

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Escuela profesional de Ingeniería de Sistemas



Una Institución Adventista

**Modelo predictivo agroclimático para detectar cambios
climatológicos con machine learning e IoT para el sector agrario
de la Región Puno**

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Por:

María Dolores Quispe Merma

John Wagner Yana Puma

Asesor:

Ing. Angel Rosendo Condori Coaquira

Juliaca, diciembre de 2020

DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS

Angel Rosendo Condori Coaquira, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: **"MODELO PREDICTIVO AGROCLIMÁTICO PARA DETECTAR CAMBIOS CLIMATOLÓGICOS CON MACHINE LEARNING E IOT PARA EL SECTOR AGRARIO DE LA REGIÓN PUNO"** constituye la memoria que presenta los Bachilleres **María Dolores Quispe Merma** y **John Wagner Yana Puma** para obtener el título de Profesional de Ingeniero de Sistemas, cuya tesis ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad de los autores, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca, a los 20 días del mes de julio del año 2021.



Ing. Angel Rosendo Condori
Coaquira



ACTA DE SUSTENTACIÓN DE TESIS

En Puno, Juliaca, Villa Chullunquián, a 28 día(s) del mes de diciembre del año 2020 siendo las 5:40 horas, se reunieron en el Salón de Grados y Títulos de la Universidad Peruana Unión, Filial Juliaca, bajo la dirección del Señor Presidente del jurado: Mtro. Lennin Henry Lenturión Julca, el secretario: Mg. Abel Angel Sullen Macalusa y los demás miembros: Ing. Jorge Eddy Ortega Luque e Ing. Eder Gutiérrez Quiroz y el asesor Ing. Angel Rosendo Londoño Loaguiva

con el propósito de administrar el acto académico de sustentación de la tesis titulada: Modelo predictivo agroclimático para detectar cambios climatológicos con machine learning e IoT para el sector agrario de la Región Puno

de el(los)/a(las) bachiller(es): a) John Wagner Yana Ruma
b) Maria Dolores Quispe Merma

conducente a la obtención del título profesional de Ingeniero de Sistemas
(Nombre del Título Profesional)

con mención en.....

El Presidente inició el acto académico de sustentación invitando al (los)/a(la)(las) candidato(a)s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/a(la)(las) candidato(a)s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): John Wagner Yana Ruma

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>14</u>	<u>C</u>	<u>decente</u>	<u>Bueno</u>

Candidato (b): Maria Dolores Quispe Merma

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>14</u>	<u>C</u>	<u>decente</u>	<u>Bueno</u>

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al(los)/a(la)(las) candidato(a)s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente

Asesor

Candidato/a (a)

Miembro

Miembro

Candidato/a (b)

DEDICATORIA

A mi familia por ser el motivo para alcanzar mis objetivos, a mis amados padres y hermanos por su apoyo infinito e incondicional.

(María Quispe)

Dedico primeramente a Dios, a mis padres, hermanos y amigos, por su apoyo en esta investigación.

(John Yana)

AGRADECIMIENTOS

Doy gracias a Dios por guiarme y protegerme, también hago llegar mis agradecimientos a mi casa de estudios Universidad Peruana Unión por acogerme durante mi formación profesional, seguidamente a mi asesor Ing. Angel Rosendo Condori Coaquira por su acompañamiento durante la elaboración de la presente investigación.

A mi madre por su constante aliento y su apoyo incondicional.
(María Quispe).

Agradecer primeramente a Dios, a mis padres por su apoyo moral, de igual manera a mi asesor Ing. Angel Rosendo Condori Coaquira por su guía durante el proceso de investigación.

(John Yana).

ÍNDICE GENERAL

DEDICATORIA	iv
AGRADECIMIENTOS	v
ÍNDICE GENERAL	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS	ix
ABREVIATURAS Y ACRÓNIMOS	xiv
RESUMEN	xv
ABSTRACT	xvi
CAPÍTULO I. El problema	17
1.1. Identificación del problema	17
1.2. Objetivos	21
1.2.1. Objetivo general	21
1.2.2. Objetivos específicos	21
1.3. Justificación	22
1.4. Presuposición filosófica	23
CAPÍTULO II. Revisión de la literatura	24
2.1 Revisión de literatura	24
2.2 Marco teórico	26
2.2.1. Machine learning	26
2.2.2. Modelo predictivo.....	27
2.2.3. Datos Históricos	28
2.2.4. Series temporales.....	28
2.2.5. Algoritmos	28
2.2.6. Multiclass Decision Jungle.....	28
2.2.7. Logistic Regression	28
2.2.8. Naive Bayes.....	29
2.2.9. Linear Regression	29
2.2.10. Internet of Things (IoT)	29
2.2.11. Plataformas para IoT.....	30
2.2.12. Azure Machine Learning Studio.....	35

2.2.13.	Microsoft-Power Bi	36
2.2.14.	Hardware libre	36
2.2.15.	Agro clima	43
2.2.16.	Estación meteorológica.....	44
2.2.17.	SENAMHI	45
2.2.18.	Metodología CRISP DM.....	45
CAPÍTULO III. Materiales y métodos		49
3.1.	Tipo de Investigación.....	49
3.2.	Arquitectura de solución	49
3.3.	Materiales.....	49
3.3.1.	Hardware	49
3.3.2.	Software.....	50
3.4.	Desarrollo de la Metodología.....	50
3.4.1.	Implementación y configuración de PWS hasta Microsoft Azure IoT Hub..	50
3.4.2.	Fases de la Metodología Crisp-DM.....	68
CAPÍTULO IV. Resultados		113
CAPÍTULO V. Conclusiones y recomendaciones		116
REFERENCIAS.....		119
ANEXOS.....		123

ÍNDICE DE TABLAS

Tabla 1. Materiales y equipos.....	49
Tabla 2. Software a utilizar en el desarrollo de la investigación.....	50
Tabla 3. Variables.....	75
Tabla 4. Resultados comparativos de los algoritmos de Regression.....	81
Tabla 5. Resultados comparativos de los algoritmos de clasificación.....	83

ÍNDICE DE FIGURAS

Figuras 1. Impacto de eventos climáticos adversos en la agricultura.....	18
Figuras 2. Número de emergencias ocurridas, según departamento - Mes agosto 2017	19
Figuras 3. Machine Learning	27
Figuras 4. Internet of Things (IoT)	30
Figuras 5. Nube de Microsoft Azure.....	31
Figuras 6. Arquitectura de Microsoft Azure IoT	32
Figuras 7. Cuenta de almacenamiento – Microsoft Azure.....	33
Figuras 8. Centro de IoT – Microsoft Azure.....	34
Figuras 9. Stream Analytics – Microsoft Azure.....	34
Figuras 10. Azure ML Studio – Microsoft	35
Figuras 11. Microsoft-Power Bi	36
Figuras 12. Placa de desarrollo ESP-32.....	37
Figuras 13. Placa de desarrollo NodeMCU-ESP8266	39
Figuras 14. Placa ArduinoUNO	39
Figuras 15. SparkFun Weather Shield	40
Figuras 16. Sensor DHT 22 y DHT 11	41
Figuras 17. Sensor BMP180.....	42
Figuras 18. Cargador de batería – TP4056.....	42
Figuras 19. Placa solar – 7v	43
Figuras 20. Batería litio recargable.....	43
Figuras 21. Metodología CRISP-DM.....	45
Figuras 22. Fase 1 de Metodología CRISP-DM.	46
Figuras 23. Fase 2 de metodología CRISP-DM.	46

Figuras 24. Fase 3 de metodología CRISP-DM.	47
Fuente: Google, tomada en el 2019.	47
Figuras 25. Fase 4 de metodología CRISP-DM	47
Fuente: Google, tomada en el 2019.	48
Figuras 26. Fase 5 de metodología CRISP-DM	48
Fuente: Google, tomada en el 2019.	48
Figuras 27. Fase 6 de metodología CRISP-DM	48
Fuente: Google, tomada en el 2019.	49
Figuras 28. Arquitectura de solución del modelo predictivo.....	49
Figuras 29. Estructura acrílica – PWS	50
Figuras 30. Diagrama de conexión del PWS	51
Figuras 31. Código fuente de PWS.....	52
Figuras 32. Librerías de conexión a ThingSpeak	53
Figuras 33. Creando un nuevo Channels.....	54
Figuras 34. Api Key Channels	54
Figuras 35. Chart de los datos capturados	55
Figuras 36. Estructura de solución IoT	55
Figuras 37. Creando un conjunto de recursos para la solución IoT	56
Figuras 38. Creando el centro de IoT para la solución IoT.....	57
Figuras 39. Creando un nuevo dispositivo en el centro de IoT	58
Figuras 40. Las claves de conexión del dispositivo del centro de IoT	59
Figuras 41. Enviando los mensajes del PWS al centro de IoT por el monitor serie....	59
Figuras 42. Recibiendo mensajes del PWS en el centro de IoT.....	60
Figuras 43. Arquitectura de solución IoT hasta Power Bi.....	60
Figuras 44. Creando el grupo de consumidor en el centro de IoT	61

Figuras 45. Creando un nuevo Stream Analytics para la configuración – Power Bi..	62
Figuras 46. Entrada de Stream Analytics para la configuración – Power Bi	64
Figuras 47. Salida de Stream Analytics para la configuración – Power Bi.....	65
Figuras 48. Consulta en Stream Analytics para la configuración – Power Bi.....	66
Figuras 49. Ejecutando el Trabajo de Stream Analytics– Power Bi.....	66
Figuras 50. Creando el informe en Power Bi	67
Figuras 51. Gráfico de valores temperatura y humedad en tiempo real -Power Bi....	68
Figuras 52. Estaciones meteorológicas del SENAMHI	69
Figuras 53. Estaciones Convencionales con recepción de datos en tiempo real - SENAMHI	70
Figuras 54. Estaciones Convencionales con recepción de datos en tiempo diferido - SENAMHI	71
Figuras 55. Estaciones Automáticas - SENAMHI	72
Figuras 56. Estaciones del Departamento de Puno - SENAMHI	73
Figuras 57. Datos de temperatura, precipitación, humedad, dirección y velocidad del viento.....	75
Figuras 58. Visualización de datos.	76
Figuras 59. Datos en formato.....	77
Figuras 60. Estructura de solución del modelo predictivo.....	78
Figuras 61. Resultado de la evaluación del modelo	79
Figuras 62. Baseline Weather Prediccion – varios algoritmos	80
Figuras 63. Modelo baseline con algoritmos de Clasificación	82
Figuras 64. Resultados para el algoritmo Two Class Logistic Regression	82
Figuras 65. Resultados para el algoritmo Two Class Decision Booted Decision	83
Figuras 66. Modelado en Azure ML Studio	84
Figuras 67. Agregando un R Script	85

Figuras 68. Diagrama de experimento predictivo - PWS	86
Figuras 69. URL y Clave Key	86
Figuras 70. Agregando un grupo de consumidor para ML - PWS.....	87
Figuras 71. Agregando un Analytics para ML - PWS	88
Figuras 72. Agregando una entrada en Stream Analytics para ML - PWS.....	89
Figuras 73. Creando una cuenta de almacenamiento para ML - PWS.....	90
Figuras 74. Agregando un contenedor en la cuenta del almacenamiento para ML - PWS	90
Figuras 75. Agregando una salida en Stream Analytics para ML - PWS.....	91
Figuras 76. Agregando una función en Stream Analytics para ML - PWS.....	92
Figuras 77. Ingresamos una consulta en Stream Analytics para ML - PWS.....	93
Figuras 78. Importar librerías para el modelo Decision tree	94
Figuras 79. Importación del dataset	94
Figuras 80. Conteo del dataset.....	94
Figuras 81. situación de precipitación	95
Figuras 82. Histograma de precipitación	95
Figuras 83. Definición de las variables para el modelo DecisionTreeClassifier.	96
Figuras 84. Separación de datos de entrenamiento y prueba/entrenamiento del algoritmo DecisionTreeClassifier.....	96
Figuras 85. Predicción con DecisionTreeClassifier	97
Figuras 86. Porcentaje de precisión del modelo.....	97
Figuras 87. Matriz de confusión.....	97
Figuras 88. Porcentaje de precisión del modelo.....	97
Figuras 89. Característica operativa del receptor	98
Figuras 90. Importación de librerías y carga de datos – Logistic regression.....	99
Figuras 91. Conteo del dataset – Logistic regression	99

Figuras 92. Histograma de precipitación – Logistic regression	100
Figuras 93. Separación de datos de entrenamiento y prueba – Logistic regression..	101
Figuras 94. Matriz de confusión y reporte de precisión del modelo – Logistic regression	102
Figuras 95. Característica operativa del receptor	102
Figuras 96. Evaluación del modelo de clasificación binaria – regresión logística.....	103
Figuras 97. Resultado de la evaluación de clasificación binaria - regresión logística.	104
Figuras 98. Porcentaje de precisión del modelo decision tree classifier.	105
Figuras 99. Porcentaje de precisión del modelo Logistic regression classifier.	105
Figuras 100. Curva del algoritmo de regresión logística sin calibrar.....	106
Figuras 101. Brier score del algoritmo de regresion sin calibar	107
Figuras 102. curva calibrada	109
Figuras 103. brier score de la curva calibrada	109
Figuras 104. Implementación del modelo.....	110
Figuras 105. Clave Api	111
Figuras 106. Resultado final del modelo predictivo.	112
Figuras 107. Resultado final del modelo de predicción climática.	115

ABREVIATURAS Y ACRÓNIMOS

- ML: Machine Learning (Aprendizaje Automático)
- IoT: Internet of Things (Internet de las Cosas)
- PWS: Prototipe Weather Station
- CC: Cambio climático
- CRISP-DM: Cross Industry Standard Process for Data Mining (Proceso estándar de la industria para la minería de datos)
- SENAMHI: Servicio Nacional de Meteorología e Hidrología
- MINAG: Ministerio de Agricultura
- DRAP: Dirección Regional Agraria - Puno
- INEI: Instituto Nacional de Estadística
- OMM: Organización Meteorológica Mundial
- MQTT - Message Queue Telemetry Transport

RESUMEN

El riesgo agroclimático en la vertiente del lago Titicaca para los cultivos de quinua y papa son esenciales para las distintas campañas agrícolas, ya que se requiere del pronóstico para diferentes épocas del año.

La presente investigación tiene como objetivo, Implementar un modelo predictivo para detectar cambios climatológicos con machine learning e IoT para el sector agrario de la región Puno; a través de algoritmos de clasificación y regresión de Machine Learning y como parte de IoT se construyó una mini estación meteorológica para capturar datos meteorológicos en tiempo real, esto con el fin de comparar con los resultados del modelo predictivo; los datos que servirán para dicho propósito son proporcionados por la institución de Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI). Para lograr el objetivo se trabaja con las siguientes variables: temperatura, humedad y del balance hídrico: precipitación

La metodología aplicada para el desarrollo de la presente investigación es CRISP-DM, que divide el proceso de minería de datos en seis fases, la metodología establece un conjunto de actividades en cada fase, pero sin especificar del cómo realizarlas. Cuyo resultado se genera los pronósticos de riesgo agroclimático para el sector agrario, dependiendo de las temporadas de variedad climática marcadas para la humedad entre diciembre, enero, febrero y marzo y secas otros meses.

Palabras claves: *Machine learning, IoT, modelo predictivo, Predicción agroclimática, senamhi.*

ABSTRACT

The agroclimatic risk on the slopes of Lake Titicaca for quinoa and potato crops are essential for the different agricultural seasons, since the forecast is required for different times of the year.

The present research aims to implement a predictive model to detect weather changes with machine learning and IoT for the agricultural sector of the Puno region; Through Machine Learning classification and regression algorithms and as part of IoT, a mini meteorological station was built to capture meteorological data in real time, this in order to compare with the results of the predictive model; The data that will serve this purpose are provided by the institution of the National Meteorology and Hydrology Service of Peru (SENAMHI). To achieve the objective, the following variables are used: temperature, humidity and the water balance: precipitation.

The methodology applied for the development of this research is CRISP-DM, which divides the data mining process into six phases, the methodology establishes a set of activities in each phase, but without specifying how to carry them out. The result of which is generated the agroclimatic risk forecasts for the agricultural sector, depending on the seasons of climatic variety marked for humidity between December, January, February and March and other dry months.

Keywords: *Machine learning, IoT, predictive model, agroclimatic prediction, senamhi.*

CAPÍTULO I. El problema

1.1. Identificación del problema

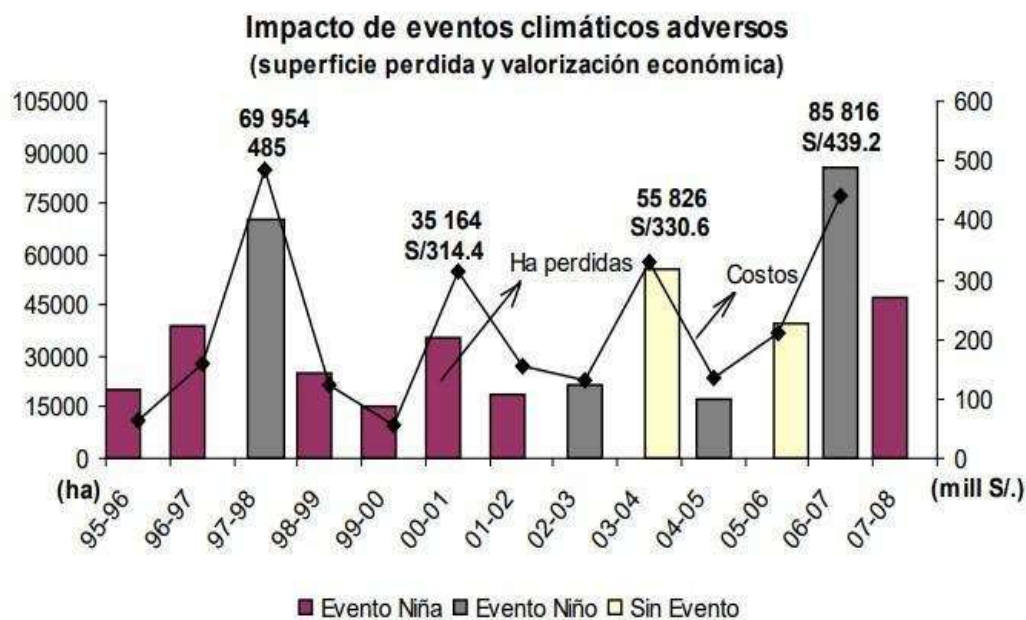
En la actualidad el cambio climático es un fenómeno que viene afectando la vida de todos los seres vivos y por ende es un grave problema ambiental y un desafío mayor a enfrentar a nivel internacional, nacional y local. La actividad humana junto a las industrias es causante de concentraciones atmosféricas de gases de efecto invernadero que van en aumento notablemente. Por tanto, la agricultura es uno de los sectores más sensibles al problema del cambio climático, ya que depende de muchas variables meteorológicas como la temperatura y precipitación, los efectos del cambio climático traen consigo pérdidas en la economía ya sea a gran escala o mediana, siendo los más afectados los agricultores.

(Carrasco, 2016) refiere que “el cambio climático afecta diversos ámbitos productivos y económicos, como el sector agropecuario, hídrico, biodiversidad, turismo, salud, entre otros. Trayendo riesgos desde disminución de producción y calidad de los alimentos, ingresos más bajos, alza de precios, sequías, mucha disponibilidad de agua debido al derretimiento de glaciares e inundaciones en áreas urbanas relacionadas con precipitación extrema”.

Según la Nasa, el clima fue cambiado conforme los años transcurren. En los últimos 650,000 años, hubo siete ciclos de retroceso y retroceso glacial como el final de la última era glacial hace aproximadamente 7,000 años, que a partir de ello se inició la era climática moderna y de nuestra civilización humana. A nivel mundial el cambio del clima es atribuido indirectamente o directamente por las actividades humanas por consiguiente trae consecuencias como las variaciones de los elementos climáticos, trayendo consigo amenazas/peligros (cambios bruscos de temperatura, granizo, vientos fuertes, lluvias intensa, heladas, retroceso de glaciares, sequías, huaycos y aluviones, etc.); trayendo consigo

impactos como aumento de plagas, cambios fenológicos, menor disponibilidad de agua, erosión, pérdida de productividad. El cambio climático trae consigo impactos en diferentes aspectos tal es el caso de la agricultura a nivel mundial, nacional y regional esto a la vez influye en el bienestar humano.

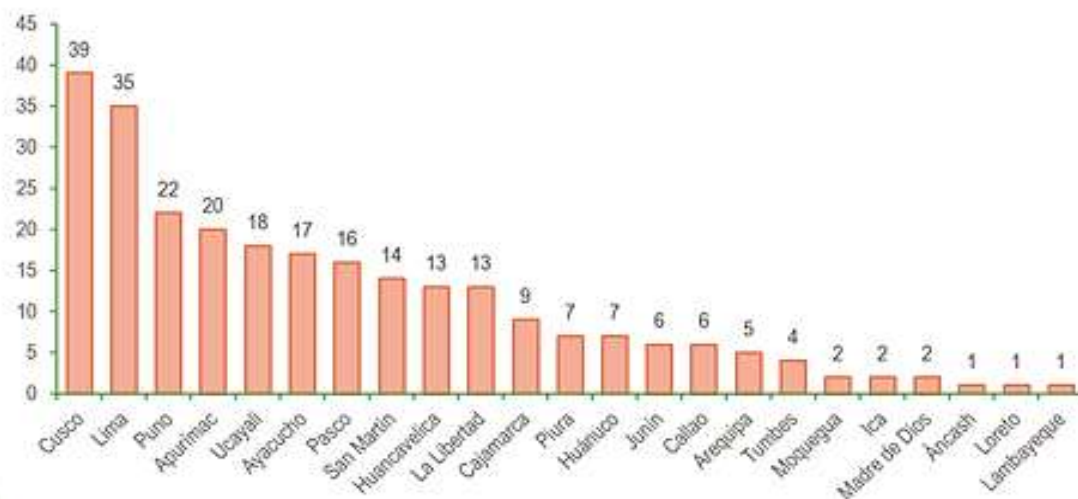
(Minagri, 2010), “resalta que tenemos una gran vulnerabilidad ante las variaciones climáticas; perdiendo más de 15 mil hectáreas por campaña agrícola. Cada dos años el sector enfrenta picos con superficie pérdidas; con costo promedio en 390 millones de soles. La mayor pérdida fue por los eventos del Niño durante los periodos 97-98 y 06-07; y en la campaña 03-04 que fue afectada por una sequía”. Ver figura 1.



Figuras 1. Impacto de eventos climáticos adversos en la agricultura

Fuente: Ministerio de Agricultura

La región puno sufre también los eventos adversos del cambio climático, enfrentándose a los cambios drásticos, INDECI registró 22 emergencias reportadas en puno para el año 2017, siendo el tercer departamento en el que se registró incidencias por el impacto del clima, ver figura 2.



Figuras 2. Número de emergencias ocurridas, según departamento - Mes agosto 2017
 Fuente: Ministerio de Agricultura

Implementar un modelo predictivo de Machine Learning para el pronóstico agroclimático con alimentación de datos históricos, el modelo será entrenado y será más inteligente y por ende mayor es su potencialidad en la toma de decisiones de poder incrementar la producción y poder anticipar los eventos adversos del cambio climático, en el sector agrario de la región Puno.

Hacer frente a los cambios climáticos, para realizar sembríos es un tema desesperante por parte de los productores. Entonces se requiere pronósticos agroclimáticos realizados con herramientas de machine learning y de acuerdo a ello dar a los agricultores recomendaciones sobre las condiciones climáticas que ocurrirían en los próximos meses con suficiente anticipación al evento climático, pues esto les ayudaría a evitar pérdidas productivas, económicas, etc.

“Un pronóstico climático permite tener información sobre las condiciones climáticas de los próximos meses y sirve para dar a los agricultores tengan información fiable sobre: ¿Cuándo sembrar? ¿Cuándo cosechar? ¿Qué variedad elegir? ¿Con qué densidad de

siembra? El cual conlleva a tomar mejores decisiones en el manejo de los cultivos, que posteriormente se evite grandes pérdidas económicas.

Las aplicaciones de Aprendizaje Automático (Machine Learning) en la agricultura, tiene relevancia. La predicción agroclimática alto andina en el Perú, es un tema importante; pronosticar o estimar el comportamiento futuro del clima (estimaciones de valores futuros), están basados en comportamientos pasados representados como SERIES DE TIEMPO. Es precisamente que Machine Learning, pone a nuestra disposición sus algoritmos para detectar patrones de comportamiento. Actualmente se hacen uso de estos modelos para la predicción y pronóstico del clima, predicción de fallos en equipos tecnológicos, etc. con la tecnología y los datos proporcionados por distintas entidades, en este caso el SENAMHI, hacer uso de los datos a favor de la sociedad.

(Iribarren, 2016), indica que “el uso de datos históricos con una suficiente calidad y cantidad es muy importancia para obtener resultados en modelación y predicción climática con el menor grado de incertidumbre posible”. “Las series temporales se encuentran vinculadas a variables climáticas del análisis de varios años de series temporales agro meteorológicas a través de algoritmos escalables de minería de datos” (EGASA, 2012).

“Existen diversos factores que pueden ser considerados como agentes forzantes del clima y consecuentemente alterarlo. Si podemos anticipar estos factores de forzamiento, entonces podremos anticipar en buena forma las fluctuaciones del sistema climático y generar una predicción, la respuesta del sistema climático a esta combinación de factores forzantes depende de los diversos tiempos de respuesta de los diferentes componentes del sistema” (Martinez, Rivadeneira, & Nieto, 2011).

“Los modelos de predicción numérica del tiempo (NWP) usan modelos matemáticos de la atmósfera para predecir el tiempo basados en las condiciones actuales del tiempo. Los

NWP son omnipresentes en la actual predicción meteorológica. Las bases teóricas de la predicción numérica del tiempo provienen de la dinámica atmosférica, la cual provee las ecuaciones que describen la evolución de la atmósfera” (Lighezzolo, 2014).

La información resultante puede ser utilizada por diferentes usuarios (Dirección regional agraria, gremios, técnicos en campo y agricultores). El acceso al conocimiento y permitir recomendar al productor sobre sus épocas de siembra para una mayor eficiencia. Siendo esto una iniciativa para una agricultura sostenible y adaptada al clima. A través del análisis de información histórica y siguiendo los principios de análisis de datos; herramientas basadas en algoritmos de aprendizaje automático arrojarán información relevante sobre los factores más limitantes en la producción de los cultivos. Esta información emitida será útil para la toma de decisiones por parte de gremios y sus agricultores, así como también podrán tener la posibilidad de adoptar buenas prácticas agronómicas adaptadas al clima futuro.

La certeza de los resultados que se obtendría, se constituirá como un gran aporte para la toma de decisiones en cuanto al manejo que debe de seguirse y de adoptar medidas de adaptabilidad frente al cambio agroclimático. Que favorecerá a diferentes cultivos alto andinos.

1.2.Objetivos

1.2.1. Objetivo general

Implementar un modelo predictivo para detectar cambios climatológicos con machine learning e IoT para el sector agrario de la región Puno.

1.2.2. Objetivos específicos

- Implementar una mini estación meteorológica para capturar datos en tiempo real.

- Transmitir y procesar los datos obtenidos para alimentar con ellos el modelo, configurando correctamente las conexiones necesarias entre el dispositivo y la plataforma en la nube.
- Analizar y preparar datos históricos del SENAMHI, para alimentar al modelo predictivo.
- Elección del algoritmo más óptimo para obtener predicciones más precisas.
- Implementar un modelo predictivo en la nube que sea preciso, y que permita obtener una predicción correcta.

1.3. Justificación

(Idexcam, 2017), advirtió que “Los efectos del cambio climático vienen impactando negativamente en la producción y exportación de algunos productos agrícolas, como es el caso del mango, espárragos, quinua, cacao, cebolla y azúcar”.

Los mercados nacionales e internacional ofrecen estaciones meteorológicas teniendo costos elevados; por lo que se propone implementar una mini estación meteorológica con un costo bajo. Utilizando el dispositivo ESP32 basado en Arduino al cual se conectarán un conjunto de sensores quienes realizarán las lecturas de los valores numéricos de las variables meteorológicas.

A través de la implementación de la mini estación meteorológica empleando hardware y software libre como el ESP32 basado en Arduino que permitirá la captura de datos en tiempo real y estos datos serán transmitidos y procesados para alimentar al modelo de ML implementado en la nube.

La minería de datos en el sector agrario es un tema emergente debido a la masiva cantidad de datos que son generados diariamente en las estaciones meteorológicas y estaciones hídricas del Perú (SENAMHI), por lo que estos datos servirían para la extracción

de conocimientos ocultos en ellos; en ese sentido se pretende implementar un modelo de ML en la nube el cual realizara predicciones, con los resultados de la predicción se permitirá entender y facilitar los procesos de variabilidad agroclimática permitiendo conducir hacia la mejor toma de decisiones por parte de los gremios agrarios y agricultores de la región.

1.4.Presuposición filosófica

Dios nos otorgó la tierra para cultivarla y cuidar de ella, con el fin de que no muramos de hambre, entonces, (Génesis 2:15), “Tomó, pues Jehová Dios al hombre, y le puso en el huerto de Edén, para que lo labrara y lo guardase”. Sin embargo, por actividades realizadas por el hombre el medio ambiente al pasar los años se ha ido deteriorando y haciendo el trabajo de cultivar cada vez más difícil por las tempestades medioambientales producida desde la revolución industrial.

CAPÍTULO II. Revisión de la literatura

2.1 Revisión de literatura

Para el respaldo de la investigación, se realizó la revisión de las diferentes fuentes comprendidas en el marco de esta investigación, que serán descritas de manera resumida.

La Investigación de (Gnoza & Barberena, 2018), con el título “Estudio de factibilidad del uso de Machine Learning con múltiples fuentes de datos en el pronóstico del tiempo” cuyo objetivo fue desarrollar una herramienta predictiva y PWS para el estudio de la factibilidad de uso de las predicciones meteorológicas en Uruguay. En la cual se realizó la aplicación de captura de datos en Bluemix de IBM y en Azure, creación del modelo predictivo y el ensamblado de la mini estación meteorológica. Como resultados se logró predecir en forma discreta (lluvia o no lluvia) y en forma continua (valor), las precipitaciones en función a las variables meteorológicas temperatura, humedad y presión atmosférica, finalmente son mostrados en la plataforma de Power BI, la lectura de datos de la PWS como el modelo y su resultado predictivo.

(Iribarren, 2016), realizó una investigación titulada “Modelo Predictivo. Machine Learning aplicado al análisis de datos climáticos capturados por una placa sparkfun”, con el objetivo de captar las condiciones meteorológicas para la predicción y luego transmitir y procesar los datos obtenidos al modelo predictivo creada en la nube, realizando las configuraciones necesarias entre el dispositivo y la plataforma en la nube y mostrar los resultados de una forma visual y sencilla de comprender. Se utilizó una metodología propia de configurar la placa sparkfun, configurar la nube IoT Hub, desarrollo del modelo Azure. preparación de datos, elección del algoritmo, y finalmente despliegue del modelo. Como resultado se obtuvo la elección del algoritmo “Multiclass decisión forest”, quien obtuvo una predicción más óptima.

(Perez, Matadamas, Morales, & Altamirano, 2017), con el título “Prototipo basado en Redes Neuronales para monitoreo y predicción de temperatura en invernaderos de los Valles Centrales de Oaxaca” con el objetivo de desarrollar un sistema para el monitoreo y la predicción de la temperatura construido con tecnología accesible que utilice Redes Neuronales Artificiales, utilizando como patrones de entrada información climática de la región de los Valles Centrales del estado de Oaxaca. Cuyo resultado fue el entrenamiento y la validación de cada una de las configuraciones, realizándose ocho pruebas, cambiando en cada una el número de neuronas en la capa oculta, para poder comparar el desempeño de la red.

(Escribano, 2017), en su investigación “Implementación de una estación meteorológica con arduino”, con el objetivo de captar parámetros meteorológicos mediante sensores conectados al Arduino, mostrando resultados en un dispositivo de telefonía móvil. Con el desarrollo de una metodología propia, realizando el análisis de las necesidades globales del sistema, elección de los dispositivos, diseño y montaje del sistema, elección del algoritmo óptimo para la implementación del código, ejecución de pruebas que afianzarán la fiabilidad del proyecto. El resultado obtenido fue la comprobación del funcionamiento del sistema por el monitor de arduino de la aplicación Weather Station generada por App Inventor.

(Ferrer, 2017), en su investigación “Desarrollo de una estación meteorológica autónoma de bajo costo, con el objetivo de desarrollar una herramienta tecnológica que permita estudiar el comportamiento de variables ambientales y de esta forma poder comprender el comportamiento de los microclimas el cual puede ser un insumo para la agricultura de precisión, realizada con la metodología propia de seis fases: selección de las variables, acondicionamiento de los sensores y módulos, diseño de la base de datos,

programación de los módulos, desarrollo de la interfaz gráfica, diseño del modelo de energía. Los resultados obtenidos fueron: desarrollo y funcionamiento de la estación meteorológica, mostrando en una página web los datos monitoreados, para así se permita un estudio más completo y detallado de los eventos que afectan los cultivos.

2.2 Marco teórico

En esta sección introduciremos los conceptos básicos y conocimientos de relevancia para esta investigación en lo que respecta a Machine Learning e IoT, los servicios de Microsoft Azure, ya que el caso de estudio consiste en un modelo predictivo y un PWS herramienta para capturar datos meteorológicos desplegada en un dispositivo IoT de Azure IoT Hub, también se describe los servicios de Microsoft Azure y el hardware que fue utilizada en dicha solución IoT, es decir, la placa y sensores meteorológicos. Finalmente, se describe la metodología de desarrollo para el Modelo predictivo y la solución IoT.

2.2.1. Machine learning

Desarrollo de técnicas para que las computadoras aprendan, basada en análisis de datos. Estudia algoritmos que tienen la capacidad de aprender a realizar una tarea automáticamente, mejorando su performance con la experiencia. El entrenamiento se realiza utilizando un algoritmo que permite obtener un modelo capaz de realizar predicciones a partir de nuevos datos. Tomar en cuenta que el algoritmo utilizado para resolver un problema específico depende del tipo de problema que se estudiara.

Permiten resolver tareas complejas, cuya solución es muy difícil o imposible de realizar manualmente. Existen dos tipos de aprendizaje automático:

2.2.1.1. Machine Learning no Supervisado

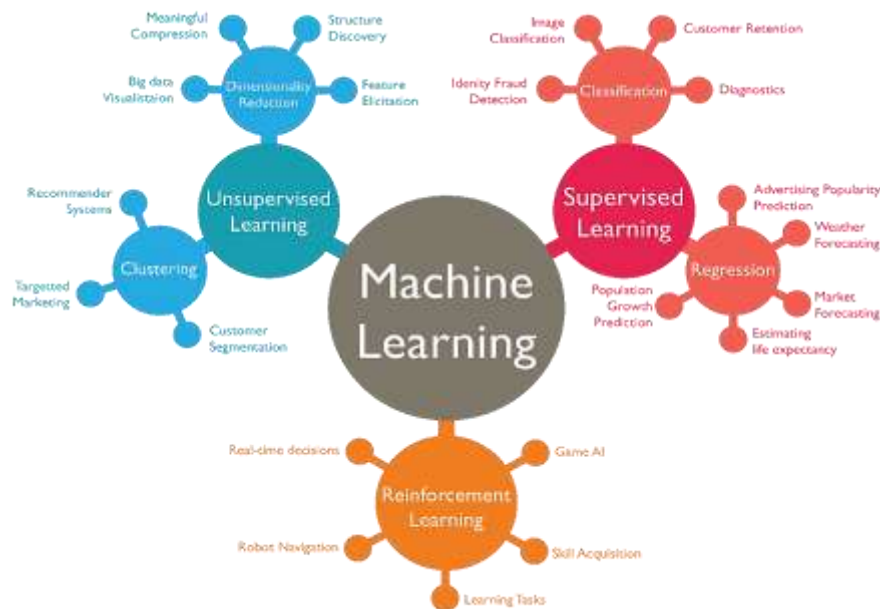
(IAAR; López, Raúl;, s.f.), se cita lo siguiente “En los problemas de aprendizaje no supervisado el algoritmo es entrenado usando un conjunto de datos que no tiene ninguna

etiqueta; en este caso, nunca se le dice al algoritmo lo que representan los datos. La idea es que el algoritmo pueda encontrar por si solo patrones que ayuden a entender el conjunto de datos”.

El objetivo es encontrar patrones a partir del análisis de datos, un ejemplo frecuente es la segmentación de clientes con atributos idénticos a los de las campañas de marketing.

2.2.1.2. Machine Learning Supervisado

“Los algoritmos trabajan con datos etiquetados, intentado encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida. El algoritmo se entrena con un histórico de datos y así aprende a asignar la etiqueta de salida adecuada a un nuevo valor (predice el valor de salida)”.(Simeone, 2018).



Figuras 3. Machine Learning
Fuente: towardsdatascience.com

2.2.2. Modelo predictivo

Es un análisis de datos (comportamiento observado) que posteriormente debe entrar en un proceso denominado entrenamiento, luego realizar un proceso de estimación del comportamiento previsto.

2.2.3. Datos Históricos

Información de diferentes tipos que fueron registrados en la historia de cualquier índole. Conjunto almacenado de información que puede constar de 5 años a más.

2.2.4. Series temporales

“Es una colección de observaciones hechas secuencialmente a lo largo del tiempo. En cada punto de medición en el tiempo, pueden ser monitoreados uno o más atributos, y la serie temporal resultante es llamada univariada o multivariada, respectivamente. En muchos casos, una secuencia de símbolos puede ser usada para representar una serie temporal”.(Mamani, 2011).

2.2.5. Algoritmos

Conjunto de cálculos y una serie de pasos ordenados o instrucciones que llevan a la solución de un problema concreto. Para el desarrollo de la investigación se plantean distintos algoritmos de Machine Learning, de las cuales se escogerá el más óptimo y que tenga porcentaje de incertidumbre mínima. Las que están dentro de las opciones se mencionan a continuación.

2.2.6. Multiclass Decision Jungle

Multiclass Decision Jungle es un modelo de machine learning que se basa en un algoritmo de aprendizaje supervisado llamado junglas de decisión. El modelo se puede usar para predecir un objetivo que tiene valores múltiples.

2.2.7. Logistic Regression

(Amat, 2016), “es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. Una de las principales aplicaciones de la regresión logística es la de clasificación binaria, en el que las

observaciones se clasifican en un grupo u otro dependiendo del valor que tome la variable empleada como predictor”.

2.2.8. Naive Bayes

Realiza la técnica de clasificación y predicción que construye modelos que predicen la probabilidad de posibles resultados. Naive Bayes hace uso de datos históricos para poder localizar las relaciones y asociaciones, para luego realizar predicciones.

2.2.9. Linear Regression

La regresión lineal se desarrolló en el campo de la estadística y se estudia como un modelo para comprender la relación entre las variables numéricas de entrada y salida, pero ha sido tomada en préstamo por el aprendizaje automático. Es a la vez un algoritmo estadístico y un algoritmo de aprendizaje automático.

2.2.10. Internet of Things (IoT)

Podemos definirla como un sistema de dispositivos interconectados, máquinas mecánicas y digitales, objetos (computadora, Tablet, Smartphone, cerradura de una puerta, etc), animales o persona, a las cuales se ha provisto de identificadores únicos (UID) y la habilidad de transferir datos sobre una red sin la necesidad de la interacción humana.

(ITU-T, 2019), “Infraestructura de red global dinámica con capacidades auto configurables, basadas en protocolos de comunicación estándares e interoperables en donde las cosas físicas y virtuales: Tienen identidades, atributos físicos y personalidades virtuales, Utilizan interfaces inteligentes y se integran a la perfección en la red de información.

El internet de las cosas tiene múltiples aplicaciones en diferentes ámbitos, tales como: en la salud, wearables (bandas fitness para el monitoreo de gasto calórico y pulsaciones cardíacas y otros) permitiendo a los médicos realizar un seguimiento de las condiciones del

Actualmente en el mercado existen diferentes plataformas y middlewares quienes pocibilitan el desarrollo de sistemas basado en IoT, alguna de estas plataformas es ofrecida por medio de servicios en la nube como: Google Cloud IoT Core, Microsoft Azure IoT, Amazon Web Services IoT.

2.2.11.1. Microsoft Azure

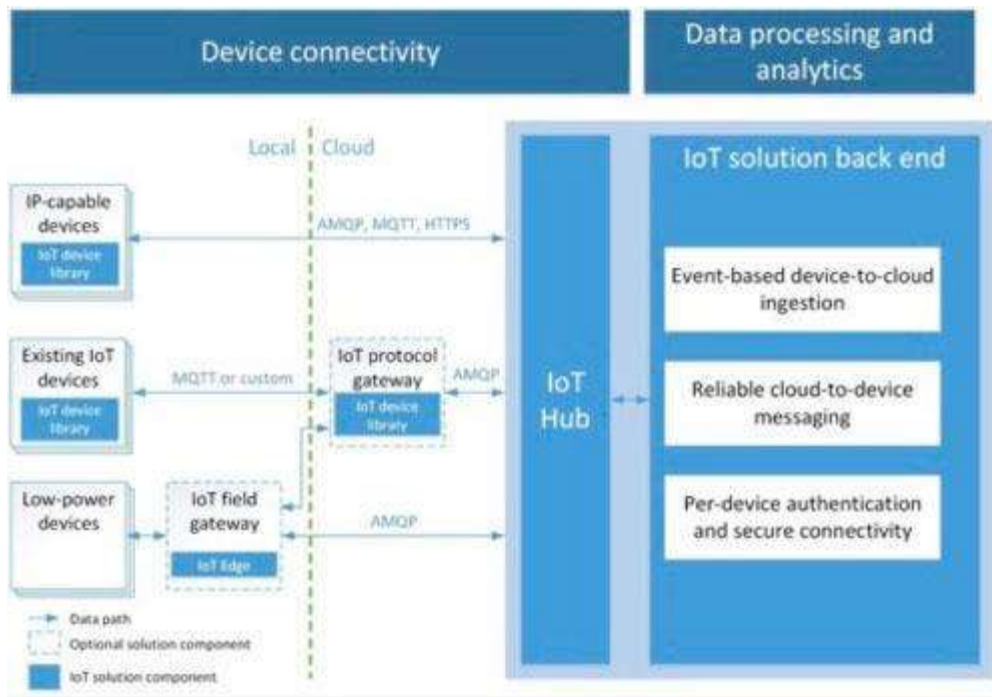
(Javier, 2019), “Microsoft Azure es conjunto en constante expansión de servicios en la nube para ayudar a su organización a satisfacer sus necesidades comerciales. Al ser un servicio de pago por uso, se adapta a las necesidades del usuario siendo escalable en tiempo real, tanto en cantidad como en aumento del rendimiento”.



Figuras 5. Nube de Microsoft Azure

Fuente: Microsoft

Es una nube híbrida que nos ofrece servicios “PaaS”. Por un lado, se creará una interfaz de “IoT” al que llegarán los datos obtenidos de los sensores de las diferentes placas como Raspberry Pi, NodeMCU ESP8266, ESP-32, MXChip de Microsoft, entre otros en tiempo real. Por otro lado, se hará uso de Machine Learning de Azure para la creación, realizar el entrenamiento y despliegue de un modelo de predicción que se basa en los datos históricos de meteorología e hidrología obtenidas de diferentes fuentes.



Figuras 6. Arquitectura de Microsoft Azure IoT
Fuente: Cioperu.pe

a) Administrador de Recursos

El administrador de recursos, nos permitió trabajar con los recursos de una aplicación como un grupo, de tal manera que se implementó, administro y superviso todos los recursos de la solución de forma singular. También se han creado grupos diferentes de entornos de desarrollo, test y producción.

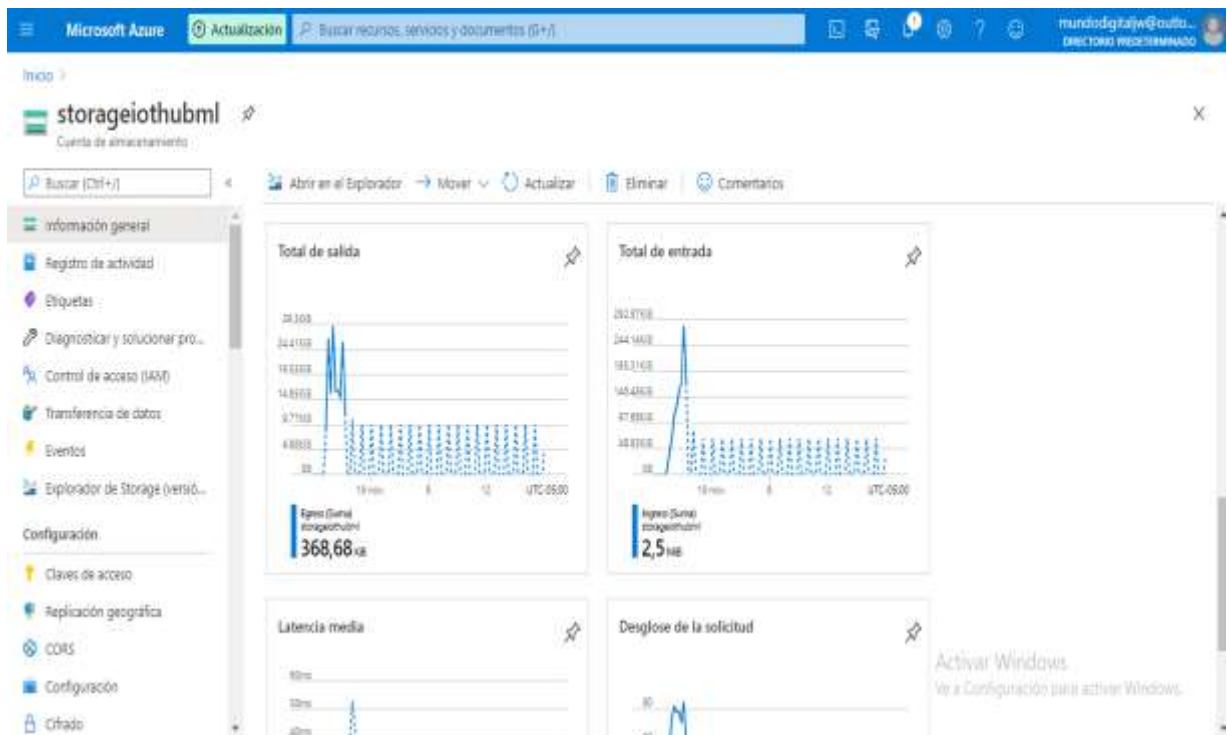
Recurso: Instancia de servicio de Microsoft Azure.

Grupo de recursos: Es un grupo lógico de recursos, con reglas a considerar:

- Compartir mismo ciclo de vida, un recurso solo existe dentro de un grupo de recursos y es factible trasladar un determinado recurso de un grupo a otro.

b) Cuenta de almacenamiento

La herramienta de Microsoft nos facilita servicios de almacenamiento persistente, escalable y a la vez se realizan backus y mantenimiento.



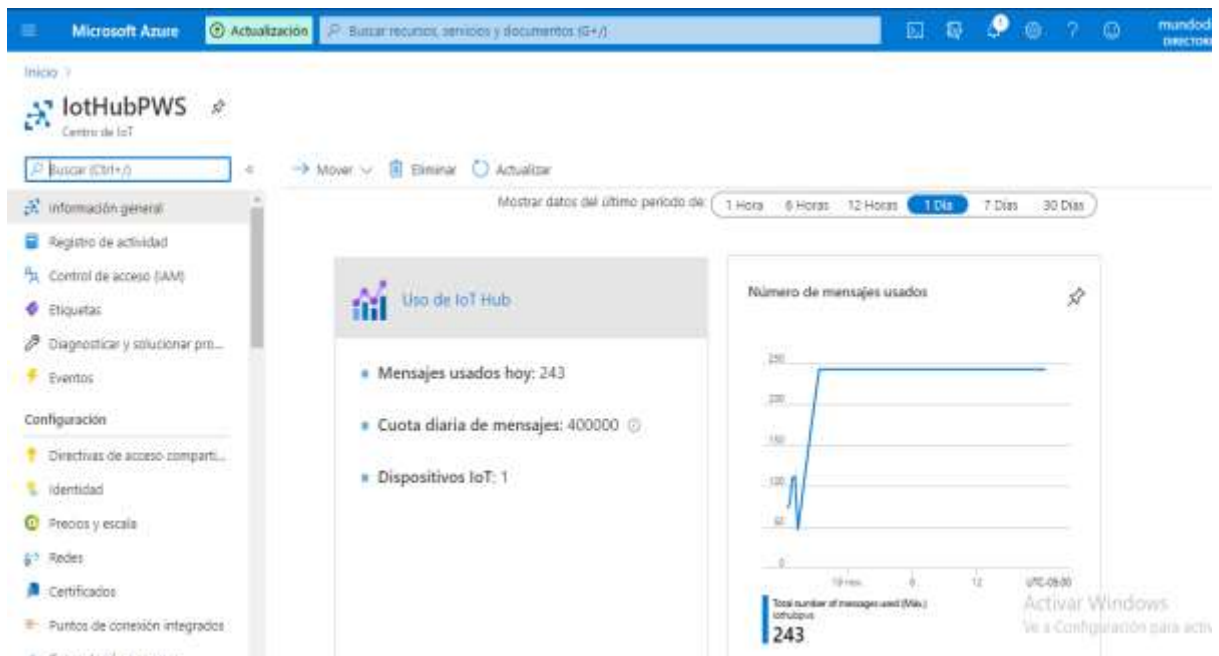
Figuras 7. Cuenta de almacenamiento – Microsoft Azure
 Fuente: Elaboración Propia

c) IoT Hub

Servicio diseñado que permite un paso de mensajes bidireccionales entre el dispositivo de IoT y la nube, las características principales son:

- Permite la comunicación entre el dispositivo y la nube y viceversa.
- Admite protocolos, AMQP MQTT y HTTP.
- Permite subida de archivos de distintos dispositivos a la nube.
- Facilita una seguridad particular por cada dispositivo.

Para crea un IoT Hub desde el portal de Azure podemos seguir la documentación oficial de Microsoft Azure en el siguiente enlace: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal>.

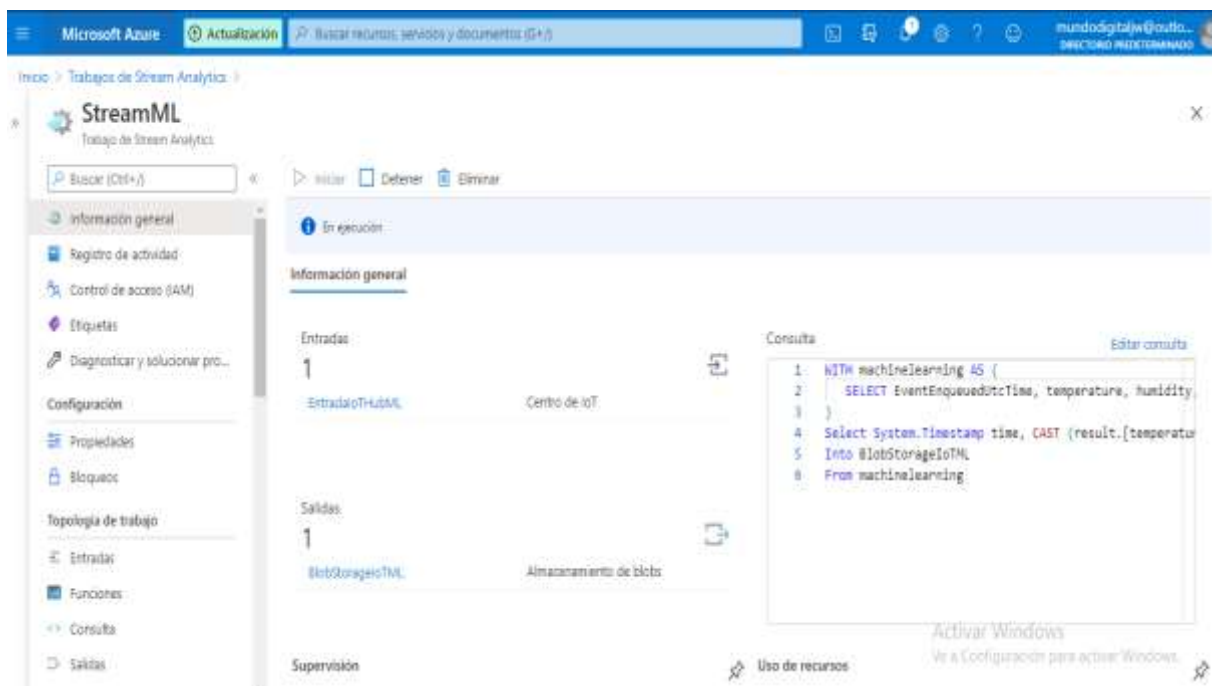


Figuras 8. Centro de IoT – Microsoft Azure

Fuente: Elaboración propia

d) Stream Analytics

Servicio de procesamiento y análisis de eventos completamente administrables en tiempo real sobre flujo de datos. Los datos pueden ser capturados de dispositivos, modulo y/o sensores, páginas web, redes sociales, etc.



Figuras 9. Stream Analytics – Microsoft Azure

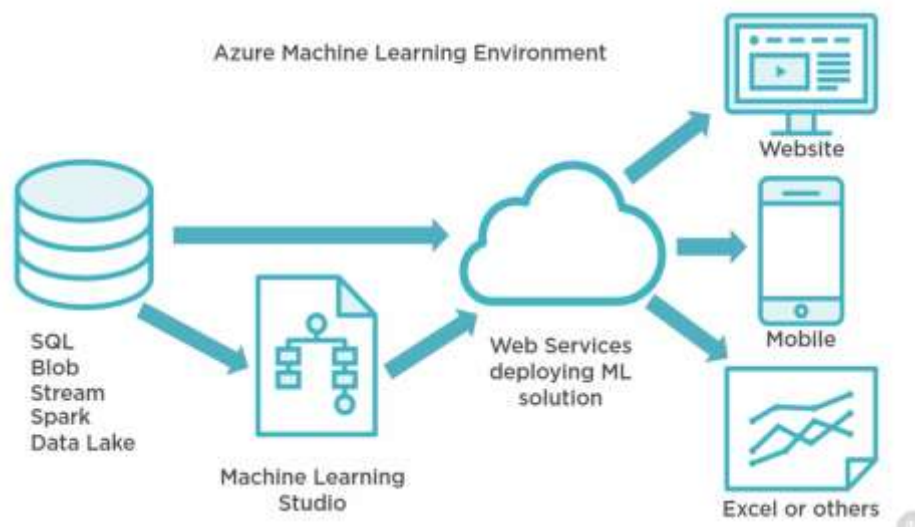
Fuente: Elaboración propia

2.2.11.2. Azure Machine Learning

Dispone de un servicio de aprendizaje que permite crear algoritmos de aplicaciones y sitios web inteligentes, utilizando Scikit-learn, TensorFlow u otra herramienta de Python y que permite entrenar los modelos de forma local o en la nube. Cognitive Services tiene un conjunto de medios para desarrollar sitios web, bots y algoritmos inteligentes con la capacidad de oír, emitir voz, aprendizaje de las necesidades de usuarios mediante interpretación del lenguaje natural.

2.2.12. Azure Machine Learning Studio

Posee un entorno de trabajo visual sumamente intuitivo en el que podrás entrenar tus modelos de aprendizaje y alimentarlo con facilidad gracias a las fuentes de datos y series de funciones de procesamiento que se encuentran disponibles. Entre sus ventajas más llamativas puede destacarse que no requerimos mayores esfuerzos de generación de código para incluir, editar y ejecutar los conjuntos de datos de nuestros modelos de machine learning.



Figuras 10. Azure ML Studio – Microsoft
Fuente: wikiazure.com

2.2.13. Microsoft-Power Bi

Es un servicio accesible y gratuito de análisis de negocio basado en la nube, permite también unir diversas fuentes de datos. Power Bi permite administrar la integridad de un negocio mediante una vista paneles en tiempo real, generar informes amigables y estos pueden ser consultados accediendo a los datos desde el punto en que uno se encuentre por medio de apps.



Figuras 11. Microsoft-Power Bi

Fuente: Power Bi/objetivoanalista.com

2.2.14. Hardware libre

2.2.14.1. Placa de desarrollo ESP-32 (DEVKIT V1 NodeMCU-32)

Según, (Naylamp Mechatronics, s.f.), es “la herramienta más potente en hardware libre para el prototipado de proyectos Internet de las cosas (IoT). la placa integra el SoM ESP-WROOM-32 que tiene como base al ESP32-SoC, el convertidor USB-serial CP2102 es necesario para la programación por USB con el módulo ESP32, los reguladores de voltaje y leds indicadores. La plataforma o placa ESP32 es la transformación del ESP8266 mejorando sus capacidades de comunicación, y a nivel de conectividad permite la utilización de diversos protocolos de comunicación inalámbrica tales como: WiFi, Bluetooth y entre

otros. Sin embargo, el procesamiento de su CPU es de 32-bit de dos núcleos de hasta 240Mhz que se pueden controlar independientemente. Además, puede ser aplicado en Mini Servidores Web, en procesamiento digital, Cámara IP, Webcams, la Domotica y la robótica móvil”. El ESP-32 es usada también como repetidores wifi, permitiendo acceder al dispositivo y configúralo.

Especificaciones del ESP-32:

- El Voltaje de Salida y entrada es: 3.3V DC
- El Voltaje de Alimentación (USB) es: 5V DC
- La Placa: ESP32 DEVKIT V1
- El SoM: ESP-WROOM-32
- El CPU: Dual core Tensilica Xtensa LX6 (32 bit)
- El Procesador secundario: realiza operaciones básicas en modo de ultra bajo consumo Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- El Modulo Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)
- La Memoria: 448 KByte ROM - 520 KByte SRAM - 16 KByte SRAM in RTC
- Los Pines Digitales GPIO son: 24 (Algunos pines solo como entrada)
- Los Pines PWM son: 16
- Los Pines Analógicos ADC son: 18 (3.3V, 12bit: 4095, tipo SAR, ganancia programable)
- Conversor Digital a Analógico DAC: 2 (8bit)
- La Seguridad es: Estándares IEEE 802.11 incluyendo WFA, WPA/WPA2 and WAPI 1024-bit OTP, up to 768-bit for customers



Figuras 12. Placa de desarrollo ESP-32

Fuente: hifi.sac.com

Para aumentar la cobertura de la comunicación wifi existen muchas tecnologías entre ellos se encuentra los repetidores wifi, tecnología para extender la cobertura de una red teniendo en cuenta las dos bandas de frecuencia 2.4 GHz y 5GHz y utilizar la tecnología según la distancia deseada.

2.2.14.2. Placa de desarrollo NodeMCU-ESP8266

Según, (Sandorobotics, 2019), “el módulo o placa Node MCU es una pequeña placa Wifi lista para ser usado en cualquier proyecto de IoT. Está diseñado por del conocido chip ESP8266, esto nos ofrece una solución completa y autónoma de redes inalámbricas Wi-Fi, lo que le permite hospedar la aplicación o servir como puente entre el microcontrolador y el Internet, y expone todos sus pines en los extremos. Además, nos ofrece más ventajas como la incorporación de un regulador de tensión de manera integrada, así como los puertos de USB para la programación. Se puede programar con LUA o mediante el IDE de Arduino”.

Especificaciones de NodeMCU-ESP8266:

- El Procesador: ESP8266 @ 80MHz (3.3V) (ESP-12E)
- 4MB de memoria de FLASH (32 MBit)
- WiFi 802.11 b/g/n
- Regulador 3.3V integrado (500mA)
- El Conversor USB-Serial CH340
- Función Auto-reset
- 9 pines GPIO con I2C y SPI

- entrada analógica (1.0V max)
- agujeros de montaje (3mm)
- 1 Botón de RESET
- 1 Entrada alimentación externa VIN (20V max)



Figuras 13. Placa de desarrollo NodeMCU-ESP8266

Fuente: hifi.sac.com

2.2.14.3. Placa Arduino

(Arduino, 2018), “Es una plataforma electrónica de código abierto basada en hardware y software que son fáciles de usar, se pueden leer entradas como: luz en un sensor, un dedo en un botón y transformarla en una salida como, por ejemplo: encender un motor, encender o apagar un LED, también se puede decir a la tarjeta qué debe hacer cuando se envía un conjunto de instrucciones al micro controlador. Para construirlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el software Arduino (IDE), basado en el procesamiento”.



Figuras 14. Placa ArduinoUNO

Fuente: <http://arduino.cl/arduino-uno/>

2.2.14.4. SparkFun Weather Shield

(Sparkfun, s.f.), “este accesorio permite medir la temperatura, humedad relativa, luminosidad y presión barométrica a través de los sensores que tiene incorporada. También hay conexiones a sensores opcionales como el pluviómetro, el GPS y la velocidad del viento, El Weather Shield utiliza el sensor de temperatura / humedad, el sensor de presión barométrica MPL3115A2 y el sensor de luz ALS-PT19. El escudo se basa en las bibliotecas de Arduino Si7021 y MPL3115A2”.



Figuras 15. SparkFun Weather Shield

Fuente: <https://www.sparkfun.com/products/13956>

2.2.14.5. Sensores

Dispositivos que miden las variables físicas y las transforma en variables eléctricas. Están siempre en contacto con la variable que miden, consiguiendo una frecuencia de obtención de medidas muy altas.

a) Sensor o módulo DHTXX

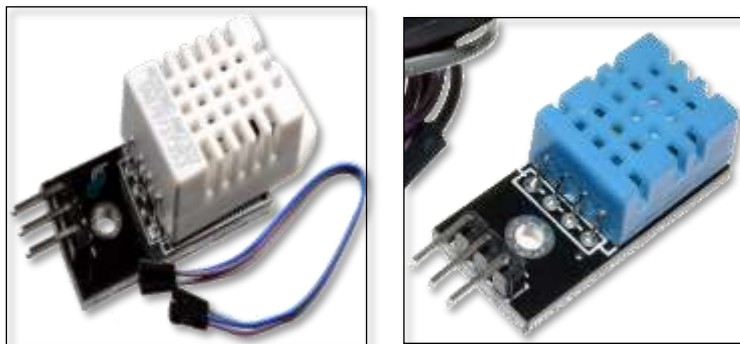
Los módulos o sensores DHT 11 y DHT 22 son dos modelos de una misma familia de sensores, que nos permiten realizar la medición simultánea de humedad y temperatura utilizando además un único hilo para la comunicación de los datos vía terminal serie, para lo cual ambos disponen de un procesador interno que realiza el proceso de medición,

proporcionando la medición mediante una señal digital, por lo que resulta muy sencillo obtener la medición desde un microprocesador como la placa de Arduino, ESP8266 o ESP-32.

El modulo o sensor DHT 11 realiza medidas de temperatura de 0 - 50 grados, con precisión de 2°C, y la humedad de 20 - 80%, con una precisión del 5% con una frecuencia de muestreo es de 1 muestra por segundo (1 Hz).

El modelo DHT 22, mide la variable de temperatura entre el rango de -40 a 125, con precisión de 0.5°C, mide la variable humedad de 0 - 100%, con una precisión del 2-5% y finalmente la frecuencia de muestreo es entre dos muestras por segundo (2 Hz).

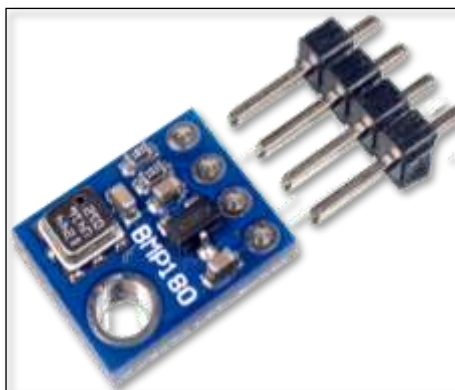
Es clara la superioridad del sensor DHT 22, quien tiene características más óptimas.



Figuras 16. Sensor DHT 22 y DHT 11
Fuente: hifi.sac.com

b) Sensor o módulo BMP

El Modulo sensor de BMP 180 calcula la presión del aire y temperatura, la clavija de alimentación debe ser conectada a la línea de 3,2 voltios de la placa Node MCU y el conector de tierra conectar al pin GND de la placa NodeMCU.



Figuras 17. Sensor BMP180

Fuente: hifi.sac.com

c) TP4056 C USB cargador de baterías de litio

El cargador de baterías que he seleccionado es el TP4056. Además de cargar la batería de forma segura y controlada, cuenta con un sistema de protección de sobretensión, que detiene la carga en cuanto la tensión de batería alcanza los 4.2V. Ofrece una corriente de carga de hasta 1 amperio.



Figuras 18. Cargador de batería – TP4056

Fuente: hifi.sac.com

d) Placa Solar

Un panel solar es un dispositivo que nos permite aprovechar la energía del sol y usarla como fuente de energía eléctrica. Estos paneles constan de celdas, dichas celdas son

dependientes del efecto fotovoltaico (células fotovoltaicas), la energía de la radiación solar ilumina las celdas y generan cargas positivas y negativas, para luego producir un campo eléctrico que luego genera energía eléctrica.



Figuras 19. Placa solar – 7v
Fuente: hifi.sac.com

e) Batería de 3.7V 3000mAH ION litio Recargable



Figuras 20. Batería litio recargable
Fuente: hifi.sac.com

2.2.15. Agro clima

Ciencia que estudia las condiciones meteorológicas, climáticas e hidrológicas y su intervención en los procesos de la producción agrícola. Estudiando el tiempo atmosférico, facilitó apoyo a la actividad agrícola.

2.2.16. Estación meteorológica

Una estación meteorológica nos permite medir y registrar distintas variables meteorológicas o climáticas de acuerdo a las necesidades de la aplicación en la que se desea emplear. La estación meteorológica está conformada por sensores de alta precisión, entre las medidas que se realizan son de las siguientes variables meteorológicas: humedad, temperatura, presión atmosférica, humedad relativa, velocidad del viento, luminosidad, ubicación, etc., seguidamente son enviados a un subsistema de consola y está a la vez envía los valores tomados al software de almacenamiento.

Las estaciones se clasifican según la Organización Meteorológica Mundial (OMM) según lo siguiente.

A. De acuerdo a su finalidad:

- ✓ Agrícolas
- ✓ Climatológicas
- ✓ Sinópticas
- ✓ Aeronáuticas
- ✓ Especiales

B. Según la magnitud de observaciones:

- ✓ Auxiliares
- ✓ Principales
- ✓ Ordinarias

C. Según el nivel de observación:

- ✓ Superficie
- ✓ Altitud

D. Por el lugar de observación:

- ✓ Terrestre
- ✓ Marítimas

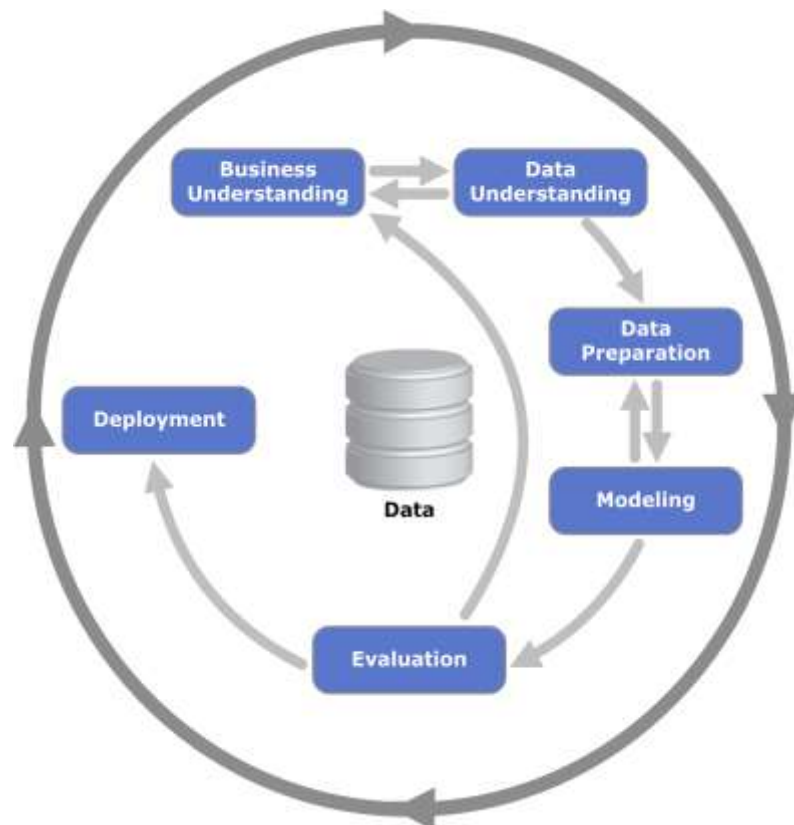
✓ Aéreas

2.2.17. SENAMHI

Es el organismo técnico especializado de nuestro estado peruano - El Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI), proporciona información del pronóstico del tiempo, brinda asesoría, estudios científicos en meteorología, hidrología, agro meteorología, temas ambientales de manera oportuna, confiable y accesible en beneficio de la sociedad.

2.2.18. Metodología CRISP DM

La metodología de trabajo a utilizar para el logro de los objetivos es CRISP-DM, este tiene organizado en forma jerárquica las actividades, la figura 21 muestra un modelo estructurado en seis fases y alguna de ellas son bidireccionales que permiten revisar la fase de forma total o parcial.

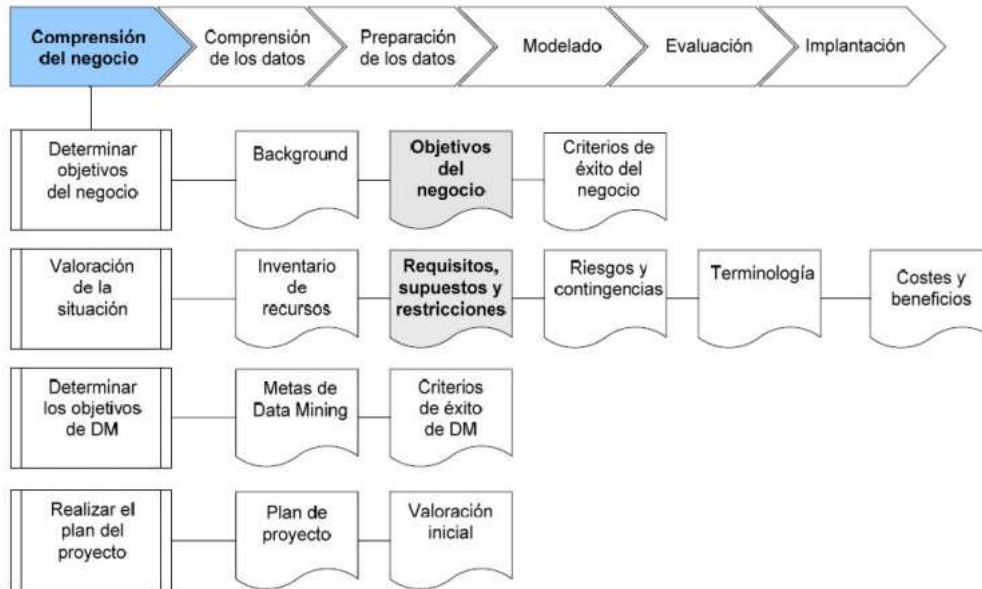


Figuras 21. Metodología CRISP-DM

Fuente: Google, tomada en el 2019.

2.2.18.1. Fases de la metodología CRISP DM

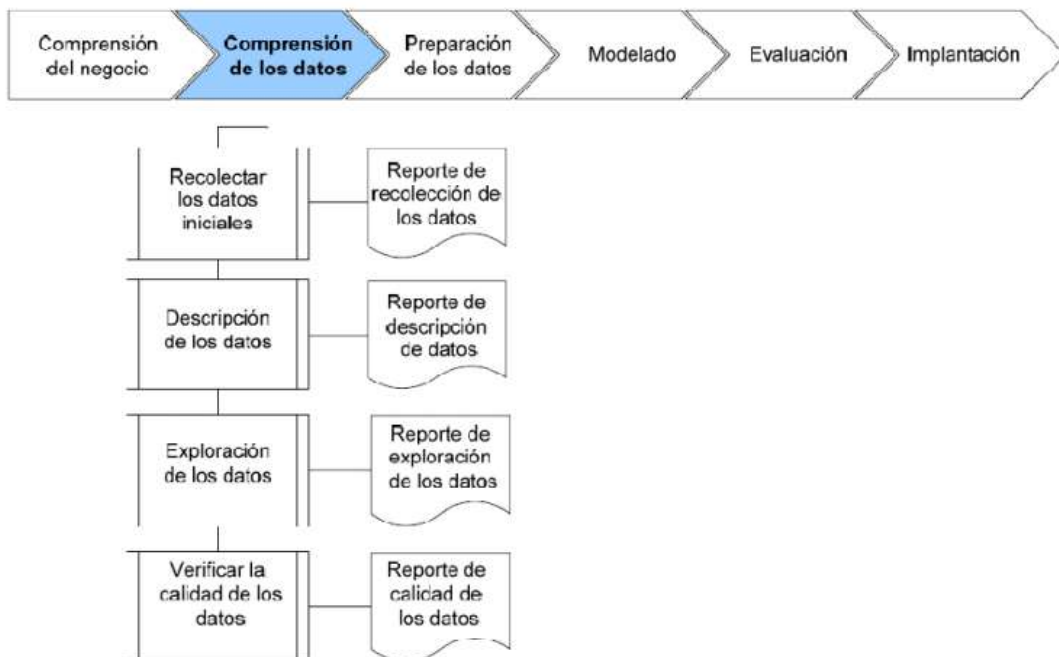
A. Comprensión del negocio



Figuras 22. Fase 1 de Metodología CRISP-DM.

Fuente: Google, tomada en el 2019.

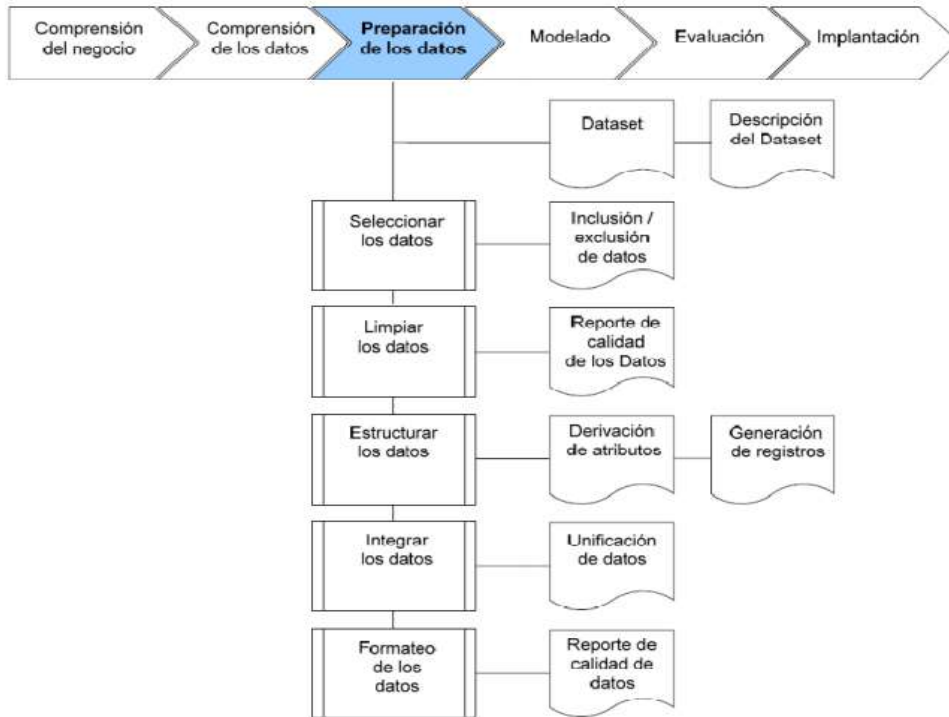
B. Comprensión de los datos



Figuras 23. Fase 2 de metodología CRISP-DM.

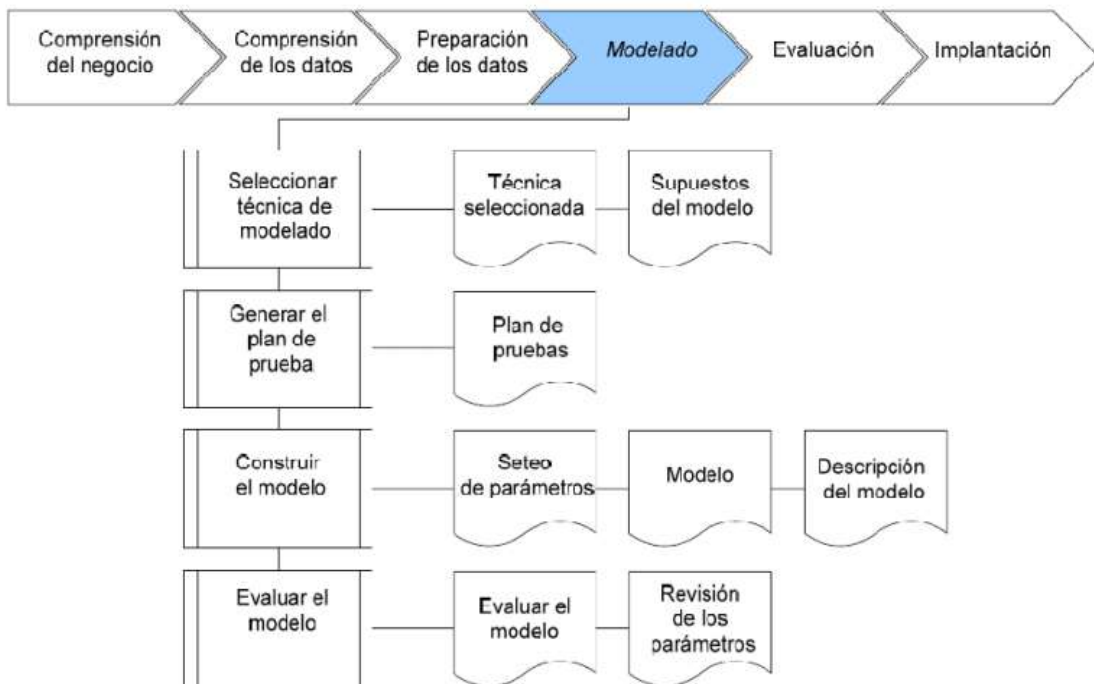
Fuente: Google, tomada en el 2019.

C. Preparación de los datos



Figuras 24. Fase 3 de metodología CRISP-DM.
Fuente: Google, tomada en el 2019.

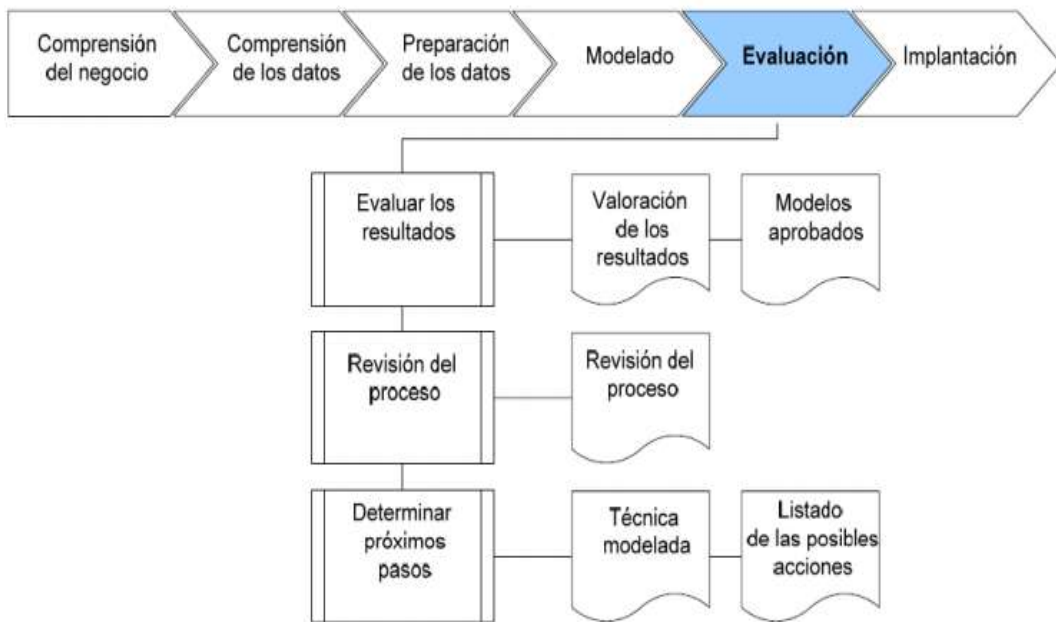
D. Modelado



Figuras 25. Fase 4 de metodología CRISP-DM

Fuente: Google, tomada en el 2019.

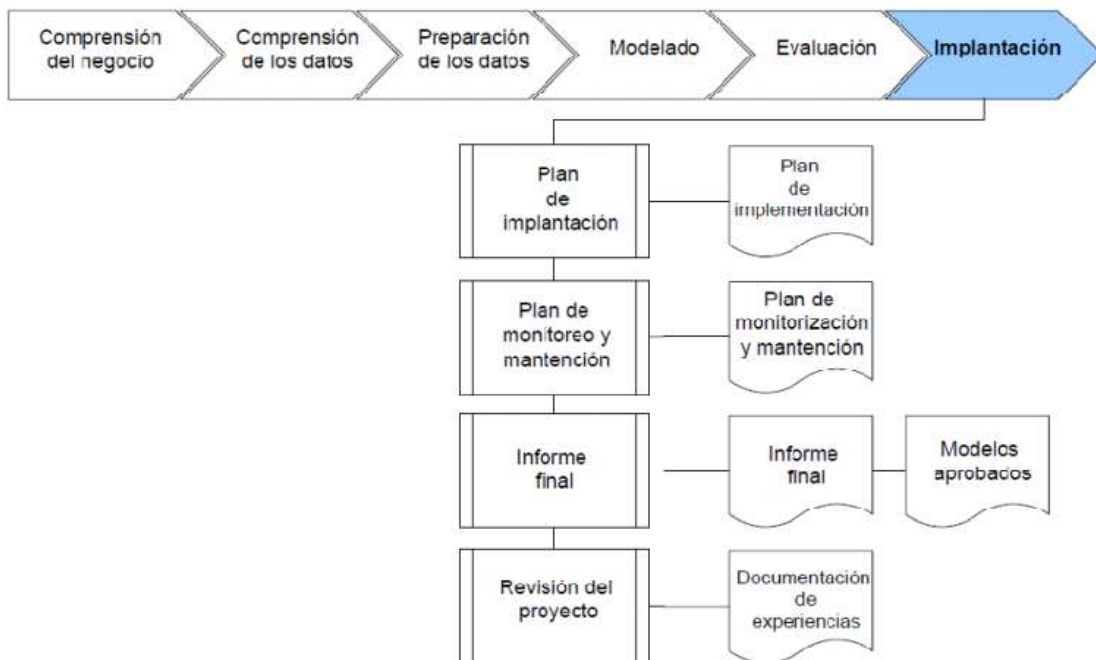
E. Evaluación



Figuras 26. Fase 5 de metodología CRISP-DM

Fuente: Google, tomada en el 2019.

F. Despliegue



Figuras 27. Fase 6 de metodología CRISP-DM

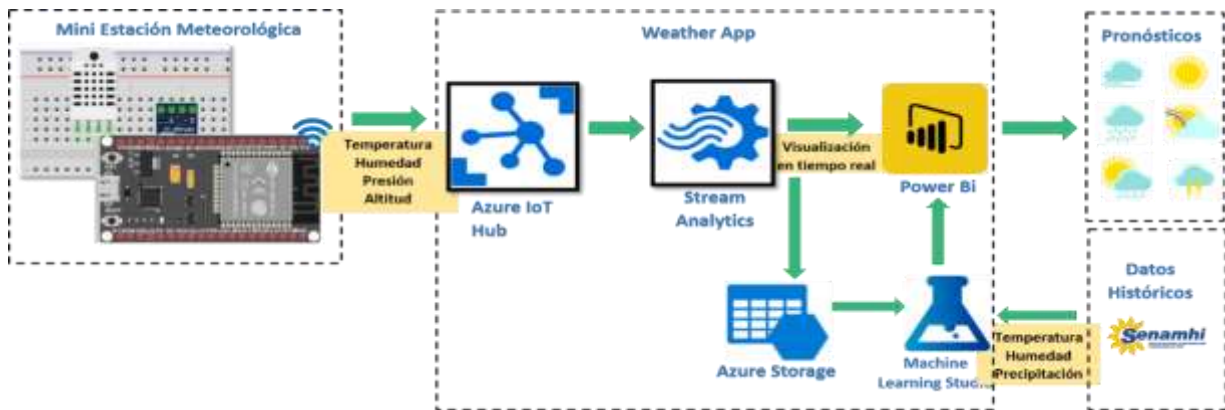
Fuente: Google, tomada en el 2019.

CAPÍTULO III. Materiales y métodos

3.1. Tipo de Investigación

La investigación que se realizó es de tipo tecnológico porque se implementó un modelo de inteligencia predictiva, cuantitativa porque se ha medido la variable de estudio y es de tipo longitudinal porque analiza en el tiempo las variables de estudio, aplicada y propositiva.

3.2. Arquitectura de solución



Figuras 28. Arquitectura de solución del modelo predictivo
Fuente: Elaboración propia

3.3. Materiales

3.3.1. Hardware

Tabla 1. Materiales y equipos

Ítem	Material/equipo Hardware	Cantidad
1	ESP-32	1
2	Sensor DHT 22	1
3	Sensor BMP 180	1
4	TP4056	1
5	Batería litio	1
6	Placa solar	1
7	Protoboard	2
8	Cables Jumper	2

Fuente: Elaboración propia

3.3.2. Software

Tabla 2. Software a utilizar en el desarrollo de la investigación

Ítem	Software
1	Plataforma Microsoft Azure
2	IoT Hub
3	Stream Analytics
4	Azure Machine Learning Studio
5	Power BI
6	Lenguaje de programación Python
7	Id Arduino
8	Plataforma de ThingSpeak

Fuente: Elaboración propia

3.4. Desarrollo de la Metodología

3.4.1. Implementación y configuración de PWS hasta Microsoft Azure IoT Hub

3.4.1.1. Construcción de la estructura acrílica

Se construyó una estructura acrílica en forma de casa para el PWS, y que servirá para colocarlo al exterior, con fin de capturar datos de manera correcta y precisa, se utilizó una plancha acrílica de 0.50 cm x 0.70 cm, silicona fría y otros.

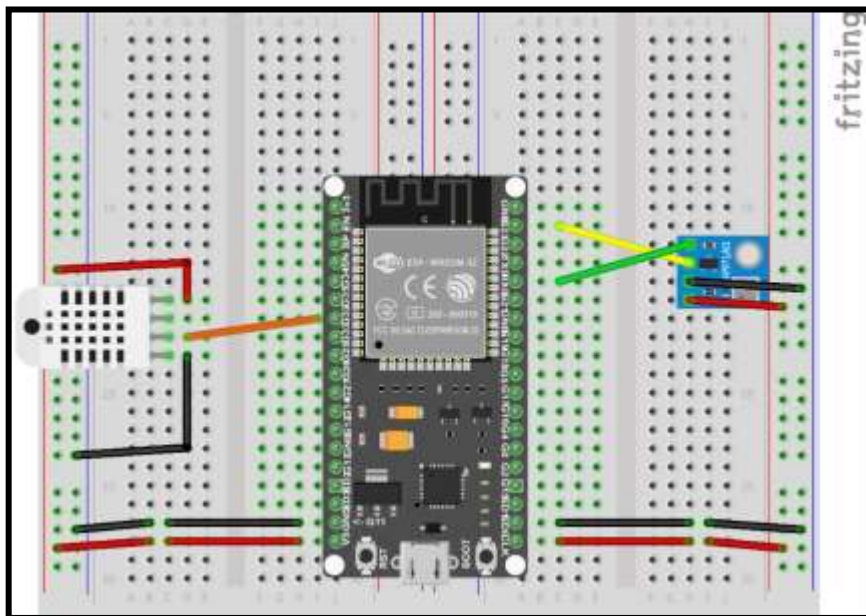


Figuras 29. Estructura acrílica – PWS

Fuente: Elaboración propia

3.4.1.2. Diagrama de conexión del PWS

Por medio de la herramienta libre fritzing se realizó el modelo inicial de la mini-estación meteorológica portable con el fin de identificar la manera óptima de realizar las conexiones necesarias entre cada uno de los dispositivos a utilizar. Por medio de la herramienta se construyó el siguiente diagrama:



Figuras 30. Diagrama de conexión del PWS

Fuente: Elaboración propia

3.4.1.3. Configuración Inicial

Para configurar la placa se realizó dos pasos usando el programa Arduino. Para esto se instalará el IDE de Arduino y los drivers de la conexión USB, luego se desarrolló el código para la toma de datos y así determinar la frecuencia a la que se medirá los parámetros meteorológicos y la configuración adecuada para enviar la información a la nube.

Para hacer uso del módulo o placa ESP 32, se requiere una librería en el IDE Arduino, esta librería se puede cargar desde el mismo IDE Arduino ingresando en Archivo / preferencias y se debe ingresar el siguiente enlace:

https://dl.espressif.com/dl/package_esp32_index.json, además se procede a cargar la librería de ESP32 en el gestor de librerías. Una vez que haya instalado los paquetes y placas ESP8266 en el IDE de Arduino, instale las siguientes librerías en el IDE de Arduino:

- ESP32_AzureIoT_Arduino
- AzureIoT Protocol_MQTT
- DHT Sensor Library
- AdafruitUnifiedSensor
- Adafruit_BMP085_Unified

3.4.1.4. Código Fuente del PWS hasta Azure IoT Hub



```
WeatherStationAzureIoT Hub
#include <WiFi.h>
#include "AzureIoT Hub.h"
#include "Esp32MQTTClient.h"

#include "DHT.h"
#include "Adafruit_BMP085.h"

#define pin2 32 //Pin del DHT22.

//Mantener activo el intervalo
#define MQTT_KEEPALIVE_INTERVAL_S 120
#define DEVICE_ID "Esp32Device"
#define MESSAGE_MAX_LEN 256

// Ingrese el SSID y la contraseña de WiFi
const char* ssid = "MOVISTAR_3C50";
const char* password = "n4Y9RUEVHIX794YUBHGW";

//Establecer msg_Interval para establecer la frecuencia de transmisión
unsigned long msg_Interval = 10100;
unsigned long msg_Timer = 0;
unsigned long stream_Interval = 20000;
unsigned long stream_Timer = 0;

/*String containing Hostname, Device Id & Device Key in the format:
```

Figuras 31. Código fuente de PWS

Fuente: Elaboración propia

Visualización de datos capturados en tiempo real con ThingSpeak

Librerías para la conexión con la plataforma de ThingSpeak

```
#include "ThingSpeak.h"
#include "WiFi.h"

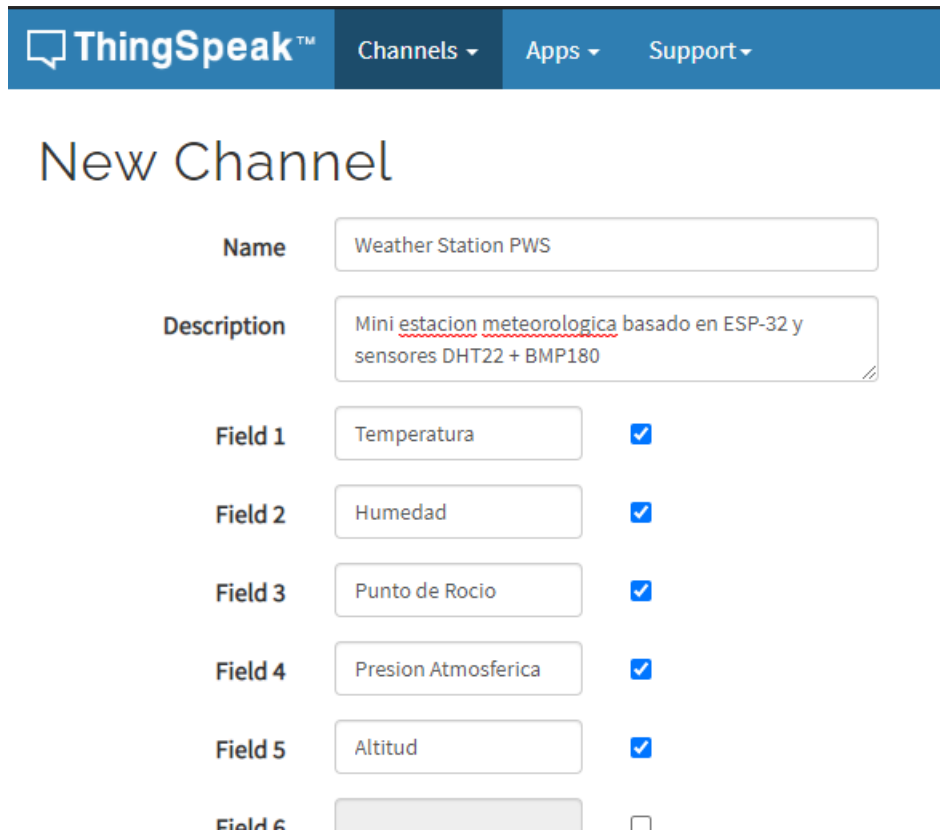
#include "DHT.h"
#include "Adafruit_BMP085.h"

-----
unsigned long channelID = 1233481;           //ID de vuestro canal.
const char* WriteAPIKey = "E75JG01DNQ6AY931"; //Write API Key de vuestro canal.
```

Figuras 32. Librerías de conexión a ThingSpeak

Fuente: Elaboración propia

Creamos un nuevo channels, para nuestras variables del clima con los Field tal como se muestra la figura 33.



ThingSpeak™ Channels Apps Support

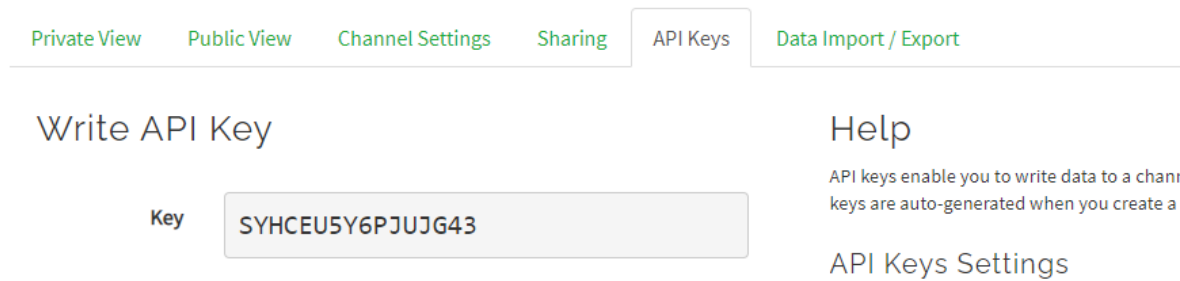
New Channel

Name	Weather Station PWS
Description	Mini estación meteorologica basado en ESP-32 y sensores DHT22 + BMP180
Field 1	Temperatura <input checked="" type="checkbox"/>
Field 2	Humedad <input checked="" type="checkbox"/>
Field 3	Punto de Rocio <input checked="" type="checkbox"/>
Field 4	Presion Atmosferica <input checked="" type="checkbox"/>
Field 5	Altitud <input checked="" type="checkbox"/>
Field 6	<input type="checkbox"/>

Figuras 33. Creando un nuevo Channels

Fuente: Elaboración propia

Es importante saber dónde se encuentra el api key el channel ID para copiar al código y esto nos permitirá conectarnos a la plataforma de ThingSpeak.



Figuras 34. Api Key Channels

Fuente: Elaboración propia

Finalmente se muestra los datos capturado en chart de cada variable que son temperatura, humedad, punto de rocío, presión y altitud.



Figuras 35. Chart de los datos capturados

Fuente: Elaboración propia

3.4.1.5. Configuración de Azure IoT Hub



Figuras 36. Estructura de solución IoT

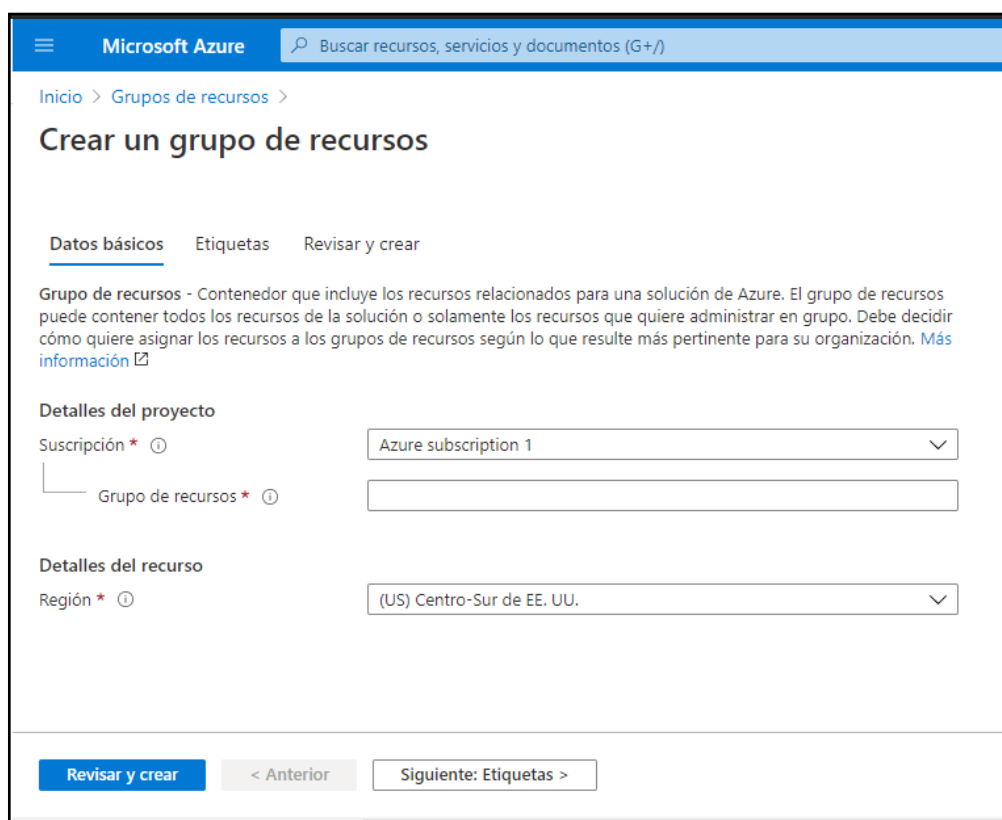
Fuente: Elaboración propia

Para la configuración del entorno en la nube, se creó y se configuro respectivamente un servicio de Azure IoT Hub en la plataforma de Azure. El servicio es administrado por la plataforma antes mencionada quien permitió una comunicación muy segura de forma bidireccional entre la solución de Azure y el dispositivo de IoT. Entre las ventajas principales tenemos:

- ✓ Monitorea y gestiona la conectividad del dispositivo.
- ✓ Librerías predeterminadas en función a los lenguajes.
- ✓ Una comunicación dispositivo-nube y viceversa de forma fiable y escalable.
- ✓ Una comunicación segura haciendo uso de credenciales y control de acceso por dispositivo.

Después de configurar el entorno, se conectó el PWS al IoT Hub haciendo uso de la plataforma de Azure IoT Hub Device Explorer. La herramienta mencionada se utilizó para gestionar los diferentes dispositivos enlazados al Azure IoT Hub. De esta forma, la información capturada por el PWS llegó a la nube, donde fue almacenada.

a) Creamos un conjunto de recursos

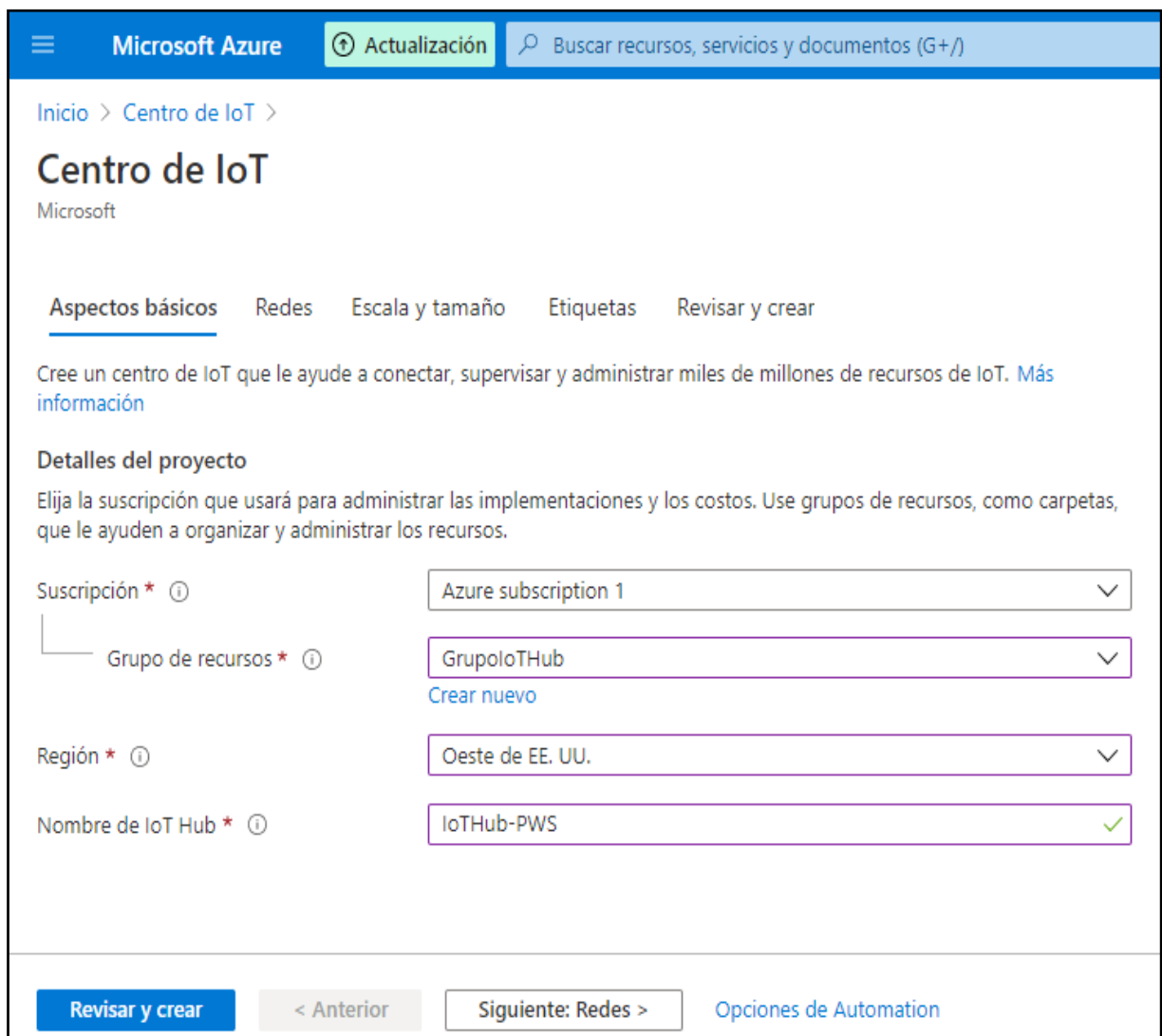


Figuras 37. Creando un conjunto de recursos para la solución IoT
Fuente: Elaboración propia

b) Creamos un centro de IoT - PWS

Describiremos cómo se procedió a crear un centro de IoT con Azure Portal.

1. En la pestaña Aspectos Básicos, complete los campos de la siguiente manera:
 - a. **Suscripción:** Seleccionamos la suscripción que se utilizara para el IoT Hub.
 - b. **Grupo de recursos:** Seleccionar un grupo de recursos o crear uno nuevo.
 - c. **Región:** Seleccionar la región en la que se ubicara el centro IoT. Seleccionar la ubicación más cercana.
2. **Nombrar al IoT Hub:** Se ingresó un nombre para el IoT Hub. Este nombre debe ser único a nivel mundial. Finalmente se creará.



The screenshot shows the 'Centro de IoT' (IoT Hub) creation page in the Microsoft Azure portal. The page is titled 'Centro de IoT' and is part of the 'Microsoft' service. The 'Aspectos básicos' (Basic) tab is selected, with other tabs including 'Redes' (Networks), 'Escala y tamaño' (Scale and size), 'Etiquetas' (Tags), and 'Revisar y crear' (Review and create). The page instructs the user to create an IoT Hub to connect, monitor, and manage millions of IoT resources. Under 'Detalles del proyecto' (Project details), the user is prompted to choose a subscription, resource group, region, and IoT Hub name. The 'Suscripción' (Subscription) is set to 'Azure subscription 1'. The 'Grupo de recursos' (Resource group) is set to 'GrupoloTHub', with a 'Crear nuevo' (Create new) link below it. The 'Región' (Region) is set to 'Oeste de EE. UU.' (West US). The 'Nombre de IoT Hub' (IoT Hub name) is set to 'IoTHub-PWS', which is marked as valid with a green checkmark. At the bottom, there are buttons for 'Revisar y crear' (Review and create), '< Anterior' (Previous), 'Siguiendo: Redes >' (Next: Networks), and 'Opciones de Automation' (Automation options).

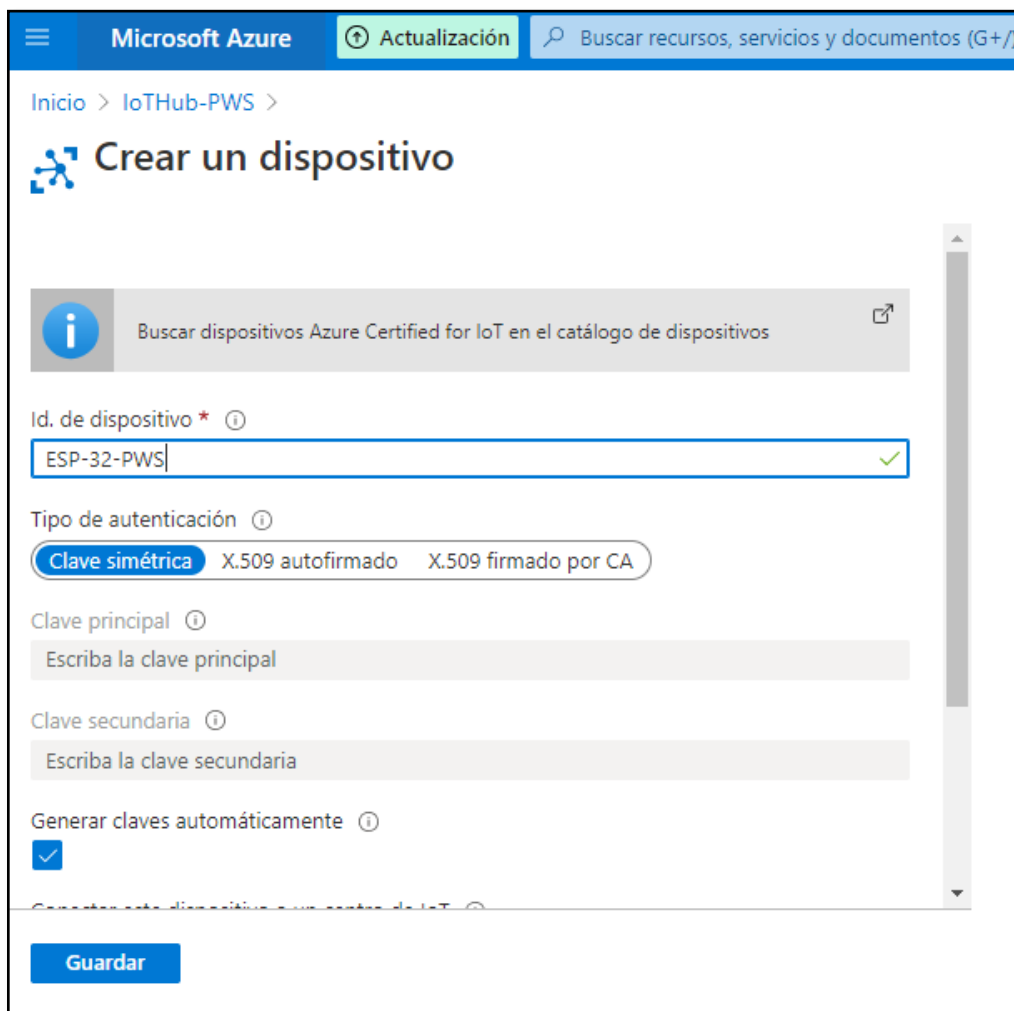
Figuras 38. Creando el centro de IoT para la solución IoT

Fuente: Elaboración propia

c) Registramos un nuevo dispositivo en IoT Hub

Se creó la identidad del dispositivo en “registro de identidad” en nuestro centro de IoT. Un dispositivo no puede conectarse a un concentrador a menos que tenga una entrada en el registro de identidad.

Abrimos Dispositivos de IoT, en el menú de navegación de su centro de IoT y luego seleccionamos Nuevo para agregar un dispositivo en IoT Hub.

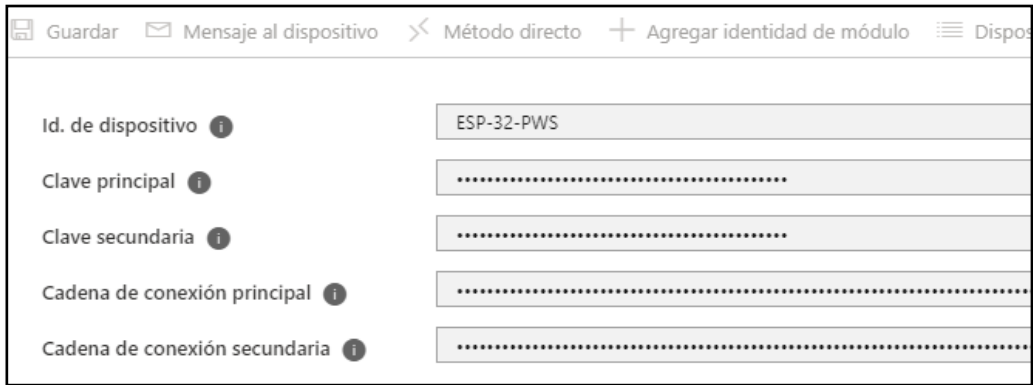


The screenshot shows the Microsoft Azure IoT Hub interface for creating a new device. The page title is "Crear un dispositivo". The breadcrumb navigation is "Inicio > IoT Hub-PWS >". The main heading is "Crear un dispositivo". Below the heading, there is a search bar with the text "Buscar dispositivos Azure Certified for IoT en el catálogo de dispositivos". The form fields are as follows:

- Id. de dispositivo ***: ESP-32-PWS (with a green checkmark)
- Tipo de autenticación**: Clave simétrica (selected), X.509 autofirmado, X.509 firmado por CA
- Clave principal**: Escriba la clave principal
- Clave secundaria**: Escriba la clave secundaria
- Generar claves automáticamente**:

At the bottom of the form, there is a "Guardar" button.

Figuras 39. Creando un nuevo dispositivo en el centro de IoT
Fuente: Elaboración propia

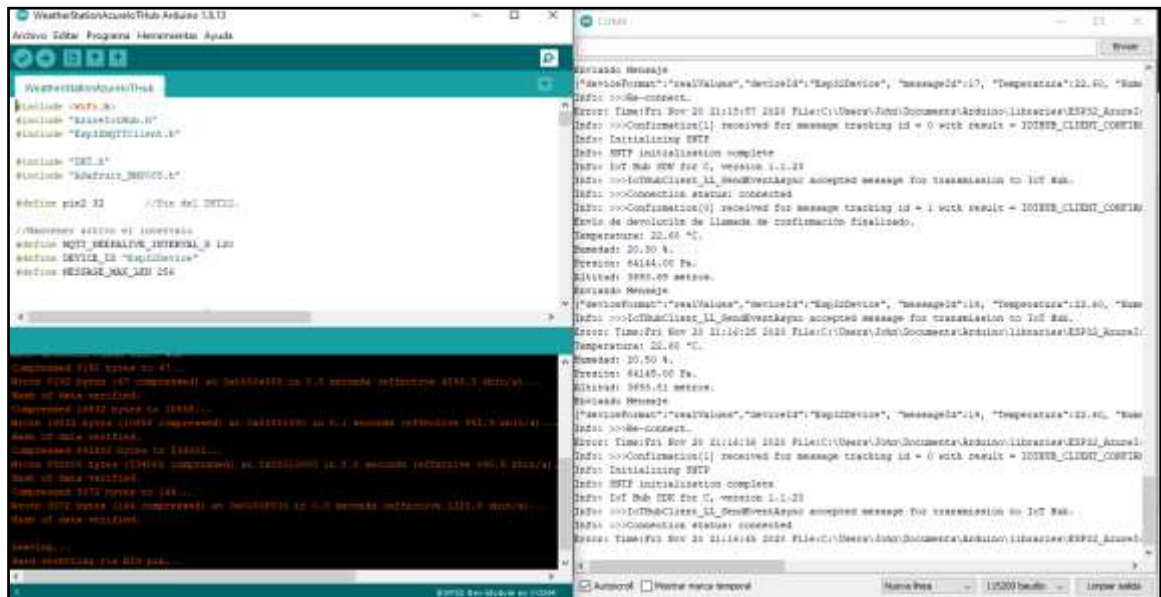


Figuras 40. Las claves de conexión del dispositivo del centro de IoT

Fuente: Elaboración propia

Aquí llegamos a ejecutar el código y podemos ver que ya se envían y reciben los primeros mensajes tal como muestra las siguientes imágenes:

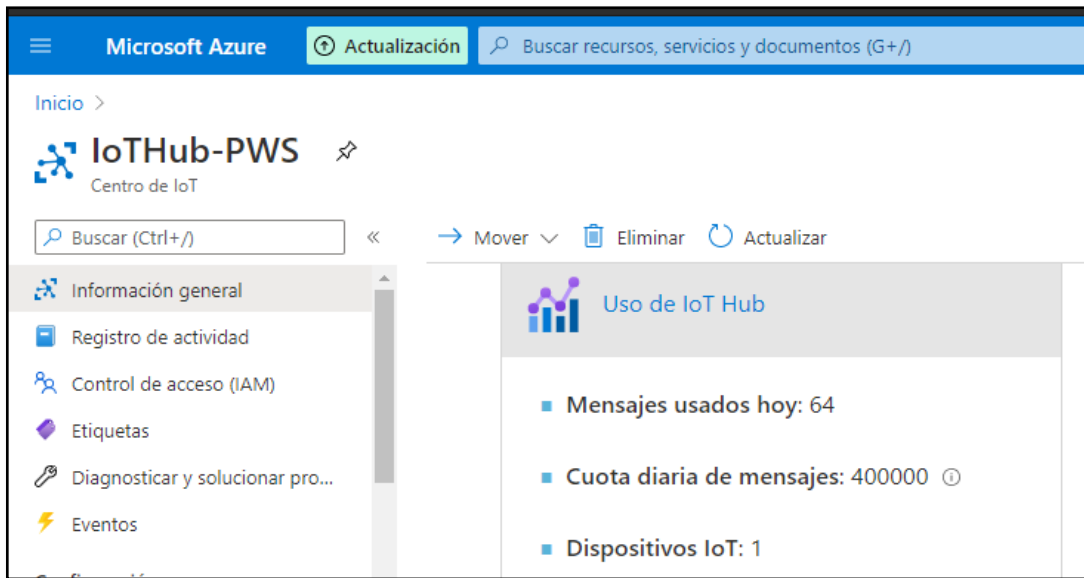
Local:



Figuras 41. Enviando los mensajes del PWS al centro de IoT por el monitor serie

Fuente: Elaboración propia

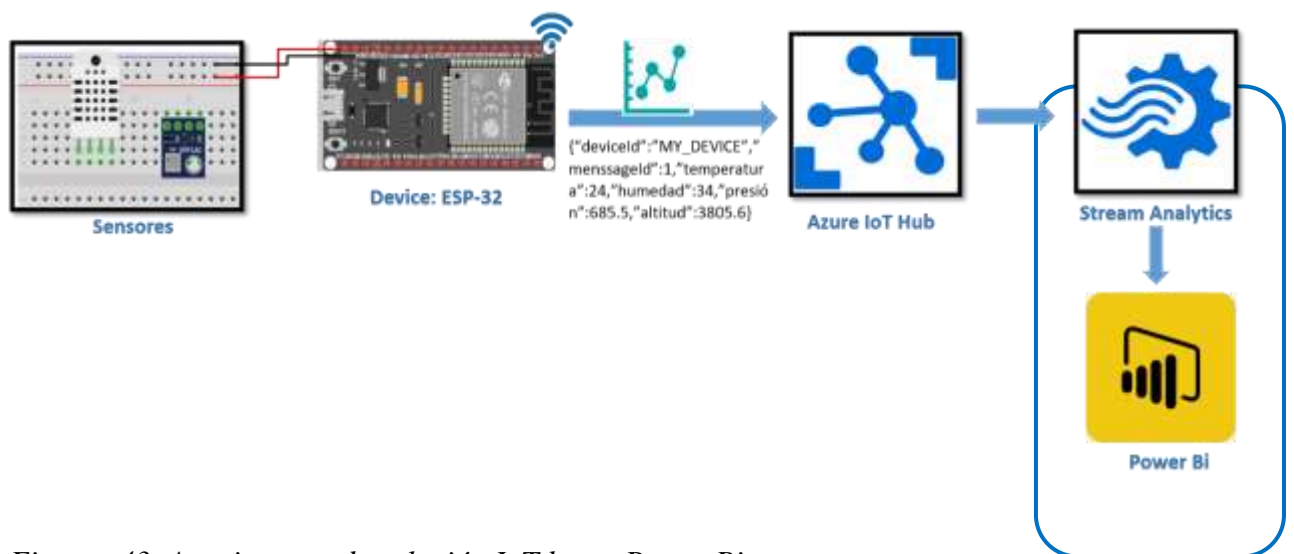
Nube:



Figuras 42. Recibiendo mensajes del PWS en el centro de IoT
Fuente: Elaboración propia

3.4.1.6. Visualización de datos en tiempo real de los sensores DHT22 y BMP 180 desde la plataforma de Azure IoT Hub con Power Bi

Se incluyó un panel realizado en PowerBi que facilito la visualización de los datos climáticos capturados en el PWS, el panel creado incluyo las medidas principales tomadas por la placa ESP-32, a continuación, se muestra arquitectura de solución en la figura 43.



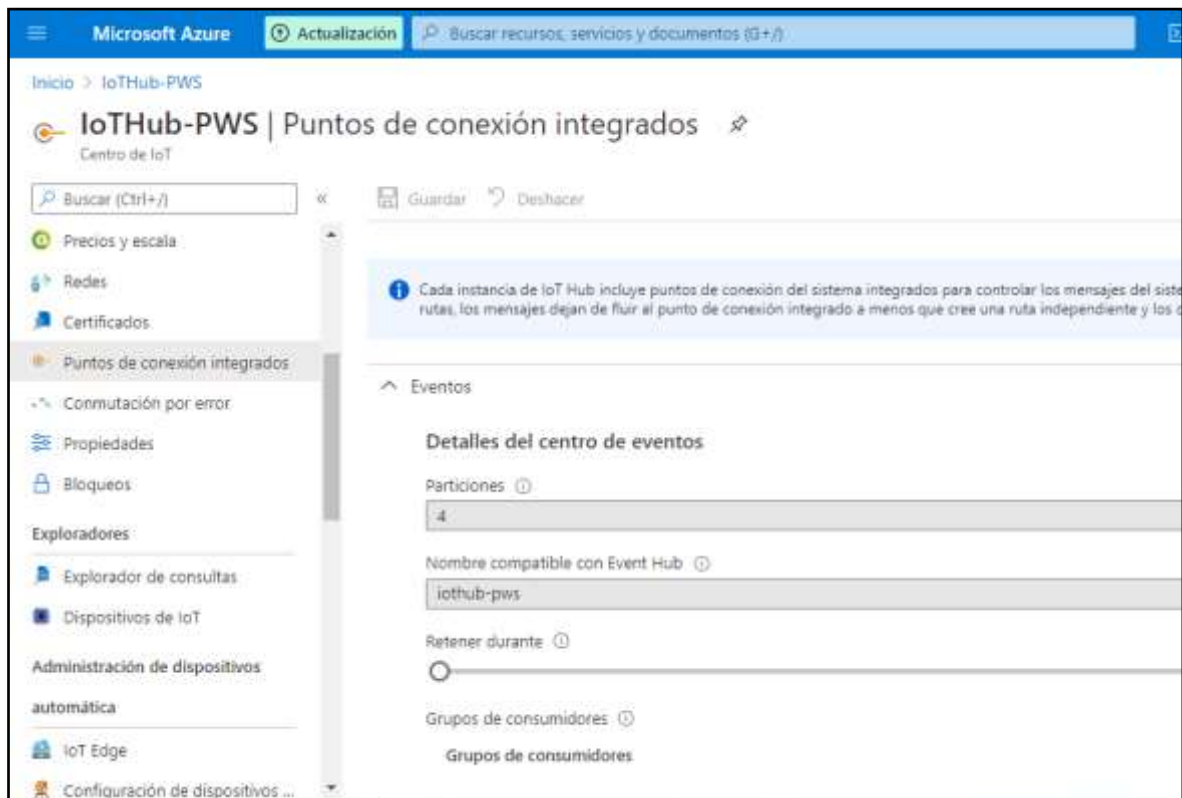
Figuras 43. Arquitectura de solución IoT hasta Power Bi
Fuente: Elaboración propia

a) Agregamos grupos de consumidores al centro de IoT Hub

Los grupos de consumidores proporcionan una vista independiente de un conjunto de eventos que permiten que las aplicaciones y servicios de Azure usen datos de forma independiente en el mismo punto de conexión del centro de eventos. Hemos agregado grupos de consumidores al punto de conexión integrado de la instancia de nuestro IoT Hub que será usada posteriormente para extraer datos del punto de conexión.

Agréguenos un grupo de consumidores al IoT Hub, siguiendo estos pasos:

1. En el panel izquierdo de IoT Hub, seleccionamos Puntos de conexión integrados, Eventos y en el panel superior derecho se escribió un nombre en Grupos de consumidores para luego guardarla.



Figuras 44. Creando el grupo de consumidor en el centro de IoT

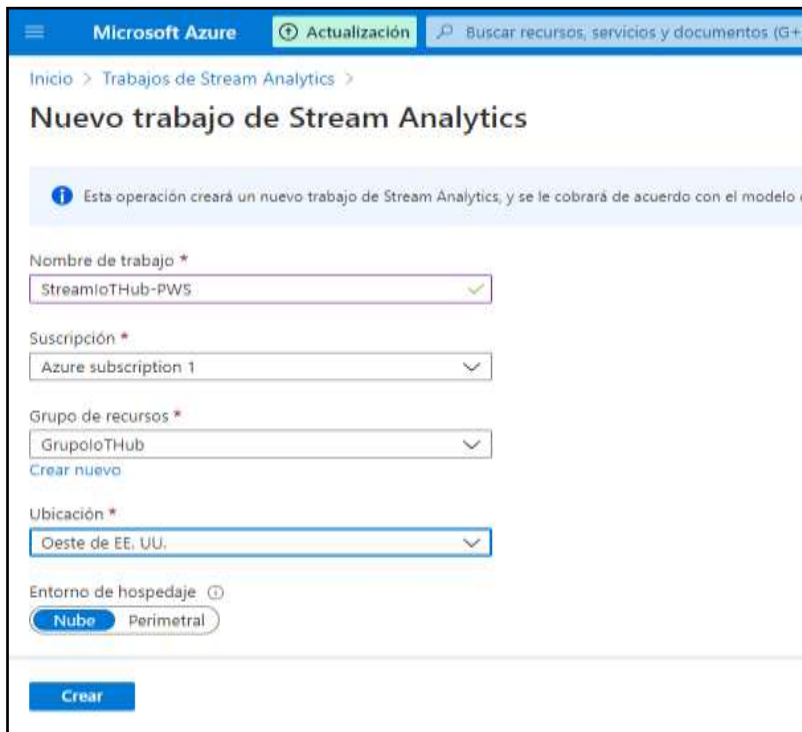
Fuente: Elaboración propia

La transmisión de los datos fue configurada desde el módulo ESP-32 hasta PowerBi. Para esto, se usó la herramienta Stream Analytics, esto nos permitió en el procesamiento y transmisión de los datos en tiempo real.

Hemos creando un trabajo de Stream Analytics y se definió salida, entradas y la consulta para recuperar datos.

b) Creación de un Stream Analytics

En el portal de Azure, creamos un recurso, colocando un nombre único para el trabajo, haciendo uso del mismo grupo de recursos del IoT y de la ubicación del grupo de recursos.



Figuras 45. Creando un nuevo Stream Analytics para la configuración – Power Bi
Fuente: Elaboración propia

Agregamos una entrada al trabajo de Stream Analytics

seleccionamos el panel de Entradas, seleccionamos agregar entrada de flujo y IoT Hub escribiendo la siguiente información:

- **Nombre de entrada:** Escribimos un alias único.
- **Suscripción:** Seleccionamos la suscripción de Azure que utilizamos.
- **IoT Hub:** Seleccionamos la instancia del centro IoT.
- **Punto de conexión sera:** Mensajería.
- **Nombre de directiva de acceso compartido:** Service.
- **Clave de directiva de acceso compartido:** Es llenado automáticamente en función de nuestra selección de nombre de directiva de acceso compartido.
- **Grupo de consumidores:** Seleccionamos el grupo de consumidores que creamos con anterioridad.
- El resto de todos los campos se dejó con sus valores predeterminados para luego guardar.

The screenshot shows the 'Centro de IoT' configuration window in the Azure portal. The window title is 'Centro de IoT' and it is labeled 'Nueva entrada'. The user's email 'johnyanapuma@outloo...' and 'DIRECTORIO PREDETERMINADO' are visible in the top right. The configuration fields are as follows:

- Alias de entrada ***: A text input field containing 'EntradaIoTHubPWS' with a green checkmark to its right.
- Indicar manualmente la configuración del Centro de IoT**: A radio button that is unselected.
- Seleccionar centro de IoT de entre las suscripciones**: A radio button that is selected.
- Suscripción**: A dropdown menu showing 'Azure subscription 1'.
- Centro de IoT**: A dropdown menu showing 'IoTHub-PWS'.
- Punto de conexión**: A dropdown menu showing 'Mensajes'.
- Nombre de directiva de acceso compartido**: A dropdown menu showing 'service'.
- Clave de directiva de acceso compartido**: A text input field with a masked password '.....'.

A blue 'Gua...' button is located at the bottom left of the form.

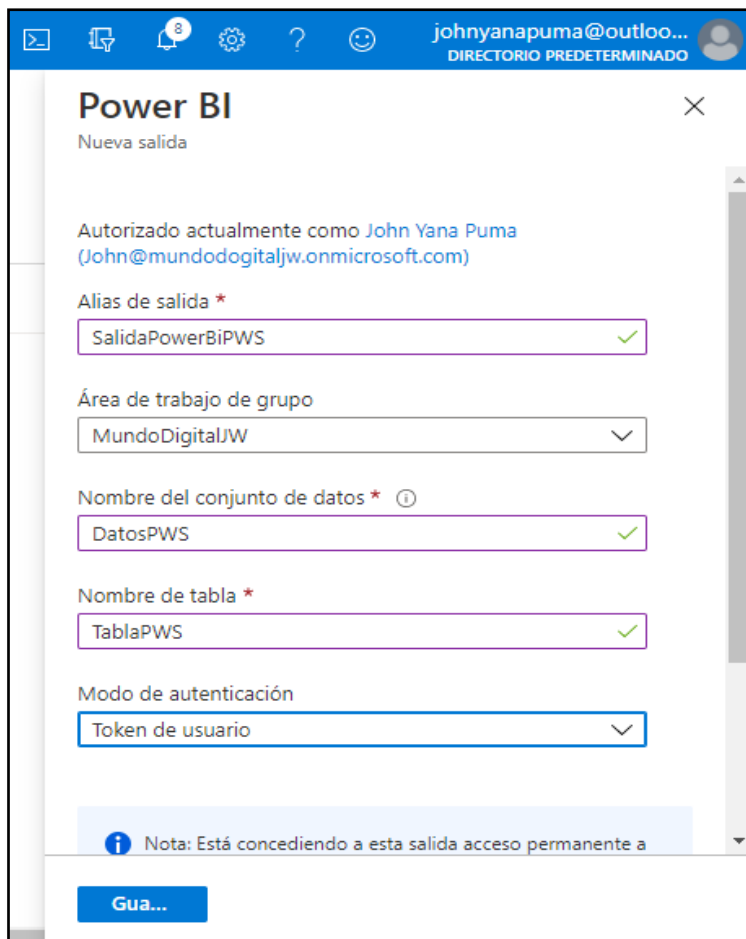
Figuras 46. Entrada de Stream Analytics para la configuración – Power Bi

Fuente: Elaboración propia

Adicionamos una salida al trabajo de Stream Analytics

En el panel Salidas en topología de Stream Analytics, hemos seleccionado Agregar Power BI, seleccionamos autorizar y continuamos con las indicaciones para el respectivo inicio de sesión en Power BI, ya con la sesión iniciada, escribiremos lo siguiente:

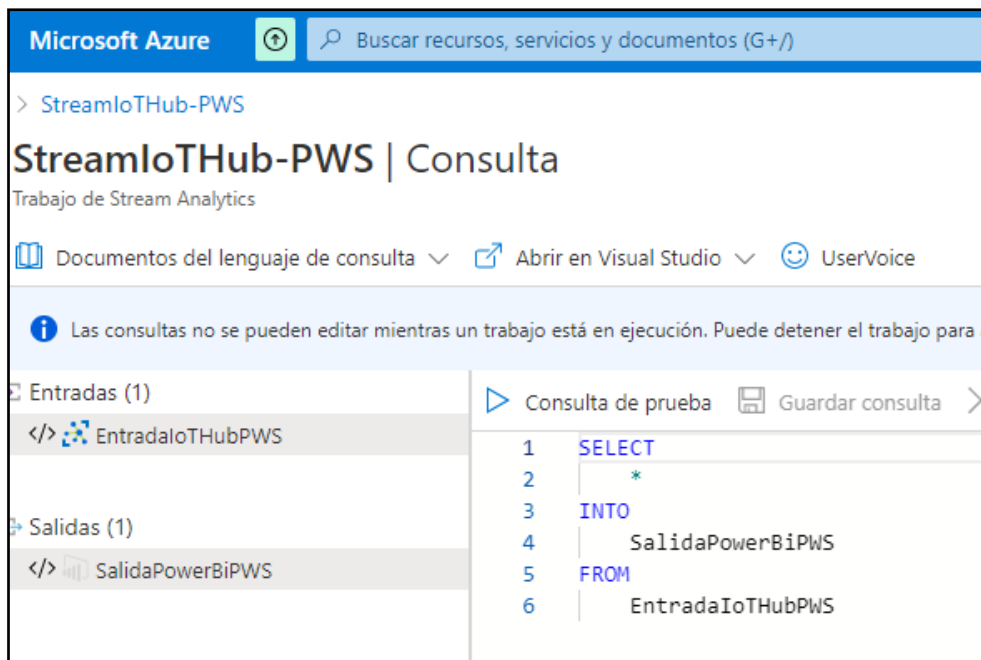
- **Nombre de salida:** Alias único para la salida.
- **Area o espacio de trabajo de grupo:** Seleccionamos el área de trabajo del grupo de destino.
- **Nombrar al conjunto de datos:** Escribimos un nombre.
- **Nombrar a la tabla:** Escribimos un nombre.
- **Modo de autenticación:** Predeterminada y guardar.



Figuras 47. Salida de Stream Analytics para la configuración – Power Bi
Fuente: Elaboración propia

c) Configurar la consulta del trabajo de Stream Analytics

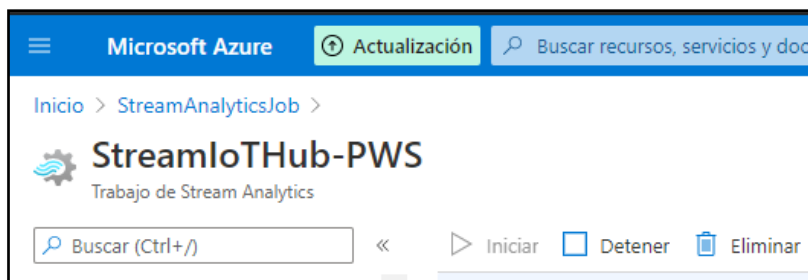
En la sección de topología de trabajo, haremos selección de consulta para luego reemplazar [YourInputAlias] por [YourOutputAlias] y guardar consulta.



Figuras 48. Consulta en Stream Analytics para la configuración – Power Bi
Fuente: Elaboración propia

d) Ejecutamos el trabajo de Stream Analytics

En esta sección el trabajo debe ser iniciada correctamente, siendo que su estado cambia de “detenido” a “en ejecución”.



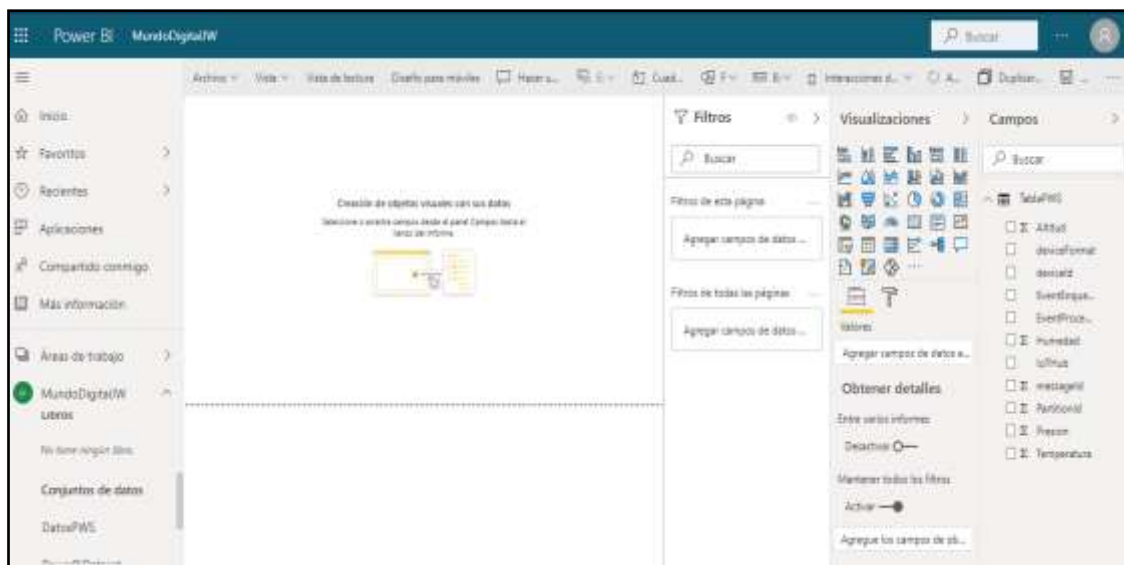
Figuras 49. Ejecutando el Trabajo de Stream Analytics– Power Bi
Fuente: Elaboración propia

Creación y publicación del informe de Power BI para la visualización de los datos

Se creó y público un informe mediante el servicio Power BI, para ello se siguió con lo siguiente:

1. Nos aseguramos de que la aplicación de muestra se haya ejecutado en el dispositivo.

2. Inicio de sesión en la plataforma de Power BI.
3. Selección del área de trabajo utilizado, **Mi área de trabajo**.
4. Selección de una dataset.
5. Muestra el grupo de los datos que se fue especificada cuándo generó la salida para la tarea de análisis de flujo.
6. Para el conjunto de datos que se creó, se realiza la selección Agregar informe.

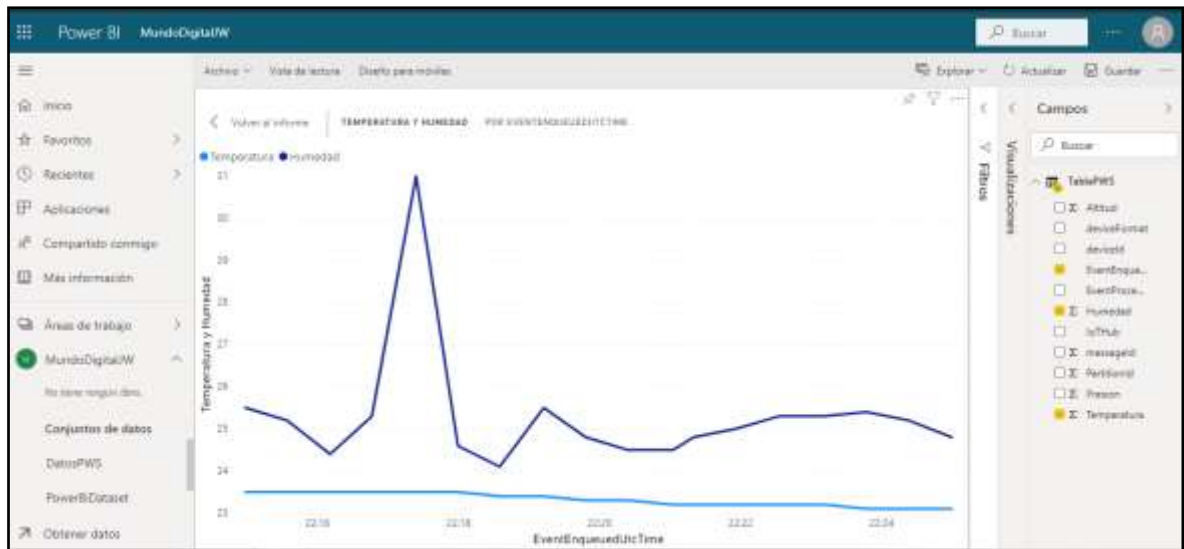


Figuras 50. Creando el informe en Power Bi

Fuente: Elaboración propia

- e) **Crear gráfico de líneas en la que muestre la temperatura en tiempo real en un determinado período.**

Se agrega un gráfico de líneas en la que el eje de la X muestra la fecha y hora en la zona horaria UTC y el eje de la Y muestra la temperatura y la humedad, como se muestra la figura 51.



Figuras 51. Gráfico de valores temperatura y humedad en tiempo real -Power Bi
Fuente: Elaboración propia

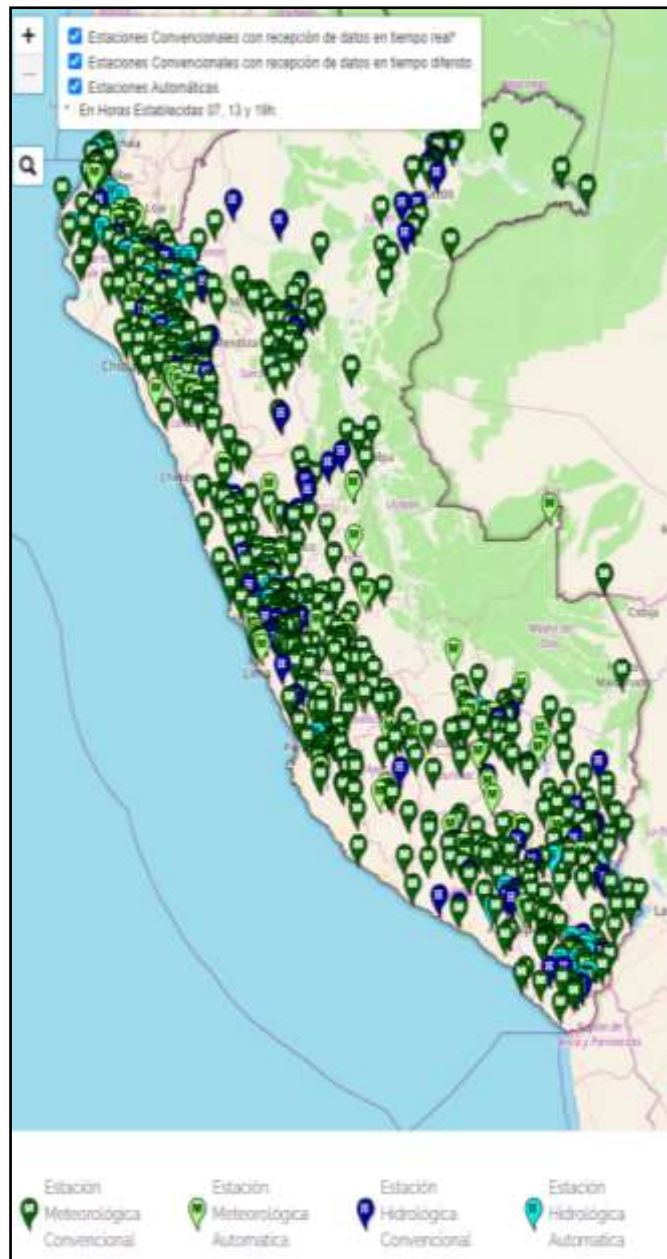
3.4.2. Fases de la Metodología Crisp-DM

3.4.2.1. Comprensión del negocio

Los objetivos de negocio son representados por los objetivos que fueron descritos en el apartado 1.2.

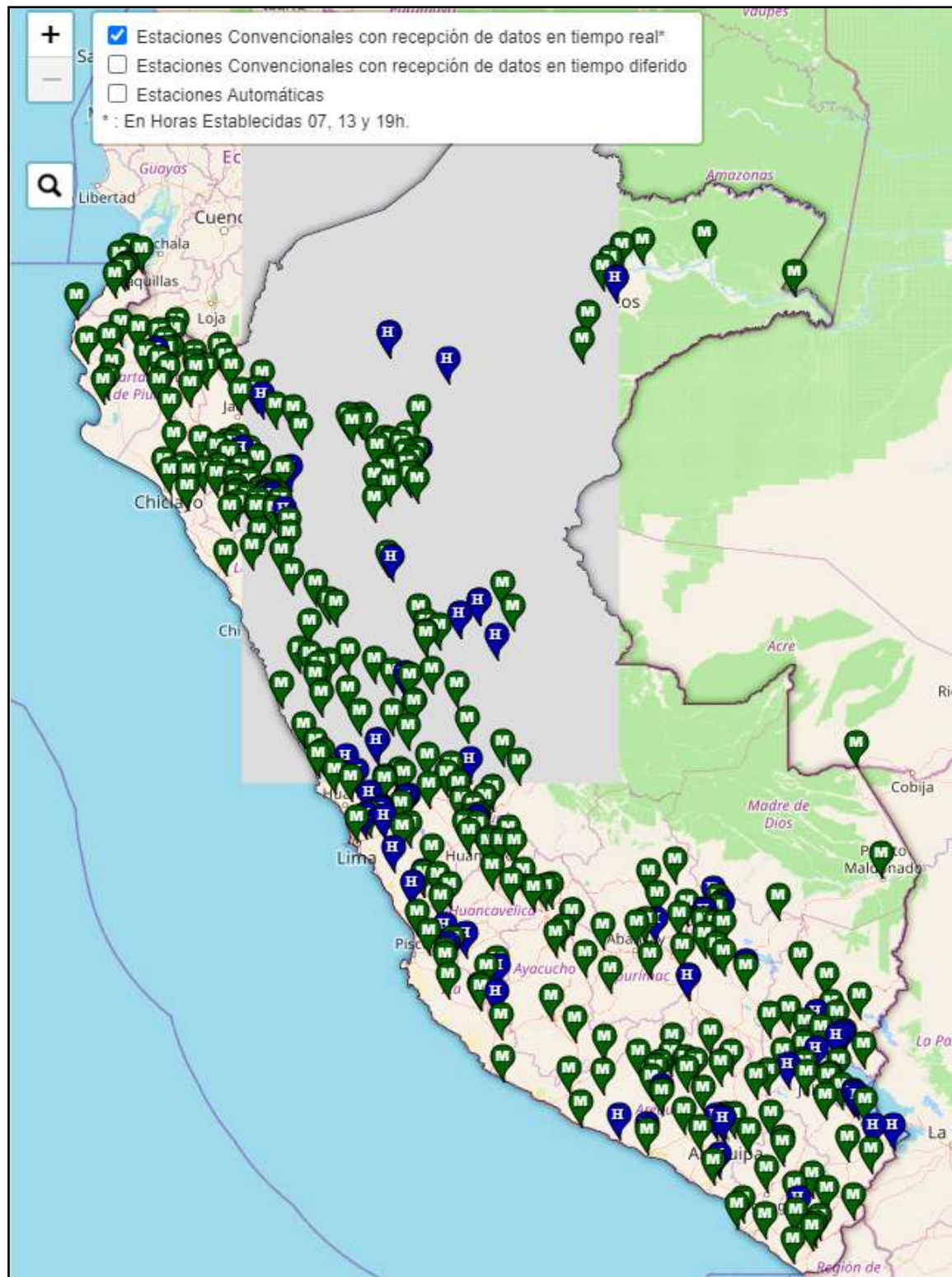
Una de las cuestiones acerca de la comprensión del negocio era definir con qué estación meteorológica trabajar, así que se optó por la estación meteorológica de Juliaca para posteriormente realizar la extracción de datos del SENAMHI y otras Apis Rest disponibles de forma gratuita.

El SENAMHI cuenta con más de 900 estaciones meteorológicas e hidrometeorológicas conforme a las normas técnicas de la OMM, tal como se muestra en la figura 52, entre convencionales y automáticas.



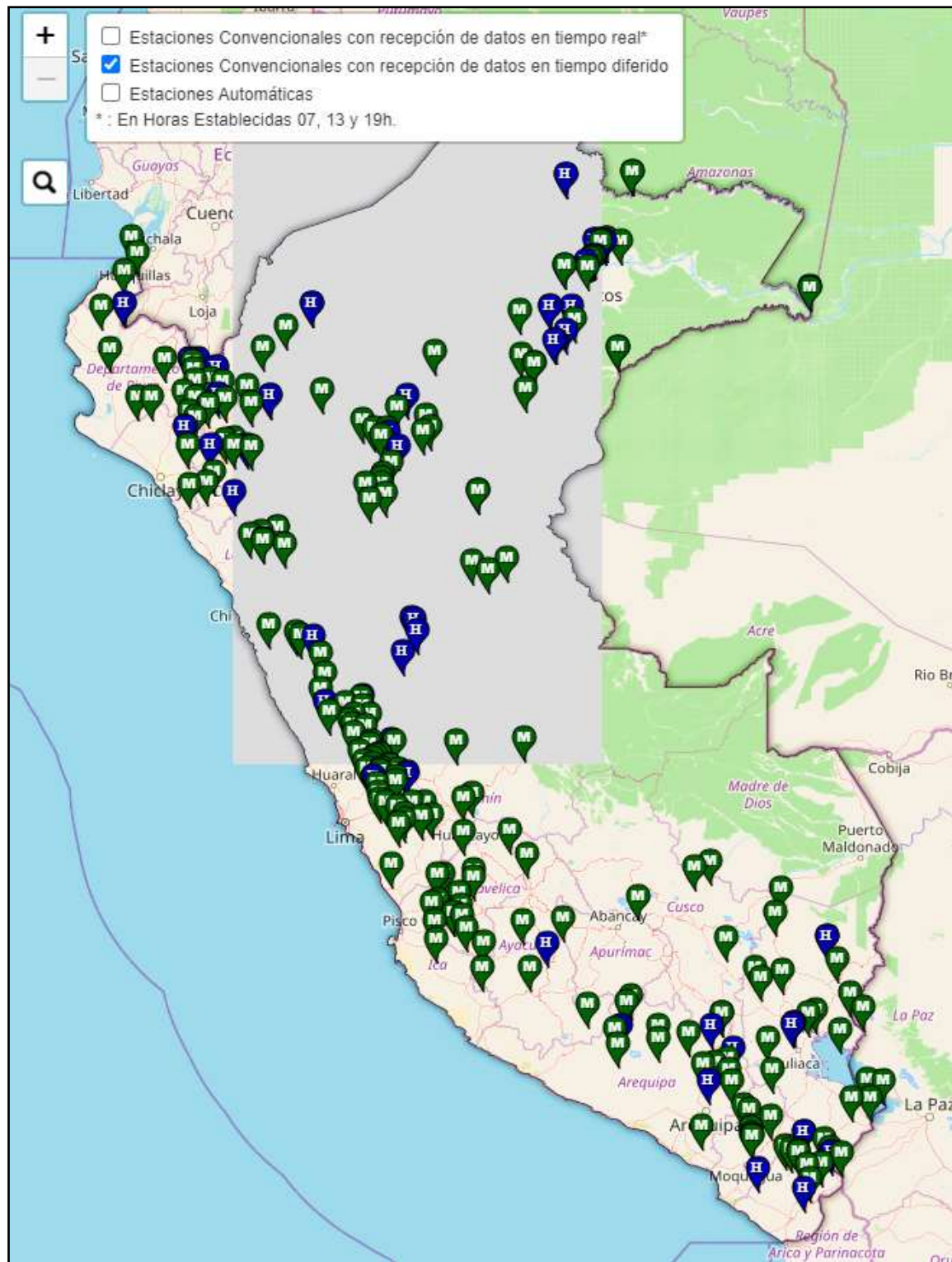
Figuras 52. Estaciones meteorológicas del SENAMHI
 Fuente: <https://www.senamhi.gob.pe/?&p=estaciones>

Estaciones Convenzionales con la recepción de los datos en tiempo real.



Figuras 53. Estaciones Convenzionales con recepción de datos en tiempo real - SENAMHI
Fuente: <https://www.senamhi.gob.pe/?&p=estaciones>

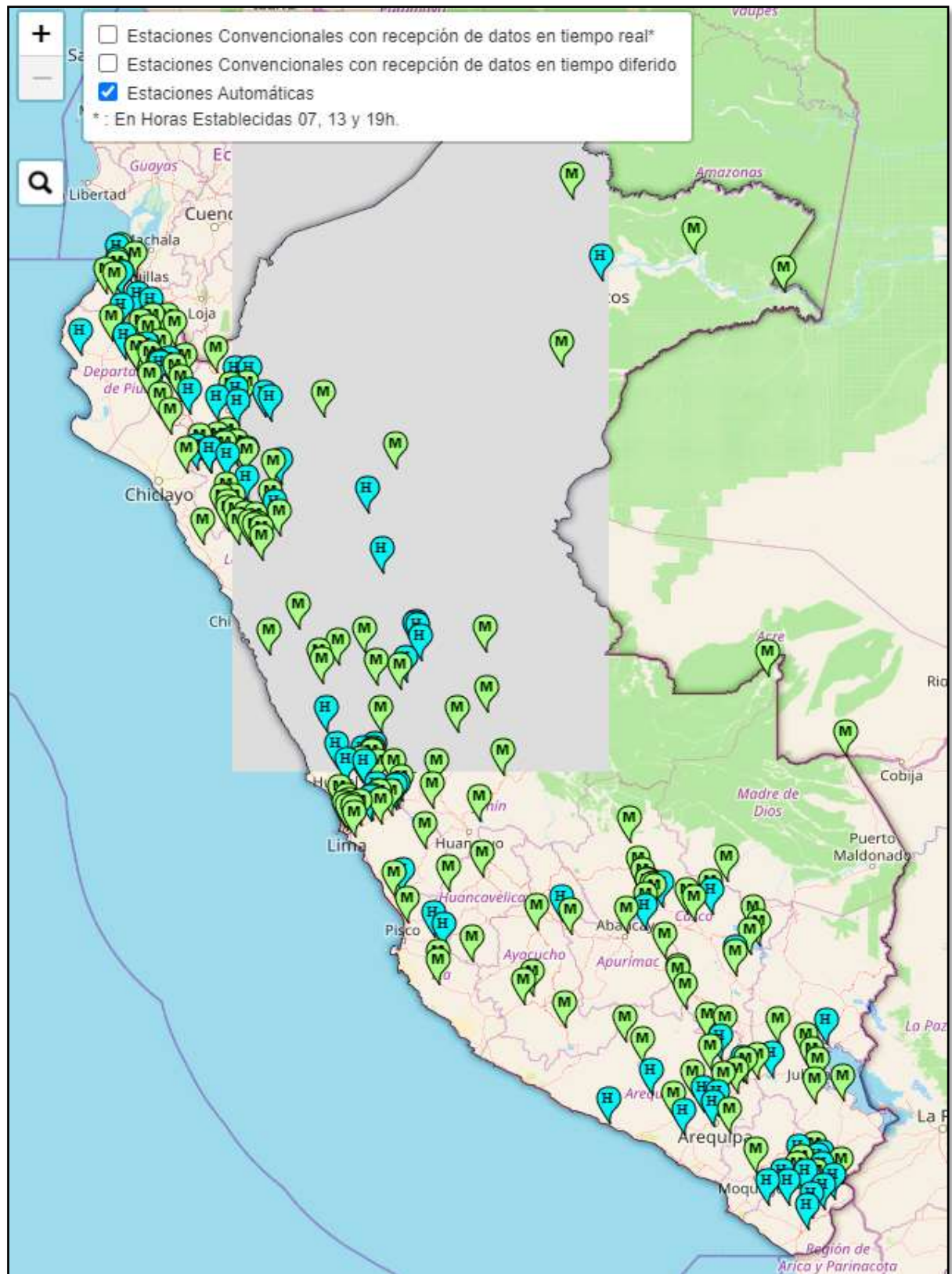
Estaciones Convenzionales con recepción de datos en tiempo diferido



Figuras 54. Estaciones Convenzionales con recepción de datos en tiempo diferido - SENAMHI

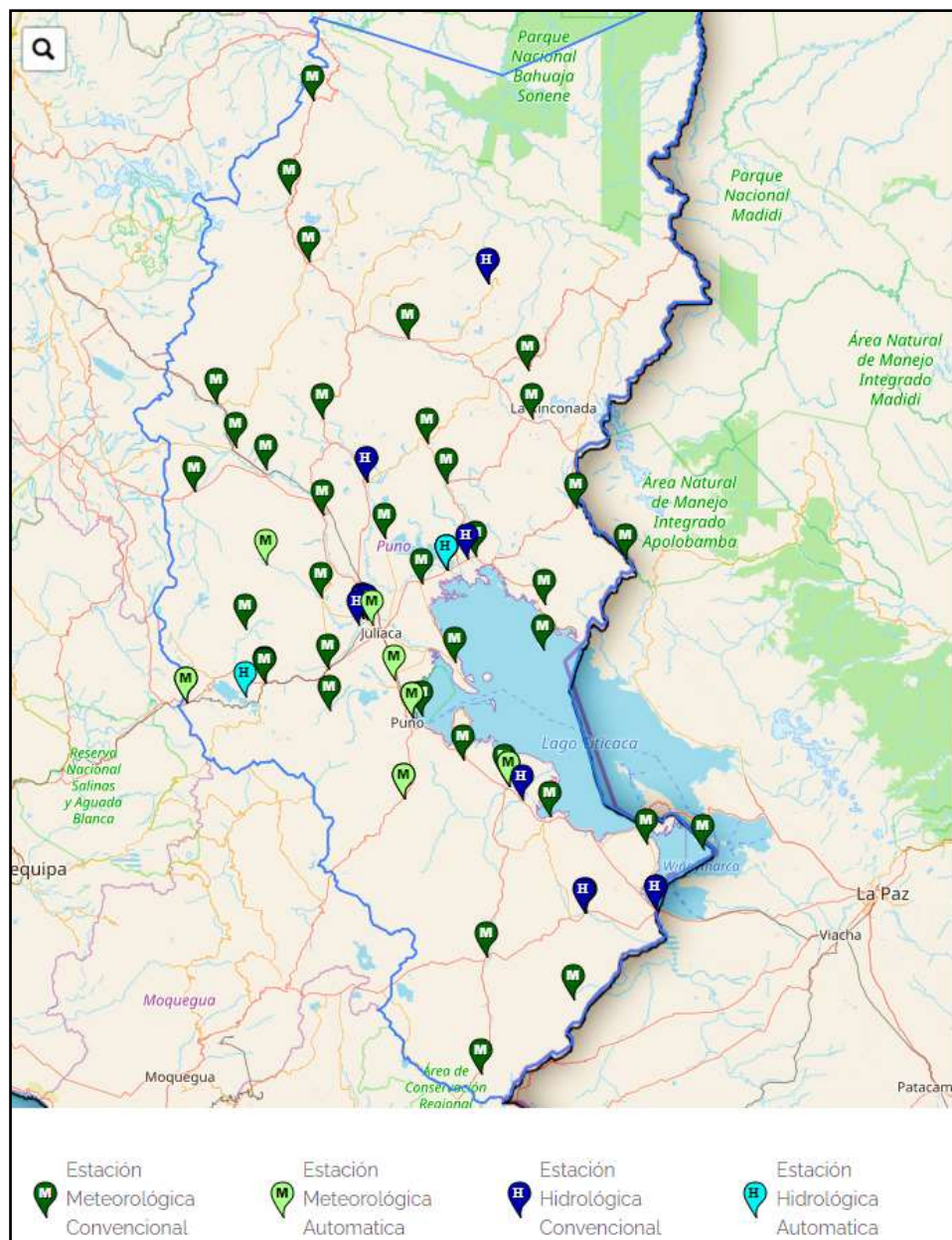
Fuente: <https://www.senamhi.gob.pe/?&p=estaciones>

Estaciones Automáticas



Figuras 55. Estaciones Automáticas - SENAMHI
Fuente: <https://www.senamhi.gob.pe/?&p=estaciones>

El SENAMHI implementó a nivel nacional un sistema de monitoreo agro meteorológico y fenológico que consta de 13 direcciones zonales, La región Puno es una de ellas, como Dirección Zonal 13 - Puno, que a la vez también cuenta con estaciones meteorológicas e hidrometeoro-lógicas, que van distribuidas en distintas provincias y distritos como muestra la figura 56, siendo 44 estaciones entre convencionales y automáticas.



Figuras 56. Estaciones del Departamento de Puno - SENAMHI
 Fuente: <https://www.senamhi.gob.pe/main.php?dp=puno&p=estaciones>

Las variables agroclimáticas son aquellos elementos cuantificables como: humedad relativa, la temperatura, precipitación, la velocidad del viento, etc., quienes están estrechamente relacionadas con el crecimiento y desarrollo de los cultivos. Estas variables climáticas al considerarse como eventos meteorológicos extremos afectan al cultivo determinando una buena o mala producción al finalizar la campaña agrícola.

Esta fase incluyó la recolección de información sobre el tema en el que se basa el proyecto para comprender los avances ya realizados y los posibles enfoques del proyecto. Se consideró finalizada al definir el tema de la investigación.

Antes del desarrollo de la fase de comprensión de datos, se tuvo que realizar la configuración del hardware la solución IoT denominado Estación Meteorológica (PWS) con herramientas de hardware de bajo costo y software libre, configurando los sistemas de transmisor y receptor. El transmisor fue el encargado de la lectura, la escritura y transmisión de los datos al receptor. Entretanto el receptor fue el encargado de recepcionar los datos para su almacenamiento, visualización y finalmente procesarlos en un servidor remoto.

3.4.2.2. Comprensión de Datos

En esta fase se tuvo contacto con los datos y se realizó una comprensión de los datos meteorológicos, realizando una exploración de cada una de las variables con sus respectivas unidades de medida y el rango de medida, y comprobando la calidad de estas.

Buscar los patrones de repeticiones en series de datos para después almacenar solo el dato adjunto al número de veces que se repite. Teniendo en cuenta que los datos pueden ser redundantes, irrelevantes, básicos y subjetivos.

En esta fase se tuvo contacto con los datos relevantes para el estudio. Corresponde a la obtención de datos, se realizó una solicitud de datos históricos al SENAMHI que

corresponde a los años 2015 al 2020, del 1 de diciembre del 2015 al 21 de noviembre del 2020, el registro de datos es de cada hora, la información incluye las siguientes variables, de las cuales solo se trabajó con las variables de interés del proyecto. descritas en la tabla 3

Tabla 3. Variables

Item	Variable	Unidad	Descripción
1	Temperatura	°C	1
2	Humedad relativa	%	1
3	Precipitación pluvial	mm	1
4	Presión atmosférica	m/s	1

Fuente: Elaboración propia

Los datos proporcionados por el SENAMHI están en formato “xlsx” como se muestra en la figura 57,

* Datos sin control de calidad. * El uso de estos datos será de entera responsabilidad del usuario. Leyenda: * S/D = Sin Datos.						
Estación: JULIACA						
Departamento: PIURA		Provincia: SAN ROMAN		Distrito: JULIACA		
Latitud: 15°28'15.8"		Longitud: 70°10'16.4"		Altitud: 3826 msnm.		
Tipo: EMA - Meteorológica		Código: 472CF72C				
AÑO / MES / DIA	HORA	TEMPERATURA (°C)	PRECIPITACIÓN (mm/hr)	HUMEDAD (%)	DIRECCION DEL VIENTO (°)	VELOCIDAD DEL VIENTO (m/s)
1/04/2019	00:00	9.7	0	88	21	0
1/04/2019	01:00	9.3	0	93	283	0.1
1/04/2019	02:00	8.1	0	100	69	3
1/04/2019	03:00	8.5	0	100	62	3.2
1/04/2019	04:00	8.3	0	99	86	2.6
1/04/2019	05:00	8.2	0	99	237	0
1/04/2019	06:00	8.3	0	99	285	2.1
1/04/2019	07:00	8.5	0	100	299	3
1/04/2019	08:00	9.8	0	93	206	0.2
1/04/2019	09:00	12.4	0	68	169	0.8
1/04/2019	10:00	13	0	62	115	0.3
1/04/2019	11:00	14.7	0	56	340	1.1
1/04/2019	12:00	15.9	0	55	73	3
1/04/2019	13:00	16.7	0	60	38	2
1/04/2019	14:00	11.8	0.8	67	96	4.7
1/04/2019	15:00	10.2	0.2	74	6	5

Figuras 57. Datos de temperatura, precipitación, humedad, dirección y velocidad del viento.

Fuente: Senamhi.

IOT. En el apartado con anterioridad se explicó sobre la implementación y configuración de PWS, ya que a través de ella se realizó la captación de datos en tiempo real mediante sensores ya descritos en el apartado 2, estos deben captar correctamente las condiciones del clima caso contrario no se podrá comparar con los datos almacenados y por ende la predicción sería errónea. en la figura 56, se muestra los datos obtenidos en tiempo real, que seguidamente es recibida por los servicios de IoT Hub y Stream Analytics y por último que se almacene en el repositorio Azure Storage.

3.4.2.3.Preparación de datos

Se decide qué datos serán usados para el análisis, para lo cual se toma en cuenta los objetivos de minería de datos (calidad, volumen de datos y tipos de datos). Realizar análisis de datos `para conocerlos mejor, y depuración correspondiente si es que es el caso, y en base a los resultados seleccionar los algoritmos que mejor se ajusta al problema a resolver. Una vez que se tiene los datos en Excel, lo que sigue es verificar los datos nulos, vacíos, etc para un buen desempeño del modelo.

AÑO / MES / DÍA	HORA	TEMPERATURA (°C)	PRECIPITACIÓN (mm/hora)	HUMEDAD (%)	DIRECCION DEL VIENTO (°)	VELOCIDAD DEL VIENTO (m/s)
1/04/2019	00:00	9.7	0	88	21	0
1/04/2019	01:00	9.3	0	93	283	0.1
1/04/2019	02:00	8.1	0	100	59	3
1/04/2019	03:00	8.5	0	100	52	3.2
1/04/2019	04:00	8.3	0	99	86	2.6
1/04/2019	05:00	8.2	0	99	237	0
1/04/2019	06:00	8.3	0	99	285	2.1

Figuras 58. Visualización de datos.
Fuente: Senamhi

Primero, se realizó un procesado inicial de datos brutos para obtener lo que se conoce como datos preparados. Este proceso inicial tiene como objetivo adaptar el formato a las necesidades del modelo y simplificar así su construcción. A partir del archivo original de

datos se generó uno nuevo, en la que se tomó en cuenta las transformaciones aplicadas para normalizar los datos.

- Convertir la fecha en formato: YYYYMMDD
- La hora en formato:hh (números enteros)
- Sustituir a la etiqueta “S/D” por un valor vacío
- Añadir una nueva columna para la categorización (LLUVIA, NO LLUVIA) TRUE es el valor de la precipitación mayor a cero y FALSE cero.
- Se realizó el cambio el formato de los datos, se añadió un campo con

Seguidamente se procedió a guardar los datos con la estructura requerida se guardó en formato .csv para poder utilizarlo como input en la plataforma de Azure, siendo el archivo resultante como muestra la figura 59.

1	ANO,MES,DIA,HORA,DIRVIENTO,VELVIENTO,PRECIPITACION,HUMEDAD,TEMPERATURA,RAIN			
2	2015,12,1,0,109,0.5,0,71,7.1,False			
3	2015,12,1,1,293,2.1,0,77,6.6,False			
4	2015,12,1,2,247,1.6,0,74,6.4,False			
5	2015,12,1,3,142,1.6,0,79,5.5,False			
6	2015,12,1,4,276,1.8,0,79,5.9,False			
7	2015,12,1,5,273,0,0,78,5.2,False			
8	2015,12,1,6,293,2.2,0,72,5.7,False			
9	2015,12,1,7,249,1.8,0,67,6.9,False			
10	2015,12,1,8,257,0.5,0,65,8.4,False			
11	2015,12,1,9,190,1.2,0,55,10.9,False			
12	2015,12,1,10,166,2.2,0,48,12.9,False			
13	2015,12,1,11,341,2.5,0,40,14.2,False			
14	2015,12,1,12,341,4,0,33,15.5,False			
15	2015,12,1,13,333,4.3,0,29,16.2,False			
16	2015,12,1,14,331,4.7,0,28,16.7,False			
17	2015,12,1,15,356,3.9,0,23,17.2,False			
18	2015,12,1,16,58,7.1,0,34,16.2,False			

Figuras 59. Datos en formato
Fuente: Elaboración propia

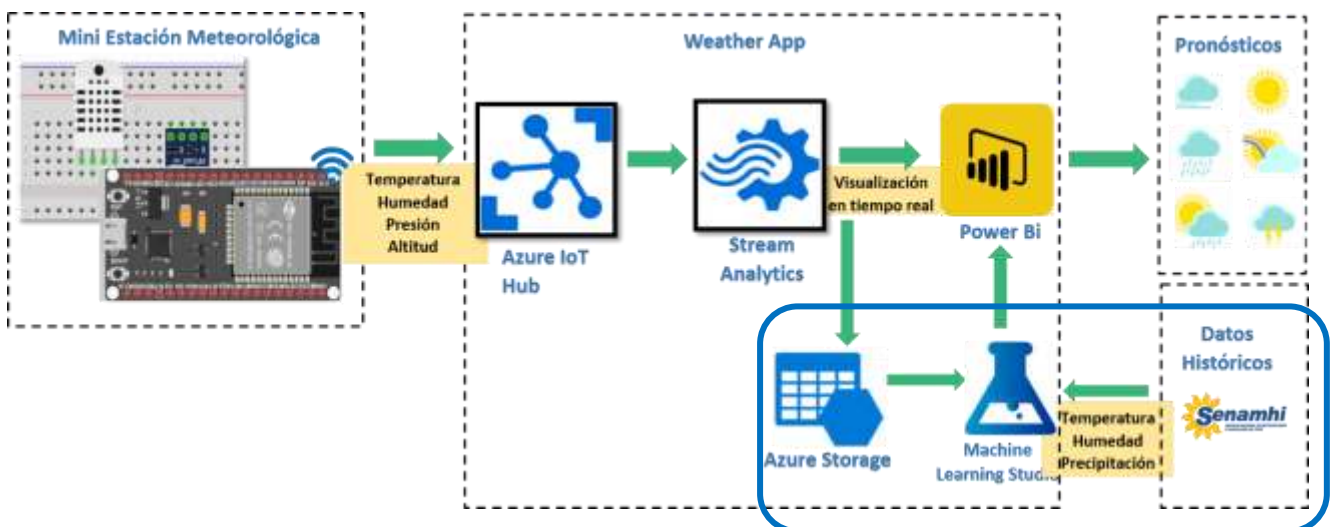
3.4.2.4. Modelado

En esta fase de la metodología, para la aplicación de los modelos está un conjunto de datos que se dividen en los subconjuntos train, test y validation, considerando la proporción (train 60%, test y validation 40%).

Se entrenó el modelo de aprendizaje supervisado mediante el conjunto de train más conocido como datos de entrenamiento, seguidamente se evaluó la eficiencia a través del test y finalmente se evaluó el desempeño con un conjunto totalmente distinto al que fue separado al comienzo del análisis (validation).

Una de las herramientas es: Microsoft Azure Machine Learning Studio, en donde se realizó la creación de modelo en la nube, esta herramienta permitirá evaluar la precisión total, precisión media y porcentaje de acierto en la clasificación. Se creó un experimento (nombre de cada archivo en Azure ML Studio), en el que contiene un algoritmo **two class logistic regresion**; esta plataforma cuenta con una serie de bloques (Score Model, Train Model, Evaluate Model, Tune Model Hyperparameters)

a) Modelado en la Plataforma de Microsoft Azure



Figuras 60. Estructura de solución del modelo predictivo
Fuente: Elaboración propia

Construcción modelo predictivo

Primer modelo (baseline) con algoritmos de Regresion

En la definición del baseline para el modelo continuo, se optó usar las variantes de árboles de regresión, los algoritmos se ejecutaron asignando al meta parámetros valores por defecto, en Azure ML, ver figura 60.

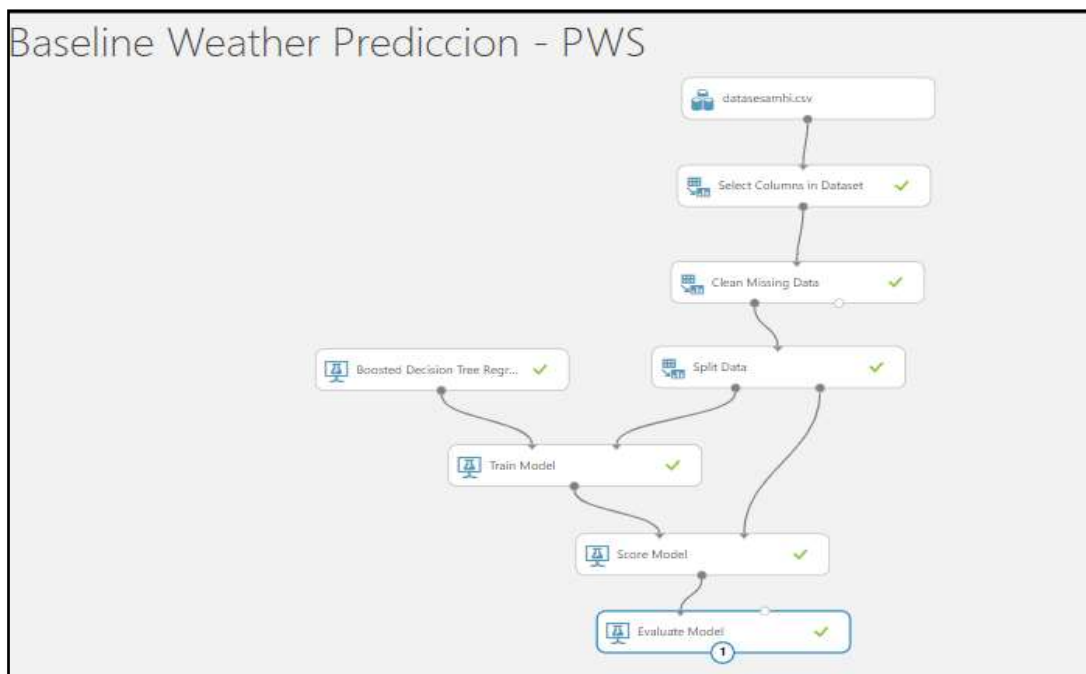


Figura 60. Baseline Weather Prediccion – PWS

Fuente: Elaboración propia

A continuación, se muestra los resultados obtenidos:

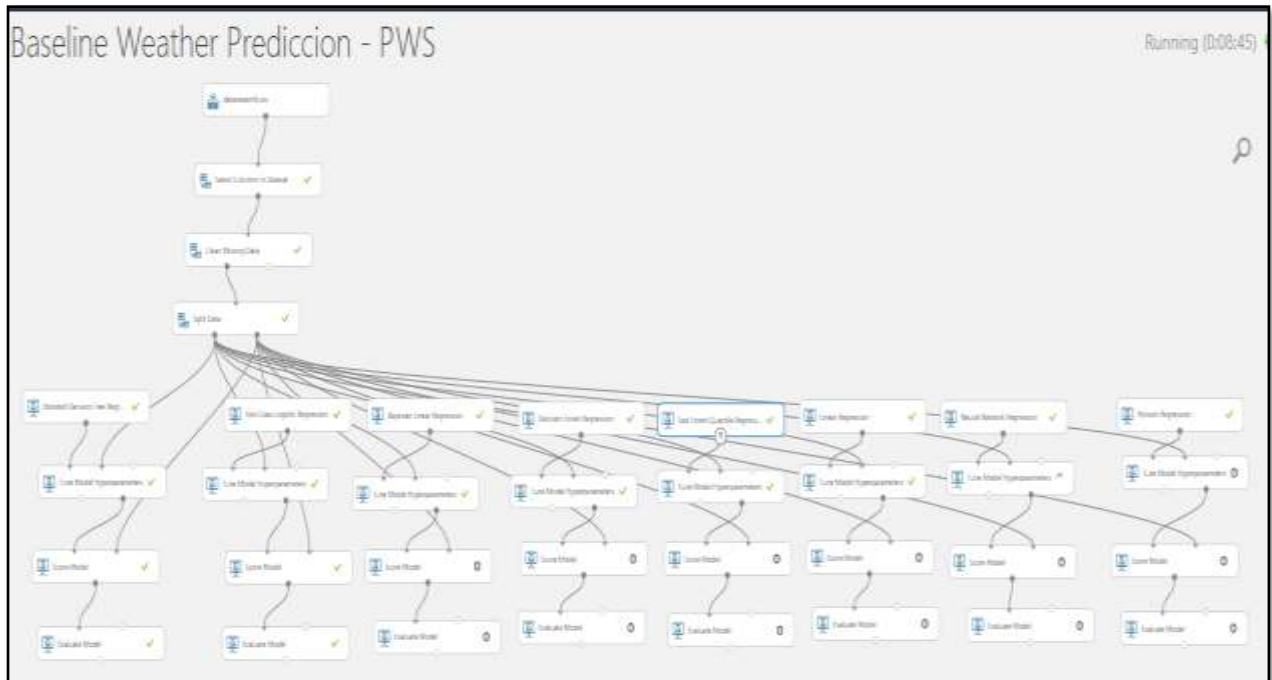
Baseline Weather Prediccion - PWS > Evaluate Model > Evaluation results	
Metrics	
Mean Absolute Error	0.10238
Root Mean Squared Error	0.234462
Relative Absolute Error	0.861956
Relative Squared Error	0.92565
Coefficient of Determination	0.07435

Figuras 61. Resultado de la evaluación del modelo

Fuente: Elaboración propia

Técnicas aplicadas

En esta sección, se ilustra algunas de las pruebas realizadas para mostrar el proceso iterativo que fue necesario realizar con el objetivo de identificar el modelo que mejor se adapte al tipo de problema realizado.



Figuras 62. Baseline Weather Predicción – varios algoritmos

Fuente: Elaboración propia

Se aplicaron los siguientes métodos variando los meta parámetros de los mismos:

Algoritmos de Regression:

1. BayesianLinearRegression
2. BoostedDecisionTree Regression
3. DecisionForestRegression
4. LinearRegression
5. Neural Network Regression
6. Poisson Regression

La opción utilizada para variar los metas parámetros fue entire grid, que lo que hace es probar con todas las combinaciones de parámetros y seleccionar la que tiene mayor desempeño de acuerdo a una métrica indicada (en nuestro caso coeficiente de determinación).

Tabla 4. Resultados comparativos de los algoritmos de Regression

Item	Técnica Aplicada	Parámetros	Mean Absolute Error	Coefficient of determination
1	Boosted Decision Tree Regression	Por defecto	0.1084	0.0995
2	Bayesian Linear Regression	Por defecto	0.1158	0.1164
3	Decision Forest Regression	Por defecto	0.104692	0.0673
4	Linear Regression	Por defecto	0.115871	0.1166
5	Neural Network Regression	Por defecto	0.109398	0.1297
6	Poisson Regression	Por defecto	0.11436	0.1111

Fuente: Elaboración propia

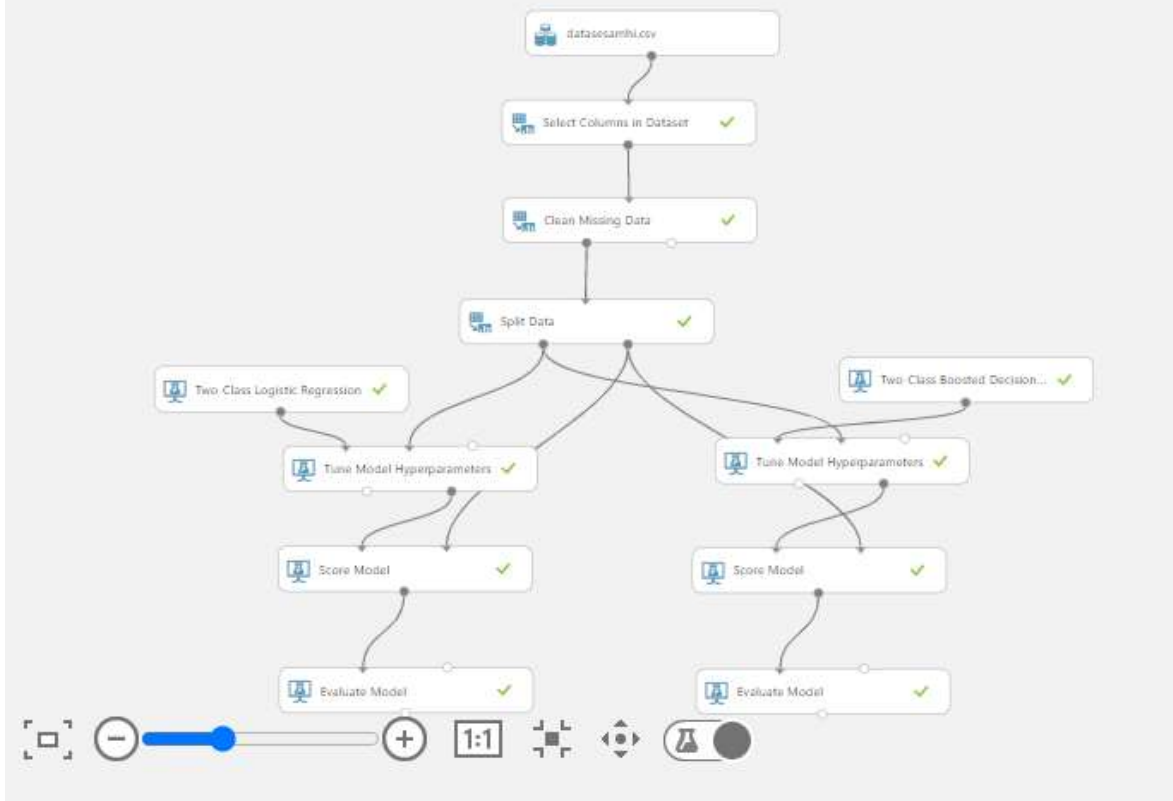
Segundo modelo (baseline) con algoritmos de Clasificación

En la definición del baseline para el modelo, se optó por usar Regresion Logistica algoritmo de clasificación, cabe aclarar que los algoritmos se ejecutaron asignando a los meta parámetros valores por defecto, en Azure ML, también se muestra algunas de las pruebas realizadas para ilustrar el proceso iterativo que es necesario realizar con el objetivo de determinar el modelo que mejor se adapta al tipo de problema a resolver.

Técnicas aplicadas

- Two Class Logistic Regression
- Two Class Decision Booted Decision

Baseline Weather Clasifi - senamhi juliaca



Figuras 63. Modelo baseline con algoritmos de Clasificación

Fuente: Elaboración propia

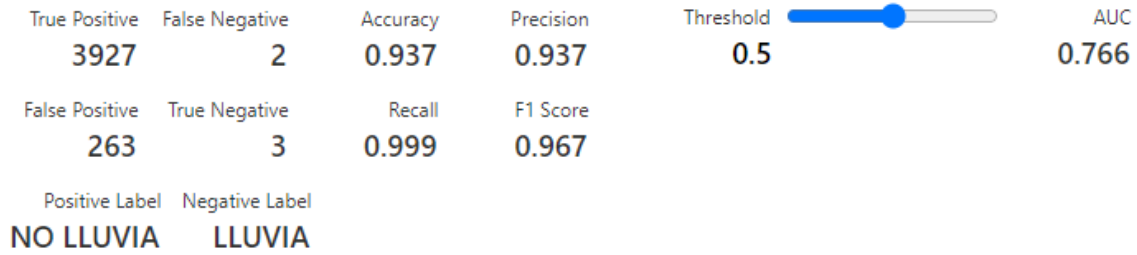
En esta sección se muestra los resultados para el algoritmo Two Class Logistic Regression

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
3928	1	0.937	0.937	0.5	0.856
False Positive	True Negative	Recall	F1 Score		
263	3	1.000	0.967		
Positive Label	Negative Label				
NO LLUVIA	LLUVIA				

Figuras 64. Resultados para el algoritmo Two Class Logistic Regression

Fuente: Elaboración propia

En esta sección se muestra los resultados para el algoritmo Two Class Decision Booted Decision.



Figuras 65. Resultados para el algoritmo Two Class Decision Booted Decision
Fuente: Elaboración propia

La opción utilizada para variar los meta parámetros fue entiere grid, que lo que hace es probar con todas las combinaciones de parámetros y seleccionar la que tiene mayor desempeño de acuerdo a una métrica indicada (en nuestro caso Accuracy y Recall).

Tabla 5. Resultados comparativos de los algoritmos de clasificación

Item	Técnica Aplicada	Parámetros	Accuracy	Recall
1	Two Class Logistic Regression	Por defecto	0.937	1.000
2	Two Class Decision Booted Decision	Por defecto	0.937	0.999

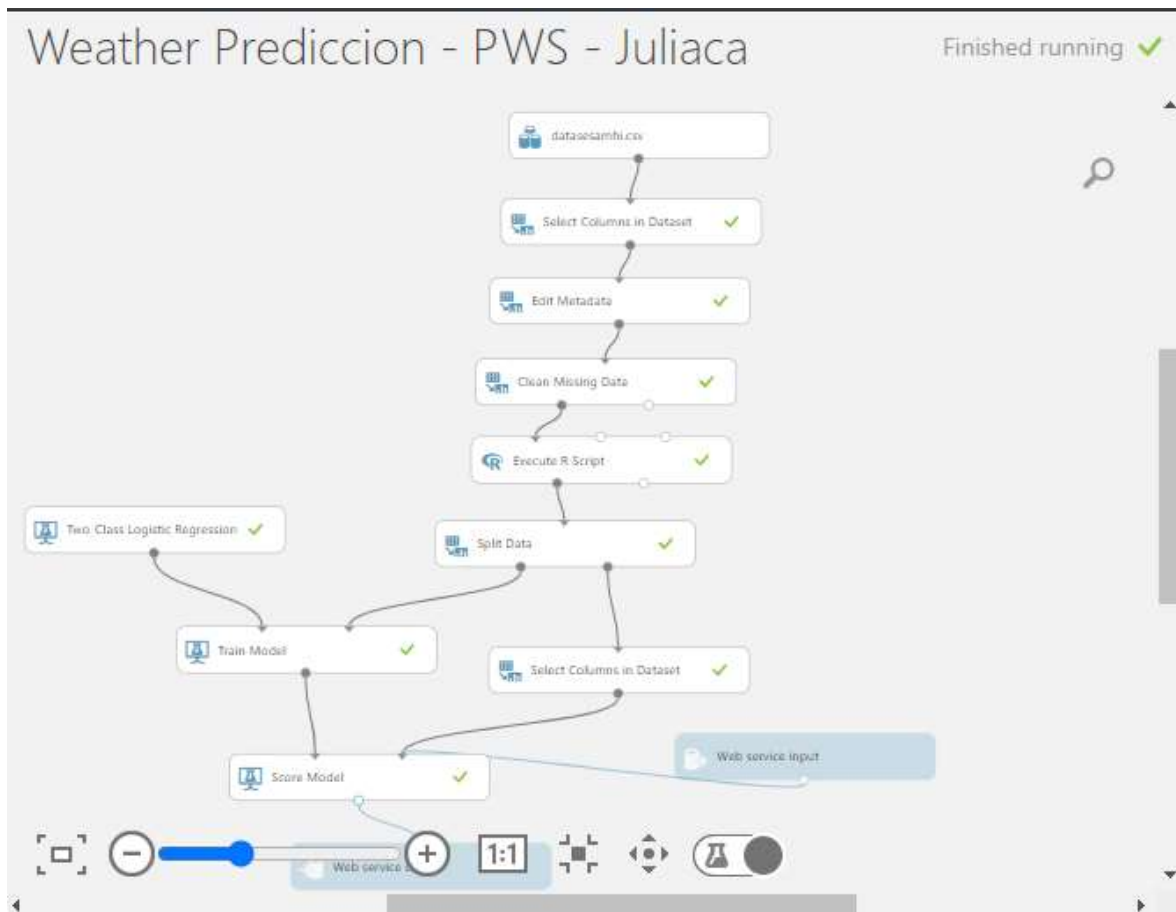
Fuente: Elaboración propia

Implementaremos el modelo de predicción meteorológica como servicio web

El modelo de predicción meteorológico propuesto está enfocado en el ámbito del Análisis Predictivo, que está dentro de Azure AI Library, su objetivo es proporcionar un esquema de trabajo para el flujo de análisis predictivo de pronóstico del tiempo en base a información real alojada en la nube. El modelo de predicción meteorológico representa la infraestructura de solución que soportará al modelo predictivo.

Para la definición de este modelo de predicción meteorológico se investigó sobre modelos de implementación que están dentro de Azure AI Library, la metodología de despliegue del mismo, además de recopilar información de la estructura de solución.

De acuerdo a nuestra investigación, la herramienta Azure ML Studio posee un escenario de implementación que va de acuerdo a las necesidades de la solución, como se observa en la figura 66.



Figuras 66. Modelado en Azure ML Studio
Fuente: Elaboración propia

Se agregó un módulo R-script para limpiar los datos de humedad y de temperatura.

Para que el modelo funcione correctamente, debe poder convertir los datos de temperatura y humedad en datos numéricos. Se agregó un módulo de secuencia de comandos R al modelo meteorológico para eliminar filas que contienen valores de datos de humedad y de temperatura que no se pueden convertir a números.

Al lado izquierdo de la ventana de Azure Machine Learning Studio (clásica), expandimos el panel de herramientas. Escribimos ejecutar en el cuadro de búsqueda, seguidamente realizamos la selección del módulo Ejecutar script de R.



Figuras 67. Agregando un R Script
Fuente: Elaboración propia

Se realizó la selección del nuevo módulo ejecutar script R y abrir la ventana de propiedades en el cual se copió y pegó el siguiente código en el cuadro R Script.

```
# Map 1-based optional input ports to variables
data <- maml.mapInputPort(1) # class: data.frame

data$temperature <- as.numeric(as.character(data$temperature))
data$humidity <- as.numeric(as.character(data$humidity))
completedata <- data[complete.cases(data), ]
maml.mapOutputPort('completedata')
```

Implementar el servicio web predictivo

Se realizó la validación del modelo, configuramos un servicio web predictivo basado en el modelo y luego implementaremos el servicio web.

Se ejecutó para poder validar los pasos del modelo, pudiendo tardar unos minutos en validar. Seleccionamos la sección SET UP WEB SERVICE/ Servicio Web Predictivo, mostrando el diagrama del experimento predictivo. En el diagrama del experimento predictivo se eliminó la conexión entre el módulo de entrada del servicio web y se realizó la

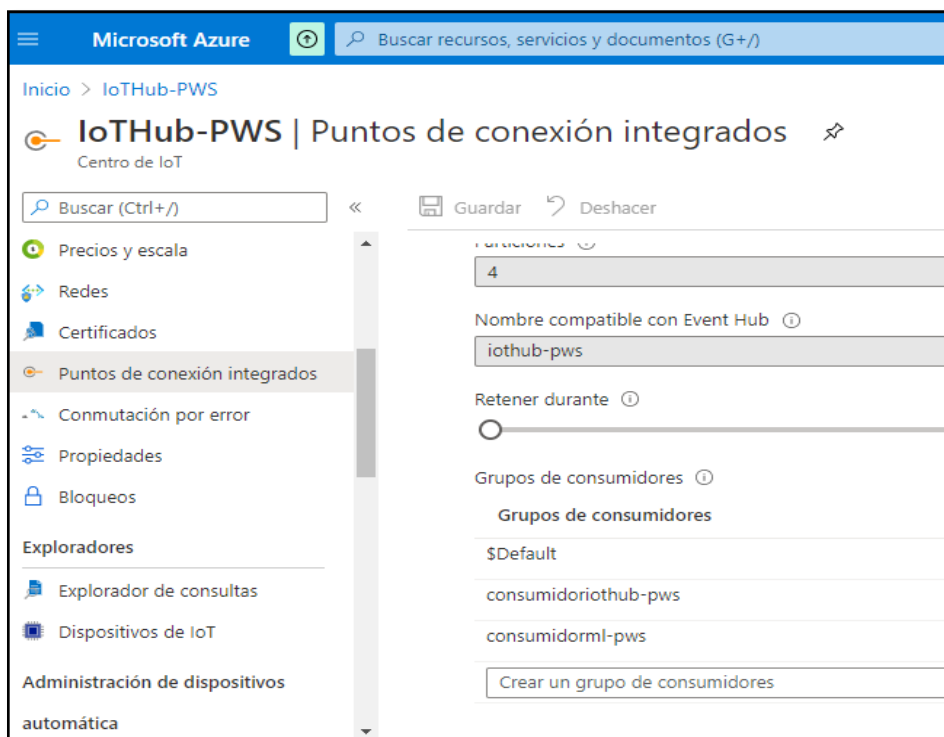
Configuración del centro IoT para comparar con datos capturados en tiempo real

Se agregó un Grupo de consumidores para ML - PWS

El grupo de consumidores nos proporcionan vistas independientes del flujo de eventos que permiten que los servicios y aplicaciones de Azure consuman datos de forma independiente desde el mismo punto de conexión de Event Hub. Entonces se creó un grupo de consumidores al punto final integrado de nuestro centro de IoT que se usó para extraer datos del punto final.

Agregando un grupo de consumidores a nuestro centro de IoT, de acuerdo a estos pasos:

1. En la parte izquierda del centro de IoT seleccionamos Puntos de conexión integrados, seleccione Eventos en el panel derecho e ingresamos un nombre en Grupos de consumidores y guardar.



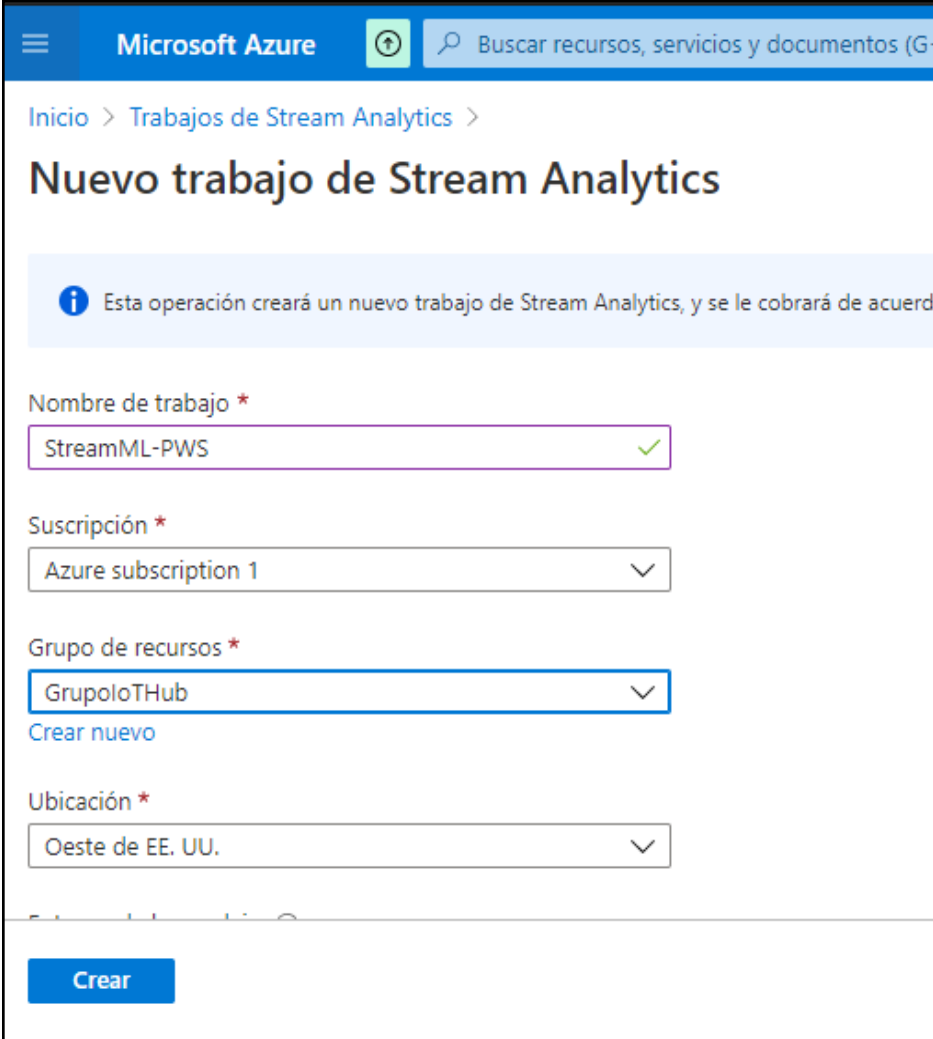
Figuras 70. Agregando un grupo de consumidor para ML - PWS

Fuente: Elaboración propia

Creacion, configuracion y ejecución del trabajo de Stream Analytics para ML - PWS

Creamos un trabajo de Stream Analytics, seguimos estos pasos para ingrese la siguiente información del trabajo.

- **Nombre del trabajo:** Nombre exclusivo.
- **Suscripción:** Suscripción si es diferente a la predeterminada.
- **Grupo de recursos:** Igual al grupo de recursos que el IoT Hub.
- **Ubicación:** Igual a la ubicación que el grupo de recursos.
- Dejar el resto de campos con el valor predeterminado y crear.



The screenshot shows the Microsoft Azure portal interface for creating a new Stream Analytics job. The breadcrumb navigation is 'Inicio > Trabajos de Stream Analytics >'. The main heading is 'Nuevo trabajo de Stream Analytics'. Below the heading is an information icon and a message: 'Esta operación creará un nuevo trabajo de Stream Analytics, y se le cobrará de acuerdo...'. The form contains the following fields:

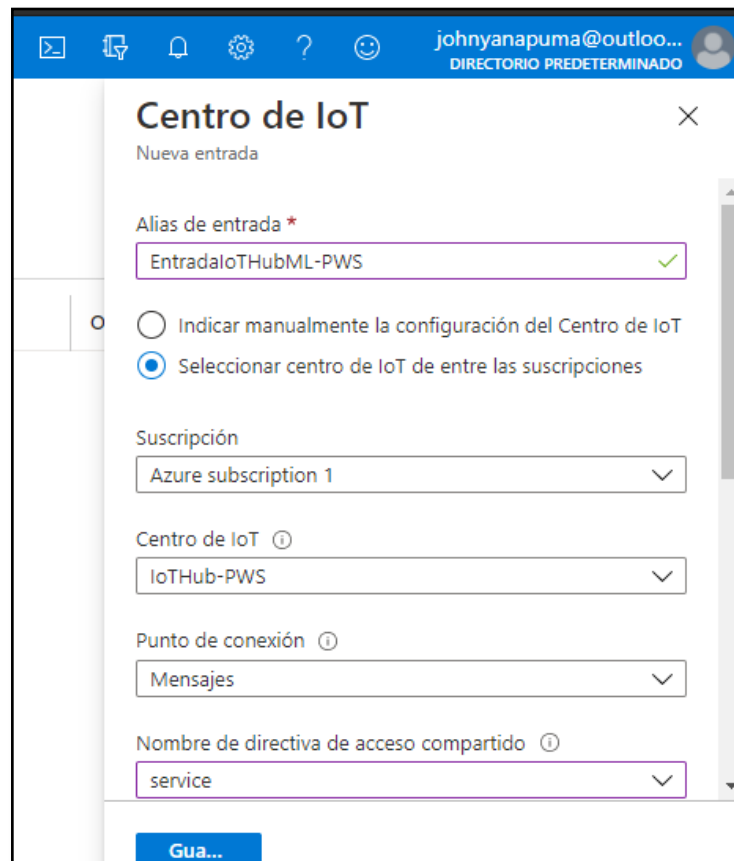
- Nombre de trabajo ***: A text input field containing 'StreamML-PWS' with a green checkmark on the right.
- Suscripción ***: A dropdown menu showing 'Azure subscription 1'.
- Grupo de recursos ***: A dropdown menu showing 'GrupoloTHub'.
- Ubicación ***: A dropdown menu showing 'Oeste de EE. UU.'.

At the bottom of the form is a blue button labeled 'Crear'.

Figuras 71. Agregando un Analytics para ML - PWS
Fuente: Elaboración propia

Agregamos una entrada al trabajo de Stream Analytics para ML - PWS

En el panel seleccionamos Entradas, agregando entradas de flujo y luego seleccione IoT Hub, ingresando información como muestra la imagen.



The screenshot displays the 'Centro de IoT' (IoT Hub) configuration interface for a new input. The title bar shows the user 'johnyanapuma@outloo...' and 'DIRECTORIO PREDETERMINADO'. The main heading is 'Centro de IoT' with a close button. Below it, the section is titled 'Nueva entrada' (New Input). The configuration fields are as follows:

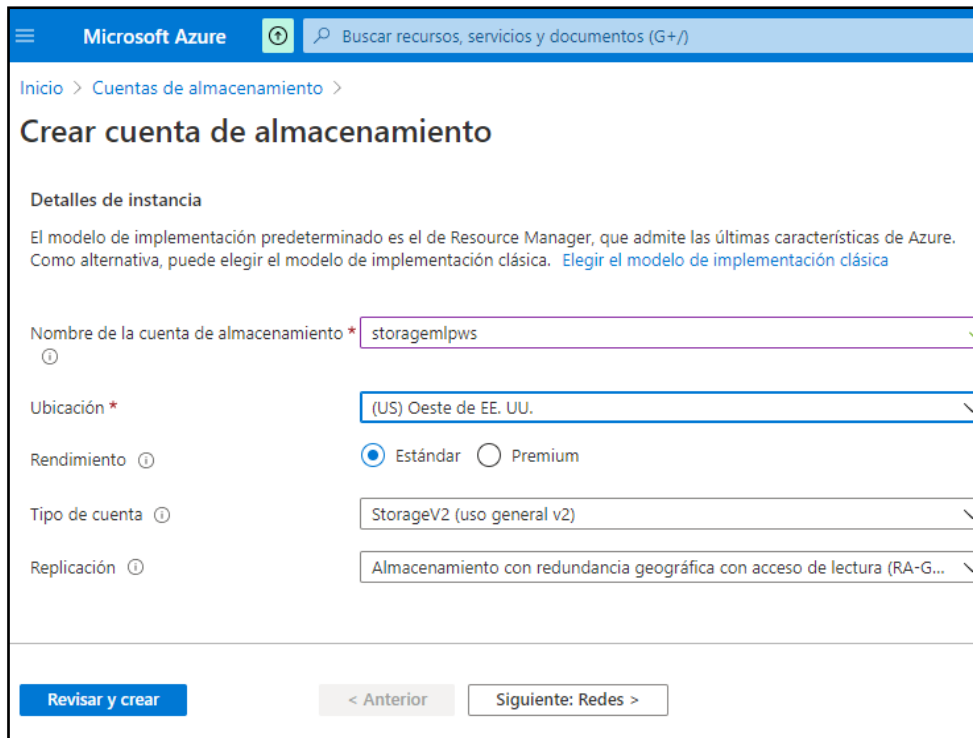
- Alias de entrada ***: EntradaIoTHubML-PWS (with a green checkmark)
- Indicar manualmente la configuración del Centro de IoT**: (unselected)
- Seleccionar centro de IoT de entre las suscripciones**: (selected)
- Suscripción**: Azure subscription 1 (dropdown)
- Centro de IoT**: IoT Hub-PWS (dropdown)
- Punto de conexión**: Mensajes (dropdown)
- Nombre de directiva de acceso compartido**: service (dropdown)

A blue button labeled 'Gua...' is located at the bottom of the form.

Figuras 72. Agregando una entrada en Stream Analytics para ML - PWS
Fuente: Elaboración propia

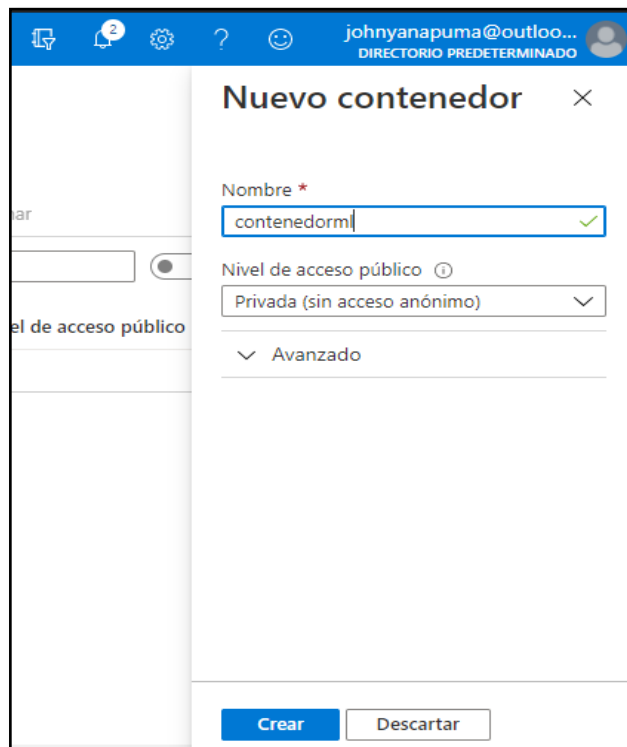
Cuenta de almacenamiento para ML – PWS

En la plataforma de Azure hemos ingresado información en el recurso “cuenta de almacenamiento” y crear, como muestra la imagen.



Figuras 73. Creando una cuenta de almacenamiento para ML - PWS
Fuente: Elaboración propia

Agregando un contenedor para ML - PWS



Figuras 74. Agregando un contenedor en la cuenta del almacenamiento para ML - PWS
Fuente: Elaboración propia

Agregar una salida al nuestro trabajo de Stream Analytics para ML - PWS

En el panel Salidas, se agregó una salida al trabajo en la que se elegirá seleccionar almacenamiento de suscripciones e ingresar la información como se muestra en la figura 75.

The screenshot shows a configuration window for a new output named 'SalidaBlobStorageML'. The user has selected the option to 'Seleccionar almacenamiento de las suscripciones' (Select storage from subscriptions). The configuration includes the following details:

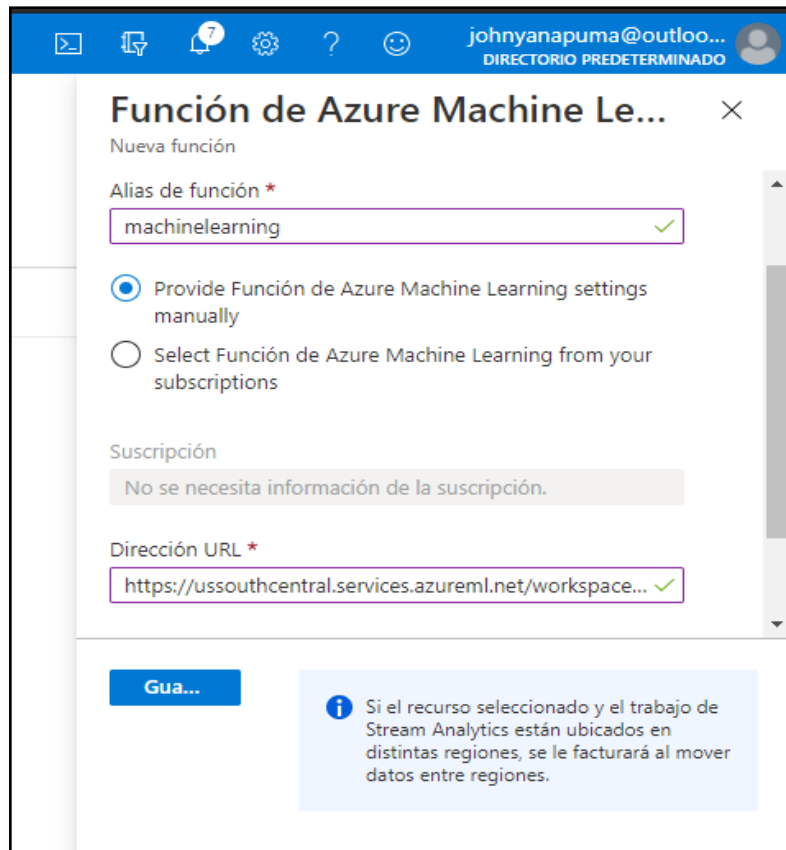
- Suscripción:** Azure subscription 1
- Cuenta de almacenamiento:** storagemlpws
- Clave de cuenta de almacenamiento:** [Redacted]
- Contenedor:** Usar existente (selected)

A 'Gua...' button is visible at the bottom of the configuration panel.

Figuras 75. Agregando una salida en Stream Analytics para ML - PWS
Fuente: Elaboración propia

Agregar una función al trabajo de Stream Analytics para llamar al servicio web que se implementó para ML - PWS

En el panel Funciones, se agregó una función y llenar con la información como muestra la figura 76.



Figuras 76. Agregando una función en Stream Analytics para ML - PWS
Fuente: Elaboración propia

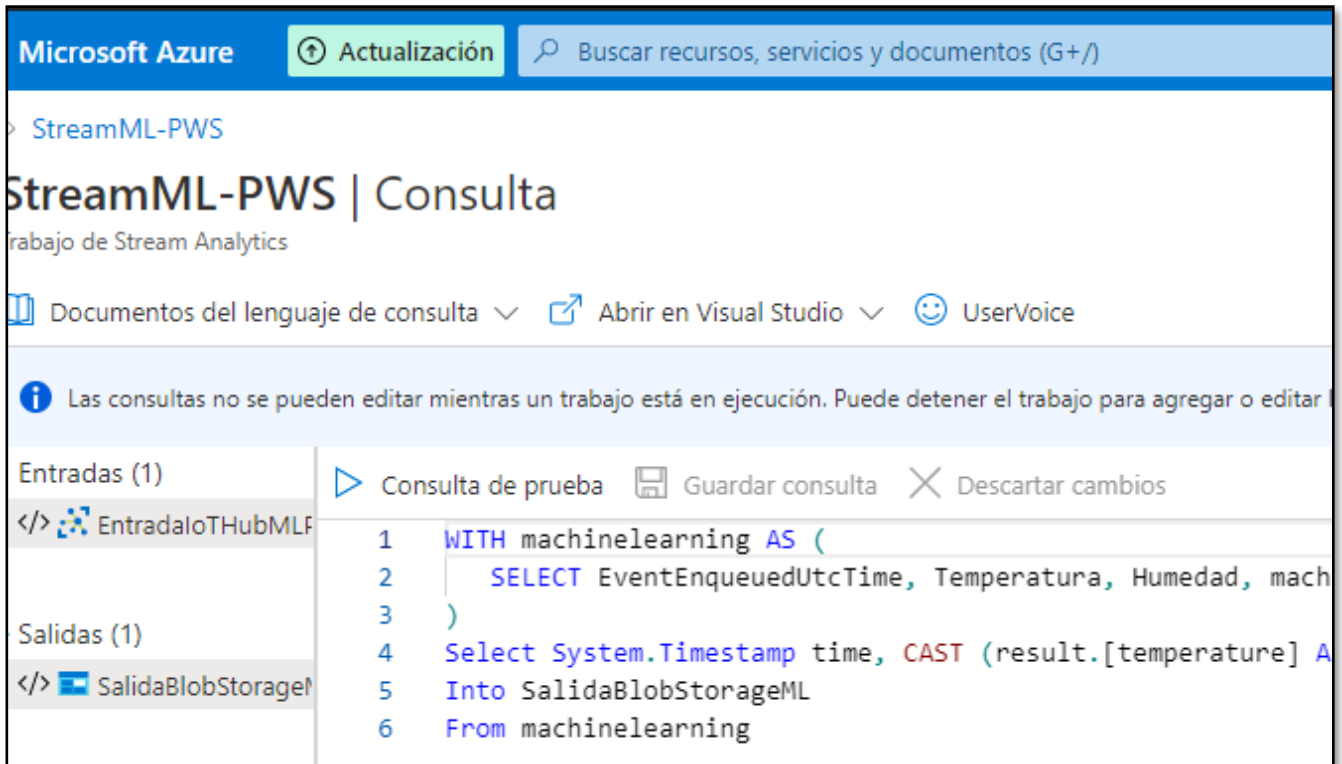
Configurar la consulta del trabajo de Stream Analytics para ML - PWS

1. En Topología del trabajo, seleccionamos Consulta.
2. Se reemplaza el código existente con el siguiente código:

```
WITH machinelearning AS (
    SELECT      EventEnqueuedUtcTime,      temperature,      humidity,
    machinelearning(temperature, humidity) as result from [YourInputAlias]
)
Select System.Timestamp time, CAST (result.[temperature] AS FLOAT) AS
temperature, CAST (result.[humidity] AS FLOAT) AS humidity, CAST
(result.[scored probabilities] AS FLOAT ) AS 'probabilities of rain'
Into [YourOutputAlias]
From machinelearning
```

- Reemplace [YourInputAlias] con el alias de entrada del trabajo.

- Reemplazar [YourOutputAlias] con el alias de salida del trabajo.



Figuras 77. Ingresamos una consulta en Stream Analytics para ML - PWS

Fuente: Elaboración propia

b) Modelado usando Sci - Kit Learn

Se utilizó también la librería de Scikit-learn, librería de mucha utilidad para Machine Learning en Python, siendo de código abierto y proporciona una amplia gama de algoritmos. Esta librería incluye los siguientes paquetes: Numpy, pandas, SciPy, matplotlib, IP[y].

Para este paso se hizo uso de uno de los componentes de scikit-learn que fue, Algoritmos de aprendizaje supervisado y los algoritmos a utilizar fueron: regresión logística y árboles de decisión. Para probar la calidad y validez del modelo se utilizó: confianza predictiva (accuracy).

- **Algoritmo Decision Tree**

Se ha creado un archivo IPython donde se codificó el modelo, y se importó las librerías a usarse, para el modelo de decision tree, como muestra la figura 78.

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
%matplotlib inline
plt.rc("font", size=14)

from sklearn import tree
import sklearn.model_selection as model
import seaborn as sns
import statsmodels.api as sm
from sklearn import metrics
import graphviz
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
```

Figuras 78. Importar librerías para el modelo Decision tree

Fuente: Elaboración propia.

```
In [4]: data = pd.read_csv("DataSenamhi3.csv", header=0)
print(data.shape)

data = data.dropna() # losing the lines that c/indv?
print(data.shape)
print(list(data.columns))

(40748, 10)
(40748, 10)
['ANO', 'MES', 'DIA', 'HORA', 'DIRVIENTO', 'VELVIENTO', 'PRECIPITACION', 'HUMEDAD', 'TEMPERATURA', 'RAIN']
```

Figuras 79. Importación del dataset

Fuente: Elaboración propia.

En la figura 79, se procedió a importar los datos y se almaceno esta información dentro de la variable data y se imprimió las columnas con la que cuenta el conjunto de datos.

```
data.head()
```

	ANO	MES	DIA	HORA	DIRVIENTO	VELVIENTO	PRECIPITACION	HUMEDAD	TEMPERATURA	RAIN
0	2015	12	1	0	109	0.5	0.0	71	7.1	False
1	2015	12	1	1	293	2.1	0.0	77	6.6	False
2	2015	12	1	2	247	1.6	0.0	74	6.4	False
3	2015	12	1	3	142	1.6	0.0	79	5.5	False
4	2015	12	1	4	276	1.8	0.0	79	5.9	False

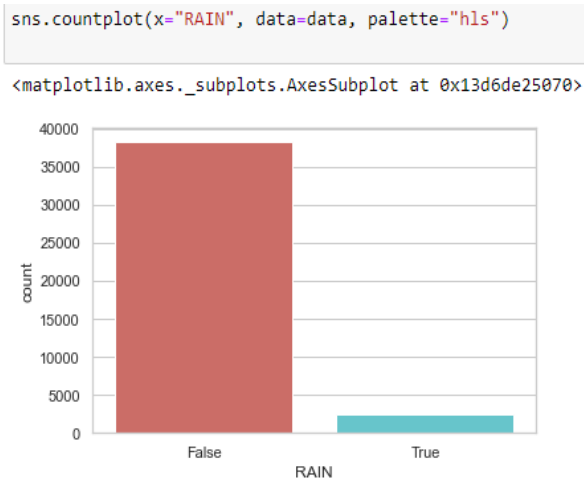
```
data['RAIN'].value_counts()

False    38244
True      2504
Name: RAIN, dtype: int64
```

Figuras 80. Conteo del dataset

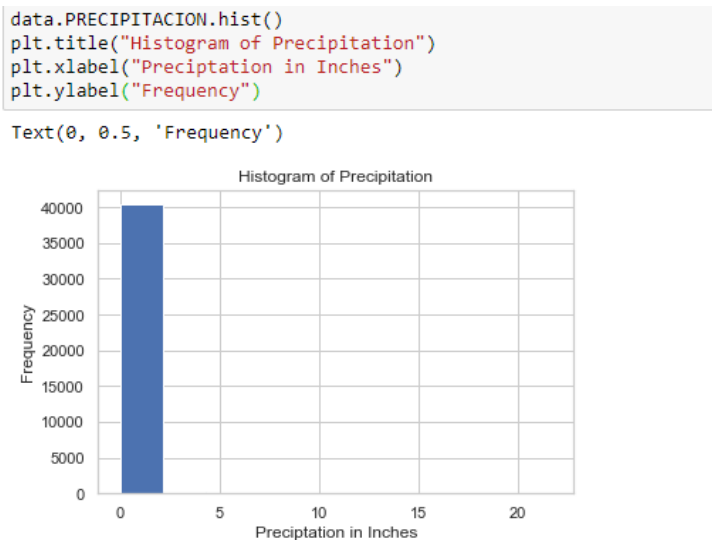
Fuente: Elaboración propia.

En la figura 80, se visualiza los primeros cinco datos al ejecutar el método head. El método value_counts devuelve una estructura conteniendo los valores presentes y el número de ocurrencias de false (no llueve) y true (llueve).



Figuras 81. situación de precipitación
Fuente: Elaboración propia.

En la figura 81 se muestra un gráfico de la variable RAIN, en la que se ve que hay más datos de false (no lluvia) con un aproximado de 38244 con respecto a true (lluvia) 2504.



Figuras 82. Histograma de precipitación
Fuente: Elaboración propia.

En la figura 82, se muestra un histograma, en la que se calcula las ocurrencias de datos de entrada de la variable precipitación.

La separación de la data frame, en datos de entrenamiento y prueba se realizó de la siguiente manera:

```
: X = data[['HUMEDAD', 'TEMPERATURA']]  
y = data["RAIN"]  
  
# convert to  
y = data.RAIN.astype(int)
```

Figuras 83. Definición de las variables para el modelo DecisionTreeClassifier.

Fuente: Elaboración propia.

En la figura 83, se procedió a definir las variables de “X” y “Y” en nuestro modelo, “X” son todas las variables de nuestro conjunto de datos (features columnas de entrenamiento), por otra parte “Y” es igual a los datos correspondientes a “target” que cuenta con una sola columna con true (llueve) y false (no llueve).

```
: X_train, X_test, y_train, y_test = model.train_test_split(X, y, test_size=0.3, random_state=0)  
clf = tree.DecisionTreeClassifier()  
clf = clf.fit(X_train, y_train)
```

Figuras 84. Separación de datos de entrenamiento y prueba/entrenamiento del algoritmo DecisionTreeClassifier.

Fuente: Elaboración propia.

En la figura 84, se realizó la separación de datos de entrenamiento y prueba para probar el algoritmo, para ello se hizo el uso de `train_test_split` para la separación de datos, tomando un 20% de los mismos para utilizarlos como prueba y el 60% para el entrenamiento del modelo. Una vez que se tiene los datos listos para empezar a construir el modelo, lo que se hizo fue importar decision tree definiendo el algoritmo a utilizar para este caso *DecisionTreeClassifier*, seguidamente se procedió a entrenar el modelo utilizando la instrucción *fit* y los datos de “X” como también “Y” de entrenamiento.

```

: y_pred = clf.predict(X_test)
print('Accuracy of decision tree classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))

```

Accuracy of decision tree classifier on test set: 0.93

Figuras 85. Predicción con DecisionTreeClassifier

Fuente: Elaboración propia.

En la figura 85 se hizo el uso de la instrucción *predict* y los datos de prueba, para realizar la predicción correspondiente.

```

from sklearn import model_selection
from sklearn.model_selection import cross_val_score
kfold = model_selection.KFold(n_splits=10, random_state=7)
modelCV = tree.DecisionTreeClassifier()
scoring = 'accuracy'
results = model_selection.cross_val_score(modelCV, X_train, y_train, cv=kfold, scoring=scoring)
print("10-fold cross validation average accuracy: %.3f" % (results.mean()))

```

10-fold cross validation average accuracy: 0.926

Figuras 86. Porcentaje de precisión del modelo

Fuente: Elaboración propia.

```

from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)

```

```

[[11274  196]
 [  704   51]]

```

Figuras 87. Matriz de confusión

Fuente: Elaboración propia

En la figura 87, se calcula las métricas para verificar el rendimiento del modelo, primero se obtiene la matriz de confusión haciendo uso de la librería *metrics* de *scikit learn*.

```

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

```

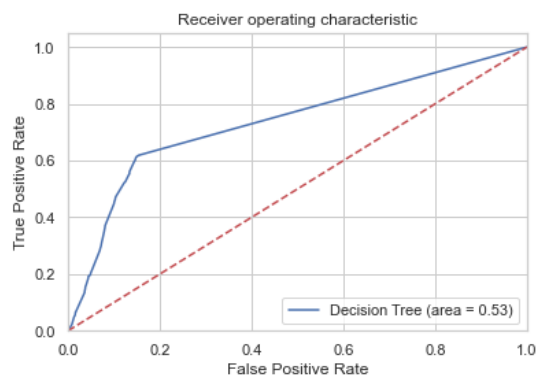
	precision	recall	f1-score	support
0	0.94	0.98	0.96	11470
1	0.21	0.07	0.10	755
accuracy			0.93	12225
macro avg	0.57	0.53	0.53	12225
weighted avg	0.90	0.93	0.91	12225

Figuras 88. Porcentaje de precisión del modelo

Fuente: Elaboración propia.

En la figura 88, calculamos la precisión del algoritmo, se usó de `accuracy_score`, teniendo una exactitud del modelo: 0.93.

```
In [18]: from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, clf.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, clf.predict_proba(X_test)[:,:1])
plt.figure()
plt.plot(fpr, tpr, label='Decision Tree (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC_DecisionTree')
plt.show()
```



Figuras 89. Característica operativa del receptor
Fuente: Elaboración propia.

- **Algoritmo Logistic Regression**

Para este algoritmo también se realiza los mismos procedimientos descritos anteriormente, comenzando con la importación de las librerías a utilizar para el algoritmo de regresión logística.

```

import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
%matplotlib inline
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
import sklearn.model_selection as model
import seaborn as sns
import statsmodels.api as sm
from sklearn import metrics

sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)

```

```

data = pd.read_csv("DataSenamhi3.csv", header=0)
print(data.shape)

data = data.dropna() # Losing the lines that clinde?
print(data.shape)
print(list(data.columns))

(40748, 10)
(40748, 10)
['ANO', 'MES', 'DIA', 'HORA', 'DIRVIENTO', 'VELVIENTO', 'PRECIPITACION', 'HUMEDAD', 'TEMPERATURA', 'RAIN']

```

Figuras 90. Importación de librerías y carga de datos – Logistic regression
Fuente: Elaboración propia.

```

data.head()

```

	ANO	MES	DIA	HORA	DIRVIENTO	VELVIENTO	PRECIPITACION	HUMEDAD	TEMPERATURA	RAIN
0	2015	12	1	0	109	0.5	0.0	71	7.1	False
1	2015	12	1	1	293	2.1	0.0	77	6.6	False
2	2015	12	1	2	247	1.6	0.0	74	6.4	False
3	2015	12	1	3	142	1.6	0.0	79	5.5	False
4	2015	12	1	4	276	1.8	0.0	79	5.9	False

```

data['RAIN'].value_counts()

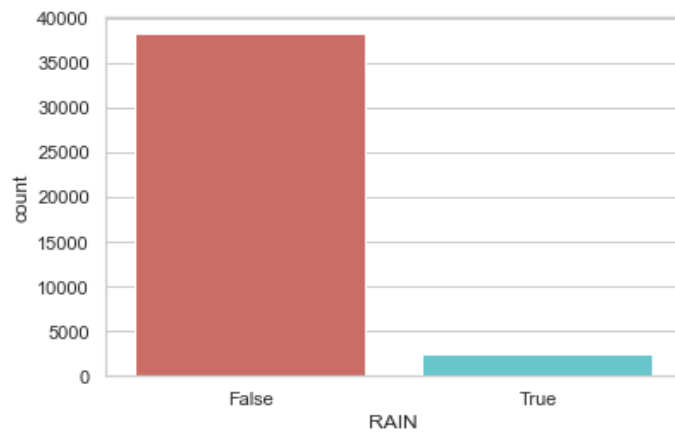
False    38244
True      2504
Name: RAIN, dtype: int64

```

Figuras 91. Conteo del dataset – Logistic regression
Fuente: Elaboración propia.

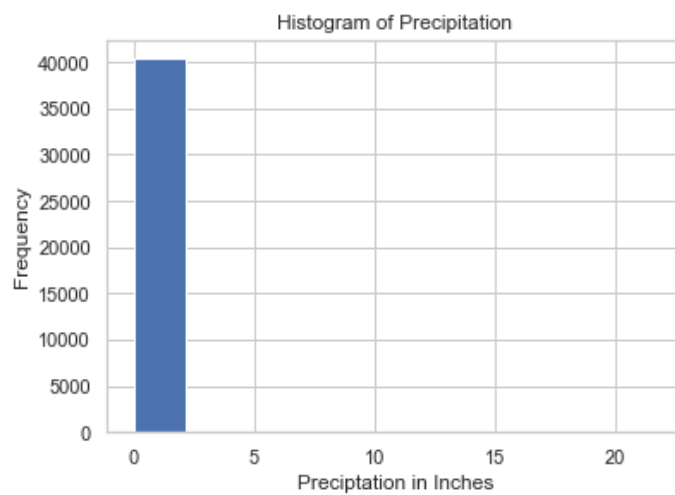
```
sns.countplot(x="RAIN", data=data, palette="hls")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fecb6061f0>
```



```
data.PRECIPITACION.hist()  
plt.title("Histogram of Precipitation")  
plt.xlabel("Precipitation in Inches")  
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```



Figuras 92. Histograma de precipitación – Logistic regression

Fuente: Elaboración propia.

```

: X = data[['HUMEDAD', 'TEMPERATURA']]
  y = data["RAIN"]

# convert to
y = data.RAIN.astype(int)

```

```

: logit_model=sm.Logit(y,X)
  result=logit_model.fit()
  print(result.summary())

```

```

Optimization terminated successfully.
Current function value: 0.283937
Iterations 7

```

```

Logit Regression Results
=====
Dep. Variable:          RAIN   No. Observations:          40748
Model:                  Logit   Df Residuals:              40746
Method:                 MLE     Df Model:                   1
Date:                   Sun, 22 Nov 2020   Pseudo R-squ.:            -0.2295
Time:                   10:47:42   Log-Likelihood:           -11570.
converged:              True     LL-Null:                   -9410.4
Covariance Type:       nonrobust   LLR p-value:              1.000
=====
                coef    std err          z      P>|z|    [0.025    0.975]
-----
HUMEDAD         -0.0212     0.000   -65.406     0.000    -0.022    -0.021
TEMPERATURA     -0.1439     0.002   -63.659     0.000    -0.148    -0.139
=====

```

*Figuras 93. Separación de datos de entrenamiento y prueba – Logistic regression
Fuente: Elaboración propia*

```

X_train, X_test, y_train, y_test = model.train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

```

```

LogisticRegression()

```

```

y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))

```

```

Accuracy of logistic regression classifier on test set: 0.94

```

```

from sklearn import model_selection
from sklearn.model_selection import cross_val_score
kfold = model_selection.KFold(n_splits=10, random_state=7)
modelCV = LogisticRegression()
scoring = 'accuracy'
results = model_selection.cross_val_score(modelCV, X_train, y_train, cv=kfold, scoring=scoring)
print("10-fold cross validation average accuracy: {:.3f} % (results.mean())")

```

```

10-fold cross validation average accuracy: 0.938

```

```

: from sklearn.metrics import confusion_matrix
  confusion_matrix = confusion_matrix(y_test, y_pred)
  print(confusion_matrix)

[[11455   15]
 [  751    4]]

```

```

from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

```

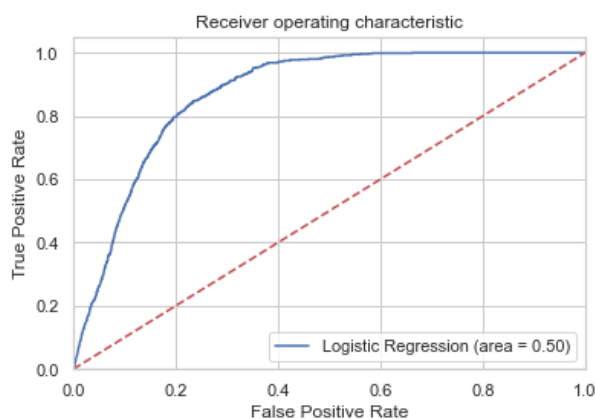
	precision	recall	f1-score	support
0	0.94	1.00	0.97	11470
1	0.21	0.01	0.01	755
accuracy			0.94	12225
macro avg	0.57	0.50	0.49	12225
weighted avg	0.89	0.94	0.91	12225

Figuras 94. Matriz de confusión y reporte de precisión del modelo – Logistic regression
Fuente: Elaboración propia

```

from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC_Logisitic')
plt.show()

```



Figuras 95. Característica operativa del receptor
Fuente: Elaboración propia

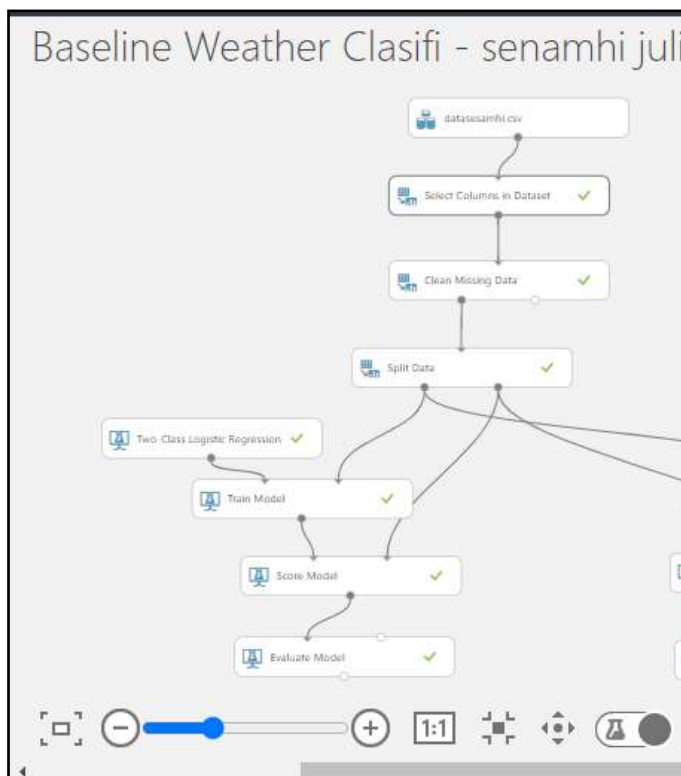
3.3.2.5.Evaluación

Evaluación de un modelo de Clasificación Binaria con Microsoft Azure Machine Learning Studio

En un escenario de clasificación binaria, la variable objetivo tiene solo dos resultados posibles, por ejemplo: {0, 1} o {false, true}, {negative, positive}, en nuestro caso sería {NO LLUVIA, LLUVIA}

Creamos el experimento: Agregamos los siguientes módulos al área de trabajo en Azure Machine Learning Studio (clásico):

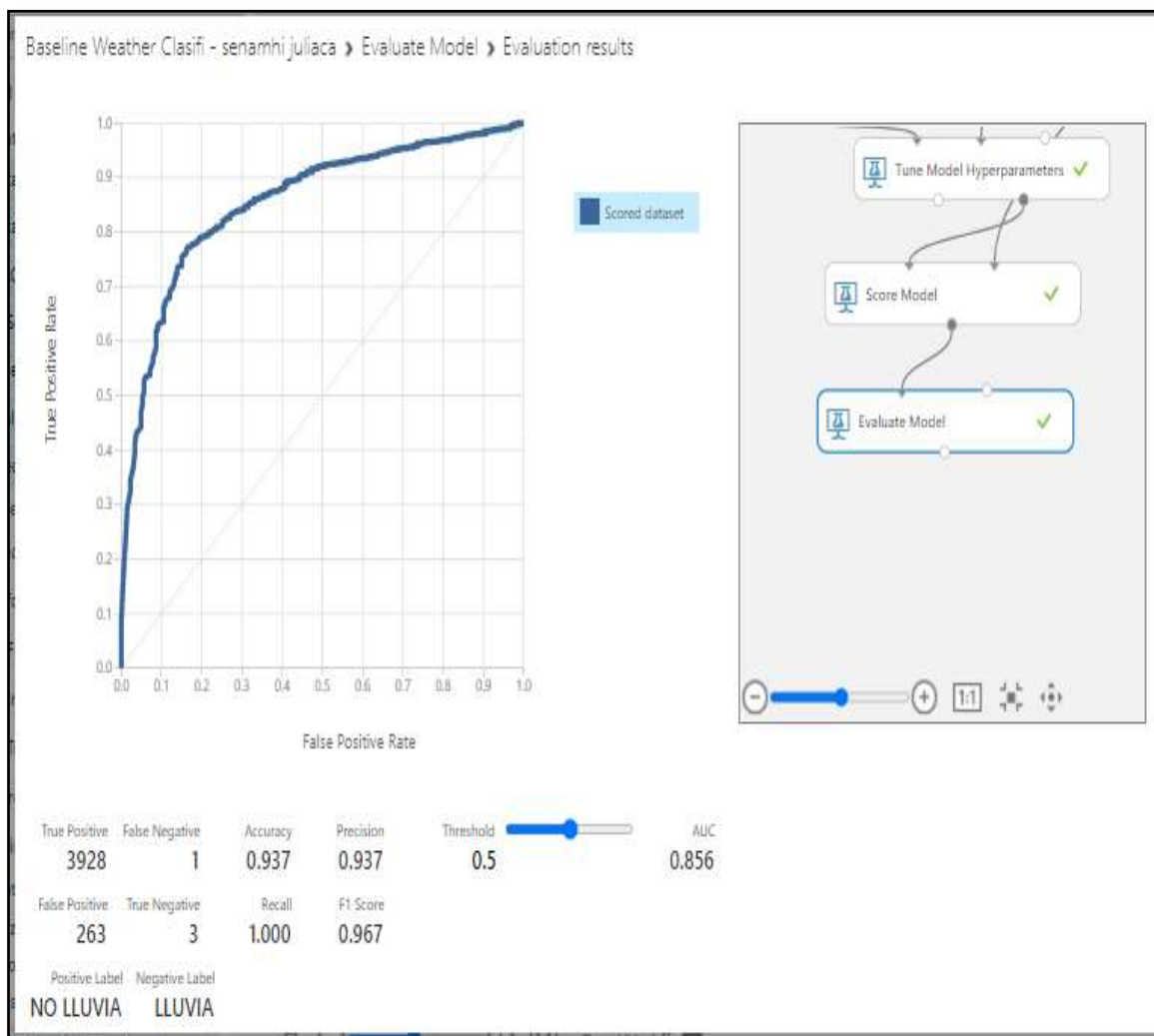
- ✓ Conjunto de datos de clasificación binaria de predicción de clima (datasenamhi.csv)
- ✓ **Regresión logística de dos clases**
- ✓ Train Model
- ✓ Score Model
- ✓ Evaluate Model



Figuras 96. Evaluación del modelo de clasificación binaria – regresión logística
Fuente: Elaboración propia

Inspección de los resultados de la evaluación

Después de ejecutar el experimento, hacemos clic en el puerto de salida del módulo Evaluar modelo y seleccionar **Visualizar** para ver los resultados de la evaluación (figura 89). Las métricas de evaluación disponibles para los modelos de clasificación binaria son: Accuracy, Precision, Recall, F1 Score y AUC; teniendo como resultado del accuracy de 0.937 equivalente a 93.7 % de precisión del algoritmo. El módulo genero una matriz de confusión que muestra el número de positivos verdaderos, falsos negativos, falsos positivos y negativos verdaderos, así como curvas ROC, Precision/Recall y Lift.



Figuras 97. Resultado de la evaluación de clasificación binaria - regresión logística.
Fuente: Elaboración Propia

Evaluación con la herramienta de Sking Learn

Se lleva a cabo la evaluación del modelo para asegurar su funcionamiento, con el modelo en despliegue se planificará su monitoreo y mantenimiento correspondiente para mejorar su desempeño del modelo. La evaluación se realizó con Accuracy Score.

```
X_train, X_test, y_train, y_test = model.train_test_split(X, y, test_size=0.3, random_state=0)
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
print('Accuracy of decision tree classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))

Accuracy of decision tree classifier on test set: 0.93
```

Figuras 98. Porcentaje de precisión del modelo decision tree classifier.

Fuente: Elaboración propia.

Se puede observar en la figura 98, que el modelo de DecisionTree tuvo un 93% de nivel de predicción, el porcentaje es considerablemente muy bueno.

```
: X_train, X_test, y_train, y_test = model.train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

: LogisticRegression()

: X_train, X_test, y_train, y_test = model.train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

: LogisticRegression()

: y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))

Accuracy of logistic regression classifier on test set: 0.94
```

Figuras 99. Porcentaje de precisión del modelo Logistic regression classifier.

Fuente: Elaboración Propia.

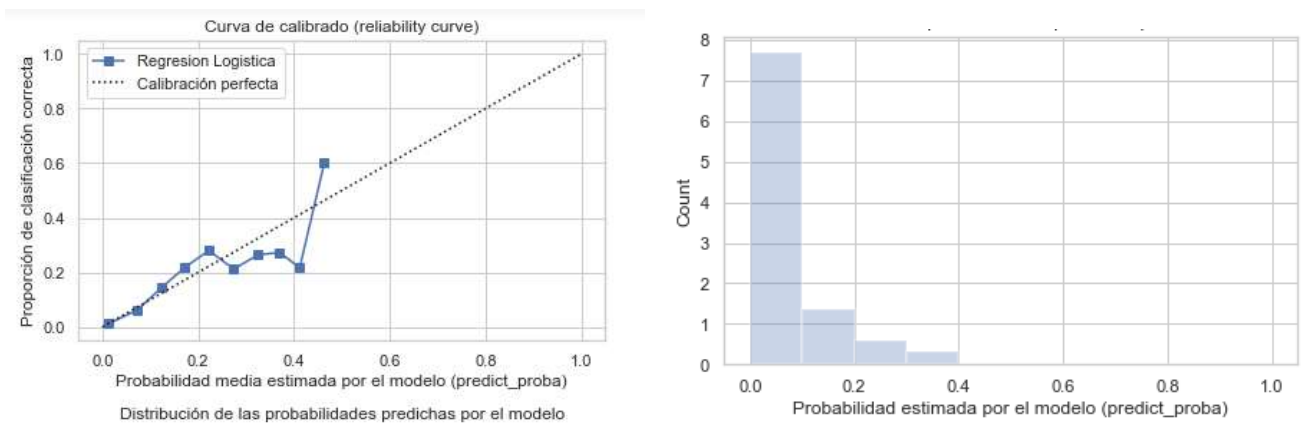
En la figura 99, se observó que se tiene un 94% de nivel de predicción, porcentaje muy bueno y más alto que la anterior.

Comparando los modelos se tiene que el modelo Logistic regression classifier tiene un nivel de confianza más alto que el otro modelo.

Se hizo uso de dos herramientas para la evaluación de los algoritmos, una en la plataforma de Microsoft Azure y la biblioteca de Scikit-learn con el fin de poder comparar nuestros algoritmos, teniendo un rango de diferencia de 0.3 % para el algoritmo de regresión logística, siendo un porcentaje mínimo y así validando que este algoritmo es óptimo.

Al tener al algoritmo de regresión logística con un porcentaje de confianza más alta que el anterior, se procedió a realizar la calibración respectiva del algoritmo antes mencionado. Para ello se ha realizado la calibración del modelo en Scikitlearn, conteniendo el modelo principal como el modelo de calibración.

Para crear el modelo calibrado se ha empleado la clase *CalibratedClassifierCV*, con los siguientes argumentos: *base_estimator* (modelo que se quiere calibrar), *method* (tipo de modelo empleado para calibrar el sigmoid o isotonic) y *cv* (estrategia de validación para el entrenamiento de ambos modelos).



Figuras 100. Curva del algoritmo de regresión logística sin calibrar
Fuente: Elaboración propia

```

brier_score = brier_score_loss(y_test, logreg.predict_proba(X = X_test)[:, 1])
print(f"Brier score = {brier_score}")
print("")
print(classification_report(y_test, logreg.predict(X = X_test)))

```

Brier score = 0.05350216839901229

	precision	recall	f1-score	support
0	0.94	1.00	0.97	11775
1	0.00	0.00	0.00	810
accuracy			0.94	12585
macro avg	0.47	0.50	0.48	12585
weighted avg	0.88	0.94	0.90	12585

Figuras 101. Brier score del algoritmo de regresión sin calibrar
Fuente: Elaboración propia

Calibración del modelo de Regresión Logística.

```

import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
%matplotlib inline
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
import sklearn.model_selection as model
import seaborn as sns
import statsmodels.api as sm
from sklearn import metrics

sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)

from sklearn.calibration import calibration_curve
from sklearn.calibration import CalibratedClassifierCV
from sklearn.metrics import brier_score_loss
from sklearn.metrics import classification_report

import warnings
warnings.filterwarnings('once')
warnings.simplefilter('ignore', (DeprecationWarning))

data = pd.read_csv("datajuliac4.csv", header=0)
print(data.shape)

data = data.dropna()
print(data.shape)
print(list(data.columns))

X = data[['HUMEDAD', 'TEMPERATURA']]
y = data["RAIN"]

# convert|
y = data.RAIN.astype(int)

```

```
X_train, X_test, y_train, y_test = model.train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

```
LogisticRegression()
```

```
# Creación del modelo calibrado
modelo_calibrado = CalibratedClassifierCV(logreg, cv=3, method='isotonic')
```

```
# Entrenamiento del modelo base y de la calibración
_ = modelo_calibrado.fit(X_train, y_train)
```

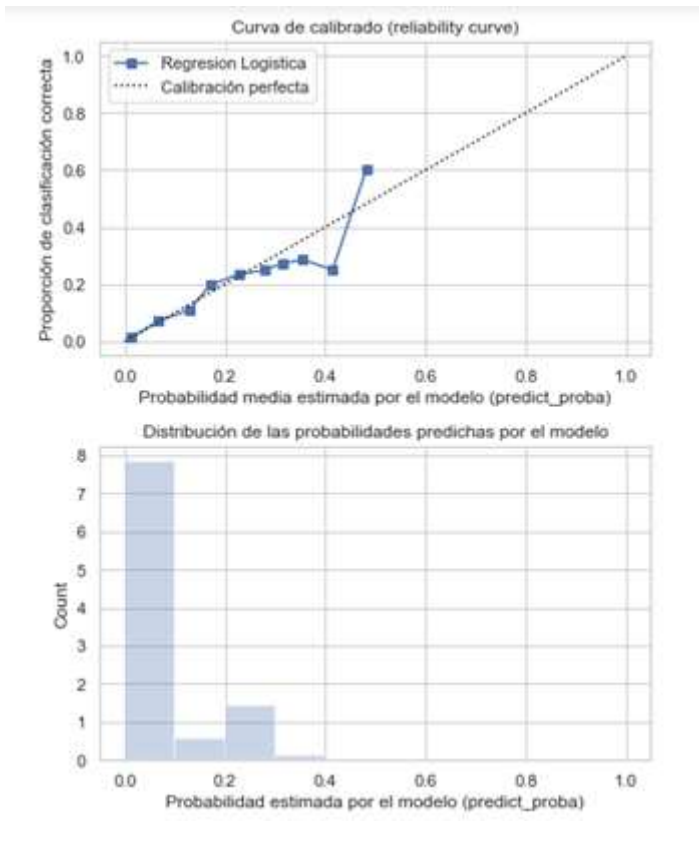
```
# Predicción con probabilidades calibradas
predicciones = modelo_calibrado.predict_proba(X = X_test)
# Se extraen las probabilidades de la clase positiva
prob_positivo = predicciones[:, 1]
```

```
fraccion_positivos, media_prob_predicha = calibration_curve(y_test, prob_positivo, n_bins=20)
```

```
fig, axs = plt.subplots(nrows=2, ncols=1, figsize=(6, 2*3.84))
|
axs[0].plot(media_prob_predicha, fraccion_positivos, "s-", label="Regresión Logística")
axs[0].plot([0, 1], [0, 1], "k:", label="Calibración perfecta")
axs[0].set_ylabel("Proporción de clasificación correcta")
axs[0].set_xlabel("Probabilidad media estimada por el modelo (predict_proba)")
axs[0].set_title('Curva de calibrado (reliability curve)')
axs[0].legend()

axs[1].hist(prob_positivo, range=(0, 1), bins=10, density=True, lw=2, alpha = 0.3)
axs[1].set_xlabel("Probabilidad estimada por el modelo (predict_proba)")
axs[1].set_ylabel("Count")
axs[1].set_title('Distribución de las probabilidades predichas por el modelo')

plt.tight_layout()
plt.show();
```



Figuras 102. curva calibrada
Fuente: Elaboración propia

```
brier_score = brier_score_loss(y_test, modelo_calibrado.predict_proba(X = X_test)[:, 1])
print(f"Brier score = {brier_score}")
print("")
print(classification_report(y_test, modelo_calibrado.predict(X = X_test)))
```

Brier score = 0.05296218092107595

	precision	recall	f1-score	support
0	0.94	1.00	0.97	11775
1	0.00	0.00	0.00	810
accuracy			0.94	12585
macro avg	0.47	0.50	0.48	12585
weighted avg	0.88	0.94	0.90	12585

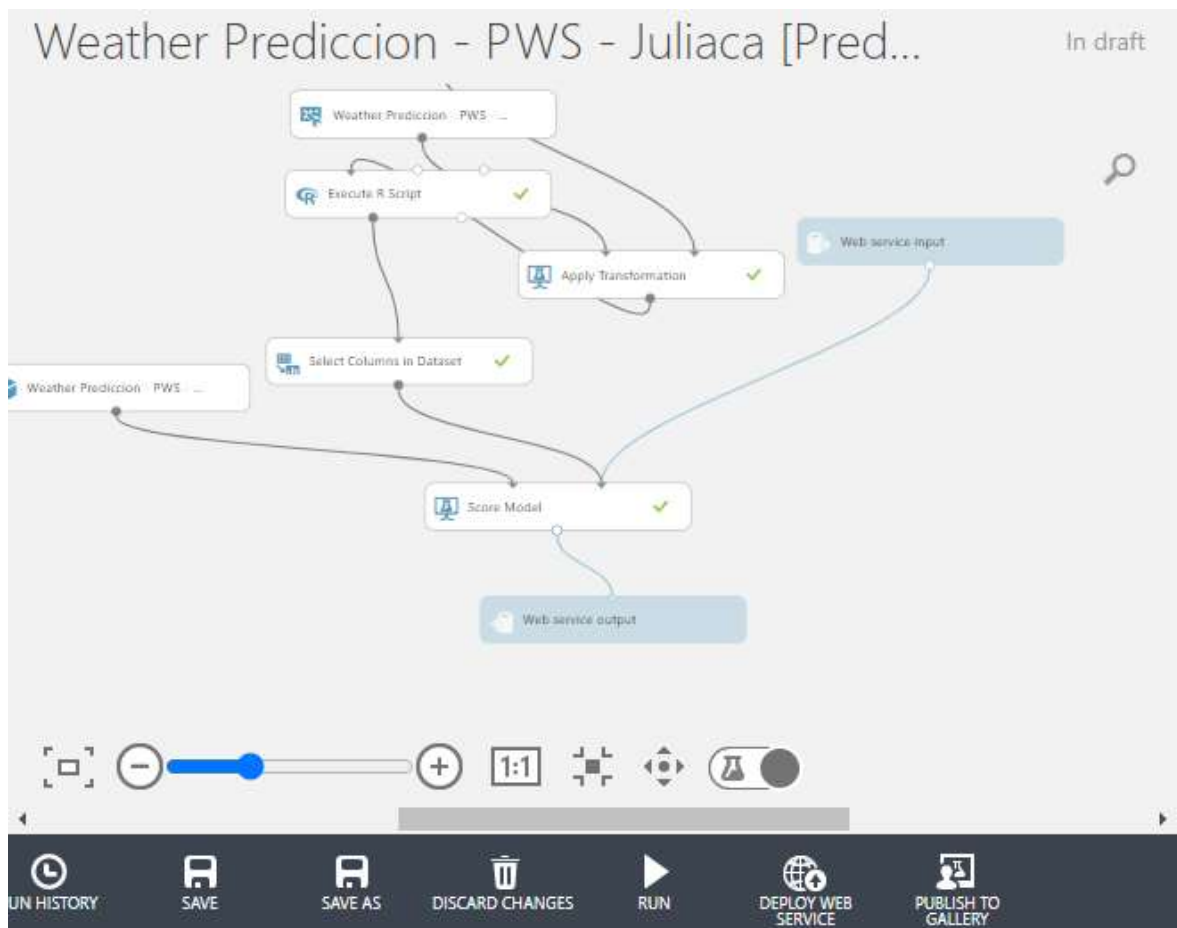
Figuras 103. brier score de la curva calibrada
Fuente: Elaboración propia

Se ha conseguido reducir el brier score, significa que ha mejorado la calibración del modelo. El accuracy del modelo no sea visto afectado por la calibración.

3.3.2.6. Implementación

Azure Machine Learning Studio (clásico) nos permitirá compilar y probar una solución de análisis predictivo. Luego implementaremos la solución como un servicio web.

Los servicios web de Machine Learning Studio (clásico) proporcionan una interfaz entre una aplicación y un modelo de puntuación de flujo de trabajo de Machine Learning Studio (clásico).



Figuras 104. Implementación del modelo.

Fuente: Elaboración propia

Las aplicaciones externas pueden realizar una comunicación en tiempo real con el modelo de puntuación de flujo de trabajo de Machine Learning Studio (clásico). Llamar a un servicio web de Machine Learning Studio (clásico) devuelve predicciones para

aplicaciones externas. Para invocar un servicio web de Machine Learning, debe pasar la clave de API que se generó cuando se implementó el servicio web. Los servicios web MachineLearningStudio (clásico) están basados en REST, una opción arquitectónica popular para proyectos de programación web.



Figuras 105. Clave Api

Fuente: Elaboración propia

Resultado final del modelo predictivo - PWS

1. Seleccione Cuentas de almacenamiento > su cuenta de almacenamiento> Blob Containers > su contenedor.
2. Descargue un archivo .csv para ver el resultado. La última columna registra la probabilidad de lluvia.

	A	B	C	D
1	time	temperature	humidity	Probabilidad de Lluvia
2	2020-11-23T02:20:00.3260000Z	21	37.5	0.958153903
3	2020-11-23T02:20:41.0200000Z	21	37.8	0.958153844
4	2020-11-23T02:21:18.6710000Z	21.1	38.1	0.958281696
5	2020-11-23T02:23:06.1460000Z	21	38.3	0.958153844
6	2020-11-23T02:24:24.7960000Z	21	38.7	0.958153784
7	2020-11-23T02:26:00.6390000Z	21	38.3	0.958153844
8	2020-11-23T02:30:47.6190000Z	20.9	38.3	0.958025634
9	2020-11-23T02:31:27.5390000Z	20.9	38.3	0.958025634
10	2020-11-23T02:34:20.4110000Z	20.8	38.3	0.957897007
11	2020-11-23T02:34:57.1210000Z	20.9	38.3	0.958025634
12	2020-11-23T02:35:33.7090000Z	20.8	38.2	0.957897067
13	2020-11-23T02:36:23.4830000Z	20.8	38.3	0.957897007
14	2020-11-23T02:37:34.6200000Z	20.8	38.3	0.957897007
15	2020-11-23T02:38:14.5500000Z	20.8	38.2	0.957897067
16	2020-11-23T02:54:57.0470000Z	7.7	27.1	0.937400699
17	2020-11-23T03:07:11.5450000Z	20.5	40.7	0.957508802

Figuras 106. Resultado final del modelo predictivo.

Fuente: Elaboración propia

CAPÍTULO IV. Resultados

Para el objetivo general, se ha logrado implementar el modelo predictivo, usando Machine Learning con la herramienta de Microsoft Azure ML Studio e IoT, implementando una mini estación meteorológica PWS, con el fin de capturar datos en tiempo real y comparar con el modelo predictivo, logrando predecir datos climáticos para la toma de decisiones para el sector agrario de la región Puno.

En cuanto al primer objetivo específico, se ha logrado implementar la mini estación meteorológica para capturar datos en tiempo real, primeramente se ha realizado un ensamblaje de conexiones en el simulador Fritzing donde se utilizó una placa ESP-32 basado en Arduino, sensores para clima como el DHT-22 que mide la temperatura y humedad y el barómetro BMP-180 que mide la presión atmosférica y la altitud, segundo se ensambló de manera física en un protoboard donde también se hizo una maqueta para usarlo en el exterior y mostrando los datos meteorológicos capturados en tiempo real haciendo uso de la plataforma de ThingSpeak, es como sigue:

Fecha	Hora	Temperatura	Humedad	Punto de rocío	Presión Atmosférica	Altitud
2020-12-06	18:43	19	51.8	8.7	64515	3650.511
2020-12-06	18:46	18.9	51.8	8.752	64514	3649.311
2020-12-06	18:49	18.6	52.8	8.758	64523	3648.711
2020-12-06	18:52	18.5	53.6	8.88	64536	3646.312

En cuanto el segundo objetivo específico, se ha logrado crear y configurar correctamente un entorno de trabajo en la nube, el cual consistió: primeramente, se creó una cuenta gratuita de 30 días en la plataforma de Microsoft Azure, segundo: para transmitir los datos capturados de la mini estación meteorológica PWS se creó un recurso de IoT Hub de Microsoft Azure, esto con el fin de procesar los datos y así nos permitirá comparar los datos climáticos con el modelo predictivo.

Con respecto al tercer objetivo específico de nuestra investigación el cual fue analizar y preparar datos históricos del SENAMHI, para alimentar al modelo predictivo, primeramente se obtuvo datos del senamhi, brindándonos información histórica en un formato “xlsx” de lecturas diarias en los últimos cinco años, del cual se realizó una limpieza de los datos en Excel, verificando si existía datos faltantes, nulos, etc. para ello tuvimos que aplicar diferentes técnicas como: completar los datos faltantes con el valor anterior, valor medio. Las etiquetas “SD” se sustituyó con un valor anterior, se agregó también una nueva columna para categorizar (true y false) correspondiente a lluvia y no lluvia, conversión de fechas. Fue así que a partir del archivo original se generó uno nuevo con la estructura requerida para tomarlo como input de los modelos predictivos guardada en formato “csv”.

Para el cuarto objetivo específico de nuestra investigación que fue, elección del algoritmo más óptimo para obtener predicciones más precisas. Se eligió el algoritmo más óptimo, basándose en el score Accuracy para tener una confianza predictiva/predicciones más precisas teniendo al modelo de DecisionTree con un 93% y Logistic regression classifier con un 94%, de las cuales se tiene que el modelo Logistic regression classifier tiene un nivel de confianza más alta y un nivel de predicción más acertada, también se ha realizado la calibración al modelo Logistic regression classifier logrando reducir el brier score.

Con respecto al quinto objetivo específico de implementar el modelo predictivo en la nube, la plataforma de cloud usada fue Microsoft Azure Machine Learning Studio, para ello se ha creado una cuenta gratuita iniciando sesión con una cuenta de Microsoft, seguidamente se ha creado un experimento (nombre que recibe un archivo en Azure ML Studio), se hizo la implantación del algoritmo de regresión logística, para mejorar su solidez sean incluido varios bloques de comprobación y entrenamiento del modelo, la plataforma nos permite evaluar el modelo analizando los parámetros como: precisión total y porcentaje de acierto. Finalmente se realizó la implementación como un servicio web de Azure siendo de esta

manera que los usuarios puedan enviar sus datos al modelo y el modelo devuelve las predicciones.

	A	B	C	D
1	time	temperature	humidity	Probabilidad de Lluvia
2	2020-11-23T02:20:00.3260000Z	21	37.5	0.958153903
3	2020-11-23T02:20:41.0200000Z	21	37.8	0.958153844
4	2020-11-23T02:21:18.6710000Z	21.1	38.1	0.958281696
5	2020-11-23T02:23:06.1460000Z	21	38.3	0.958153844
6	2020-11-23T02:24:24.7960000Z	21	38.7	0.958153784
7	2020-11-23T02:26:00.6390000Z	21	38.3	0.958153844
8	2020-11-23T02:30:47.6190000Z	20.9	38.3	0.958025634
9	2020-11-23T02:31:27.5390000Z	20.9	38.3	0.958025634
10	2020-11-23T02:34:20.4110000Z	20.8	38.3	0.957897007
11	2020-11-23T02:34:57.1210000Z	20.9	38.3	0.958025634
12	2020-11-23T02:35:33.7090000Z	20.8	38.2	0.957897067
13	2020-11-23T02:36:23.4830000Z	20.8	38.3	0.957897007
14	2020-11-23T02:37:34.6200000Z	20.8	38.3	0.957897007
15	2020-11-23T02:38:14.5500000Z	20.8	38.2	0.957897067
16	2020-11-23T02:54:57.0470000Z	7.7	27.1	0.937400699
17	2020-11-23T03:07:11.5450000Z	20.5	40.7	0.957508802

Figuras 107. Resultado final del modelo de predicción climática.
Fuente: Elaboración Propia

CAPÍTULO V. Conclusiones y recomendaciones

Conclusiones:

Con respecto al objetivo general se concluye que la integración de Machine Learning e IoT es una buena opción para predecir pronósticos del clima, ya que al crear una mini estación meteorológica PWS nos permite realizar captura de datos en tiempo real de diferentes variables y esta es contrastada con los datos predichos y estos nos contribuye en la toma de decisiones para el sector agrario de la región Puno.

Con respecto al primer objetivo específico, se concluye que al implementar el prototipo funcional mini estación meteorológica basada en la tecnología Internet de las Cosas, el módulo ESP-32 necesariamente tiene que conectarse a internet a través de un módulo wifi que tiene integrada, para el envío de la capturar los datos climáticos en tiempo real a ThingSpeak, también es de vital importancia proteger los elementos del PWS de las inclemencias del tiempo dentro de una pequeña maqueta cubierta con lamina acrílica, ya que los sensores no pueden ser expuestas al aire libre porque pueden dañarse. Al tener una batería y un panel solar, la placa ESP-32 y los sensores no necesita que estén conectados a una fuente de alimentación lo que ofrece la autonomía energética para poder aislado en campo.

Con respecto al segundo objetivo específico, se concluye que se logro configurar y conectar correctamente los dispositivos de toma de datos (PWS) con la Plataforma de Microsoft Azure, con el recurso de IoT Hub, esta plataforma es una herramienta que habilita una comunicación bidireccional, muy segura y confiable entre la aplicación de Internet de las Cosas (IoT) y los dispositivos que administra, la información recopilada por las cosas se puede analizarla y convertirla en conocimiento útil por el personal o mediante inteligencia artificial.

Con respecto al tercer objetivo específico sobre el análisis de los datos del senamhi, se concluye que es importante el tratamiento de los datos, ya que, si no es así, la predicción sería errónea y por ende la toma de decisiones no es acertada, además de ello se tiene que adaptar los datos de acuerdo a las necesidades del modelo.

Con respecto al cuarto objetivo específico, se llega a la conclusión de que el algoritmo con un alto porcentaje de precisión fue el algoritmo de regresión logística, entonces se concluye que los pronósticos están con un nivel alto de confiabilidad.

Con respecto al quinto objetivo específico, se logró crear un modelo predictivo en la nube, se concluye que es importante que el modelo implementado sea desplegado como un “Web service”, permitiendo acceder al modelo como un recurso más de la web.

Recomendaciones

Se recomienda utilizar Azure IoT Edge en vez de Azure IoT Hub, ya que, al llevar establecidas cargas de trabajo hacia el perímetro de la red, los dispositivos pierden menos tiempo en la comunicación con la nube y reaccionan con rapidez a los cambios locales y funcionan de manera confiable, y se encuentra más cerca del dispositivo.

Se recomienda que, para realizar el modelo predictivo, los datos deben ser más uniforme, ya que nos encontramos con un problema que los datos con información histórica de algunas estaciones (Apis gratuitas) a razón de dos llamadas por hora, en forma diaria y los del SENAMHI por hora, en ese caso es necesario uniformizar los datos.

Se recomienda registrar la mini estación meteorológica en Wundergroup, OpenWeatherMap para una ubicación específica, para así consumir los datos mediante Apis.

Se recomienda, disponer de más fuentes de datos tales como Apis de OpenWeatherMap, Weather Company y otros, también poner a disposición la información relativa a otros motivos o causas fenológicas de los cultivos de la región de Puno.

Se recomienda usar otras plataformas de cloud como Amazon Machine Learning, para crear y entrenar modelos predictivos y alojar aplicaciones en una solución escalable en la nube.

Se recomienda utilizar repetidores wifi para ampliar la cobertura de la red para el envío de datos captados de las mini estaciones meteorológicas, teniendo en cuenta la tecnología wifi a usar de acuerdo a la distancia deseada.

Se recomienda seguir con la investigación para mejorar el destino final de la predicción, realizando una aplicación móvil.

REFERENCIAS

- Aguilar, S. (2020). DISEÑO DE UNA SOLUCIÓN BASADA EN EL INTERNET DE LAS COSAS (IoT) EMPLEANDO LORAWAN PARA EL MONITOREO DE CULTIVOS AGRÍCOLAS EN PERÚ. Lima.
- Amat, J. (agosto de 2016). *Ciencia de datos*. Obtenido de https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple
- Andrade, L. (29 de mayo de 2019). Implementación de una plataforma colaborativa del Internet de las cosas para la captura de variables ambientales para el Municipio Santiago de Cali. Santiago de Cali.
- Arduino. (febrero de 2018). *What is Arduino?* Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- Báez, E. D., & Nicolalde, G. E. (30 de agosto de 2018). Construcción de un prototipo inalámbrico de monitoreo meteorológico conformado por plataformas de hardware y software libre, asociado al análisis de sus datos para la predicción de temperatura mediante un método basado en Machine Learning. Quito. Obtenido de <http://bibdigital.epn.edu.ec/handle/15000/19707>
- Carrasco, F. (2016). Efectos del cambio climático en la producción y rendimiento de la quinua en el distrito de Juli, periodo 1997 - 2014. Puno, Perú.
- Cavassa, F. (junio de 2019). *CIOPERU*. Obtenido de <https://cioperu.pe/articulo/26149/6-plataformas-de-iot-de-escala-empresarial/?p=10>
- EGASA. (2012). Proyecto: Modelado Distribuido de un Sistema Inteligente de Gestión del Recurso Hídrico, caso: Cuenca del Río Chili. Arequipa.

- Escribano, J. (2017). Implementación de una estación meteorológica con arduino.
- Ferrer, R. D. (agosto de 2017). Desarrollo de una estación meteorológica autónoma de bajo costo. Bucaramanga.
- Gaviria, R. A. (2018). PROTOTIPO FUNCIONAL ESTACIÓN METEOROLÓGICA PORTABLE CON DISPOSITIVOS DE BAJO COSTO (ARDUINO). Pereira.
- Gnoza, N., & Barberena, M. E. (2018). Estudio de factibilidad del uso de Machine Learning con múltiples fuentes de datos en el pronóstico del tiempo. Uruguay. Obtenido de <http://hdl.handle.net/20.500.11968/3761>
- IAAR; López, Raúl;. (s.f.). *Introducción al Machine Learning*. Obtenido de <https://iaarbook.github.io/ML/>
- Idexcam, I. d. (7 de marzo de 2017). *Gestión*. Obtenido de <https://gestion.pe/economia/cambio-climatico-afecta-cultivos-seis-agroexportaciones-peru-son-130193-noticia/?ref=gesr>
- Iribarren, D. (Julio de 2016). MODELO PREDICTIVO. MACHINE LEARNING APLICADO AL ANALISIS DE DATOS CLIMATICOS CAPTURADOS POR UNA PLACA SPARKFUN. Madrid.
- ITU-T. (2019). Global Standards for the Internet of Things. Recuperado el junio de 2019, de <https://www.itu.int/en/ITU-T/techwatch/Pages/internetofthings.aspx>
- Javier, A. (24 de julio de 2019). *Hiberus*. Obtenido de <https://www.hiberus.com/crecemos-contigo/azure-la-plataforma-cloud-de-microsoft/>
- Lighezzolo, R. A. (22 de Abril de 2014). Integración de modelos numéricos de predicción meteorológica en sistemas de alerta temprana a emergencias. (Tesis de maestría).

- Mamani, A. V. (2011). Soluções aproximadas para algoritmos escaláveis de mineração de dados em domínios de dados complexos usando GPGPU. São Carlos. doi:10.11606/D.55.2011.tde-22112011-132339
- Martinez, R., Rivadeneira, A., & Nieto, J. J. (2011). Guía de buenas practicas para la predicción estacional en latinoamérica. (ISBN-978-9978-9934-3-9), 1-55. CIIFEN.
- Minagri. (2010). La fenología como herramienta en la agroclimatica. *Componente Agroclimatica - Boletín Técnico N° 02 - 2010*, 3. (D. d. Agraria, Ed.) Perú.
- Mora, A. (2017). Diseño e Implementación de una Aplicación IoT en la Nube de Azure para el Análisis de Imagen . Madrid.
- Naylamp Mechatronics. (s.f.). *Controladores*. Obtenido de <https://naylampmechatronics.com/expressif-esp/384-nodemcu-32-esp32-wifi.html>
- Perez, A., Matadamas, I., Morales, M., & Altamirano, M. (2017). Prototipo basado en redes neuronales para monitoreo y prediccion de temperaturas en invernaderos de los valles centrales de Oaxaca. Oaxaca.
- Pérez, N. A., Fierro, D., & Ramírez, S. (julio de 2011). Pronóstico de riesgo agroclimático para el cultivo de papa en la cuenca de río Urubamba. Cusco, Perú.
- Programmerclick. (s.f.). Obtenido de <https://programmerclick.com/article/78201096843/>
- Rodríguez, C. (s.f.). Modelos no lineales de pronóstico de series temporales basados en inteligencia computacional para soporte en la toma de decisiones agrícolas. Cordoba, Argentina. Obtenido de <http://hdl.handle.net/11086/4604>
- Rosas, J. L., & Mora, H. V. (2019). Diseño, desarrollo e implementación de una red de sensores inalámbricos (WSN) para el control, monitoreo y toma de decisiones

aplicado en la agricultura de precisión basado en internet de las cosas (IOT). – Caso de estudio cultivo de frijol. Perú.

Sandorobotics. (2019). *Módulo NodeMCU V3 ESP8266 – CH340*. Obtenido de <https://sandorobotics.com/producto/hr0126/>

Simeone, O. (2018). A Very Brief Introduction to Machine Learning with Applications to Communication Systems. 1-20. (P. Recuero, Trad.) Obtenido de <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/>

Sparkfun. (s.f.). *SparkFun Weather Shield*. Obtenido de <https://www.sparkfun.com/products/13956>

Tobajas, A. (2016). Diseño e Implementación de una Estación Meteorológica utilizando Micro. Catalunya, Perú.

Ungerovivh, M., & Barreiro, M. (2017). Predicción climática estacional de precipitación acumulada en primavera y verano en el sur de Uruguay. doi:<https://doi.org/10.1590/0102-77863230004>

ANEXOS

Materiales y equipos

Material/equipo Hardware	Cantidad	Costo (Soles)
ESP-32	1	34
Sensor DHT22	1	23
Sensor BMP180	1	2.80
TP4056	1	5
Bateria litio	1	17
Placa solar	1	12
Protoboard	2	14
Cables Jumper	2	12.30
Plancha acrilica	1	20

Fuente: Elaboración propia