De Dispositivos En Azure Iot Central®

En esta sección se muestra el proceso para la creación de una aplicación de Azure IoT Central bajo un enfoque visual, además se explican los conceptos para la creación, provisionamiento y obtención de las claves de acceso para publicación de datos en la plataforma mediante el uso de métodos del protocolo HTTPS.

Figura 16

Vista portal Azure®



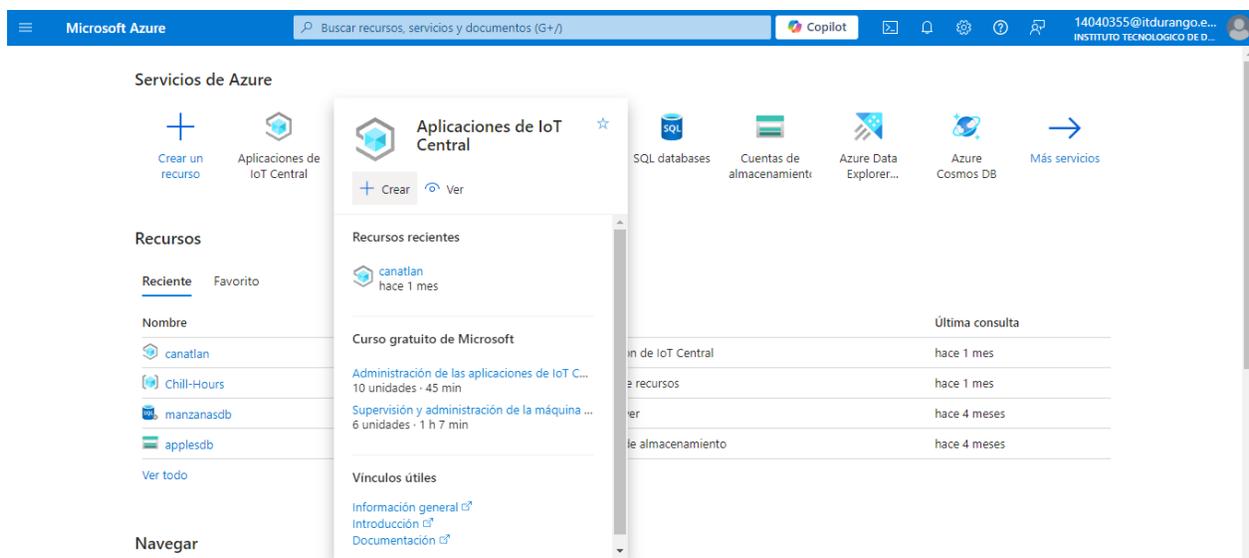
En la figura 16, se muestra el entorno visual del portal Azure, dentro de la plataforma se encuentran las aplicaciones de utilidad, como servidores SQL, servicios de aprendizaje de máquina, bases de datos, entre otras, algunas de estas aplicaciones se presentan como un servicio gratuito y algunas otras requieren de una suscripción de pago asociada a la cuenta, durante el desarrollo de este proyecto se hizo uso de algunos de los servicios de Azure mediante una suscripción del tipo estudiantil utilizando el correo institucional.

La creación de un recurso de aplicación de IoT Central para la conexión de dispositivos IoT que permita el monitoreo y el almacenamiento de datos se detalla a continuación:

Dentro del portal Azure se colocó el cursor encima del recurso Aplicaciones de Azure IoT Central y se dio clic en crear, figura 17.

Figura 17

Creación de aplicación IoT Central®



En el asistente para la creación de una aplicación de IoT Central se llenaron los campos con la información requisitada que implica el uso de una suscripción del tipo estudiantil, la selección del nombre y la asignación URL del recurso, figura 18. Antes de la creación de la aplicación el asistente pide la confirmación de aceptación de términos y muestra la información general ingresada para confirmar la solicitud, figura 19. Una vez creada la aplicación un nuevo recurso aparece en el portal de Azure con el nombre “horas-frio”, figura 20, al seleccionar este recurso se accede a información general donde se muestra el URL de la aplicación, figura 21.

Figura 18

Selección de la suscripción para la creación de la aplicación

portal.azure.com/#create/Microsoft.IoTCentral

Microsoft Azure

Buscar recursos, servicios y documentos (G+J)

Copilot

14040355@itdurango.e...
INSTITUTO TECNOLÓGICO DE D...

Inicio >

Aplicación de IoT Central

Select the subscription to manage the deployed IoT Central resource and costs. Use resource groups like folders to organize and manage all your resources.

Suscripción *

Grupo de recursos *
[Crear nuevo](#)

Instance details

Nombre de recurso *

Dirección URL de la aplicación *
azureiotcentral.com

Plantilla *

Region *

Plan de precios *

< Anterior Siguiente **Revisar y crear**

[Enviar comentarios](#)

Figura 19

Aceptación de términos previa a la creación de la aplicación

portal.azure.com/#create/Microsoft.IoTCentral

Microsoft Azure

Buscar recursos, servicios y documentos (G+J)

Copilot

14040355@itdurango.e...
INSTITUTO TECNOLÓGICO DE D...

Inicio >

Aplicación de IoT Central

TÉRMINOS

Al hacer clic en "Crear", (a) acepto los términos legales y las declaraciones de privacidad relacionados con cada oferta de Marketplace que se enumeró previamente; (b) autorizo a Microsoft a facturar con mi método de pago actual las cuotas relacionadas con las ofertas, con la misma frecuencia de facturación que mi suscripción de Azure; y (c) autorizo a Microsoft a compartir mi información de contacto y los datos de transacción y uso con los proveedores de dichas ofertas. Microsoft no proporciona derechos sobre ofertas de terceros. Para obtener información adicional, consulte los [Términos de Azure Marketplace](#).

Basics

Suscripción	Azure for Students
Grupo de recursos	Ejemplo
Nombre de recurso	horas-frio
Dirección URL de la aplicación	horas-frio
Plantilla	Aplicación personalizada
Region	Centro de EE. UU.
Plan de precios	Estándar 2

< Anterior Siguiente **Crear**

[Enviar comentarios](#)

Figura 20

Recurso horas-frío en el portal Azure

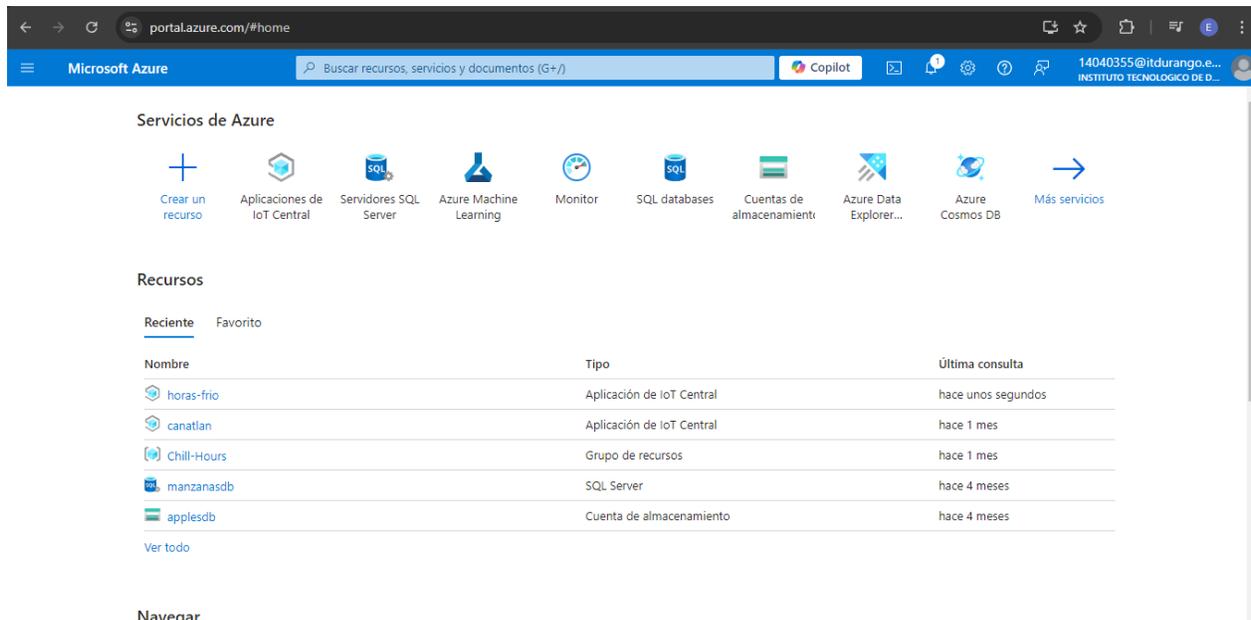


Figura 21

Acceso al recurso mediante la dirección URL de la aplicación

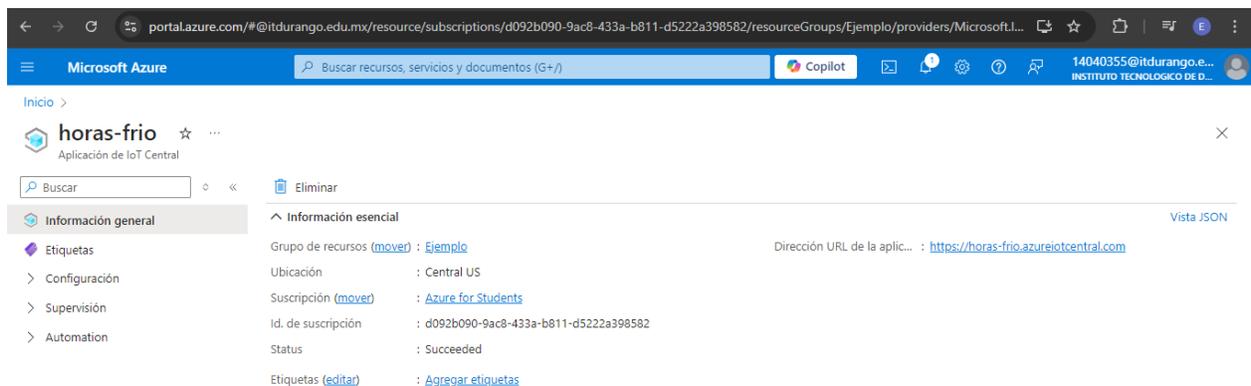
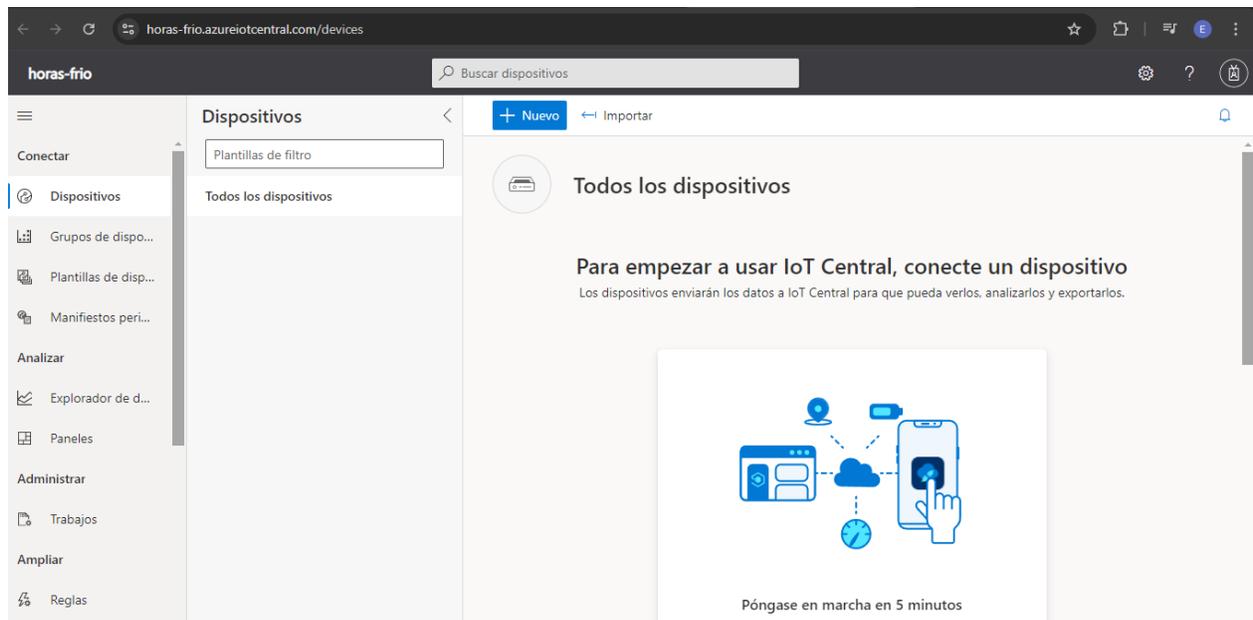


Figura 22

Vista general de la aplicación de Azure IoT Central®



Una vez dentro de la aplicación, figura 22, se creó un dispositivo nuevo asignando “*Nodo sensor LoRa*” al nombre del dispositivo y “*Nodo_LoRa*” como identificador del dispositivo, figura 23, una vez creado el dispositivo, figura 24, se obtuvieron los parámetros necesarios para realizar el provisionamiento dando clic en el dispositivo creado y posteriormente en el botón conectar que despliega la información necesaria para la conexión del dispositivo, figura 25.

Figura 23

Creación de un nuevo dispositivo

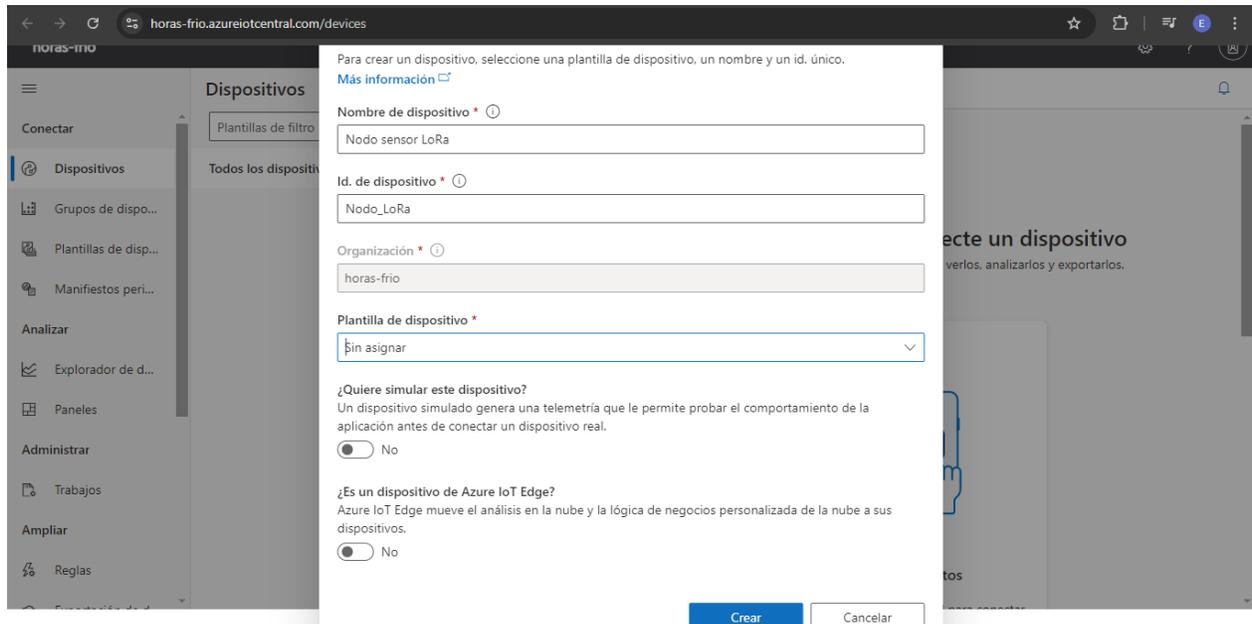


Figura 24

Dispositivo creado sin provisionamiento

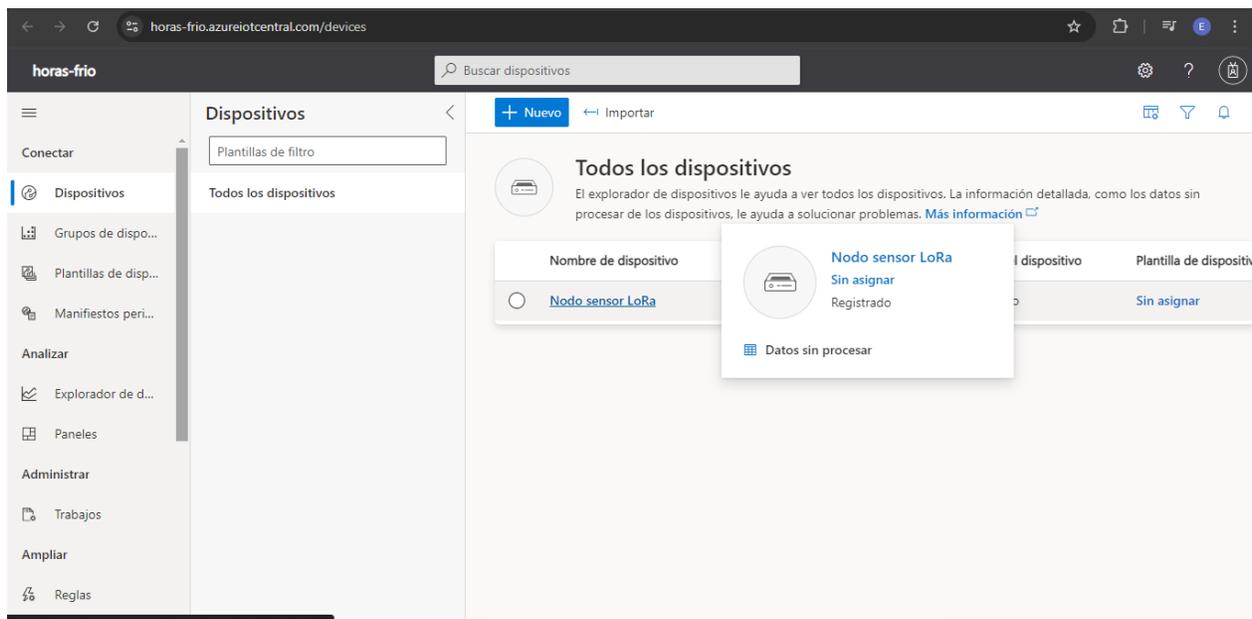


Figura 25

Obtención de los parámetros para el aprovisionamiento y la conexión del dispositivo

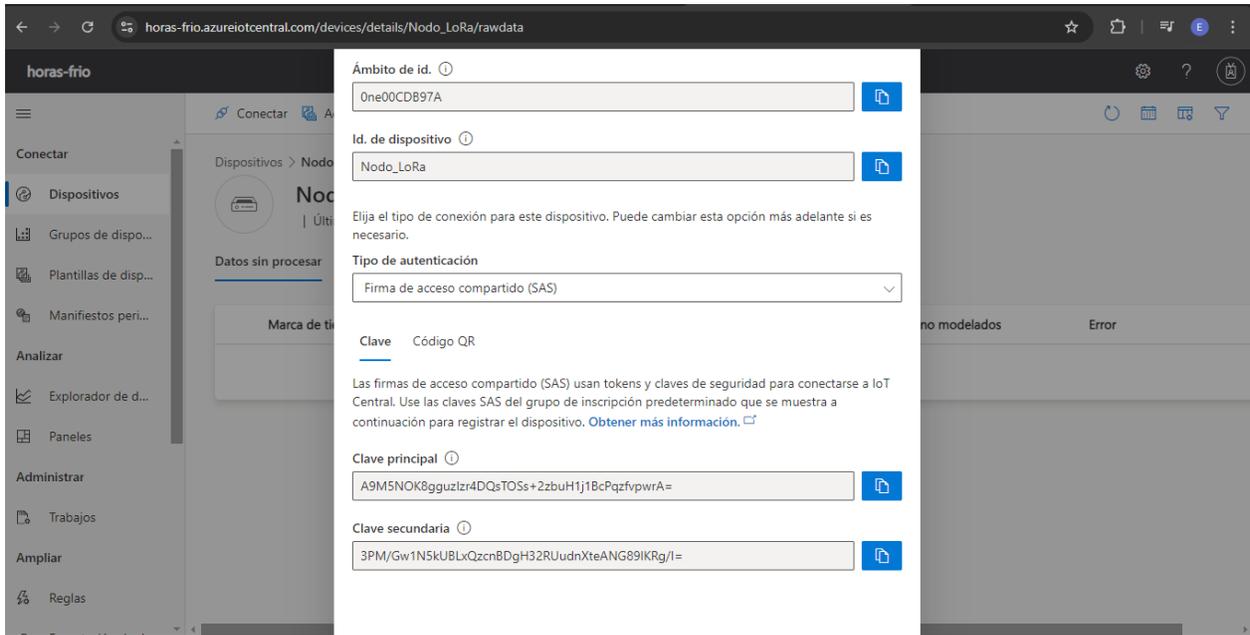
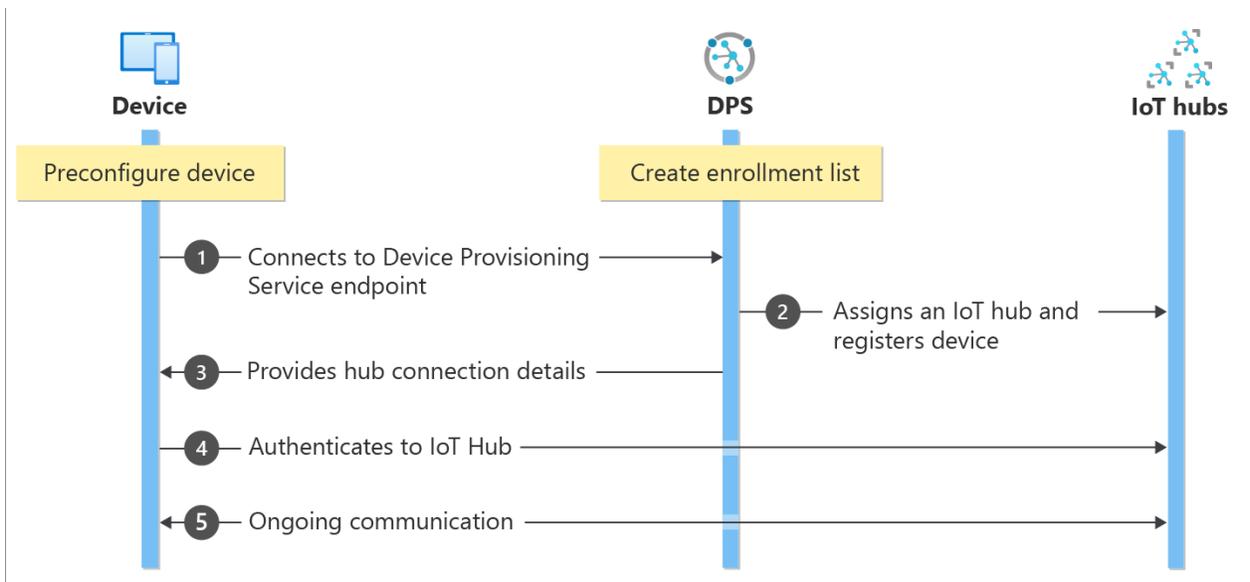


Figura 26

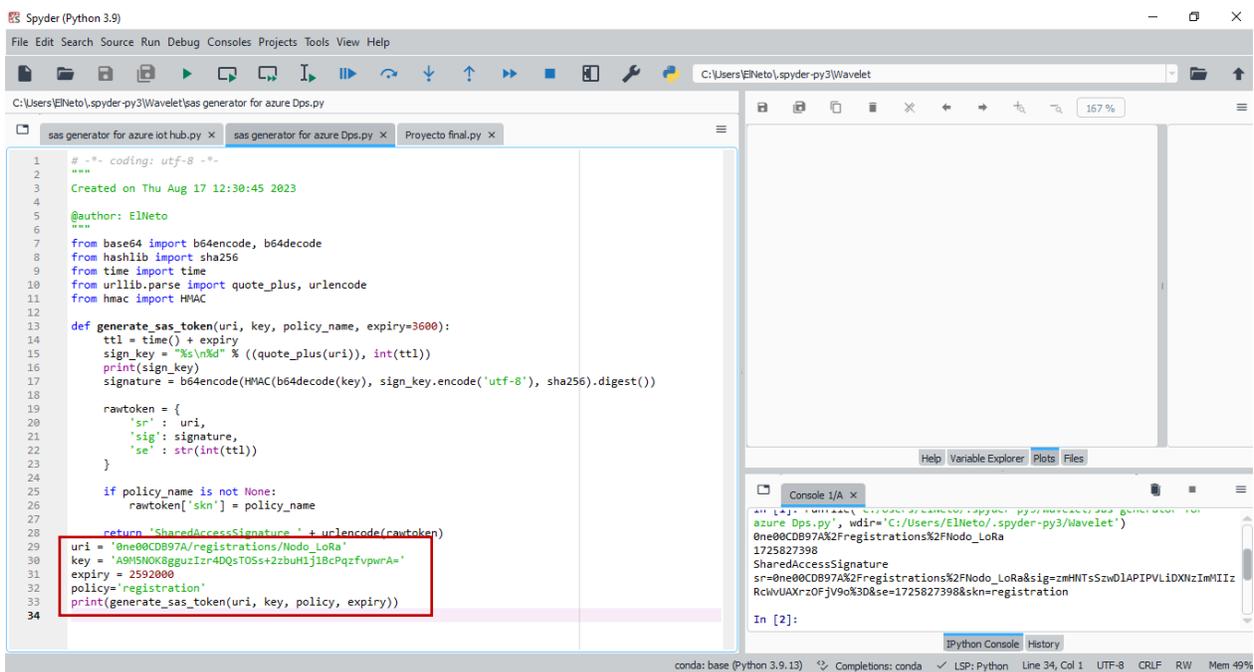
Diagrama de seguimiento para la comunicación con la plataforma



En la figura 26 se muestra el esquema general para el aprovisionamiento y conexión del dispositivo creado, esto se lleva a cabo mediante peticiones del protocolo HTTPS a través del servicio de aprovisionamiento de dispositivos (DPS) utilizando la información de conexión obtenida previamente, figura 25.

Figura 27

Creación de token para la conexión segura al servicio DPS



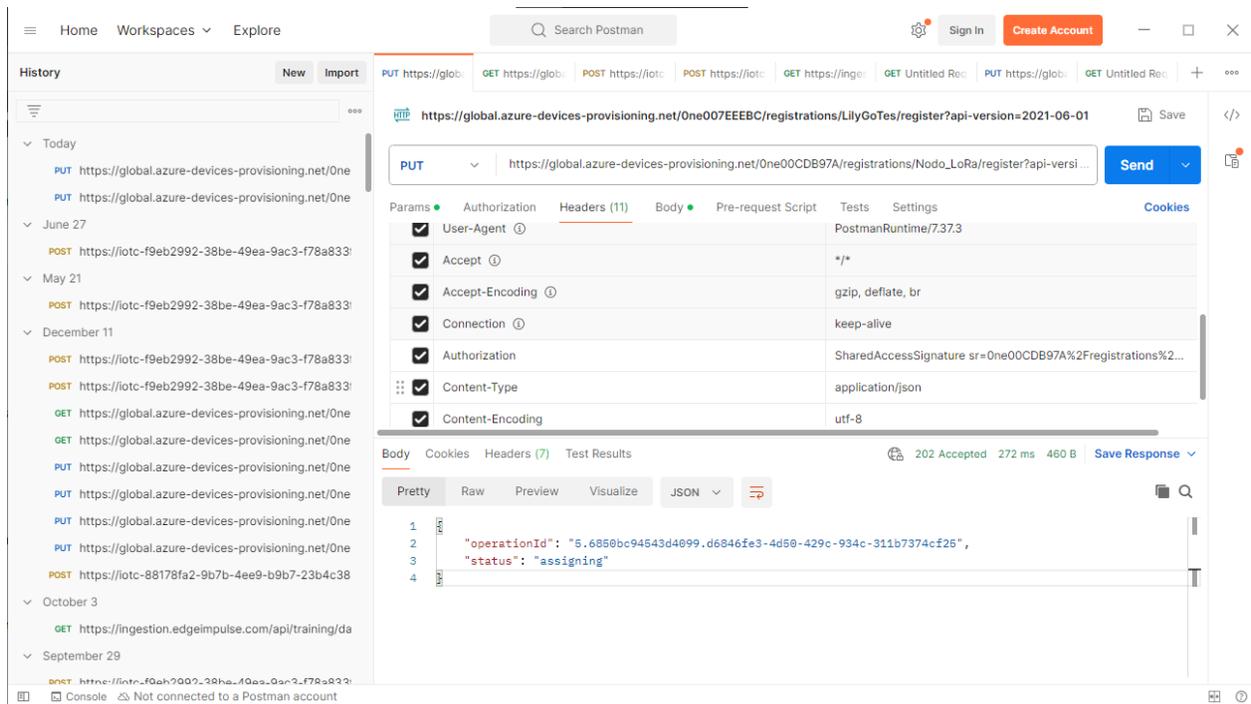
```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Aug 17 12:30:45 2023
4
5  @author: ElNeto
6  """
7  from base64 import b64encode, b64decode
8  from hashlib import sha256
9  from time import time
10 from urllib.parse import quote_plus, urlencode
11 from hmac import HMAC
12
13 def generate_sas_token(uri, key, policy_name, expiry=3600):
14     ttl = time() + expiry
15     sign_key = "%s\n%d" % ((quote_plus(uri)), int(ttl))
16     print(sign_key)
17     signature = b64encode(HMAC(b64decode(key), sign_key.encode('utf-8'), sha256).digest())
18
19     rawtoken = {
20         'sr': uri,
21         'sig': signature,
22         'se': str(int(ttl))
23     }
24
25     if policy_name is not None:
26         rawtoken['skn'] = policy_name
27
28     return "SharedAccessSignature " + urlencode(rawtoken)
29
30 uri = '0ne00CDB97A/registrations/Nodo_LoRa'
31 key = 'A9H5NOK8gguzIzr4DQsT05s+2zbuH1j1BcPqzfvpwA='
32 expiry = 2592000
33 policy='registration'
34 print(generate_sas_token(uri, key, policy, expiry))
```

```
Console 1/A x
In [1]: generate_sas_token('C:/Users/ElNeto/.spyder-py3/Wavelet/azure Dps.py', wdIr='C:/Users/ElNeto/.spyder-py3/Wavelet')
0ne00CDB97A%2Fregistrations%2FNodo_LoRa
1725827398
SharedAccessSignature
sr=0ne00CDB97A%2Fregistrations%2FNodo_LoRa&sig=zmHNTsSzw01AP1PVL1DXNzImMI1zRclwUAXrz0FjV9o%3D&se=1725827398&skn=registration
In [2]:
```

El envío de la información segura dentro del protocolo HTTPS se realiza añadiendo un encabezado denominado *Authorization* que contiene un token generado en lenguaje Python con los datos correspondientes al ámbito de id, id de dispositivo y la clave principal, además de estos datos el token requiere el parámetro de tiempo de expiración y el *polycname* cuyo valor corresponde a “*registration*”, figura 27.

Figura 28

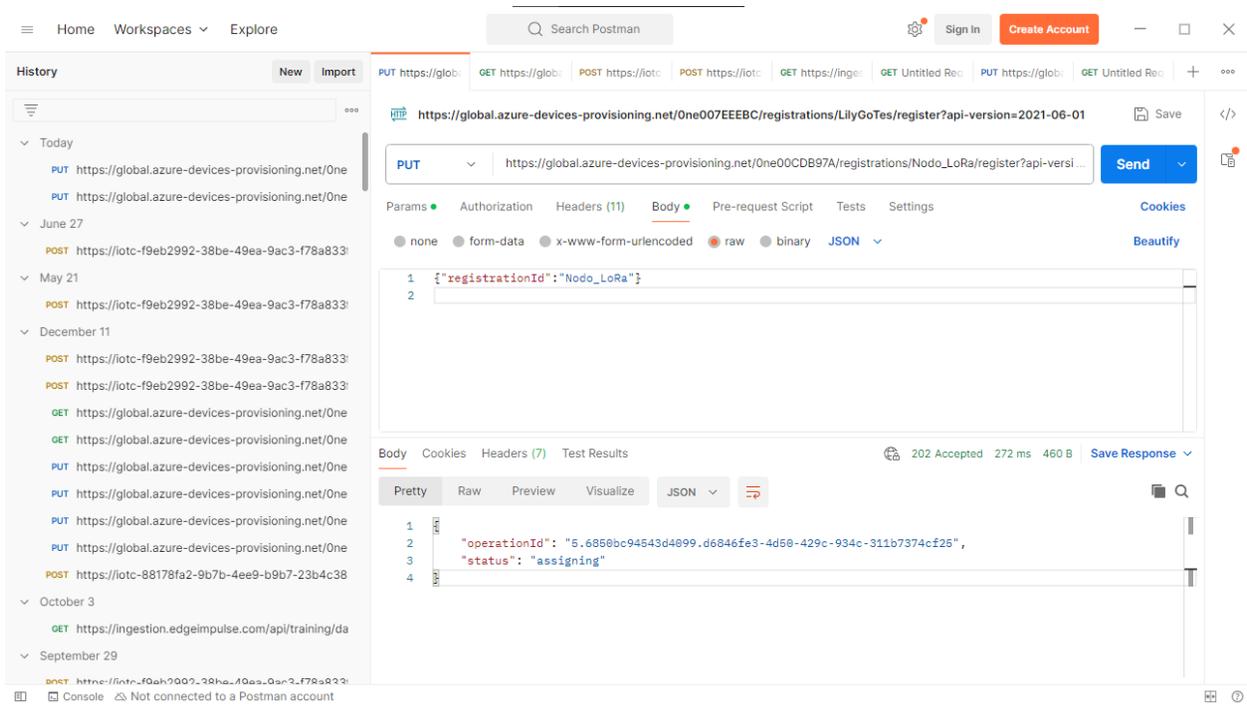
Software POSTMAN para la obtención del id de operación mediante petición PUT



En la figura 28, se muestran los encabezados para el proceso de obtención del *id de operación* mediante una petición tipo *PUT* del protocolo *HTTPS* en el URL https://global.azure-devices-provisioning.net/One00CDB97A/registrations/Nodo_LoRa/register?api-version=2021-06-01 con ayuda del programa *POSTMAN*, en el cuerpo de la petición se agrega la información del nombre del dispositivo que se desea registrar como se muestra en la figura 29.

Figura 29

Cuerpo de la petición PUT para el registro del dispositivo



La respuesta de la solicitud arroja como resultado el id de operación que es indispensable para el envío de datos de telemetría mediante peticiones tipo *POST*. Los encabezados de la siguiente petición y el token son los mismos que se aplicaron anteriormente, sin embargo, los valores de los subdirectorios de la URL son distintos, figura 30. El id de operación se envía en el cuerpo de la solicitud *HTTPS* tipo *GET* para la obtención del *Hub* asignado, figura 31.

Figura 30

Encabezados petición GET para la obtención del Hub asignado

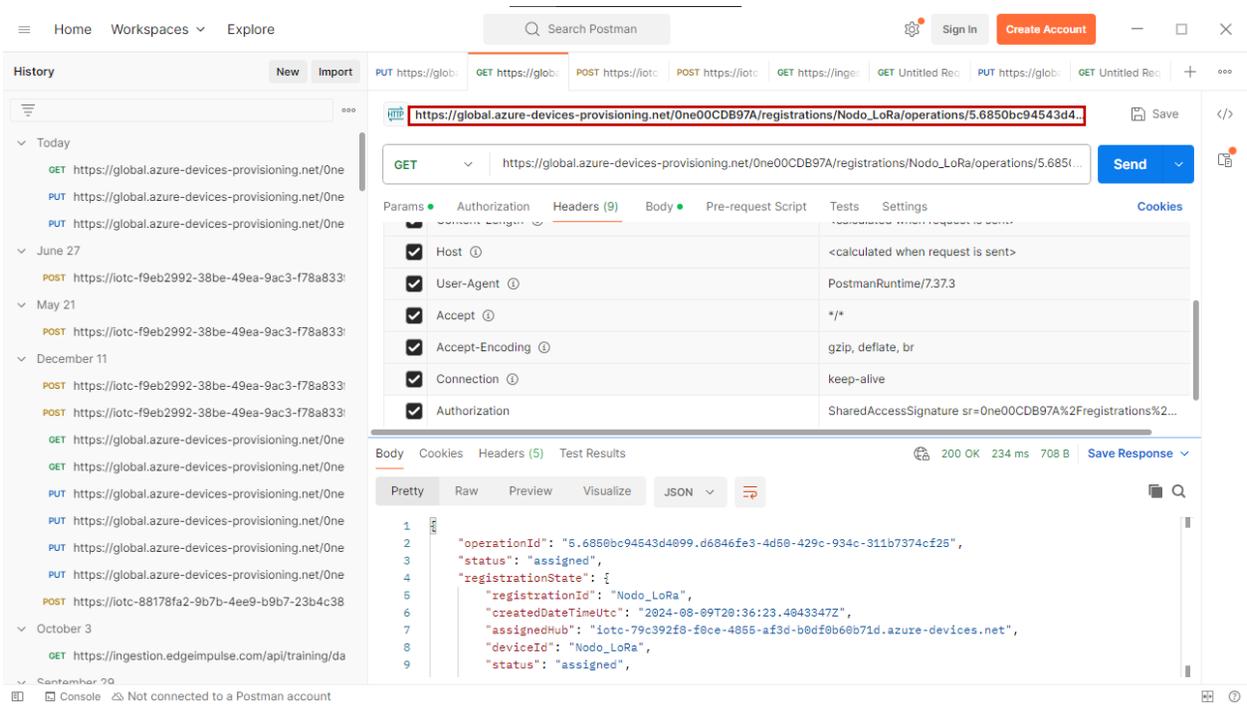
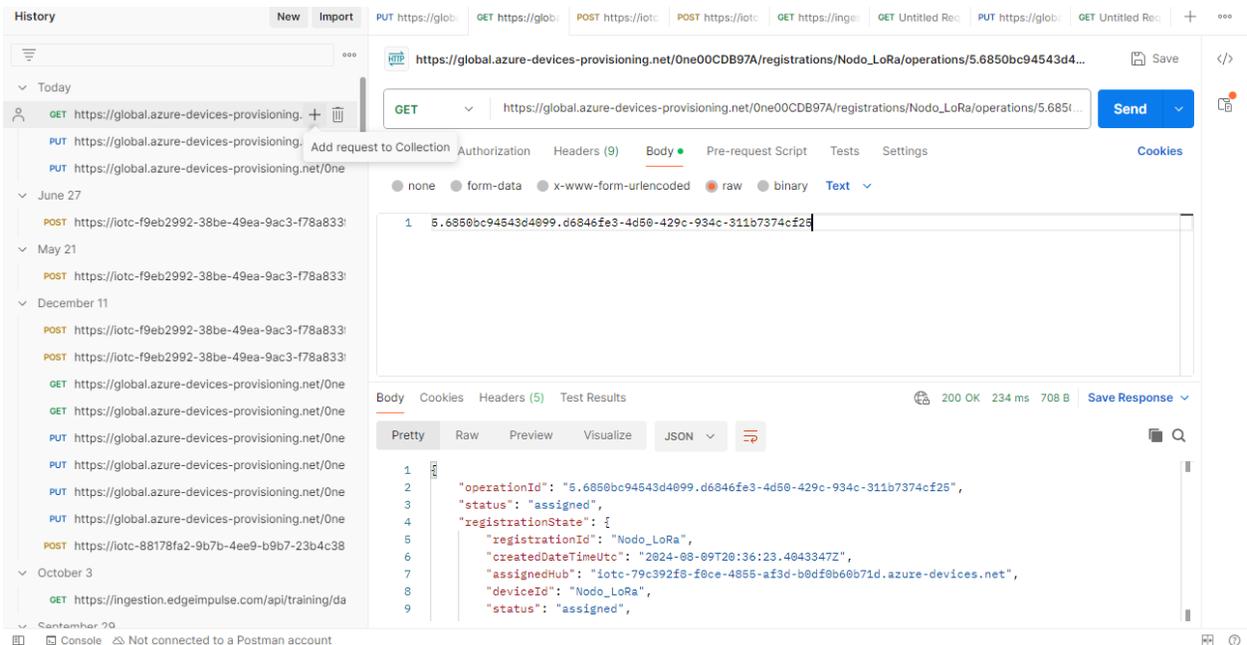


Figura 31

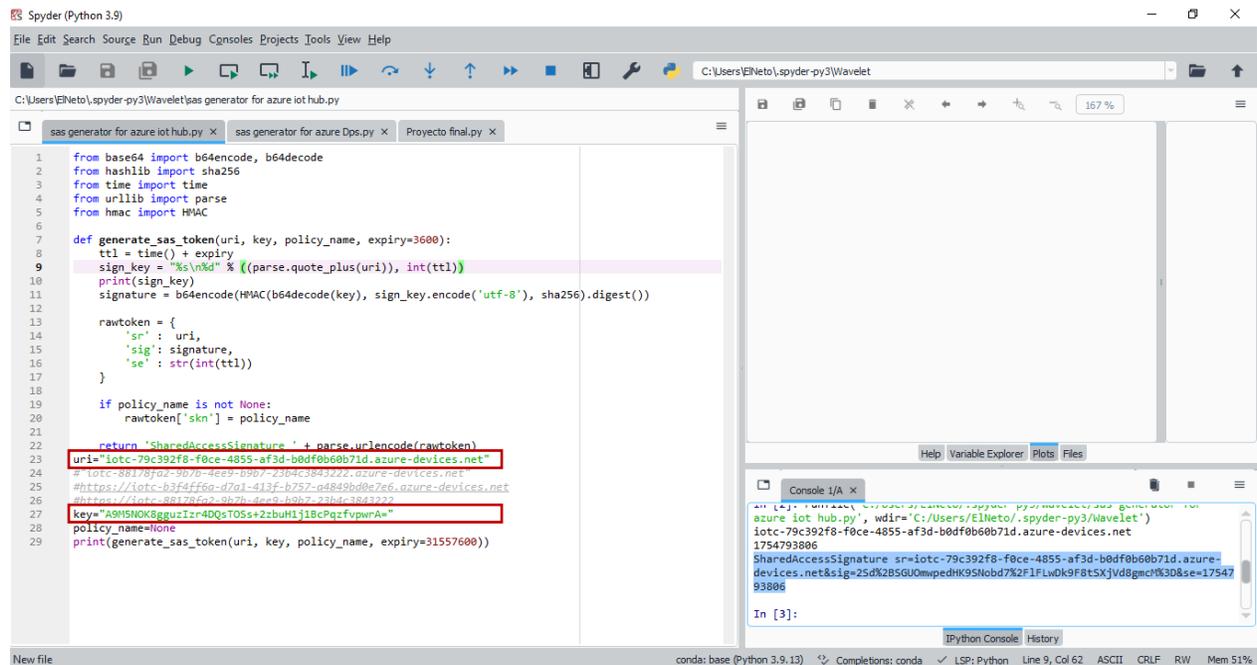
Cuerpo de la petición GET para la obtención del Hub asignado



Antes de comenzar con el envío de telemetría es necesario generar un nuevo Token utilizando la información obtenida mediante el servicio *DPS* que corresponde al *Hub* asignado y la clave primaria obtenida desde el entorno de la aplicación de Azure IoT Central, figura 32.

Figura 32

Creación de token para la publicación de los datos mediante peticiones tipo POST



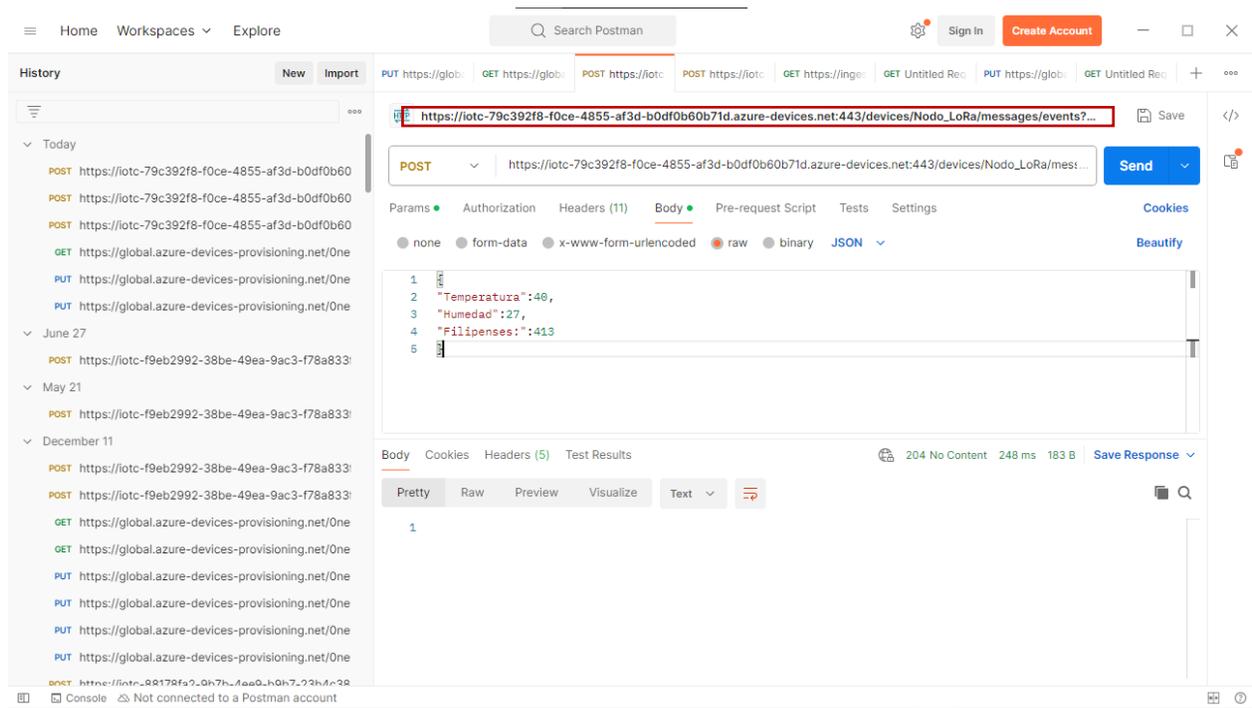
```
spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\EI\Neto\spyder-py3\Wavelet
C:\Users\EI\Neto\spyder-py3\Wavelet\sas generator for azure iot hub.py
sas generator for azure iot hub.py x sas generator for azure Dps.py x Proyecto final.py x
1 from base64 import b64encode, b64decode
2 from hashlib import sha256
3 from time import time
4 from urllib import parse
5 from hmac import HMAC
6
7 def generate_sas_token(uri, key, policy_name, expiry=3600):
8     ttl = time() + expiry
9     sign_key = "%s\n%s" % ((parse.quote_plus(uri)), int(ttl))
10    print(sign_key)
11    signature = b64encode(HMAC(b64decode(key), sign_key.encode('utf-8'), sha256).digest())
12
13    rawtoken = {
14        'sr': uri,
15        'sig': signature,
16        'se': str(int(ttl))
17    }
18
19    if policy_name is not None:
20        rawtoken['skn'] = policy_name
21
22    return 'SharedAccessSignature ' + parse.urlencode(rawtoken)
23
24 uri="iotc-79c392f8-f0ce-4855-af3d-b0df0b60b71d.azure-devices.net"
25 # iotc-85178f02-9b7b-4e69-89b7-2394c394322a.azure-devices.net
26 #https://iotc-b3f4ff6a-d7a1-413f-b757-a4849b0e7e6.azure-devices.net
27 #https://iotc-86178fa2-9b7b-4e69-89b7-2394c394322a.azure-devices.net
28 key="A9MSNOK8gguzIzr4DQsTOSs+2zbuH1j18cPqzfvpwA="
29 policy_name=None
30 print(generate_sas_token(uri, key, policy_name, expiry=31557600))

Console 1/A x
In [1]: generate_sas_token(uri, key, policy_name, expiry=31557600)
azure iot hub.py', wdir='C:\Users\EI\Neto\spyder-py3\Wavelet')
iotc-79c392f8-f0ce-4855-af3d-b0df0b60b71d.azure-devices.net
1754793806
SharedAccessSignature sr=iotc-79c392f8-f0ce-4855-af3d-b0df0b60b71d.azure-devices.net&sig=25d%2B5GU0mpeDHK95Nobd7%2F1FLw0k9FBt5XjVd8gmcl%308se=1754793806
In [3]:
IPython Console History
conda: base (Python 3.9.13) Completions: conda LSP: Python Line 9, Col 62 ASCII CRLF RW Mem 51%
```

El envío de los datos de prueba de telemetría se realiza mediante una solicitud tipo *POST* reemplazando el Token generado anteriormente en el encabezado *authorization* y agregando los valores de las variables en formato JSON en el cuerpo de la solicitud al URL de la figura 33.

Figura 33

Test para el envío de datos a la plataforma mediante peticiones POST en formato JSON



En la figura 34 se muestran los datos que fueron enviados desde el software *POSTMAN* sin procesar en la plataforma, para la exhibición por medio de elementos gráficos de los datos recibidos se requiere la asociación del dispositivo a un *template* que permite la interpretación de la información recibida mediante un enfoque visual, la creación y asignación de esta plantilla se muestra en las figuras 35-43.

Figura 34

Recepción exitosa de los datos en la plataforma Azure IoT Central sin estructurar

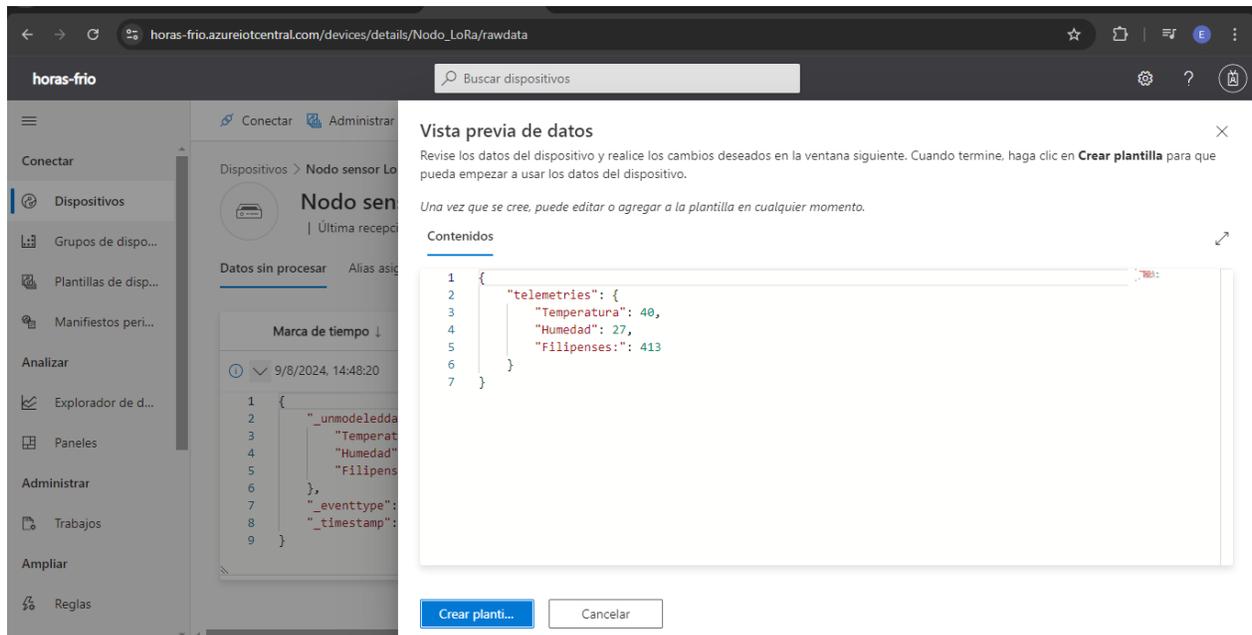


Figura 35

Creación de una plantilla de dispositivo

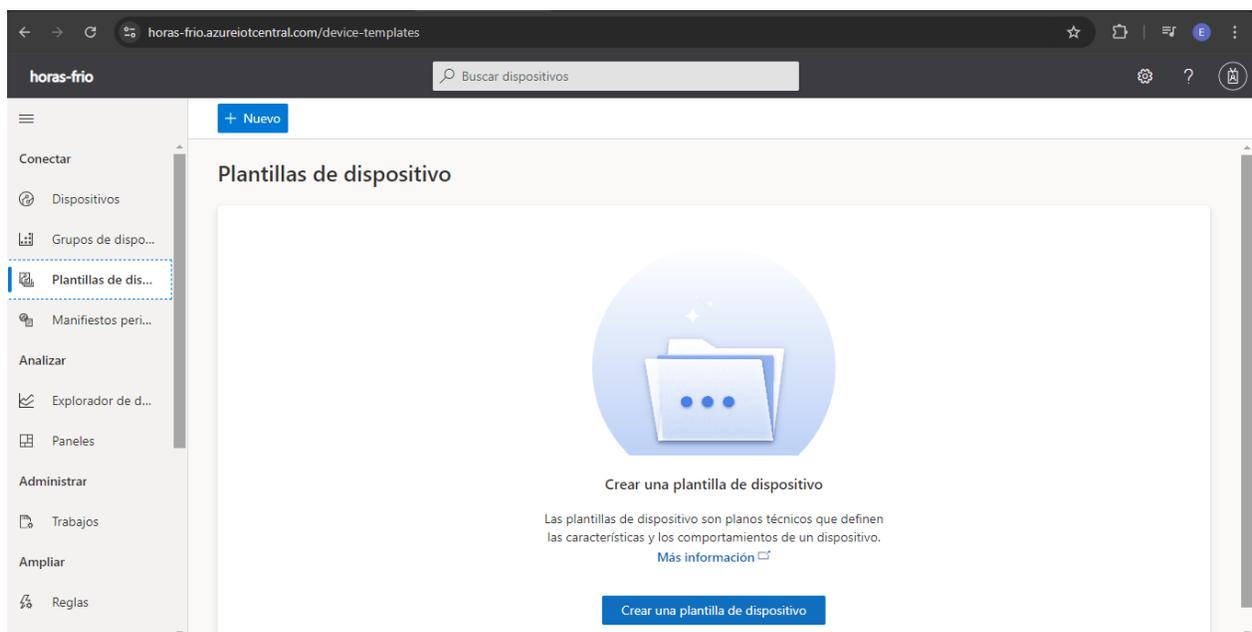


Figura 36

Selección del tipo de dispositivo

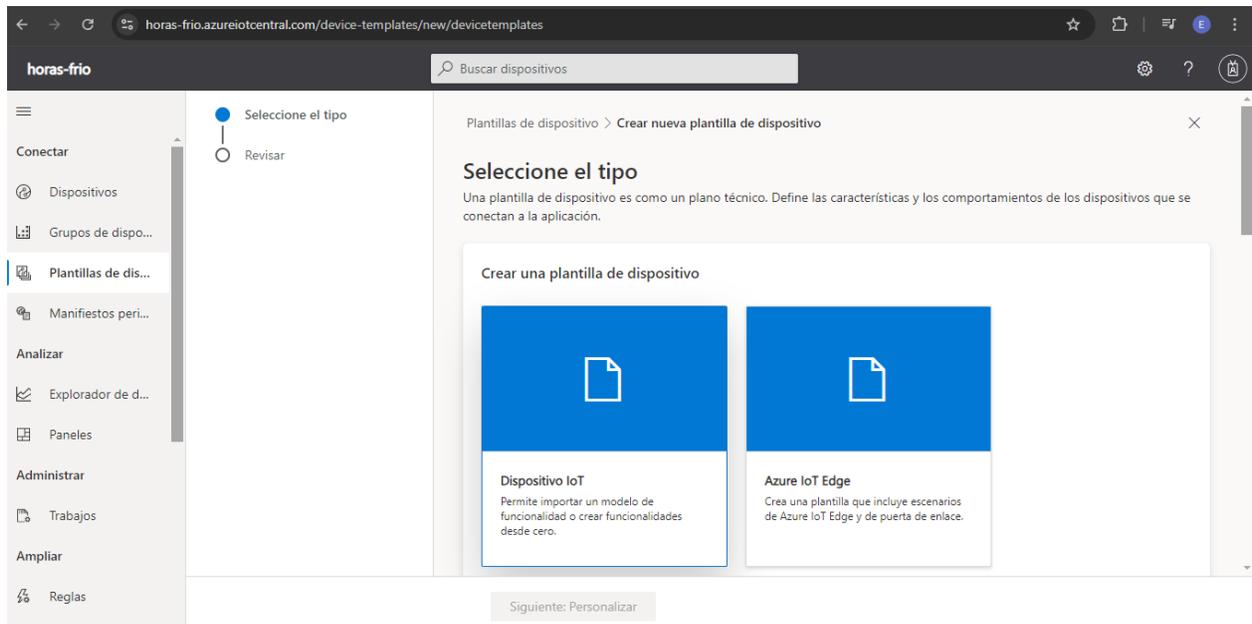


Figura 37

Asignación de nombre de plantilla

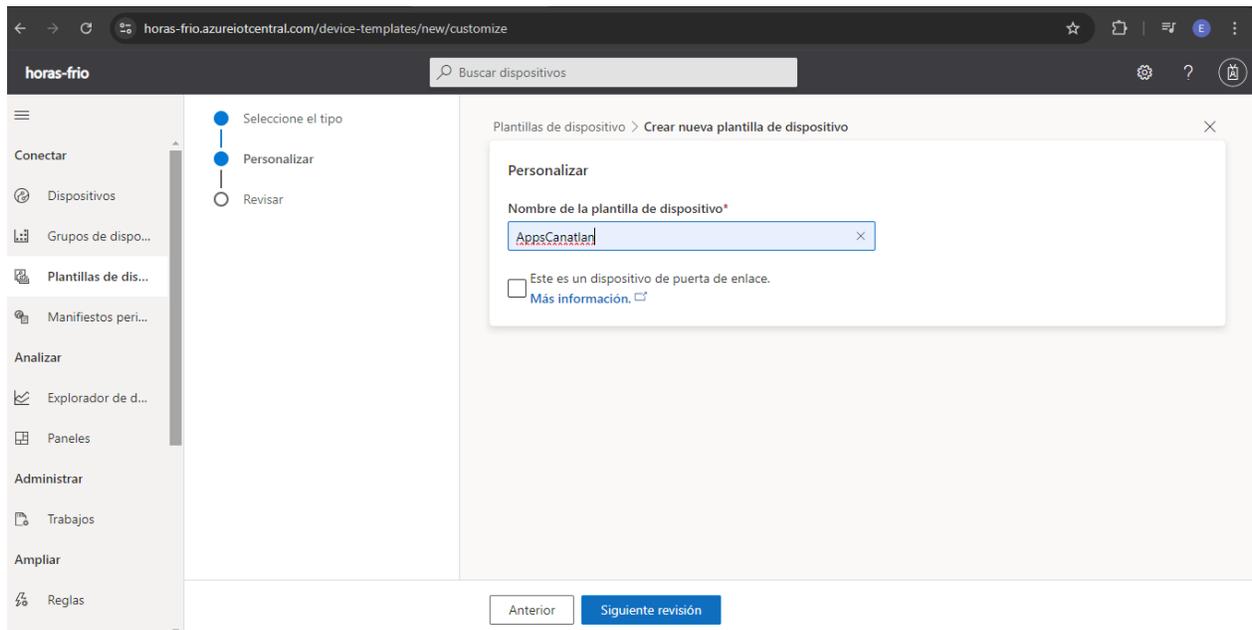


Figura 38

Selección de un modelo personalizado

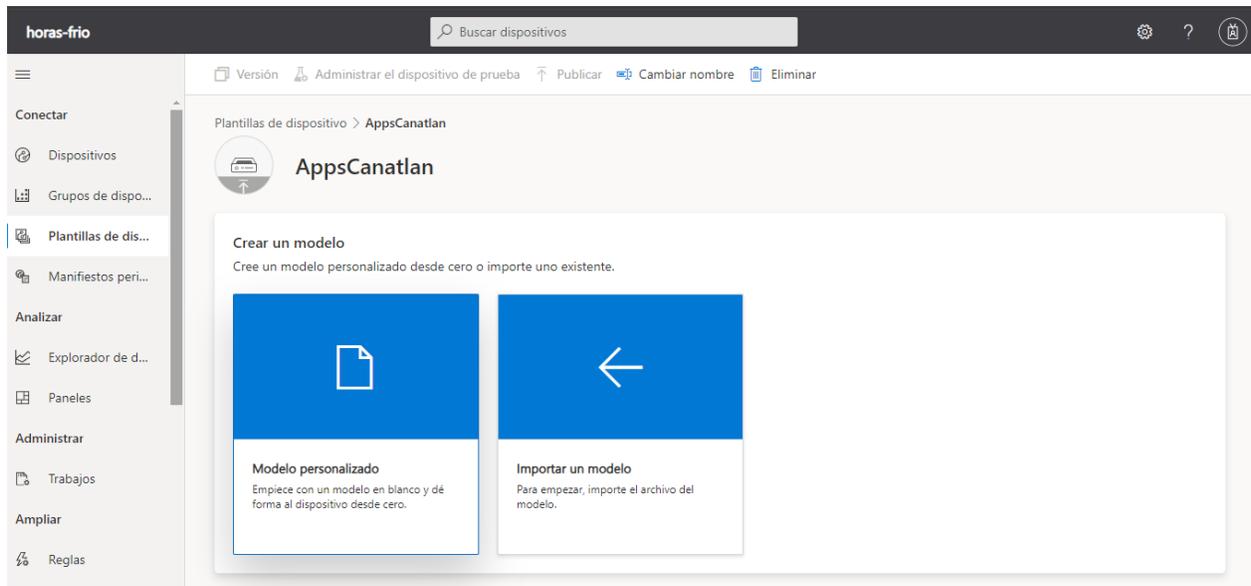


Figura 39

Asignación de funcionalidades para la recepción de telemetría

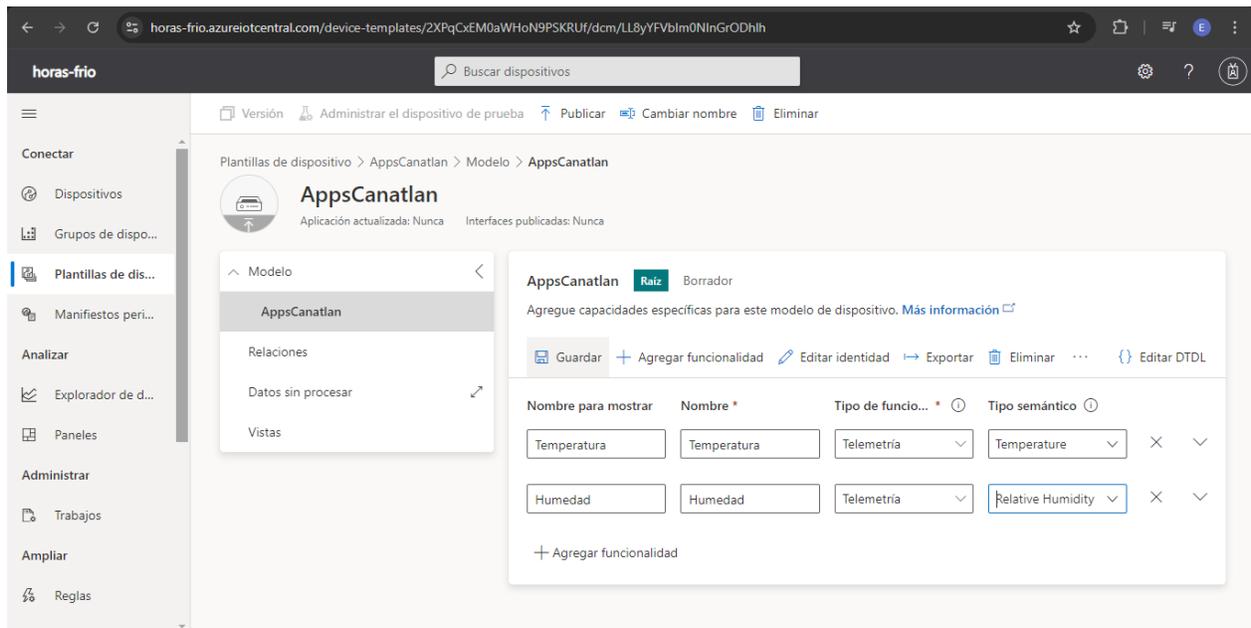


Figura 40

Selección de vistas predeterminadas

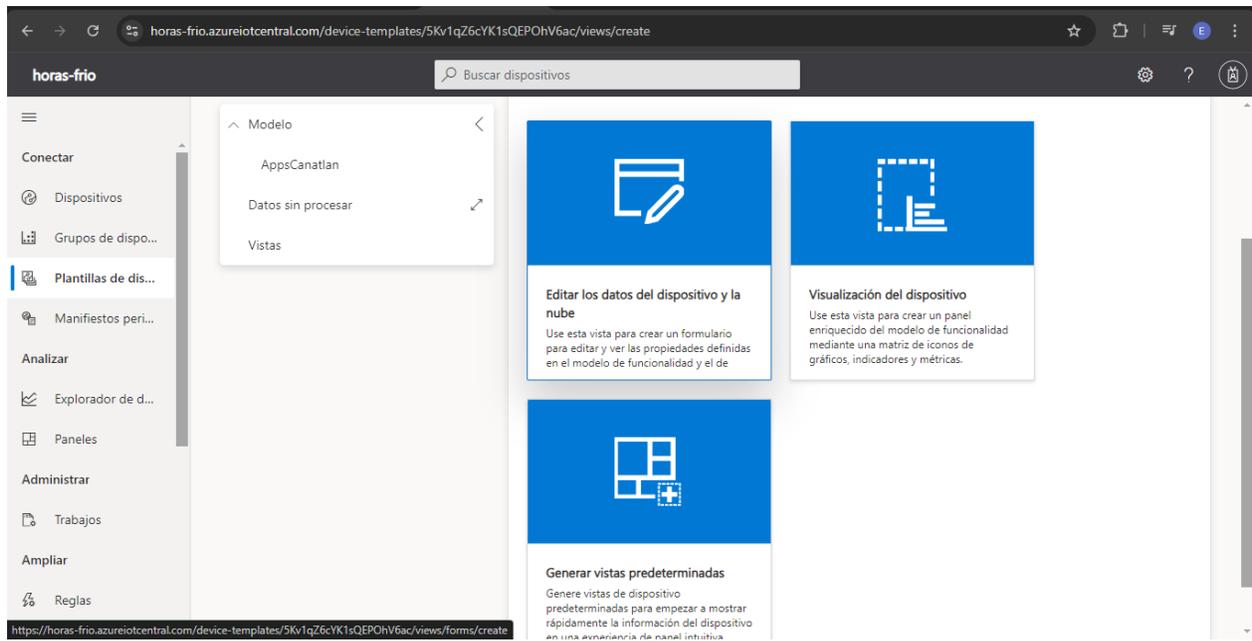


Figura 41

Generación de los elementos visuales para la exhibición de las variables de telemetría

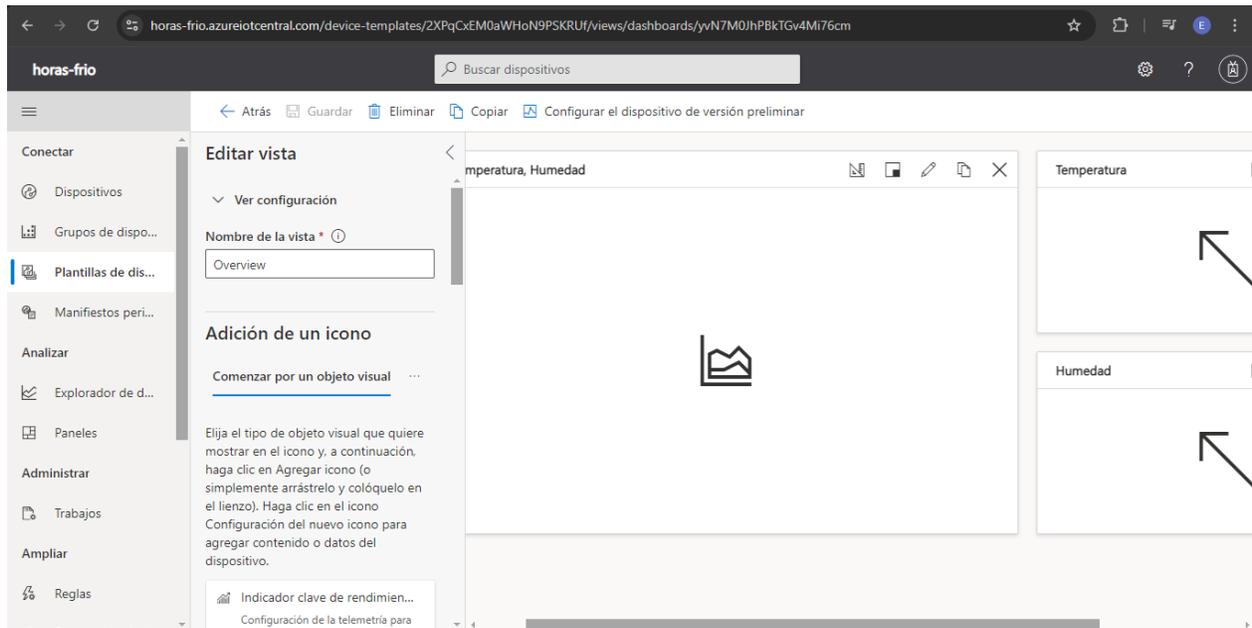


Figura 42

Publicación de la plantilla creada

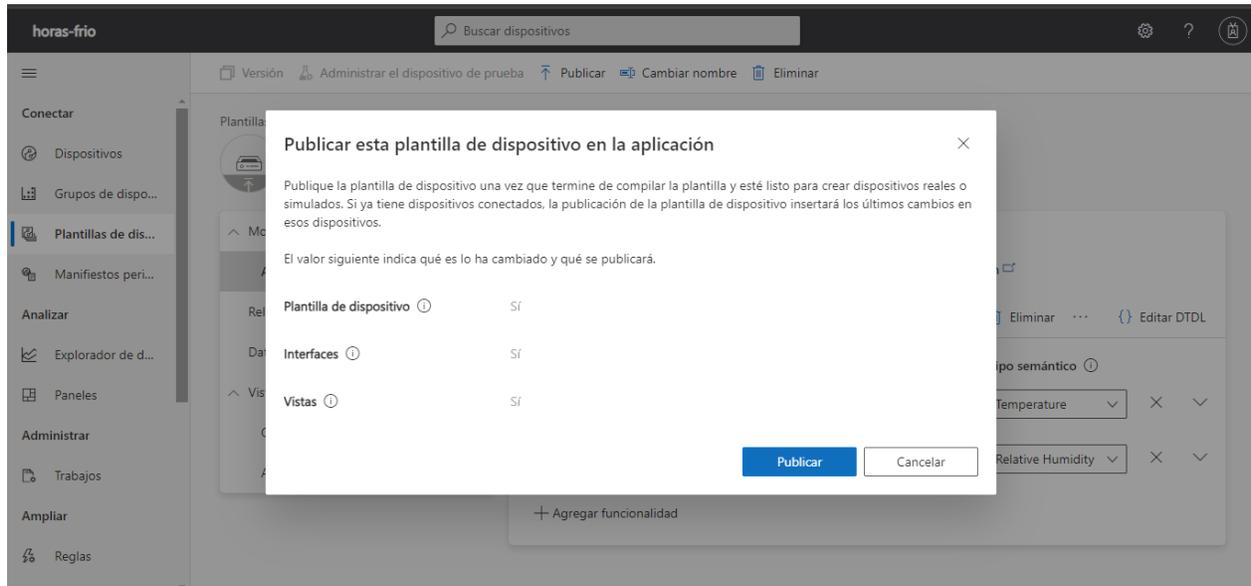
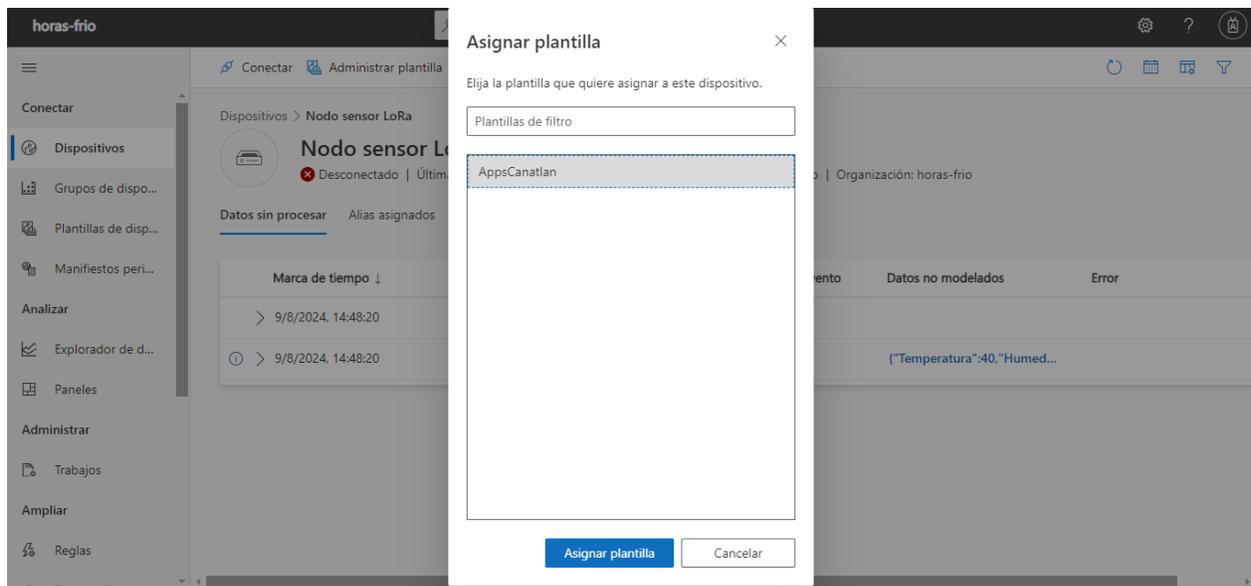


Figura 43

Asignación de plantilla al dispositivo



Los datos de prueba recopilados que se encuentran asociados a una funcionalidad de la aplicación se exhiben de forma tabular y grafica como se aprecia en las figuras 44-45, respectivamente.

Figura 44

Vista de la plataforma para los datos sin procesar con la funcionalidad asignada

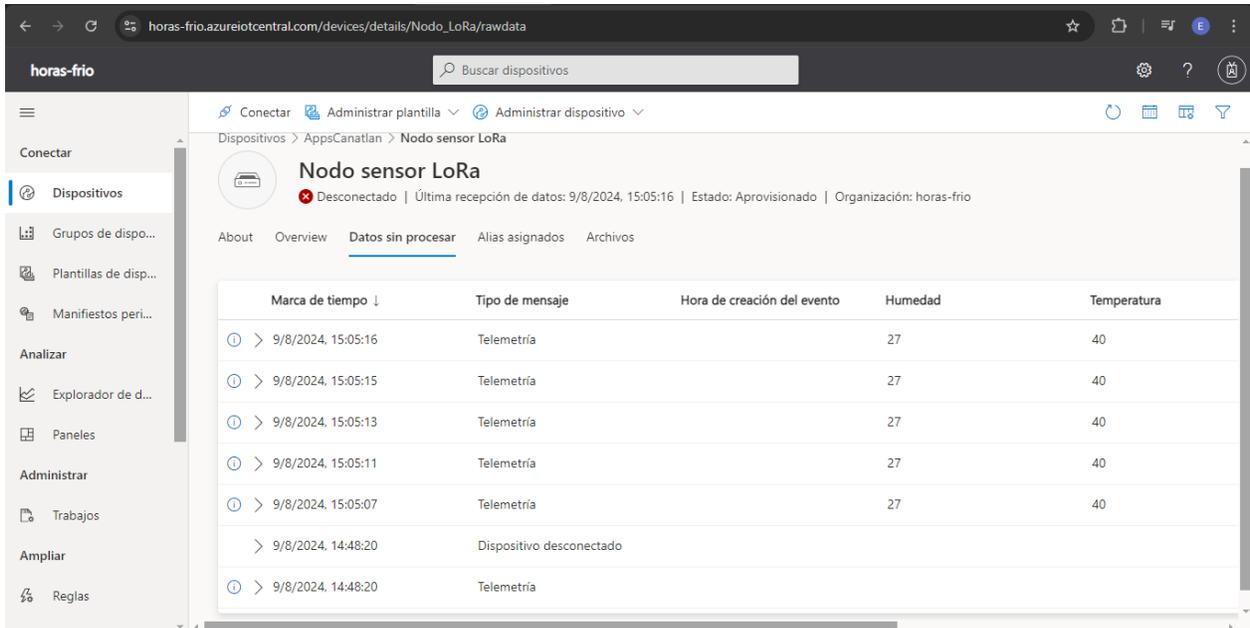
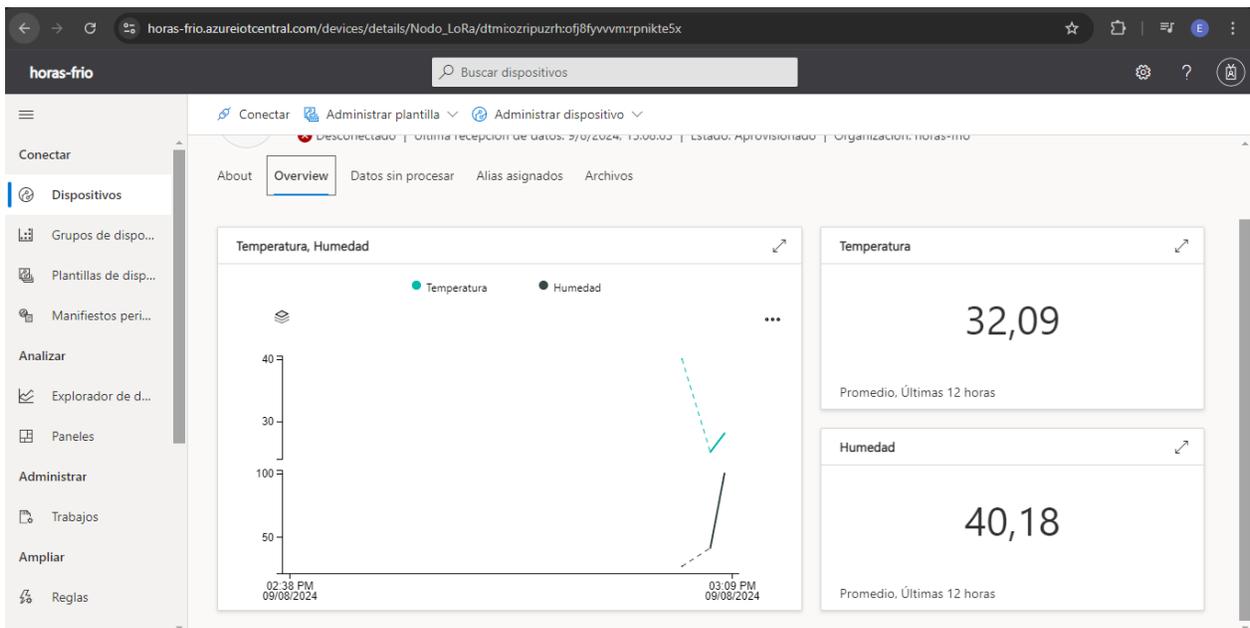


Figura 45

Vista grafica de los datos tipo test recibidos

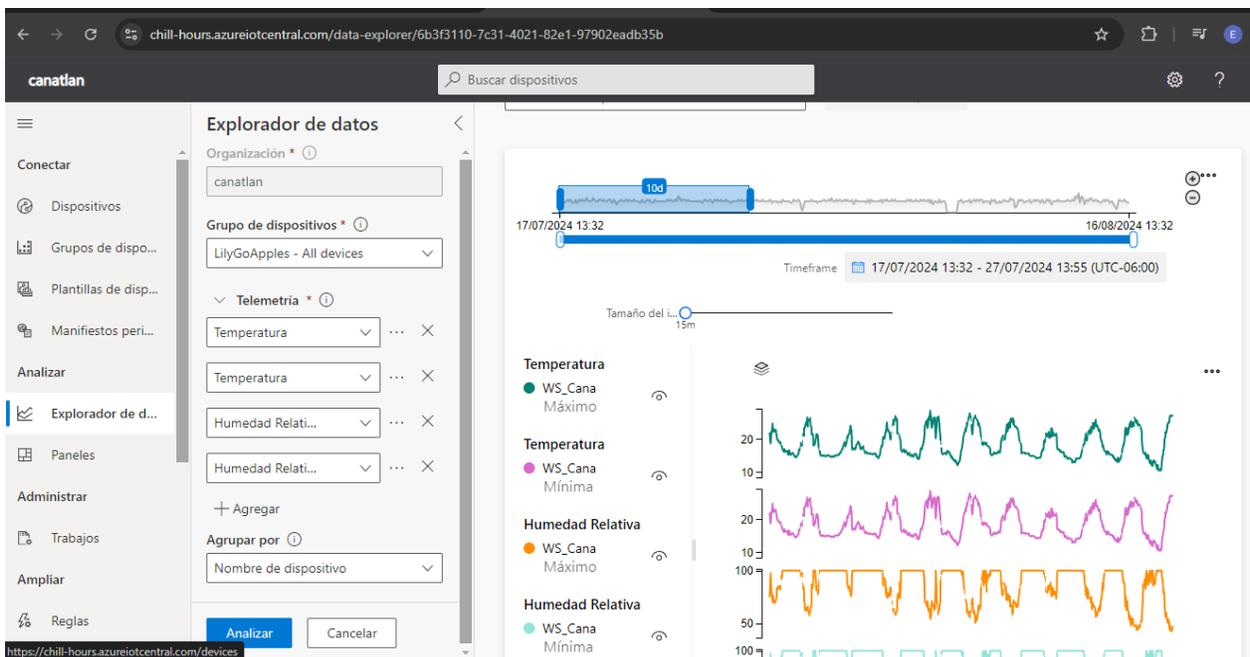


3.2 Adquisición y Depuración De Registros Recuperados

Mediante la funcionalidad del explorador de datos de la plataforma Azure IoT Central[®] se obtuvieron los registros que representan los datos enviados por el nodo sensor LoRa en periodos quinceminutales, figura 46.

Figura 46

Adquisición de registros mediante explorador de datos



Los datos obtenidos en formato CSV desde la plataforma Azure IoT Central fueron procesados mediante el programa Excel, figura 47, los espacios en blanco representan las mediciones faltantes y su estimación se realizó con el promedio de los datos adyacentes, debido principalmente a la baja tasa de cambio en la temperatura durante la noche y que las mediciones faltantes no exceden los 30 minutos y a expensas de un evento extraordinario la probabilidad de que el valor desconocido se encuentre lejos de la estimación es casi nula.

Figura 47

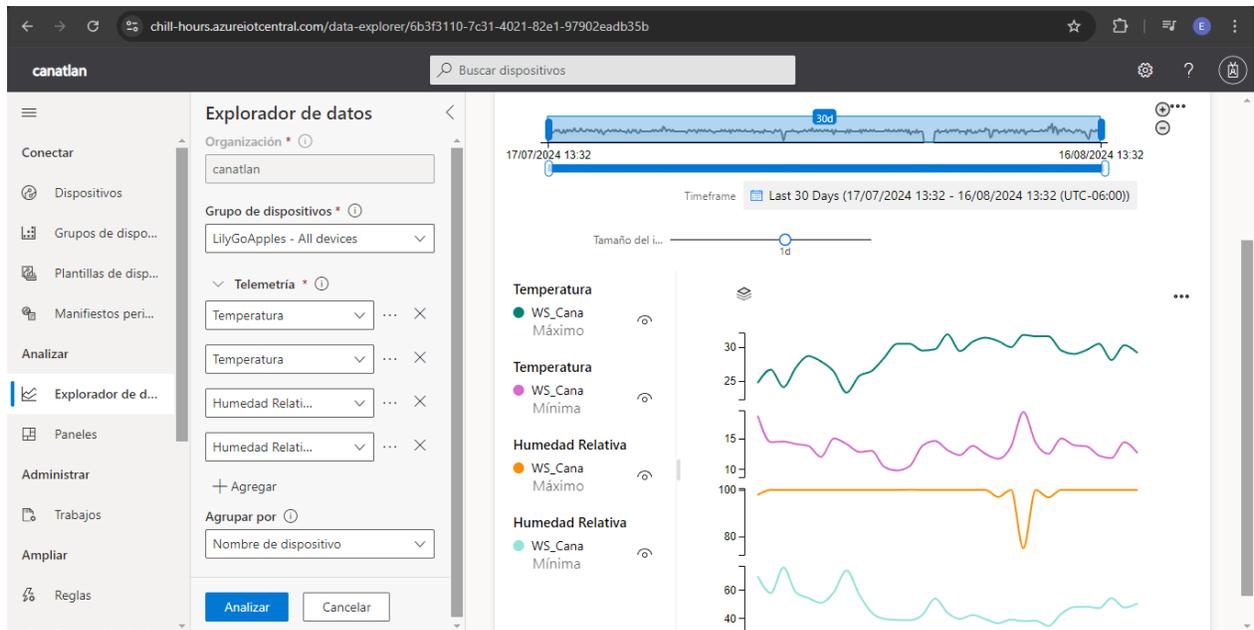
Imputación de registros faltantes

= SI(ESBLANCO(ED14), (ED13+ED15)/2, "OK.")					
EC	ED	EE	EF	EG	EH
Interval	Temperatura	Compl.Prom	Interval	Temperatura	Compl.Prom
29/04/2024 00:00	17.3	OK.	30/04/2024 00:00	19.4	OK.
29/04/2024 00:15	17.5	OK.	30/04/2024 00:15	19.1	OK.
29/04/2024 00:30	17.6	OK.	30/04/2024 00:30	15.5	OK.
29/04/2024 00:45	16.9	OK.	30/04/2024 00:45	13.9	OK.
	16.85	OK.	30/04/2024 01:00	13	OK.
29/04/2024 01:15	16.8	OK.	30/04/2024 01:15	11.1	OK.
29/04/2024 01:30	15.8	OK.	30/04/2024 01:30	11.6	OK.
29/04/2024 01:45	12.1	OK.	30/04/2024 01:45	12.3	OK.
29/04/2024 02:00	11.6	OK.			11.
29/04/2024 02:15	11.2	OK.	30/04/2024 02:15	11.1	OK.
29/04/2024 02:30	11.3	OK.	30/04/2024 02:30	9.9	OK.
29/04/2024 02:45	10.1	OK.	30/04/2024 02:45	11.5	OK.
		= SI(ESBLANCO(ED14), (ED13+ED15)/2, "OK.")			
29/04/2024 03:15	10.5	OK.			
29/04/2024 03:30	10.7	OK.	30/04/2024 03:30	8	OK.
29/04/2024 03:45	7.7	OK.	30/04/2024 03:45	7.7	OK.
29/04/2024 04:00	8	OK.	30/04/2024 04:00	7.7	OK.
		7.8	30/04/2024 04:15	6.6	OK.
29/04/2024 04:30	7.6	OK.	30/04/2024 04:30	5.9	OK.
29/04/2024 04:45	7.2	OK.	30/04/2024 04:45	6.7	OK.
29/04/2024 05:00	6.2	OK.	30/04/2024 05:00	6.1	OK.
29/04/2024 05:15	5.5	OK.	30/04/2024 05:15	5.5	OK.

Además de los registros quinceminutales, por medio de la funcionalidad del explorador de datos se obtuvieron las mediciones máximas y mínimas diarias de la temperatura y la humedad relativa del aire, figura 48.

Figura 48

Valores máximos y mínimos diarios de la temperatura y humedad relativa del aire



3.3 Cuantificación De Horas-Frío

La cuantificación diaria de horas-frío se realizó mediante los registros de temperatura y humedad relativa medidos en periodos de 15 min y la ecuación (1):

$$\text{Horas Frio (diarias)} = (S - I) * f \quad (1)$$

Donde:

S = Numero de registros que se encuentran por debajo de los 7.2 °C.

I = Numero de registros que se encuentran por debajo de los 2.5 °C.

f = frecuencia de medición en horas.

El valor obtenido tras la aplicación de la ecuación (1) se redondea al valor entero más cercano. Un factor importante para la correcta aplicación de la fórmula es que los registros deben estar completos al menos durante el periodo de tiempo donde se registran las temperaturas que se encuentran dentro del rango de interés.

3.4 Preprocesamiento y Exportación De Datos A La Plataforma Edge Impulse®

Antes de realizar la ingesta de los datos a la red neuronal es necesario que estos cumplan con una estructura y estén ordenados según lo establece la plataforma Edge Impulse® para su procesamiento, los datos recopilados anteriormente en la sección 3.2 y la metodología aplicada para la cuantificación de horas-frío de la sección 3.3 permitieron el acondicionamiento del conjunto de datos necesarios para realizar un entrenamiento supervisado, usando como variables predictoras los máximos y mínimos de temperatura y humedad relativa del aire para la predicción de la cantidad de horas-frío acumuladas diariamente.

Como parte de la normativa para la importación de los datos a la plataforma Edge impulse® fue necesaria la conversión de la etiqueta de tiempo a el formato de fecha y hora según la norma ISO 8601, la estructura para los datos de entrenamiento seleccionada se muestra en la tabla 9.

Tabla 9

Estructura para los datos de entrenamiento

Intervalo de tiempo	Temperatura máx.	Temperatura mín.	Humedad máx.	Humedad mín.	Horas-frío acumuladas
2024-03-11	25.6	2.5	77.4	17.6	7
2024-03-12	23.8	4.4	76.32	25.6	5
2024-03-13	25.5	1	71.8	16.9	8
2024-03-14	24.2	2	72.8	16.4	6
2024-03-15	20.8	7.3	43.8	15.7	1
2024-03-16	25.9	5.2	62.5	25.3	3
2024-03-17	26.55	3.05	64.15	17.45	6
2024-03-18	27.2	0.9	65.8	9.6	7
2024-03-19	24.6	7.6	51.8	8.3	0

La primera columna corresponde a la etiqueta del tiempo siguiendo la normativa antes mencionada, la segunda columna corresponde al valor máximo registrado de temperatura durante ese día, la

tercera columna corresponde al valor mínimo de temperatura registrado y la cuarta y quinta columna corresponden al valor máximo y mínimo de humedad relativa respectivamente, por último, la columna número 6 corresponde a la cantidad de horas-frío acumuladas durante ese día.

El procedimiento para la creación de un nuevo proyecto y la importación del dataset en la plataforma Edge Impulse® se muestra en las figuras 49-60.

Figura 49

Creación de un nuevo proyecto en Edge Impulse®

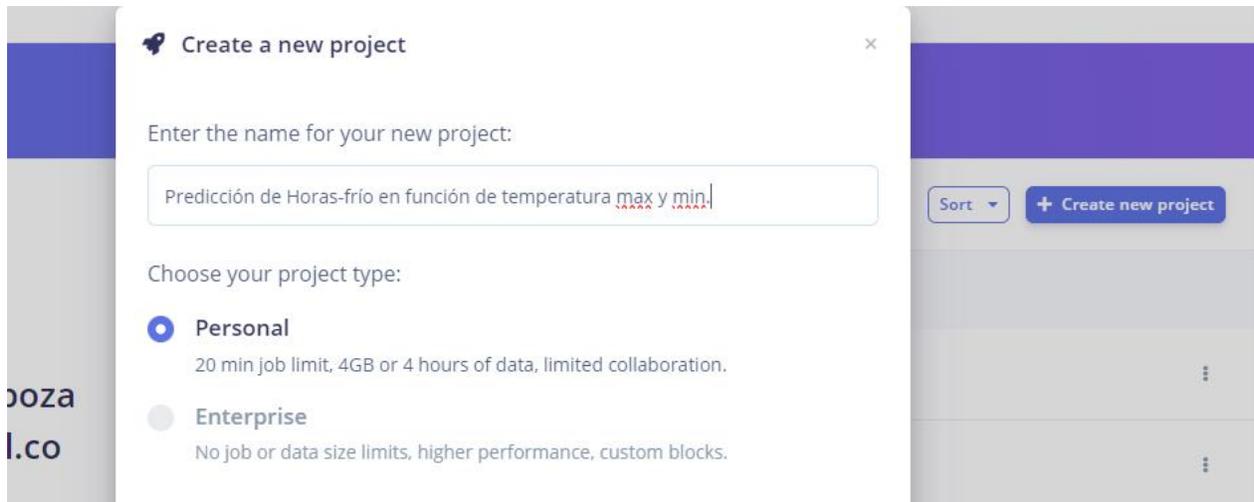


Figura 50

Construcción del dataset

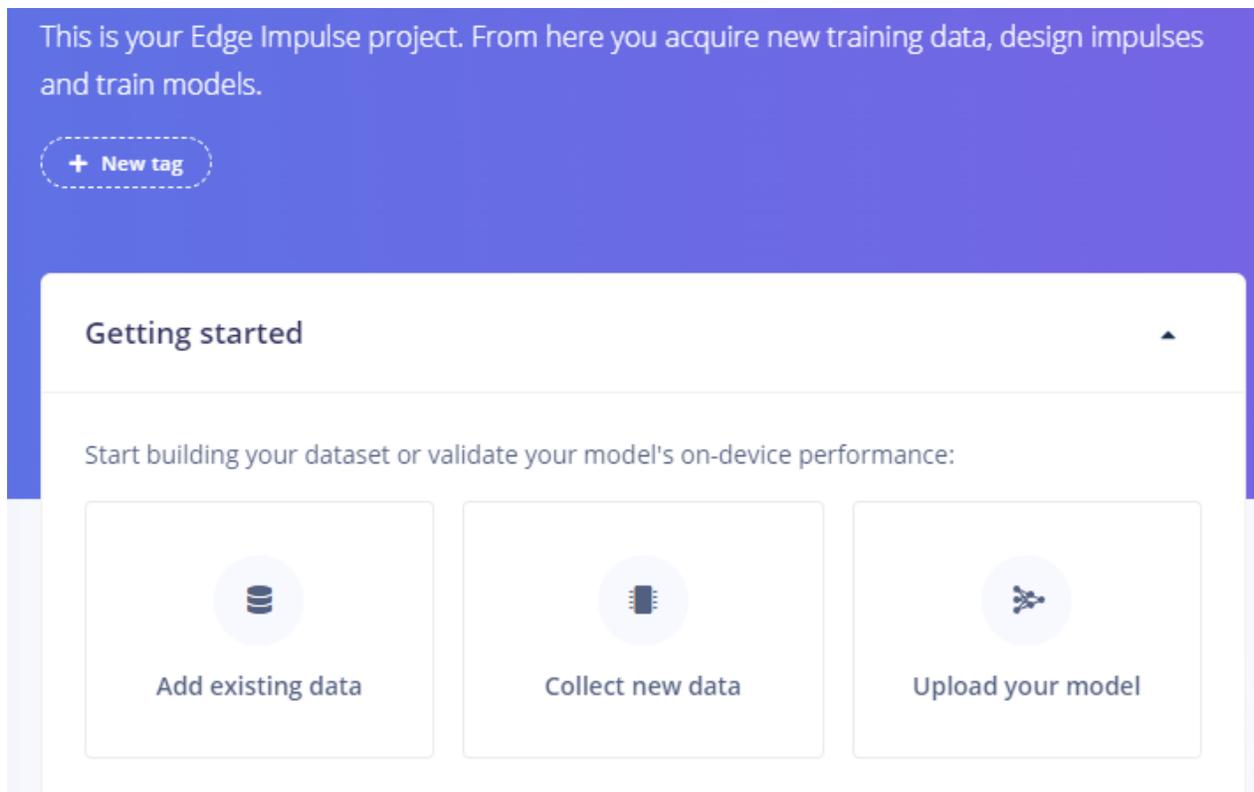


Figura 51

Carga de información del dataset

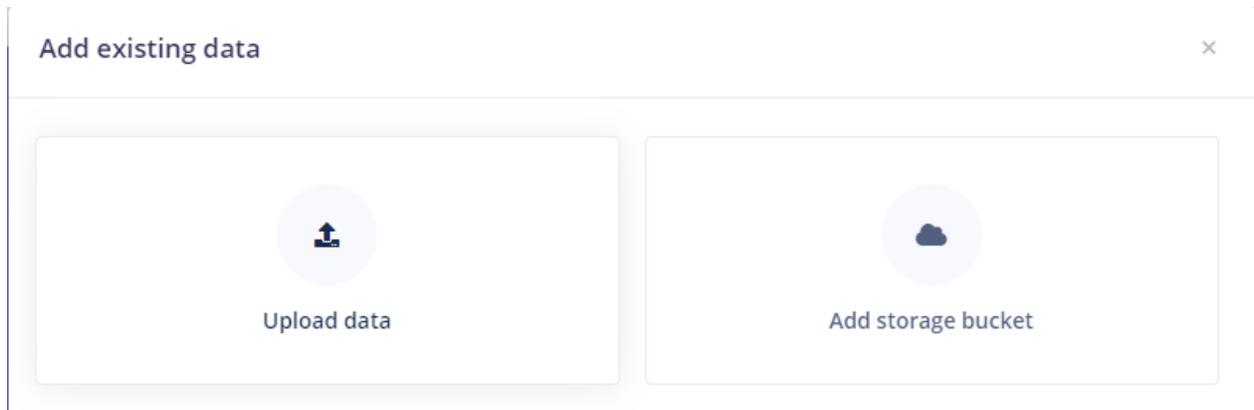


Figura 52

Inicio del asistente para la importación de datos

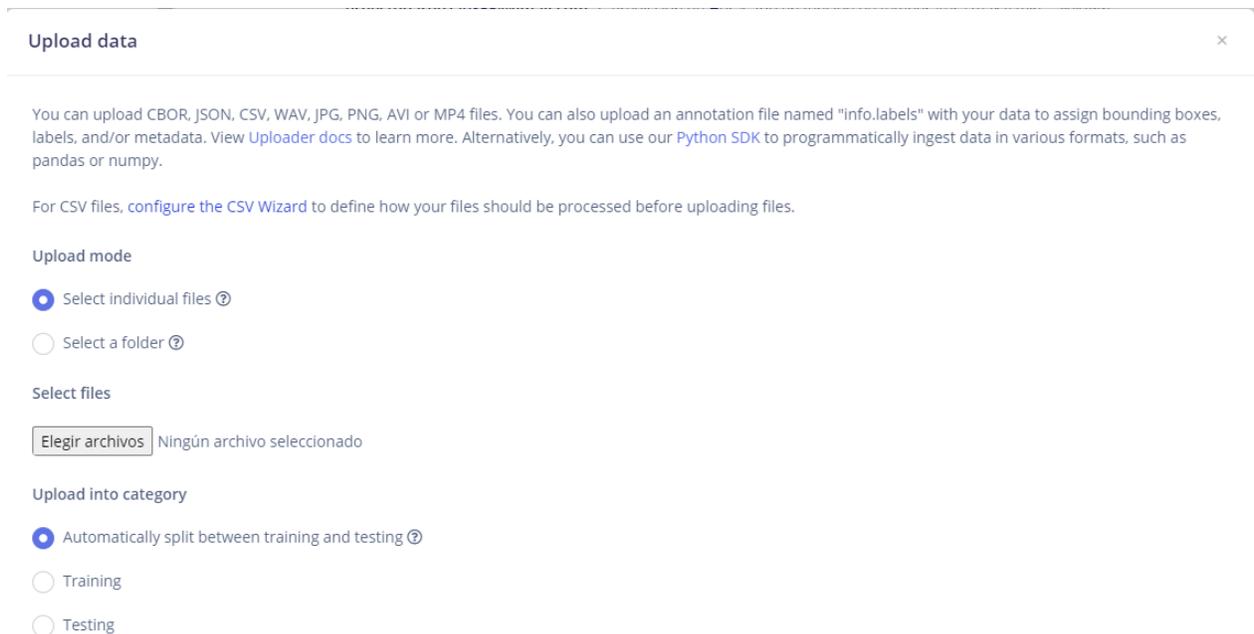


Figura 53

Selección de archivo de información para la configuración

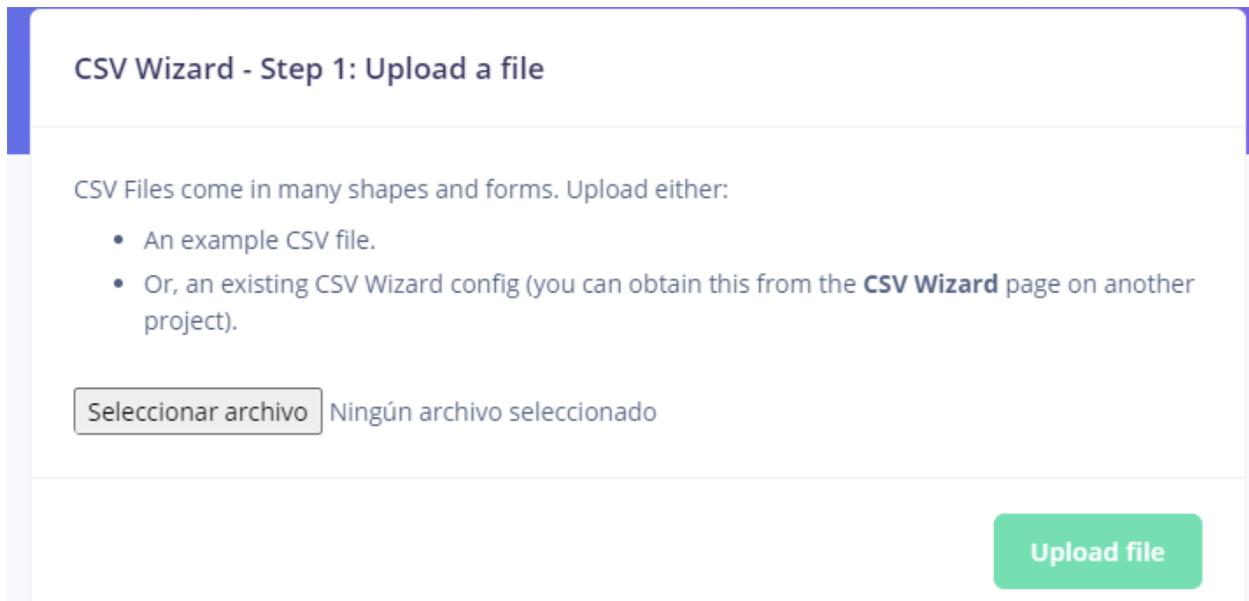


Figura 54

Selección del archivo almacenado en la PC

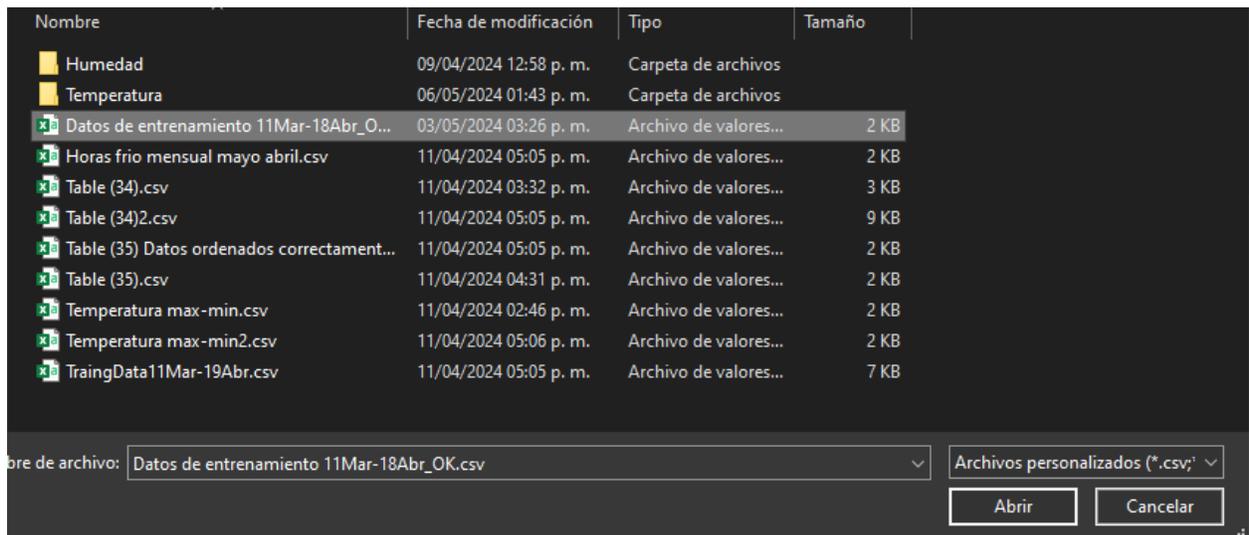


Figura 55

Vista preliminar del dataset cargado

Here are the first 200 lines of your CSV file. You should see clear tabular data with named columns.

Delimiter:

Skip first lines:

No header row:

ROW #	INTERVAL	TEMPERATURA/WS_CANAL MÁXIMO	TEMPERATURA/WS_CANAL MÍNIMA	HUMEDAD/WS_CANAL MÁXIMO	HUMEDAD/WS_CANAL MÍNIMA	HF
# 1	2024-03-11	25.6	2.5	77.4	17.6	7
# 2	2024-03-12	23.8	4.4	76.32	25.6	5
# 3	2024-03-13	25.5	1	71.8	16.9	8
# 4	2024-03-14	24.2	2	72.8	16.4	6
# 5	2024-03-15	20.8	7.3	43.8	15.7	1
# 6	2024-03-16	25.9	5.2	62.5	25.3	3
# 7	2024-03-17	26.55	3.05	64.15	17.45	6

[Different format?](#) Looks good, next

Figura 56

Asignación de características del dataset

CSV Wizard - Step 3: About your data

Let's learn a bit about the structure of your data.

Is this time-series data?

Yes, this is time-series data (either raw sensor data, or processed features)

No

How is your time-series data formatted?

Each row contains a reading, and sensor values are columns.

Example:

TIMESTAMP	TEMPERATURE	HUMIDITY
2022-11-24 21:59:00	28.04	9.81
2022-11-24 21:59:10	28.24	9.79
2022-11-24 21:59:20	28.31	9.61

Datos de entrenamiento 11Mar-18Abr_OK.csv

TIME (MS.)	TEMPERATURA/WS_CANAL MÁXIMO	TEMPERATURA/WS_CANAL MÍNIMA
0	25.6	2.5
86400000	23.8	4.4
172800000	25.5	1
259200000	24.2	2
345600000	20.8	7.3
432000000	25.9	5.2
518400000	26.55	3.05
604800000	27.2	0.9
691200000	24.6	7.6
777600000	24.9	13.5
864000000	24.3	13.9

Figura 57

Asignación de la columna de marca de tiempo y frecuencia

Do you have a timestamp or time elapsed column?

Yes, it's

No

What type of data is in your timestamp column?

Full timestamp

Time elapsed in seconds

Time elapsed in milliseconds

Detected frequency: 0.000011574074074074073 Hz.

[Different format?](#) Great, let's look at your values

Figura 58

Selección de variables predictoras y variable a predecir

CSV Wizard - Step 4: About your values

Almost there. Let's parse your values.

Do you have a column that contains the label (the value you want to predict)?

Yes, it's

No (no worries, you can provide this when you upload data)

Which columns contain your values?

Temperatura/WS_Cana.Máximo

Temperatura/WS_Cana.Mínima

Humedad/WS_Cana.Máximo

Humedad/WS_Cana.Mínima

[Different format?](#) Next, split up into samples

Example sample

TIME (MS.)	LABEL (HF)	TEMPERATURA/WS_CANA.MÁXIMO	TEMPERATUR
0	7	25.6	2.5
86400000	5	23.8	4.4
172800000	8	25.5	1
259200000	6	24.2	2
345600000	1	20.8	7.3
432000000	3	25.9	5.2
518400000	6	26.55	3.05

Sample graph



Figura 59

Asignación del tamaño de muestra

CSV Wizard - Step 5: Split up your samples

If you have a lot of data we can automatically split one CSV file up in multiple samples.

How long do you want your samples to be?

Unlimited

Limit to ms.

How should we deal with multiple labels in a sample?

The sample should have multiple labels ([Professional & enterprise only.](#))

Use the *last* value of "HF" as the label for each sample (see the table on the right).

[Different format?](#) Finish wizard

Example sample

TIME (MS.)	LABEL (HF)	TEMPERATURA/WS_CANA.MÁXIMO	TE
Sample 1 (label: 7)			
0	7	25.6	2.3
Sample 2 (label: 5)			
0	5	23.8	4.4
Sample 3 (label: 8)			
0	8	25.5	1
Sample 4 (label: 6)			

Sample graph

Figura 60

Finalización de configuración del asistente para la importación del dataset

CSV Wizard - Step 6: Finished

Any CSV files that you upload into your project - whether it's through the uploader, the CLI, the API or through data sources - will now be processed according to the rules you set up here.

Upload some data

3.5 Creación Del Modelo De Red Neuronal Artificial Mediante Edge Impulse®

Una vez importado el dataset mediante el asistente de la plataforma Edge Impulse® se creó una red de neuronal artificial mediante un enfoque visual como se muestra en las figuras 61-68, los parámetros de diseño y configuración se muestran en la tabla 10.

Tabla 10

Parámetros de diseño y configuración de red neuronal artificial

Parámetro/configuración	Valor	Descripción	Figura
Impulse	Time series data	La selección del impulso depende de la estructura y el tipo de datos, esto se define mediante la configuración del asistente para la importación de datos.	61
Processing block	Raw data	Permite obtener información de datos sin procesar que mejora el desempeño de modelos de aprendizaje profundo	62
Learning block	Regression	Este bloque permite la selección del modelo de regresión lineal de Keras importado desde Tensor Flow®	63
Windows size	86,400,000 ms	Selecciona el tamaño de ventana	64
Windows increase	1000 ms	Selecciona el incremento de ventana	64
Frequency	0.0000115740 Hz	Selección de frecuencia	64
Generate features	Default	Los parámetros preestablecidos son usados para generar	65

		características asociadas a los datos	
Number of training cycles	100	Selección de un valor entero que representa el número de ciclos de entrenamiento	66
Learning rate	0.005	Permite la selección de la tasa de aprendizaje	66
Optimizer	Adam	Selección de optimizador Adam	67
Activation function	Relu	Selección de función de activación tipo Relu	67
Hidden layer 1	20	Primera capa oculta de 20 neuronas	68
Hidden layer 2	10	Segunda capa oculta con diez neuronas	68
Output layer	1	Capa de salida que permite la obtención del valor estimado	68

Figura 61

Creación del impulso

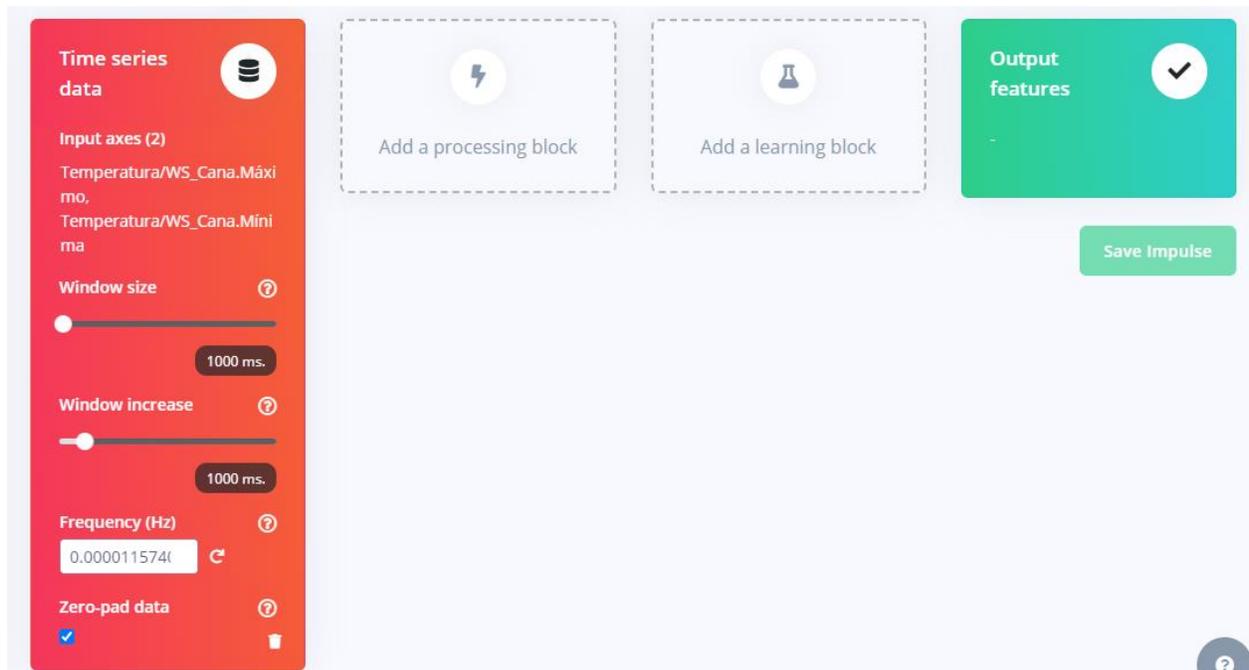


Figura 62

Selección de bloque de preprocesamiento de datos en forma lineal

Spectral Analysis OFFICIALLY SUPPORTED Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.	Edge Impulse	Add
Spectrogram OFFICIALLY SUPPORTED Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies.	Edge Impulse	Add
IMU (Syntiant) OFFICIALLY SUPPORTED Syntiant only. Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.	Syntiant	Add
Raw Data OFFICIALLY SUPPORTED Use data without pre-processing. Useful if you want to use deep learning to learn features.	Edge Impulse	Add

Figura 63

Selección de bloque de regresión lineal

Add a learning block

Did you know? You can [bring your own model](#) in PyTorch, Keras or scikit-learn.

DESCRIPTION	AUTHOR	RECOMMENDED
Regression OFFICIALLY SUPPORTED Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.	Edge Impulse	★ Add
Classification OFFICIALLY SUPPORTED Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.	Edge Impulse	Add
Anomaly Detection (GMM) PROFESSIONAL ENTERPRISE Find outliers in new data. A Gaussian mixture model (GMM) models the shape of data using a probability distribution. New data that is unlikely according to this model can be considered anomalous.	Edge Impulse	

Figura 64

Selección de frecuencia de muestreo

Time series data

Input axes (2)
Temperatura/WS_Cana.Máximo,
Temperatura/WS_Cana.Mínima

Window size 86.400.000 ms.

Window increase 1000 ms.

Frequency (Hz) 0.000011574

Raw Data

Name Raw data

Input axes (2)
 Temperatura/WS_Cana.Máximo
 Temperatura/WS_Cana.Mínima

Regression

Name Regression

Input features
 Raw data

Output features
1 (Scalar value 0...8)

Output features

1 (Scalar value 0...8)

Save Impulse

Figura 65

Generación de características del dataset

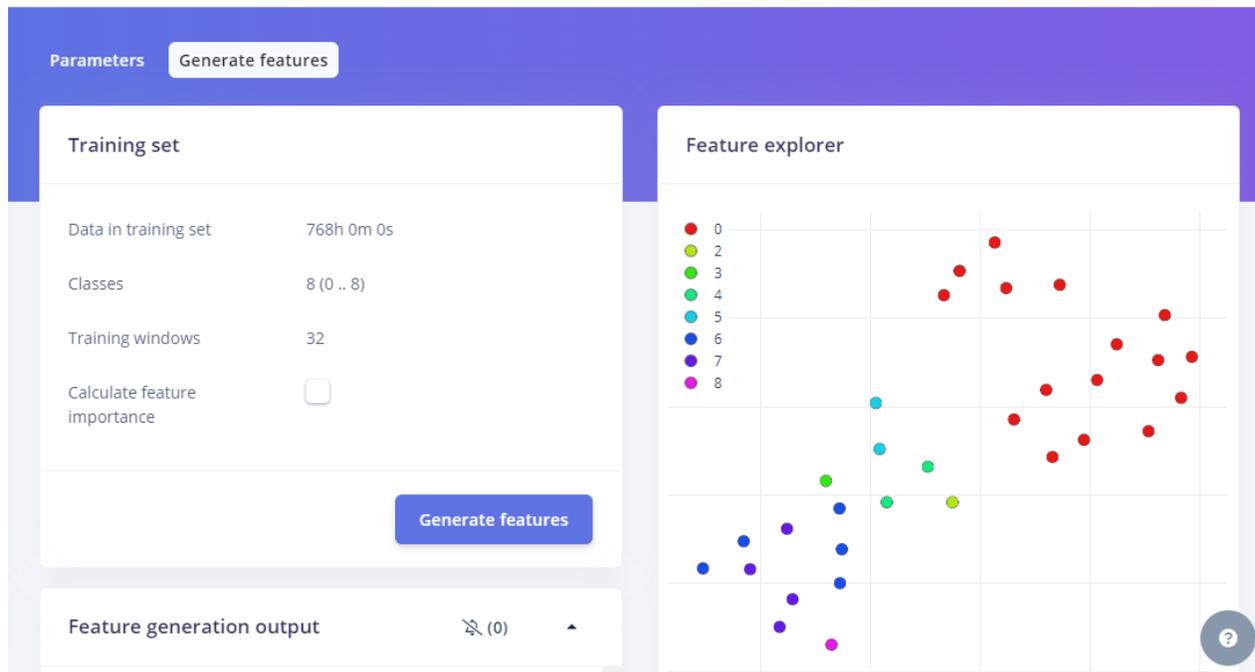


Figura 66

Selección de los parámetros de entrenamiento

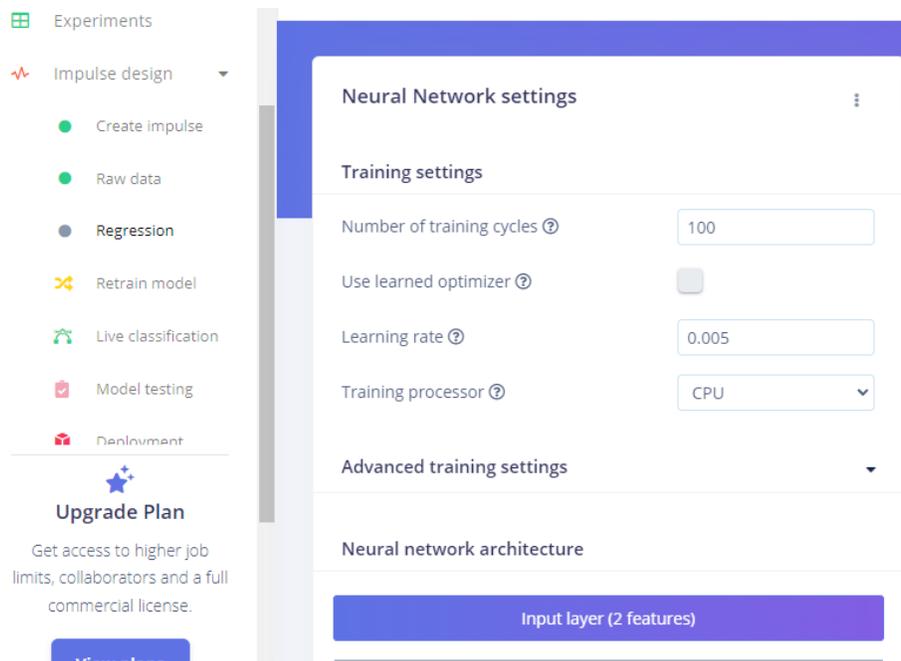


Figura 67

Visualización en modo experto para la selección de la función de activación

The screenshot shows a web interface for configuring a neural network. On the left is a sidebar with navigation options: Dashboard, Devices, Data acquisition, Experiments, and Impulse design (with sub-options: Create impulse, Raw data). Below the sidebar is an 'Upgrade Plan' section. The main area is titled 'Neural network architecture' and contains a code editor with the following Python code:

```
6 EPOCHS = args.epochs or 100
7 LEARNING_RATE = args.learning_rate or 0.005
8 # If True, non-deterministic functions (e.g. shuffling
9 # batches) are not used.
10 # This is False by default.
11 ENSURE_DETERMINISM = args.ensure_determinism
12 # this controls the batch size, or you can manipulate the
13 # tf.data.Dataset objects yourself
14 BATCH_SIZE = args.batch_size or 32
15 if not ENSURE_DETERMINISM:
16     train_dataset = train_dataset.shuffle(buffer_size
17     =BATCH_SIZE*4)
18 train_dataset=train_dataset.batch(BATCH_SIZE,
19 drop_remainder=False)
20 validation_dataset = validation_dataset.batch(BATCH_SIZE,
21 drop_remainder=False)
22
23 # model architecture
24 model = Sequential()
25 model.add(Dense(20, activation='relu'))
26 model.add(Dense(10, activation='relu'))
27 model.add(Dense(classes, name='y_pred'))
28
29 # this controls the learning rate
30 opt = Adam(learning_rate=LEARNING_RATE, beta_1=0.9,
31 beta_2=0.999)
32 callbacks.append(BatchLoggerCallback(BATCH_SIZE,
33 train_sample_count, epochs=EPOCHS, ensure_determinism
34 =ENSURE_DETERMINISM))
35
36 # train the neural network
37 model.compile(loss='mean_squared_error', optimizer=opt,
38 metrics=None)
39 model.fit(train_dataset, epochs=EPOCHS, validation_data
40 =validation_dataset, verbose=2, callbacks=callbacks)
41
```

Figura 68

Selección de la estructura de la red neuronal artificial

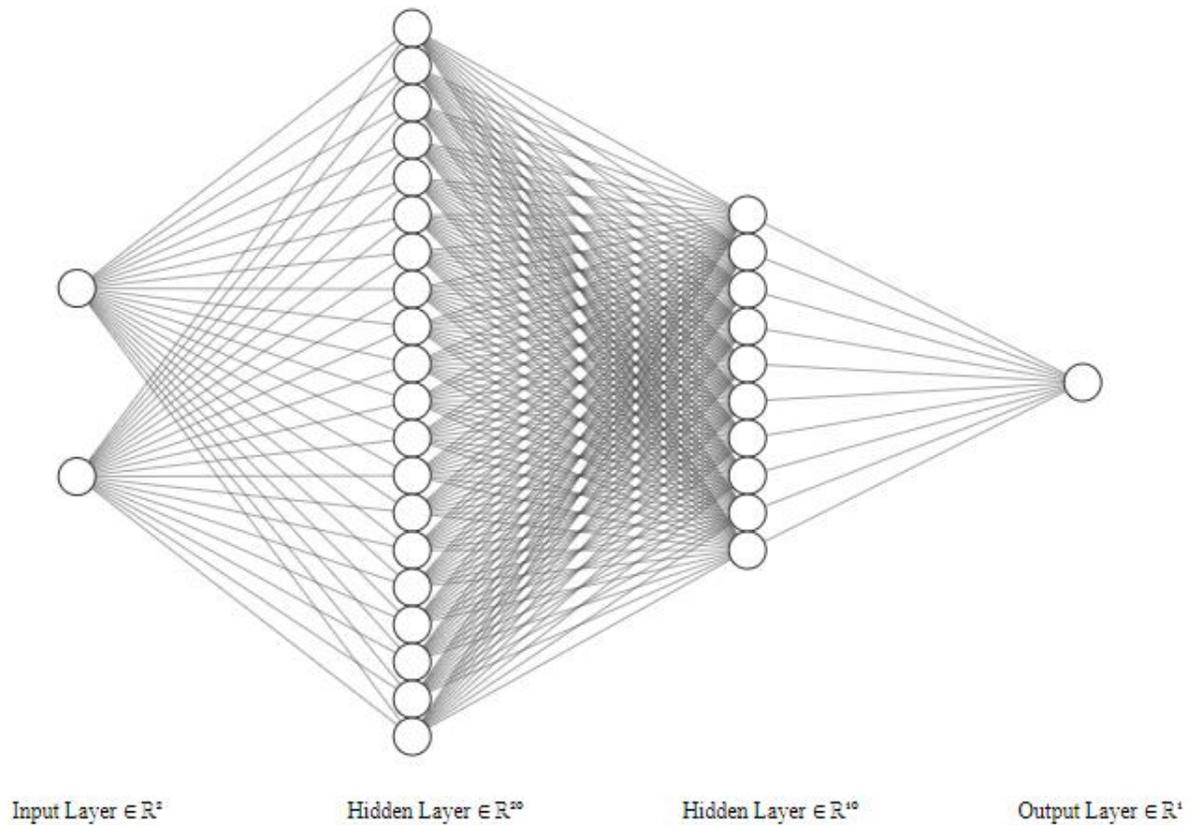
The screenshot shows a web interface for configuring a neural network. On the left is a sidebar with navigation options: Impulse design (with sub-options: Create impulse, Raw data, Regression, Retrain model, Live classification, Model testing, Deployment) and Upgrade Plan. The main area is titled 'Advanced training settings' and contains a section for 'Neural network architecture'. The architecture is visualized as a sequence of layers:

- Input layer (2 features)
- Dense layer (20 neurons)
- Dense layer (10 neurons)
- Add an extra layer (dashed box)
- Output layer (1 value)

A 'Start training' button is located at the bottom of the architecture section.

Figura 69

Esquema de la estructura de la red neuronal artificial seleccionada



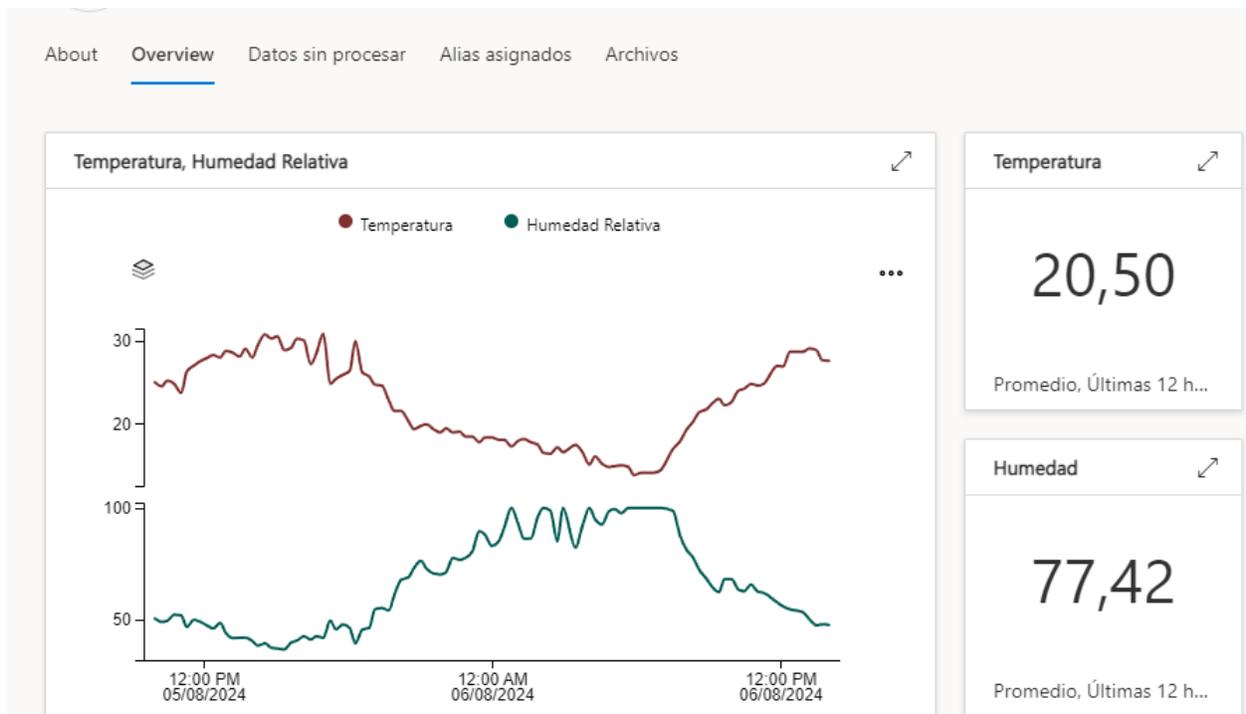
El esquema general de la red neuronal artificial para la cuantificación de horas-frío de la figura 69 consta de cuatro capas, una capa de entrada con dos variables predictoras que corresponden a la temperatura máxima y mínima de cada día, dos capas ocultas con 20 y 10 neuronas respectivamente y la capa de salida donde se obtiene el resultado de la predicción. Una vez que se determinó que la red neuronal de cuatro variables obtuvo mejores resultados de estimación en comparación con la de dos variables, se entrenaron otras redes usando la misma estructura, pero usando los máximos y mínimos de las variables predictoras de humedad relativa del aire en conjunto con las de la temperatura, es decir, la estructura de la red permaneció casi idéntica con la única excepción de que la capa de entrada estaba conformada por cuatro variables en lugar de dos.

Capítulo 4 Resultados y Discusión

Se obtuvo una herramienta tecnológica para la adquisición y monitoreo de variables climáticas que satisface las necesidades básicas de energía y comunicación dentro del sitio de estudio, operante bajo los protocolos de comunicación LoRa y GSM, el sistema de medición permite la obtención de información de temperatura y humedad relativa del aire en periodos quince minútales para la derivación de la cantidad de horas-frío acumuladas, con la información recopilada por la estación meteorológica se creó el set de datos para el entrenamiento supervisado de una red neuronal artificial que considera los registros máximos y mínimos diarios para la estimación de la cantidad de horas-frío, figura 70.

Figura 70

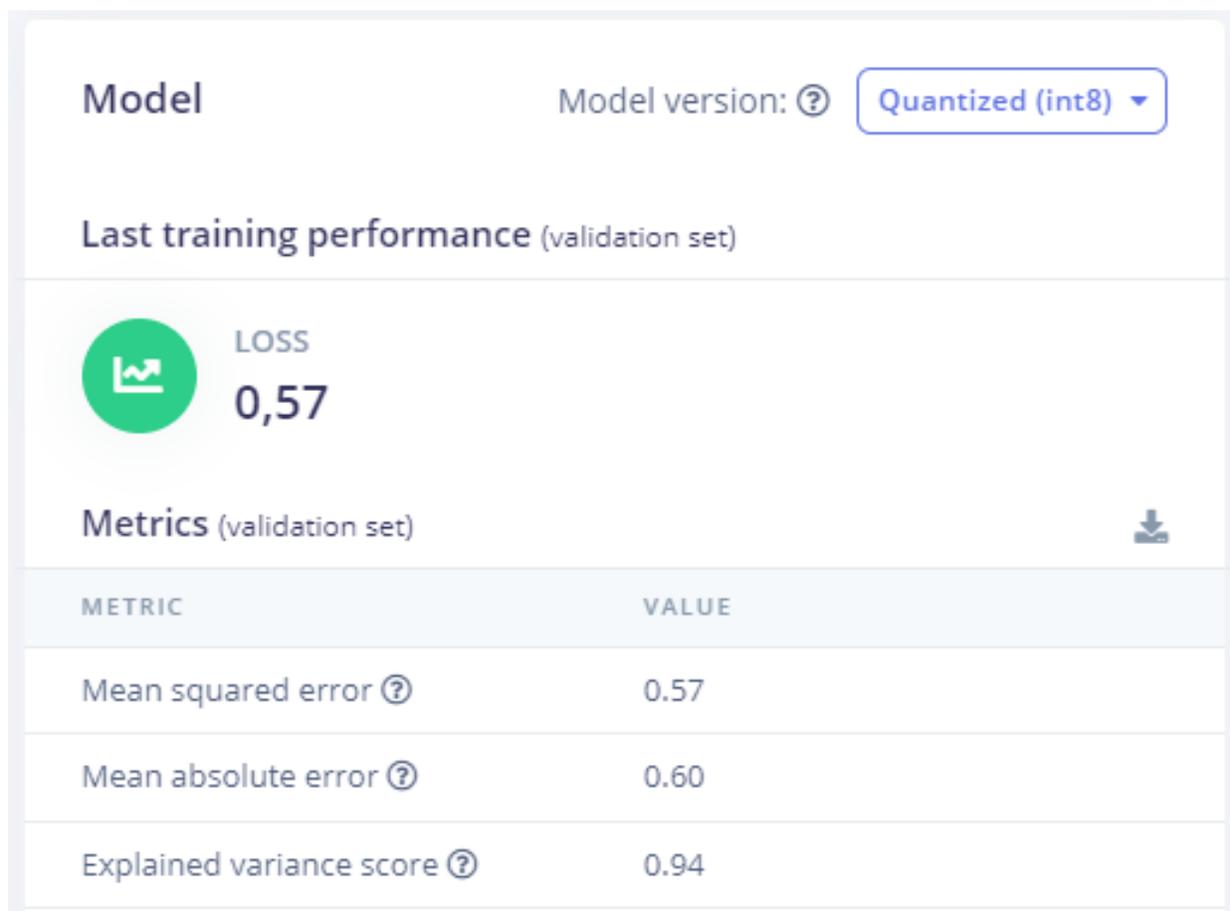
Sistema de adquisición y monitoreo de datos climatológicos en Azure IoT Central



Según la metodología propuesta en el capítulo 3 sección 3.5, se entrenó una red neuronal artificial usando como variables de entrada los valores máximos y mínimos de la temperatura del aire y una segunda red añadiendo los valores máximos y mínimos de la humedad relativa del aire, los resultados obtenidos se muestran a continuación:

Figura 71

Métricas de desempeño de la red neuronal de dos variables (validación)



En la figura 71, se observan las métricas de desempeño del modelo de red neuronal artificial en la etapa de validación, que considera los valores máximos y mínimos de temperatura del aire para la estimación de la cantidad de horas-frio acumuladas diariamente.

Figura 72

Análisis visual del desempeño de la red de dos variables (validación)



Los resultados de la clasificación en la etapa de validación se muestran con puntos verdes para aquellas estimaciones dentro del rango de error permitido (0.8 unidades) y con color rojo los resultados que sobrepasan el margen de error, figura 72.

En la figura 73, se muestran las métricas de desempeño de ambas redes para la predicción de la cantidad de horas-frío acumuladas diariamente durante el periodo del 11 de marzo al 11 de abril del año 2024, en la tabla 11, se muestra una comparación en la configuración y el desempeño obtenido.

Figura 73

Comparación de redes mediante las métricas de desempeño

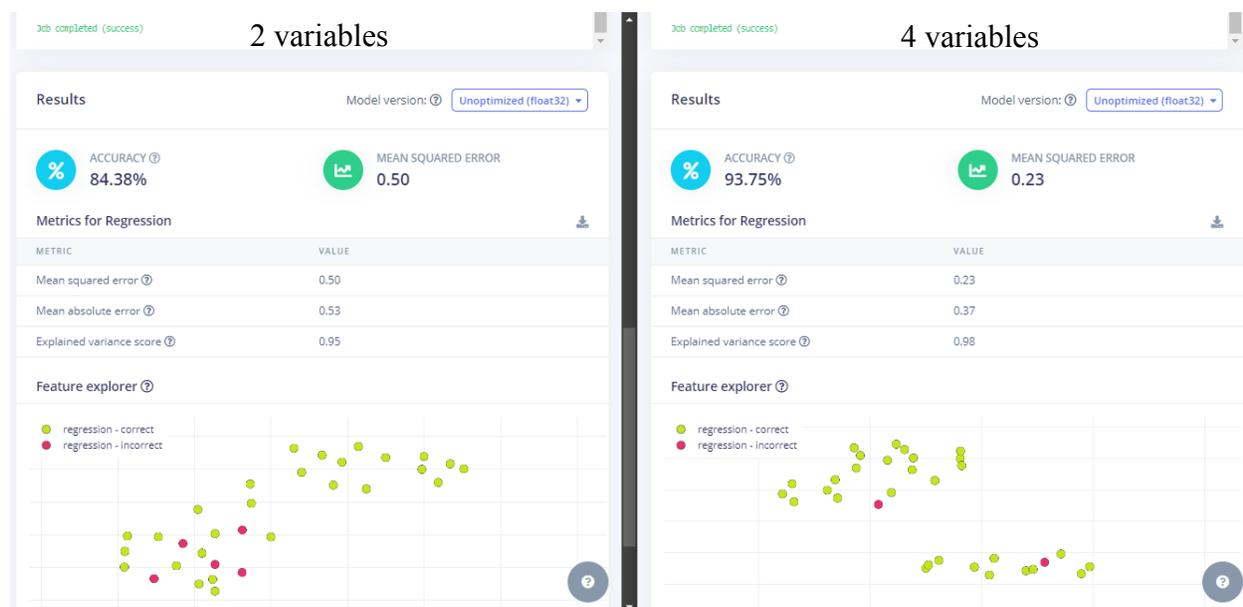


Tabla 11*Comparación del desempeño/configuración de redes neuronales de dos y cuatro variables*

Parámetros de red	2 variables	4 variables
Variables predictoras	Temperatura máx. y mín.	Temperatura máx. y mín. Humedad máx. y mín.
Impulso	Datos de series de tiempo	Datos de series de tiempo
Bloque de procesamiento	Raw data	Raw data
Bloque de aprendizaje	Regression	Regression
Ciclos de entrenamiento	800	800
Tasa de aprendizaje	0.005	0.005
Optimizador	Adam	Adam
Función de activación	Relu	Relu
Perdida (set de validación)	0.54	0.37
Estructura de red Tipo de capa: Número de neuronas	CE:2 CO1:20 CO2:10 CS:1	CE:2 CO1:20 CO2:10 CS:1
Precisión	84.38%	93.75%
MSE	0.50	0.23
MAE	0.53	0.37

Debido a que tras la revisión del estado del arte se observó una ambigüedad en el concepto de horas-frío, se entrenaron dos redes más de cuatro variables predictoras para la estimación de horas-frío bajo los criterios de discriminación de temperaturas fuera del rango de 2.5-7°C y de 0-10°C respectivamente, los resultados obtenidos y una comparación con el desempeño del modelo que considera solo temperaturas inferiores a 7°C se muestra en la tabla 12.

Tabla 12

Aproximaciones a la cantidad de horas-frío acumuladas bajo los diferentes criterios

Temperaturas consideradas	Precisión	MSE	MSA	Acumulación por cuantificación de registros	Obtenido por sumatoria de predicciones
Inferiores a 7.2°C	93.75%	0.23	0.37	116	118.35
Mayores a 2.5°C e inferiores a 7.2°C	84.38%	0.91	0.69	85	87.64
Mayores a 0°C e inferiores a 10°C	68.75%	1.28	0.83	165	150.22

La estimación mensual se realizó mediante la sumatoria de las predicciones de los modelos mostrando una diferencia de 2.35 por encima de las 116 horas-frío para el modelo que considera las temperaturas inferiores a 7.2°C, 2.64 superior a la del modelo que considera el rango entre 2.5°C y 7.2°C, en el caso del modelo que considera el rango de temperaturas de 0-10°C la diferencia fue de 14.78 horas-frío. Estos valores fueron obtenidos siguiendo la metodología del capítulo 3 secciones 3.2 y 3.3 para el procesamiento de los datos recopilados por el sistema de medición.

Tras la comparación de las estimaciones mensuales bajo los diferentes modelos con los modelos de Damota, Crossa-Raynaud, Sharp, y el modelo modificado de Sharp por Morales se obtuvo la tabla 13.

Tabla 13

Comparación de los diferentes modelos con los modelos matemáticos de Da Mota, Sharp, y Crossa-Raynaud

Método	Horas-frío
Directo (<7.2°C)	116
ANN (<7.2°C)	118.35
Directo (2.5-7.2°C)	85
ANN (2.5-7.2°C)	87.64
Directo (0-10°C)	165
ANN (0-10°C)	150.22
Da Mota	24.6199618
Sharp (Torres,R)	98.8364476
Sharp (Morales, M)	98.8864476
Crossa-Raynaud	199.757178

Conclusiones

El monitoreo de las variables climatológicas y la cuantificación de horas-frío brinda información indispensable para la correcta toma de decisiones durante el proceso de producción de cultivos caducifolios. Los resultados obtenidos demuestran la viabilidad para la estimación de la cantidad de horas-frío tanto mediante el uso de tecnología IoT como por medio del uso de técnicas de aprendizaje de máquina.

A pesar de que la precisión del modelo que considera el rango de 2.5-7.2°C (84.38%) para una estimación diaria es inferior a la del modelo que considera solo las temperaturas menores a 7.2 (93.75%), la diferencia para una estimación mensual no es significativa (ver tabla 12), ambos tienen un error inferior a tres unidades de horas-frío al compararlos con el valor obtenido mediante el método directo correspondiente.

El método presenta una gran flexibilidad para afrontar la ambigüedad en el margen de medición de las horas-frío, esto derivó en un intervalo de estimación con un límite mínimo de 85 horas-frío y como máximo 165 horas-frío. Solo los métodos derivados de Sharp se encuentran dentro de este rango de estimación bajo los diferentes criterios para la cuantificación de horas-frío, por lo tanto, las estimaciones de los modelos matemáticos de Da Mota y Crossa-Raynoud con 24.61 y 199.75 horas-frío respectivamente, no presentan estimaciones confiables en la región geográfica donde se realizó el estudio.

El sistema de alerta por mensajes SMS permitió reducir la pérdida de datos derivados de fallos previstos o imprevistos durante el periodo en que se realizó el estudio. La metodología aplicada en esta investigación se presenta como una alternativa a los métodos tradicionales y su desempeño no está restringido a una región específica.

Referencias

- Alvarez, A., Arellano, M. E., & Bugarín, R. (2019). Acumulación de horas frío para la producción de arándano en Nayarit, Méx. *Revista Mexicana de Ciencias Agrícolas*, 175-185.
- Azari, B., Hassan, K., Pierce, J., & Ebrahimi, S. (2022). Evaluation of Machine Learning Methods Application in Temperature Prediction. *COMPUTATIONAL RESEARCH PROGRESS IN APPLIED SCIENCE & ENGINEERING*, 1-12.
- Brown, P., & Luedeling, E. (2011). A global analysis of the comparability of winter chill models for fruit and nut trees. *International Journal of Biometeorology*, 411-421.
- Calderón, E. (1992). *Manual del Fruticultor Moderno*. Mexico, D.F.: Limusa.
- Edian, F., & Ramos, J. (2019). Aprendizaje de máquina y aprendizaje profundo en biotecnología: aplicaciones, impactos y desafíos. *Ciencia, Ambiente y Clima*, 7-26.
- Egas, A. K., & Roque, C. T. (2020). Diseño de un modelo predictivo basado en técnicas de machine learning que permita determinar la temperatura usando los datos de una mini estación meteorológica en la ciudad de Guayaquil. *UNIVERSIDAD DE GUAYAQUIL/FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES*.
- Fadón, E., Herrera, S., Guerrero, B. I., Guerra, E. M., & Javier, R. (2020). Chilling and Heat Requirements of Temperate Stone Fruit Trees (*Prunus* sp.). *Agronomy*. doi:<https://doi.org/10.3390/agronomy10030409>
- Flores, P. C. (2007). Requerimiento de Frío en Frutales: Efectos Negativos sobre la Producción de Fruta. *Agromensajes*, 13-14.
- García, E., & Flego, F. (s.f.). *Universidad de Palermo*. Recuperado el 04 de Junio de 2024, de <https://www.palermo.edu/ingenieria/downloads/pdfwebc&T8/8CyT12.pdf>
- Gómez, R. J., & Morales, M. L. (s.f.). *Huerto Fenológico del colegio de geografía*. Recuperado el 05 de Junio de 2024, de ALGUNAS CONSIDERACIONES SOBRE EL CONCEPTO DE HORAS-FRÍO Y SUS MÉTODOS DE CÁLCULO. ALTERNATIVA EN SU ESTUDIO.: <http://huertofenologico.filos.unam.mx/files/2017/05/CONCEPTO-DE-HORAS-FRIO.pdf>
- González, C. P. (2016). *Agricultura de Precisión*. Obtenido de cicy: https://www.cicy.mx/Documentos/CICY/Eventos/2016/Taller-Nanomateriales/Ponentes/Agricultura-de-Precision_Ivan-Gonzalez.pdf
- INCYTU. (2018). *Agricultura de Precisión*. Ciudad de México.
- Konstantinos, G. L., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine Learning in Agriculture: A Review. *Sensors*, 1-29.
- Maio, S. (2020). Estimación de valores agroclimáticos de horas de frío en San Pedro (Provincia de Buenos Aires, Argentina) y escenario futuro. *Revista Argentina de Agrometeorología*, 45-53.

- Maulud, D. H., & Abdulazeez, A. M. (2020). A Review on Linear Regression Comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends*, 140-147.
- Mendoza, G. E. (2020). CUANTIFICACIÓN Y ACUMULACIÓN DE HORAS-FRÍO Y DÍAS-GRADO EN EL VALLE ALTO DE COCHABAMBA. *Revista de Investigación e Innovación Agropecuaria y de Recursos Naturales, La Paz*, 144-150.
- NCDC. (2008). *National Oceanic and Atmospheric Administration*. Obtenido de <https://www.noaa.gov/>
- Ocampo, Velázquez, Rosalía V.; Rojas Reséndiz, Andrea L.; Villagómez Aranda, Ana Laura. (2022). Breve historia de la agricultura. En R. G. Guevara González, & I. Torres Pacheco, *Manejo del estrés vegetal como una estrategia para una agricultura sostenible* (págs. 13-52). Almería.
- Pertille, R. H., Sachet, M. R., Guerrezi, M. T., & Citadin, I. (2019). An R package to quantify different chilling and heat models for temperate fruit trees. *Computers and Electronics in Agriculture*. doi:<https://doi.org/10.1016/j.compag.2019.105067>
- Ramirez, C. I. (2016). *El desarrollo de la agricultura y el impacto que tendría en las finanzas públicas de México*. México.
- SADER. (26 de septiembre de 2021). *Gobierno de México*. Obtenido de <https://www.gob.mx/agricultura/prensa/estima-agricultura-crecimiento-de-dos-digitos-en-produccion-nacional-de-manzanas-al-cierre-del-ciclo-agricola-2021>
- SADER. (2024). *Gobierno de México*. Obtenido de <https://www.gob.mx/agricultura/es/articulos/principales-exportaciones-de-mexico>
- SADER. (11 de Febrero de 2024). *Gobierno de México*. Obtenido de Rompen récord exportaciones agroalimentarias en 2023, superan los 51 mil mdd: Agricultura: <https://www.gob.mx/agricultura/prensa/rompen-record-exportaciones-agroalimentarias-en-2023-superan-los-51-mil-mdd-agricultura#:~:text=La%20balanza%20comercial%20agropecuaria%20y,m%C3%A1s%20alta%20en%2031%20a%C3%B1os>.
- Segrelles, J. A. (2001). PROBLEMAS AMBIENTALES, AGRICULTURA Y GLOBALIZACIÓN EN AMÉRICA LATINA. *REVISTA ELECTRÓNICA DE GEOGRAFÍA Y CIENCIAS SOCIALES*.
- Servicio de Información Agroalimentaria y Pesquera. (15 de Mayo de 2017). *Gobierno de México*. Obtenido de Blog: <https://www.gob.mx/siap/articulos/manzana-mexico-produjo-716-930-toneladas-en-2016?idiom=es>
- Soto, P. J., Flores, C. M., Sanchez, C. E., & Piña, R. F. (2020). Compensadores de frío en manzano Golden Glory desarrollo y producción. *Revista Mexicana de Ciencias Agrícolas*, 11, 69-82.

Apéndice A

```
Test_TransmitLoRaPacket_ANEvery | Arduino IDE 2.0.4
File Edit Sketch Tools Help
Arduino Nano Every
Test_TransmitLoRaPacket_ANEvery.ino Test_TransmitLoRaPacket_ANEvery.ino
1 #include <ArduinoJson.h>
2 #include<SoftwareSerial.h>
3 #include<DHT.h>
4 SoftwareSerial e5(2, 3);//Rx,Tx;
5 //Constants
6 #define DHTPIN 4 // what pin we're connected to
7 #define DHTTYPE DHT22 // DHT 22 (AM2302)
8 DHT dht(DHTPIN, DHTTYPE); //// Initialize DHT sensor for normal 16mhz Arduino
9 float hum; //Stores humidity value
10 float temp; //Stores temperature value
11 static char recv_buf[512];
12 static bool is_exist = false;
13 static int led = 13;
14 int counter = 0;
15 static int at_send_check_response(char *p_ack, int timeout_ms, char *p_cmd, ...)
16 {
17     int ch;
18     int num = 0;
19     int index = 0;
20     int startMillis = 0;
21     va_list args;
22     memset(recv_buf, 0, sizeof(recv_buf));
23     va_start(args, p_cmd);
24     e5.print(p_cmd);
25     Serial.print(p_cmd);
26     va_end(args);
27     delay(200);
28     startMillis = millis();
29
30     if (p_ack == NULL)
31         return 0;
32 }
```

```
File Edit Sketch Tools Help
Arduino Nano Every
Test_TransmitLoRaPacket_ANEvery.ino Test_TransmitLoRaPacket_ANEvery.ino
33 do
34 {
35     while (e5.available() > 0)
36     {
37         ch = e5.read();
38         recv_buf[index++] = ch;
39         Serial.print((char)ch);
40         delay(2);
41     }
42
43     if (strstr(recv_buf, p_ack) != NULL)
44         return 1;
45
46 } while (millis() - startMillis < timeout_ms);
47 Serial.println();
48 return 0;
49 }
50
51 static void recv_prase(char *p_msg)
52 {
53     if (p_msg == NULL)
54     {
55         Serial.println("Received null");
56         return;
57     }
58     char *p_start = NULL;
59     char data[128]; // To hold the received bytes as characters
60
61     int bytes_len=0;
62     p_start = strstr(p_msg, "TX");
63     //Serial.println(p_start);
64     if (p_start != NULL) {
65         while (e5.available() > 0) {
66             ch = e5.read();
67             data[bytes_len++] = ch;
68             Serial.print((char)ch);
69             delay(2);
70         }
71         Serial.println(data);
72     }
73 }
```

Output

```
File Edit Sketch Tools Help
Arduino Nano Every
Test_TransmitLoRaPacket_ANEvery.ino Test_TransmitLoRaPacket_ANEvery.ino
64     if (p_start && (1 == sscanf(p_start, "+TEST: TX %s", &data)))
65     {
66         //Serial.println(data);
67     }
68 }
69
70 void setup(void)
71 {
72     Serial.begin(9600);
73     dht.begin();
74     pinMode(led, OUTPUT);
75     digitalWrite(led, LOW);
76     e5.begin(9600);
77     Serial.print("E5 LOCAL TEST\r\n");
78     at_send_check_response("+AT: OK", 100, "AT\r\n");
79     at_send_check_response("+ID: AppEui", 1000, "AT+ID\r\n");
80     at_send_check_response("+MODE: TEST", 1000, "AT+MODE=TEST\r\n");
81     at_send_check_response("+CH: NUM, 1-2", 1000, "AT+CH=NUM,0-2\r\n");
82     delay(200);
83     digitalWrite(led, HIGH);
84 }
85
86 void loop(void)
87 {
88     //Read data and store it to variables hum and temp
89     hum = dht.readHumidity();
90     temp= dht.readTemperature();
91     //Print temp and humidity values to serial monitor
92     Serial.print("Humidity: ");
93     Serial.print(hum);
94     Serial.print(" %, Temp: ");
95     Serial.print(temp);
```

Output

```
File Edit Sketch Tools Help
Arduino Nano Every

Test_TransmitLoRaPacket_ANEvery.ino Test_TransmitLoRaPacket_ANEvery.ino
88 //Read data and store it to variables hum and temp
89 hum = dht.readHumidity();
90 temp= dht.readTemperature();
91 //Print temp and humidity values to serial monitor
92 Serial.print("Humidity: ");
93 Serial.print(hum);
94 Serial.print(" %, Temp: ");
95 Serial.print(temp);
96 Serial.println(" Celsius");
97 delay(10000); //Delay 2 sec.
98 char cmd[256];
99 char Shum[12];
100 char Stemp[12];
101 char Readings[24]="";
102 dtostrf(hum,5,2,Shum);// variables a string
103 dtostrf(temp,5,2,Stemp);
104 strcat(Readings, Shum);
105 strcat(Readings, Stemp);//concatenar ambas strings para el envio en un solo paquete
106 Serial.println(Readings);
107 counter=counter+1;
108 // Transmit HEX Value
109 sprintf(cmd, "AT+TEST=TXLRSTR, \"%s\" \r\n", Readings);
110
111 int ret = at_send_check_response("+TEST: TXLRSTR", 5000, cmd);
112
113 if (ret)
114 | Serial.println("Sent");
115 else
116 | Serial.println("Send failed!\r\n\r\n");
117 delay(5000);
118 }
119

Output
```

Apéndice B

```
Filipenses413_v10 | Arduino IDE 2.0.4
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
1 #include <ArduinoJson.h>
2 #include<SoftwareSerial.h>
3 /* More information at: https://www.aeq-web.com/
4 * ESP32 SIM800 HTTP POST TEST | V 1.0_2020-AUG-18
5 */
6
7 #define SIM800L_RX 27
8 #define SIM800L_TX 26
9 #define SIM800L_PWRKEY 4
10 #define SIM800L_RST 5
11 #define SIM800L_POWER 23
12 //DS Mode Configuration
13 #define uS_TO_S_FACTOR 1000000 /* Conversion factor for micro seconds to seconds */
14 #define TIME_TO_SLEEP 900 /* Time ESP32 will go to sleep (in seconds) */
15 RTC_DATA_ATTR int bootCount = 0;
16 /*"Connection: keep-alive\r\n Content-Encoding: utf-8\r\n Accept-Encoding: gzip, defla
17 String apn = "internet.itelcel.com"; //APN
18 String apn_u = "webgprs"; //APN-Username
19 String apn_p = "webgprs2002"; //APN-Password
20 String url = "https://iotc-f9eb2992-38be-49ea-9ac3-f78a833f7f90.azure-devices.net:443/
21 String temperature = "0";
22 float humidity = 0;
23 SoftwareSerial e5(25, 26); //Rx,Tx
24 char CodeResponse[256];
25 static char recv_buf[512];
26 static char recv_buf2[512];
27 static char CR[3];
28 int ErrorC = 0;
29 String CRIS = "0";
30 int CRI = 0;
31 static char CHum[5];
32 float Humedad;
33 static char CTemp[5];
34 float Temperatura;
35 char tmp[2] = " ";
```

```
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
35 char tmp[2] = ".";
36 static int led = 13;
37 String Numero_cliente = "6181818356";
38 int Posicion = 200;
39 String CodeError = "";
40 String data = "";
41 String temperatura16 = "";
42 String humedad16 = "";
43 String temperatura = "";
44 String humedad = "";
45 String temp2 = "";
46 String hum1 = "";
47 String temp1 = "";
48 String entrada = "";
49 int PosC = 0;
50 int P1 = 0;
51 int P2 = 0;
52 bool band=false;
53 int cont=0;
54 String tf="";
55 static int at_send_check_response(char *p_ack, int timeout_ms, char *p_cmd, ...)
56 {
57     int ch;
58     int num = 0;
59     int index = 0;
60     int startMillis = 0;
61     va_list args;
62     memset(recv_buf, 0, sizeof(recv_buf));
63     va_start(args, p_cmd);
64     e5.print(p_cmd);
65     va_end(args);
66     delay(200);
67     startMillis = millis();
68
69     if (p_ack == NULL)
```

```
Filipenses413_v10.ino Filipenses413_v10.ino
68
69     if (p_ack == NULL)
70         return 0;
71
72     do
73     {
74         while (e5.available() > 0)
75         {
76             ch = e5.read();
77             recv_buf[index++] = ch;
78             //Serial.print(char(ch));
79             if (ch == 34){
80                 band=true;
81                 //break;
82             }
83             if (band){
84                 temp2=temp2+String(char(ch));
85                 //Serial.println(char(hexToDec(temp2)));
86                 //tf=tf+char(hexToDec(temp2));
87             }
88             delay(2);
89         }
90         if (strstr(recv_buf, p_ack) != NULL)
91             return 1;
92     } while (millis() - startMillis < timeout_ms);
93     Serial.println();
94     return 0;
95 }
96
97 void setup(void)
98 {
99     ++bootCount;
100     Serial.println("Boot number: " + String(bootCount));
101     esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
102     Serial.begin(9600);
```

```
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
101 esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * SLEEP_FACTOR);
102 Serial.begin(9600);
103 pinMode(led, OUTPUT);
104 digitalWrite(led, LOW);
105 e5.begin(9600);
106 Serial.print("E5 LOCAL TEST\r\n");
107 at_send_check_response("+AT: OK", 100, "AT\r\n");
108 at_send_check_response("+ID", 100, "AT\r\n");
109 at_send_check_response("+ID: AppEui", 1000, "AT+ID\r\n");
110 at_send_check_response("+MODE: TEST", 1000, "AT+MODE=TEST\r\n");
111 at_send_check_response("+CH: NUM, 1-2", 1000, "AT+CH=NUM,0-2\r\n");
112 at_send_check_response("+LOG: ", 1000, "AT+LOG=QUIET\r\n");
113 at_send_check_response("+TEST: TXLRSTR", 5000, "AT+TEST=RXLRPKT\r\n");
114 delay(200);
115 digitalWrite(led, HIGH);
116 pinMode(SIM800L_POWER, OUTPUT);
117 digitalWrite(SIM800L_POWER, HIGH);
118 Serial.println("ESP32+SIM800L AT CMD Test");
119 Serial2.begin(9600, SERIAL_8N1, SIM800L_TX, SIM800L_RX);
120 delay(2000);
121 while (Serial2.available()) {
122   Serial.write(Serial2.read());
123 }
124 delay(2000);
125
126 gsm_config_gprs();
127 }
128 void SHEXtoASCII(String data)
129 {
130   humedad="";
131   temperatura="";
132   humedad16="";
133   temperatura16="";
134   entrada=data.substring(1,data.length());
135   int punto = entrada.indexOf("E");
```

```
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
135 int punto = entrada.indexOf("E");
136 humedad16 = entrada.substring(0,punto+5);
137 temperatura16 = entrada.substring(punto+5,punto+14)+"0";
138 //entrada.length()
139 for (int i = 0; i < humedad16.length(); i += 2)
140 { String doscar = humedad16.substring(i, i + 2);
141   int valdecimal = strtol(doscar.c_str(), NULL, 16);
142   humedad=humedad+char(valdecimal);
143 }
144 for (int i = 0; i < temperatura16.length(); i += 2)
145 { String doscar = temperatura16.substring(i, i + 2);
146   int valdecimal = strtol(doscar.c_str(), NULL, 16);
147   temperatura=temperatura+char(valdecimal);
148 }
149 Serial.println(humedad);
150 Serial.println(temperatura);
151 }
152 void loop(void)
153 {
154   int ret = at_send_check_response("+TEST: RX", 1000, "");
155   if (ret){
156     Serial.println("Incoming Data...");
157     SHExtoASCII(temp2);
158     gsm_http_post("{\"Temperatura\":\"+ temperatura +\",\"Humedad\":\"+ humedad +\",\"filipenses\":\"413}");
159     digitalWrite(led, LOW);
160     delay(500);
161     digitalWrite(led, HIGH);
162     delay(14500);
163   }
164   else
165   {
166     Serial.println("LoRa e5 Fuera de rango ¿Esta conectado?...");
167     digitalWrite(led, LOW);
168     delay(15000);
169   }
}
```

```
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
169     }
170     temp2="";
171     band=false;
172 }
173
174 void gsm_http_post(String input) {
175     String Head = "Authorization: SharedAccessSignature sr=iotc-f9eb2992-38be-49ea-9ac3
176     Serial.println(" --- Start GPRS & HTTP --- ");
177     gsm_send_serial("AT+SAPBR=1,1");
178     gsm_send_serial("AT+SAPBR=2,1");
179     gsm_send_serial("AT+HTTTPINIT");
180     gsm_send_serial("AT+HTTTPARA=CID,1");
181     gsm_send_serial("AT+HTTTPARA=URL," + url);
182     gsm_send_serial("AT+HTTTPARA=CONTENT,application/x-www-form-urlencoded");
183     gsm_send_serial("AT+HTTTPARA=USERDATA," + Head + "\r\n");
184     gsm_send_serial("AT+HTTTPDATA="+ String(input.length()) + ",5000");
185     gsm_send_serial(input);
186     gsm_send_serial("AT+HTTTPSSL=1");
187     gsm_send_serial("AT+SSLOPT=0,1");
188     gsm_send_serial("AT+HTTTPACTION=1");
189     gsm_send_serial("AT+HTTTPREAD");
190     gsm_send_serial("AT+HTTTPTERM");
191     gsm_send_serial("AT+SAPBR=0,1");
192 }
193
194 void gsm_config_gprs() {
195     Serial.println(" --- CONFIG GPRS --- ");
196     gsm_send_serial("AT+SAPBR=3,1,Contype,GPRS");
197     delay(2000);
198     gsm_send_serial("AT+SAPBR=3,1,APN," + apn);
199     if (apn_u != "") {
200         gsm_send_serial("AT+SAPBR=3,1,USER," + apn_u);
201     }
202     if (apn_p != "") {
```

```
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
203     gsm_send_serial("AT+SAPBR=3,1,PwD," + apn_p);
204     }
205 }
206
207 void gsm_send_serial(String command) {
208     memset(recv_buf2, 0, sizeof(recv_buf2));
209     int index2 = 0;
210     int code;
211     Serial.println("Send ->: " + command);
212     Serial2.println(command);
213     if (command == "AT+HTTPREAD")
214     {
215         long wtimer = millis();
216         while (wtimer + 10000 > millis()) {
217             while (Serial2.available()) {
218                 CodeError = CodeError+char(Serial2.read());
219                 //Serial.write(Serial2.read());
220                 delay(2);
221             }
222         }
223         delay(1000);
224         Serial.println(CodeError);
225         DepError(CodeError);
226         CodeError="";
227     }
228     else{
229         long wtimer = millis();
230         while (wtimer + 3000 > millis()) {
231             while (Serial2.available()) {
232                 Serial.write(Serial2.read());
233                 delay(2);
234             }
235         }
236     }
237 }
```

```
Filipenses413_v10 | Arduino IDE 2.0.4
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
237 }
238 void DepError(String data){
239     int coma = data.indexOf(",")+1;
240     CRIS=data.substring(coma,coma+3);
241     CRI=CRIS.toInt();
242     switch (CRI) {
243     case 204:
244         sprintf(CodeResponse, "CODIGO %d : Enviado con exito a Azure Iot Central...", CRI);
245         Serial.println(CodeResponse);
246         delay(2000);
247         esp_deep_sleep_start();
248         break;
249     case 401:
250         sprintf(CodeResponse, "CODIGO %d : El token ha caducado! FLP-4:13", CRI);
251         Serial.println(CodeResponse);
252         Enviar_msj(Numero_cliente, "E401: El token ha caducado... FLP-4:13");
253         delay(20000);
254         ESP.deepSleep(0);
255         break;
256     case 404:
257         sprintf(CodeResponse, "CODIGO %d : No se encontro el server... FLP-4:13", CRI);
258         Serial.println(CodeResponse);
259         Enviar_msj(Numero_cliente, "E404: No se encontro el server... FLP-4:13");
260         delay(20000);
261         ESP.restart();
262         break;
263     case 603:
264         sprintf(CodeResponse, "CODIGO %d : Comportamiento esperado reiniciando ESP32LilyGO... FLP-4:13", CRI);
265         Serial.println(CodeResponse);
266         Enviar_msj(Numero_cliente, "E603: Comportamiento esperado reiniciando ESP32LilyGO... FLP-4:13");
267         delay(20000);
268         ESP.deepSleep(0);
269     break;
270     default:
271         Serial.println(CRI);
```

```
File Edit Sketch Tools Help
Select Board
Filipenses413_v10.ino Filipenses413_v10.ino
269 break;
270 default:
271 Serial.println(CRI);
272 Serial.println("ERROR! Datos no enviados... \n");
273 ErrorC++;
274 Serial.println(ErrorC);
275 delay(2000);
276 if (ErrorC == 10)
277 {
278 Enviar_msj(Numero_cliente, "EUK: Error desconocido... FLP-4:13");
279 delay(20000);
280 ESP.restart();
281 }
282 }
283 }
284 void Enviar_msj(String numero, String msj)
285 {
286 //Se establece el formato de SMS en ASCII
287 String config_numero = "AT+CMGS=\"" + numero + "\"\r\n";
288 Serial.println(config_numero);
289 //configurar modulo como modo SMS
290 Serial2.write("AT+CMGF=1\r\n");
291 delay(1000);
292 //Enviar comando para un nuevos SMS al numero establecido
293 Serial2.print(config_numero);
294 delay(1000);
295 //Enviar contenido del SMS
296 Serial2.print(msj);
297 delay(1000);
298 //Enviar Ctrl+Z
299 Serial2.write((char)26);
300 delay(1000);
301 Serial.println("Mensaje enviado");
302 }
```