



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: “ANÁLISIS PREDICTIVO SOBRE FLUJO DE DATOS
OBTENIDOS DE UNA WIRELESS SENSOR NETWORK. CASO DE
ESTUDIO.”**

AUTOR: CARRERA MENA, RENÉ ALEXANDER

GARCÍA PACHECO, EDGAR ESTEBAN

DIRECTOR: Ing. DÍAZ ZÚÑIGA, MAGI PAÚL Msc, MBA

SANGOLQUÍ

2019

CERTIFICADO



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "ANÁLISIS PREDICTIVO SOBRE FLUJO DE DATOS OBTENIDOS DE UNA WIRELESS SENSOR NETWORK. CASO DE ESTUDIO." fue realizado por los señores **René Alexander Carrera Mena** y **Edgar Esteban García Pacheco**, el mismo que ha sido revisado en su totalidad y analizado por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 13 de febrero del 2019

Firma:



Ing. Paúl Díaz Msc, MBA

1707249072



AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, **René Alexander Carrera Mena** y **Edgar Esteban García Pacheco**, declaramos que el contenido, ideas y criterios del trabajo de titulación "**ANÁLISIS PREDICTIVO SOBRE FLUJO DE DATOS OBTENIDOS DE UNA WIRELESS SENSOR NETWORK. CASO DE ESTUDIO.**" es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es verás.

Sangolquí, 13 de febrero del 2019

Firma.

René Alexander Carrera Mena

C.C.: 1721790101

Edgar Esteban García Pacheco

C.C.: 1720254810

AUTORIZACIÓN



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotros, **René Alexander Carrera Mena** y **Edgar Esteban García Pacheco**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “ANÁLISIS PREDICTIVO SOBRE FLUJO DE DATOS OBTENIDOS DE UNA WIRELESS SENSOR NETWORK. CASO DE ESTUDIO.” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Sangolquí, 13 de febrero del 2019

Firmas.

René Alexander Carrera Mena

C.C.: 1721790101

Edgar Esteban García Pacheco

C.C.: 1720254810

DEDICATORIA

El presente trabajo lo dedico a mi familia, especialmente a mi querido padre René y a mi hermosa hija Paulette, a quienes he tenido que restarles valioso tiempo para culminar con tan grandiosa carrera y tan anhelado sueño, pero es un logro que les llenará de orgullo y felicidad, teniendo en cuenta que ellos son los motores de mi vida para alcanzar y seguir alcanzando mis metas día a día.

A la Ingeniera Wendy Ayala, persona muy importante a lo largo de mi vida y de todo este proceso de aprendizaje, quien siempre me brindó su apoyo incondicional, tiempo y conocimientos para siempre alcanzar mis metas y destacar en todo momento, y a quien debo el gusto por el campo de la informática, la programación, la investigación y el ser mejor cada día.

Tnte. René Carrera

Esta tesis está dedicada a mis padres Edgar y María Soledad quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más, gracias por inculcar en mí, el ejemplo de esfuerzo y valentía, de no temer las adversidades porque Dios está conmigo siempre.

A mi esposa Andrea por su amor y apoyo incondicional, durante todo este proceso, por estar conmigo en todo momento y brindarme cada día palabras de aliento.

Tnte. Esteban García

AGRADECIMIENTO

Agradezco a mi amada familia, que han sido mi apoyo y mi fortaleza en toda circunstancia y dificultades que se han presentado a lo largo de este proceso académico, y que han sabido guiarme en todo momento para salir adelante.

Al Ingeniero Paúl Díaz, tutor de tesis, docente y amigo que siempre denotó su capacidad para transmitir sus conocimientos exigiéndome ser el mejor y quien depositó su confianza en mis capacidades en incontables ocasiones permitiéndome representar a esta universidad internacionalmente.

A la Ingeniera Elizabeth Urbano, docente y amiga que nos abrió sus puertas y nos compartió sus ideas innovadoras, que permitieron que este trabajo vaya más allá de la parte académica y se convierta en toda una experiencia de aprendizaje.

Tnte. René Carrera

Quiero expresar mi gratitud a Dios, quien con su bendición llena siempre mi vida y a toda mi familia por estar siempre presentes.

A mis profesores en especial a la Ing. Elizabeth Urbano, Ing. Paúl Días e Ing. Mauricio Campaña; quienes, con la enseñanza de sus valiosos conocimientos, hicieron que pueda crecer día a día como profesional, gracias a cada una de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

Finalmente quiero expresar mi más grande y sincero agradecimiento a mi compañero de tesis René Carrera por su principal apoyo, amistad y compañerismo durante todo este proceso; quien, con sus conocimientos y empeño, permitió el desarrollo de este trabajo.

Tnte. Esteban García

ÍNDICE DE CONTENIDOS

CARÁTULA

CERTIFICADOi

AUTORÍA DE RESPONSABILIDADii

AUTORIZACIÓN iii

DEDICATORIAiv

AGRADECIMIENTO v

ÍNDICE DE CONTENIDOSvi

ÍNDICE DE TABLASxi

RESUMENxiv

ABSTRACTxv

INTRODUCCIÓN..... 1

a) Antecedentes..... 1

b) Planteamiento del problema 2

c) Justificación..... 3

d) Objetivos..... 4

i) General..... 4

ii) Específicos 4

e) Alcance..... 4

f)	Factibilidad.....	5
i)	Factibilidad Técnica	5
ii)	Factibilidad Económica	6
iii)	Factibilidad Operativa	7

CAPÍTULO I

MARCO TEÓRICO

1.1	INTERNET DE LA COSAS	1
1.2	WIRELESS SENSOR NETWORK (WSN).....	2
1.3	FLUJOS DE DATOS Y TÉCNICAS DE DATA STREAM MINING	3
1.3.1	Estudios realizados	5
1.4	SISTEMAS DE MONITORIZACIÓN	7
1.4.1	Sistemas basados en IoT, Wsn Y Data Stream Mining.....	8
1.4.2	Estudios realizados	9
1.5	APACHE SPARK STREAMING	9
1.6	LENGUAJE DE PROGRAMACIÓN SCALA	10
1.7	METODOLOGÍA TRADICIONAL: MODELO EN “V”	11

CAPÍTULO II

DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DEL SISTEMA

2.1	CONCEPTO DE OPERACIÓN DEL SISTEMA	14
-----	---	----

2.1.1	Perspectiva del Sistema.....	14
2.1.2	Funcionalidad del Sistema	15
2.1.3	Características de los usuarios.....	16
2.1.4	Restricciones del sistema	16
2.1.5	Suposiciones y dependencias	17
2.2	REQUERIMIENTOS Y ARQUITECTURA DEL SISTEMA	17
2.2.1	Requisitos funcionales.....	17
2.2.2	Requisitos no funcionales.....	19
2.2.3	Arquitectura del sistema.....	20
2.3	DISEÑO DETALLADO	22
2.3.1	Diseño de hardware.....	22
2.3.2	Diseño de software	23
2.4	IMPLEMENTACIÓN	24
2.4.1	Diseño de la red de sensores (WSN).....	24
2.4.2	Diseño y desarrollo de la base de datos.....	25
2.4.3	Diseño y desarrollo del software del nodo de sensores ARD-I (Arduino).....	27
2.4.4	Diseño y desarrollo del software del servidor RAS-I (Raspberry/Python).....	31
2.4.5	Diseño y desarrollo del software de análisis SPK-I (Apache Spark/ Scala).....	35
2.5	INTEGRACIÓN, PRUEBA Y VERIFICACIÓN DE COMPONENTES	39

2.5.1	Integración.....	39
2.5.2	Pruebas	42

CAPÍTULO III

CASO DE ESTUDIO:

“OBTENCIÓN, PROCESAMIENTO Y DATA STREAM MINING”

3.1	DESCRIPCIÓN GENERAL DEL CASO DE ESTUDIO	45
3.1.1	Resumen de la situación.....	45
3.2	CARACTERÍSTICAS DEL ÁREA DE IMPLEMENTACIÓN.....	45
3.2.1	Ubicación	45
3.2.2	Descripción del lugar.....	46
3.3	DISTRIBUCIÓN Y ORGANIZACIÓN DE LOS DISPOSITIVOS.....	47
3.4	INICIO DE OPERACIÓN DEL SISTEMA SIS-I	48
3.4.1	Activación del sistema para la captura de datos.....	48
3.4.2	Procesamiento del flujo de datos.....	50
3.4.3	Data Stream Mining	52
3.5	VISUALIZACIÓN DEL STREAMING DE DATOS	58
3.6	VALIDACIÓN Y VERIFICACIÓN DEL SISTEMA SIS-I.....	62
3.7	MANTENIMIENTO DEL SISTEMA SIS-I.....	63

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1	CONCLUSIONES	65
4.2	RECOMENDACIONES	66

ÍNDICE DE TABLAS

Tabla 1 <i>Especificaciones Técnicas de los dispositivos</i>	5
Tabla 2 <i>Presupuesto del sistema SIS-I</i>	7
Tabla 3 <i>Requerimientos para realizar Data Stream Mining</i>	4
Tabla 4 <i>Componentes del sistema</i>	14
Tabla 5 <i>Características de los usuarios</i>	16
Tabla 6 <i>Atributos para la base de datos</i>	25
Tabla 7 <i>Atributos de sistema ARD-I</i>	28
Tabla 8 <i>Métodos empleados para el ARD-I</i>	29
Tabla 9 <i>Atributos del sistema RAS-I</i>	33
Tabla 10 <i>Atributos del sistema SPK-I</i>	37
Tabla 11 <i>Consideraciones especiales</i>	44
Tabla 12 <i>Descripción del lugar del caso de estudio</i>	45
Tabla 13 <i>Información de captura de datos</i>	50
Tabla 14 <i>Descripción de datos generados y obtenidos</i>	54
Tabla 15 <i>Operaciones importantes en Spark Streaming</i>	54
Tabla 16 <i>Revisión de la documentación</i>	62
Tabla 17 <i>Requisitos del paradigma de Data Stream Mining</i>	63
Tabla 18 <i>Puntos clave para el mantenimiento del SIS-I</i>	64

ÍNDICE DE FIGURAS

<i>Figura 1</i> Arquitectura de Spark Core	10
<i>Figura 2</i> Metodología de desarrollo “Modelo en V”	11
<i>Figura 3</i> Diagrama de caso de uso del SIS-I	15
<i>Figura 4</i> Arquitectura del sistema	21
<i>Figura 5</i> Componentes hardware-software	21
<i>Figura 6</i> Componente CTE-1	22
<i>Figure 7</i> Componente CTE-2	23
<i>Figura 8</i> Diseño de la Wireless Sensor Network.....	24
<i>Figura 9</i> Datos Sensores.....	26
<i>Figura 10</i> Script de creación de la tabla datos_sensores	26
<i>Figura 11</i> Diagrama de flujo del ARD-I	28
<i>Figura 12</i> Codificación del programa ARD-I	31
<i>Figure 13</i> Procesos y subprocesos del sistema RAS-I	32
<i>Figura 14</i> Codificación del programa ARD-I	35
<i>Figura 15</i> Procesos y subprocesos del sistema SPK-I:.....	36
<i>Figura 16</i> Codificación del programa SPK-I.....	39
<i>Figura 17</i> Flujograma Integración componentes.....	40
<i>Figura 18</i> Integración de Componentes	41
<i>Figure 19</i> Ubicación IASA.....	46
<i>Figura 20</i> Invernadero	47
<i>Figura 21</i> Ubicación equipo	47

Figura 22 Ubicación equipo invernadero	48
Figura 23 Toma de datos.....	49
Figura 24 Cadena de caracteres generada en el flujo	50
Figura 25 Datos almacenados en CTE-2.....	51
Figura 26 Datos enviados mediante socket	51
Figura 27 Flujo Generado	52
Figura 28 Spark Streaming.....	53
Figura 29 Implementación del RDD.....	53
Figura 30 Ingreso del flujo de datos.....	56
Figura 31 Definición de la técnica de análisis.....	56
Figura 32 Flujo de entrenamiento	56
Figura 33 Flujo de testing aplicado el algoritmo	57
Figura 34 Inicio del proceso de streaming en Spark- Sockets sin datos	58
Figura 35 Recepción de Sockets y procesamiento de datos.....	59
Figura 36 Procesamiento en tiempo real de datos.....	60
Figura 37 Finalización de procesamiento.....	61

RESUMEN

El uso de IoT (Internet de las cosas), está logrando resultados muy alentadores en relación a la determinación de costos, predicciones, situaciones que disminuyen la productividad y la rentabilidad de los cultivos. En el Ecuador, los floricultores se ven limitados en el uso de medios tecnológicos, en muchas ocasiones, por el desconocimiento de herramientas automatizadas. La cosecha de la rosa cortada se correlaciona con variables como: temperatura, humedad, luminosidad, radiación, calidad de aire, entre otros factores climáticos relacionados con la fisiología del crecimiento de las plantas. El uso de sensores especializados e implementados en campo que monitoreen variables climáticas permitirá que los floricultores se anticipen a posibles problemas mediante la predicción de cosechas, presencia de plagas, épocas con falta de luz, reducir costo de implementación y comprender la fisiología de la planta para una efectiva gestión del invernadero de rosas.

PALABRAS CLAVE:

- **INTERNET DE LA COSAS**
- **RED DE SENSORES INALÁMBRICOS**
- **MINERÍA DE FLUJO DE DATOS**

ABSTRACT

The use of IoT (Internet of things), is achieving very encouraging results regarding costing, predictions, situations that reduce crops' productivity and profitability. In Ecuador, growers are limited about the use of technological means, often by the lack of automated tools. Harvest cut rose correlates with variables such as temperature, humidity, light, radiation, air quality, among other climatic factors related to the physiology of plant growth. The use of specialized sensors and its implementation on field that monitor climate variables allow growers, to anticipate possible problems by predicting crops, presence of pests, seasons with lack of light, reducing cost implementation and understanding the physiology of the rose and the effective management of greenhouse of roses.

KEY WORDS:

- **INTERNET OF THINGS**
- **WIRELESS SENSOR NETWORK**
- **DATA STREAM MINING**

INTRODUCCIÓN

a) Antecedentes

Los datos, son el pilar fundamental para la optimización de procesos y para la toma de decisiones debido a que estos tienen una capacidad especial para describir el mundo. Hoy en día todo es un dato y todo puede ser considerado como un dato, lo cual, ha hecho que, en la actualidad, esta temática sea de gran importancia a nivel mundial, y sea adoptada en varios ámbitos alrededor del mundo.

Al hablar de datos, es hablar de un tweet en Twitter, un like en Facebook, una dirección en GPS, un click o un touch, una medida de temperatura, etc. Es decir, todo es un dato y, por consiguiente, todo puede ser analizado, permitiendo que un sistema complejo e interconectado como la sociedad o la producción, puedan ser descritas por medio de estos datos.

Empresas como Facebook, Amazon, Netflix dedicadas a la comunicación, ventas y entretenimiento respectivamente, usan los datos de sus clientes para dar recomendaciones, adelantarse a necesidades e inclusive para mejorar su productividad. Otras empresas relacionadas al área agrícola como HelixBios, aplican el análisis de datos para poder predecir el rendimiento de la producción de los cultivos con una mayor precisión.

Esto nace gracias al avance tecnológico, a la facilidad de acceso a la información, facilidad de acceso a herramientas informáticas y al “crecimiento exponencial de la capacidad de procesamiento que actualmente poseen los dispositivos electrónicos y computadores”. (Mohamed, 2012).

Sin embargo, estas tecnologías no están plenamente desarrolladas en el Ecuador, o a su vez, las pocas instituciones o empresas que lo usan representan la minoría, debido a los costos que estos representan.

b) Planteamiento del problema

Esta investigación esta aplicada en un entorno controlado en donde existen fluctuaciones en temperatura, humedad, luminosidad, radiación uv y calidad de aire, de la misma forma, en el lugar se realiza la producción de flores, la cual está relacionada directamente con los factores abióticos presentes en el lugar ya que, estos inciden en su fisiología, crecimiento y en la producción.

Debido a esto, esta área no dispone de un sistema que permita controlar y monitorear las variaciones de los factores abióticos, ni tampoco, se pueda estimar la producción que se obtendrá al finalizar su ciclo de crecimiento hasta su cosecha.

Actualmente, el proceso de monitoreo y toma de datos ambientales se la realiza de forma manual o con el uso de dispositivos electrónicos que necesariamente requieren de la presencia de una persona para su manipulación y que, además, está limitada a tan solo tres capturas de datos en el día durante todo el ciclo de producción, y a la vez que, esto se ha venido realizando desde algunos años atrás.

El cambio o variación de los factores abióticos, inciden directamente en la producción de flores, ya que, estas están sujetas a márgenes y rangos de tolerancia para su optimo crecimiento y que un monitoreo inadecuado puede producir la disminución de la producción o la pérdida total de la misma.

La escasez de herramientas electrónicas y principalmente de sistemas informáticos personalizados o específicos que puedan predecir y estimar los niveles de producción, así como

también el monitoreo en tiempo real de los factores abióticos del lugar, han generado que la producción no se maneje en base a parámetros óptimos y que tampoco se pueda estimar adecuadamente y técnicamente la futura producción.

c) Justificación

El proceso de monitoreo y toma de datos referentes a factores abióticos como la temperatura, humedad, luminosidad, rayos uv y calidad del aire, está orientada a productores medianos, debido a que este proceso se lo realiza de forma manual conjuntamente con el uso de dispositivos electrónicos, limitándose a capturas de datos solo en horas específicas y teniendo pérdida de datos en los intervalos de captura de los mismos.

Así mismo, no se dispone de un sistema que analice aquellos datos y permitan predecir ciertos eventos futuros sobre la producción y que, además, sean los mostrados oportunamente para una mejor toma de decisiones.

Por lo tanto, es de suma importancia disponer de un sistema que permita la recolección de estos datos en forma automática y continua, de tal manera, que se pueda disponer de un información veraz y oportuna para la toma de decisiones inmediata. Igualmente, el uso de aplicaciones enfocadas en el análisis de flujos de datos, es una temática que se están adoptando en varios ámbitos y brindan un aporte valioso a la producción.

Es importante señalar, que esta investigación permitirá disponer de un sistema que conjuga un prototipo de hardware, software y comunicaciones encargado de gestionar, analizar, almacenar y disponer de resultados orientados a optimizar la producción y tomar mejores decisiones.

Finalmente, esta investigación brinda un gran aporte al sector florícola, principalmente a pequeños productores, debido a que este sistema tiene la cualidad de ser de bajo costo, alta flexibilidad, alta usabilidad, disponibilidad entre otras.

d) Objetivos

i) General

Aplicar técnicas de análisis predictivo sobre un flujo de datos provenientes de una Wireless Sensor Network (Red Inalámbrica de Sensores), que permitan mejorar la toma de decisiones aplicadas a un caso de estudio

ii) Específicos

- Aplicar técnicas de data stream mining mediante el empleo de software especializado para realizar las predicciones.
- Generar la visualización de los datos y de los resultados obtenidos mediante gráficas interactivas que permitan describir el análisis realizado y mejorar la toma de decisiones.
- Validar el sistema predictivo, mediante la implementación del prototipo y las técnicas de data stream mining en un caso de estudio.

e) Alcance

- Diseñar un prototipo funcional de hardware y software basado en una Wireless sensor network que permita la gestión del dispositivo, el proceso de obtención, transmisión, análisis y almacenamiento de datos.
- Establecer una arquitectura básica para el análisis del flujo de datos que permitirá el empleo de técnicas de data stream mining y el desarrollo de un modelo predictivo en el caso de estudio.
- Realizar la visualización y la descripción de los resultados obtenidos en el caso de estudio.

f) Factibilidad

i) Factibilidad Técnica

Para la realización de este proyecto, se dispone de dos computadores para la comunicación con los dispositivos que proveerán el flujo de datos y a su vez otros periféricos encargados de la captura, transmisión y recepción de datos detallados en capítulos futuros. Ambos computadores soportan las herramientas especificadas anteriormente, además, se dispone de un lugar controlado (invernadero) para la respectiva captura de datos en donde se colocarán los dispositivos de medición.

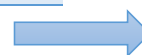
Las especificaciones de los dispositivos son las siguientes:

Tabla 1

Especificaciones Técnicas de los dispositivos

Ítem	Dispositivo	Características
1	Computador Portátil	<ul style="list-style-type: none"> - Marca: HP Envy 15 X360 - Procesador: Intel i7 - Ram: 12Gb - Disco Duro: 1Tb - Tarjeta de Video: 4Gb Intel Graphics - Velocidad: 2.21Ghz – 3.5Ghz - Sistema Operativo: Windows 10

CONTINÚA



2	Computador Compacto	<ul style="list-style-type: none">- Marca: Raspberry Pi- Procesador: Chipset Broadcom BCM2387- Ram: 1Gb- Micro SD: Samsung 32Gb UHS Clase 10- Gpu: Dual Core Video Core IV- Velocidad: 1.2Ghz- Sistema Operativo: Raspbian
3	Arduino Mega	<ul style="list-style-type: none">- Microcontrolador: ATmega328- Voltaje: 5V- Digital I/O Pins: 14 (de los cuales 6 son salida PWM)- Entradas Analógicas: 6- Clock Speed: 16 MHz
4	Conjunto de Sensores	<ul style="list-style-type: none">- Sensor de Temperatura y Humedad (DHT22)- Sensor de Luminosidad (MH Series Flying Fish)- Sensor de Índice Ultravioleta- Sensor de Calidad de Aire (MQ135)

ii) Factibilidad Económica

El desarrollo del proyecto contempla la adquisición de varios dispositivos electrónicos, capacitación y otros insumos necesarios y que se encuentran descritos a continuación:

Tabla 2
Presupuesto del sistema SIS-I

Ítem	Descripción	Cantidad	Valor	Observación
1	Computadora No.1	1	\$1,100.00	Propiedad del Investigador
2	Raspberry Pi 3	1	\$85.00	Propiedad del Investigador
3	Arduino Uno	1	\$13.00	Propiedad del Investigador
4	Arduino Mega	1	\$17.00	Propiedad del Investigador
5	Sensor de Humedad y Temperatura DHT22	1	\$7.50	Propiedad del Investigador
6	Sensor de luminosidad MH Flying Fish	1	\$7.50	Propiedad del Investigador
7	Sensor Ultravioleta	1	\$7.50	Propiedad del Investigador
8	Sensor de Calidad de Aire MQ135	1	\$7.50	Propiedad del Investigador
9	Antenas de transmisión de datos NRF24L01	2	\$37.00	Propiedad del Investigador
10	Impresora Epson L355	1	\$350.00	Propiedad del Investigador
11	Internet	-	\$477.60	Propiedad del Investigador
12	Servicios Básicos	-	\$240.00	Propiedad del Investigador
13	Investigación y desarrollo	-	\$1,340.00	Propiedad del Investigador
14	Cursos de análisis de datos	3	\$124.00	Propiedad del Investigador
15	Implementos de Oficina	-	\$50.00	Propiedad del Investigador
TOTAL			\$3,841.10	

iii) Factibilidad Operativa

Para este proyecto se ha considerado las siguientes competencias requeridas para su adecuado desarrollo:

- Conocimiento y capacidad de aplicación de nuevas herramientas y tecnologías orientadas para el análisis de datos.

- Conocimiento de varios lenguajes de programación principalmente aquellos orientados al análisis de datos como Scala, Python o R.
- Conocimiento y capacidad de aplicación de dispositivos electrónicos como arduino, raspberry pi, varios tipos de sensores y a su vez, su respectiva programación, integración e implementación.

Estas competencias y destrezas han sido adquiridas en diferentes actividades académicas y mediante cursos de capacitación, lo que permite que el desarrollo de esta propuesta sea operativamente factible.

CAPÍTULO I

MARCO TEÓRICO

1.1 INTERNET DE LA COSAS

El termino Internet de las Cosas (IoT), es un concepto que parte fundamentalmente del término Computación Ubicua, la cual se refiere a dotar de habilidades que antes solo tenía una computadora a cualquier dispositivo que nos rodea, y cuyo objetivo radica en el “aprovechamiento de la información ofrecida por dispositivos de computación distribuidos en el entorno y que a su vez, son capaces de adaptarse de forma transparente en función del contexto, del tipo de usuario y del dispositivo utilizado”. (Cano, Calafate, Malumbres, & Manzoni, 2003)

En este sentido, el Internet de la Cosas representa un gran avance en la capacidad para la recolección y captura de datos, procesamiento y análisis y a la difusión y distribución de los datos con la finalidad de obtener información valiosa y conocimiento dentro de ella. Además, en “el IoT se busca que los objetos se vuelvan de cierta manera más inteligentes” (Vazhnov, 2016), de tal forma que puedan percibir y sentir su entorno, y tengan la capacidad de interacción y comunicación entre otros dispositivos, sistemas en internet y también con otros usuarios.

En la actualidad “el IoT representa la próxima evolución de Internet” (Evans & Grupo de Soluciones Empresariales para Internet, 2011) lo que permitirá generar beneficios a varios sectores y facilitar el entendimiento y control de todo lo que nos rodea. Entonces, es necesario concebir la idea que todo es un sensor desde el punto de vista de la obtención de datos; un Smartphone, una computadora, una Tablet, una mascota o inclusive una persona; esto quiere decir, que permitirá recolectar todo tipo de datos, transmitirlos, analizarlos y visualizarlos de tal forma que brindaran la capacidad de mantener un control permanente de fenómenos, actividades o anomalías en

cualquier campo ya sea este industrial o de la vida cotidiana, así como también generar predicciones para actuar oportunamente y mejorar la toma de decisiones.

Por lo tanto, es importante recordar que: "Con un billón de sensores integrados en el entorno, todos conectados por sistemas informáticos, software y servicios, se podrá escuchar el pulso de la Tierra, lo que tendrá un efecto tan profundo en la interacción humana con el planeta como profunda ha sido la revolución que ha significado Internet para las comunicaciones". Peter Hartwell (Jefe Investigador de laboratorios HP).

El uso de sensores interconectados, al igual que nuestros sentidos, ayudan a entender y disfrutar la vida, pero también nos informaran todo acerca de cómo interactuamos con el mundo.

1.2 WIRELESS SENSOR NETWORK (WSN)

Las Redes de sensores inalámbricos han sido objeto de interés en varias disciplinas ya que permite disponer de información en tiempo real del entorno donde estén siendo empleadas. "Las WSN consisten en pequeños nodos de sensores, que actúan como generadores de datos y relés de red" (Akyildiz & Vuran, 2010), por tal motivo, las WSN han podido ser adoptadas en diferentes aplicaciones que poco a poco han ido convirtiéndose en parte de la vida cotidiana.

Una WSN se compone principalmente de sistemas individuales embebidos que son capaces de interactuar con el entorno a través de varios tipos de sensores, procesando la información localmente y comunicando esta información de manera inalámbrica.

Una WSN se basa en nodos, estos a su vez se conforman de tres componentes que se integran formando un solo sistema, los cuales son:

- Módulos Wireless son la parte fundamental de la red de sensores ya que esta posee la capacidad de comunicación.

- Tablero de Sensores conforma el circuito dotado de varios tipos de sensores que se acoplados al circuito y estos a su vez capturan los datos del entorno.
- Tablero de Programación provee las diferentes interfaces que permiten programar cada nodo para que ejecuten las diferentes tareas en donde sea aplicado.

El surgimiento de las WSN ha permitido acoplar y adaptar un sin número de sensores lo que ha dado como resultado que haya sido empleado en varios ámbitos como en la creación de aplicaciones militares, medioambientales, médicas, para uso doméstico, industriales y otras aplicaciones comerciales.

1.3 FLUJOS DE DATOS Y TÉCNICAS DE DATA STREAM MINING

El gran crecimiento de dispositivos electrónicos, la implementación de nuevas tecnológicas y el acceso a teléfonos inteligentes, han hecho que la sociedad hoy en día sea considerada como la sociedad de la información. Esta mención se la adoptó, gracias a la inmensa cantidad de datos que el mundo entero genera al interactuar con el mundo mediante el uso de dispositivos tecnológicos de forma directa o indirecta.

Debido a esta particularidad, “la extracción de conocimiento se está convirtiendo en una tarea importante para mucha gente” (Bifet, Holmes, Kirkby, & Pfahringer, 2011). Si bien cierto, años atrás, era impensable que la cantidad de información generada “en tan solo cinco años (periodo 2007-2011), llegase a supera en diez veces el universo de información hasta el 2007” (Bifet, Holmes, Kirkby, & Pfahringer, 2011) fecha en la cual también esta misma cantidad de información generada, llegó a sobrepasar por primera vez la capacidad y disponibilidad de almacenamiento.

Por estos obstáculos suscitados hace algunos años atrás, nace la idea del Green Computing o Computación Verde, que no es más que el uso eficiente de los recursos informáticos basados

principalmente en eficiencia algorítmica y que buscan reducir el impacto energético y medioambiental ante la gran demanda de procesamiento y almacenamiento de datos.

El enfoque de minería de flujo de datos nace de la necesidad de solventar el análisis y captación de datos que, debido a su gran tamaño, sobrepasan la capacidad de memoria. Los algoritmos desarrollados para el análisis en stream, están orientados a manejar esta gran cantidad de datos, considerando la existencia de un flujo de datos ilimitado con una alta velocidad que cambian rápidamente en el tiempo.

Para poder dar una solución a este paradigma, existen varios algoritmos tanto de predicción como de clasificación, que principalmente deben atender ciertos requerimientos descritos a continuación:

Tabla 3
Requerimientos para realizar Data Stream Mining

Ítem	Requerimiento	Descripción
1	Procesar un ejemplo a la vez e inspeccionarlo máximo una sola vez	Cada data deber ser aceptado como llegue en el orden que llegue; y éste puede ser inspeccionado o no, pero una vez procesado y se haya descartado no puede se lo puede recuperar.
2	Usar una cantidad de memoria limitada	La memoria es una restricción física que los algoritmos deben manejar eficientemente en donde pueda realizar los análisis estadísticos y almacenar el modelo de predicción a la vez.
3	Trabajar en una cantidad de tiempo limitada	Los algoritmos deben ser capaces de procesar la información en tiempo real ya que si su procesamiento no es lo suficientemente rápido se reflejaría en una inevitable perdida de datos.
4	Estar listo para predecir en cualquier punto	El algoritmo debe ser capaz de generar el mejor modelo y debe ser manipulado directamente desde la memoria en donde se realizará toda la estadística.

FUENTE: (Bifet, Holmes, Kirkby, & Pfahringer, 2011)

Para atender estos requerimientos, se han desarrollado varios algoritmos en lo que respecta a la minería de flujo de datos, uno de ellos es la Regresión Lineal con Gradiente de Descenso

Estocástico. Este algoritmo actualiza los coeficientes de regresión conforme ingresan los datos. Además, este algoritmo trabaja en memoria de cada una de las maquinas distribuidas, lo que se traduce en una reducción de tiempo de ejecución de la regresión lineal realizada.

1.3.1 Estudios realizados

Data Stream Mining: A practical approach.: (Bifet, Holmes, Kirkby, & Pfahringer, 2011)

Este estudio muestra detalladamente una herramienta que es comúnmente conocida en el análisis de flujos de datos denominada MOA (Massive Online Analysis). Esta herramienta permite “implementar algoritmos y realizar experimentaciones para un aprendizaje en línea desde flujos de datos cambiantes.” (Bifet, Holmes, Kirkby, & Pfahringer, 2011).

Además, establece ciertas pautas considerando un escenario típico de aprendizaje en el que se pueden aplicar ciertas estrategias de predicción.

Advances in data stream mining.: (Mohamed, 2012)

Este artículo menciona la velocidad con la que el desarrollo del hardware y software han contribuido a la generación de datos. Por lo que, “En muchas aplicaciones, es inevitable analizar este flujo de datos (data stream) en tiempo real.” (Mohamed, 2012).

Además, un concepto que es importante considerar es que “las técnicas de minería de datos (data mining) no han respondido a las necesidades de la extracción de flujos de datos” (Mohamed, 2012). Por lo que, analiza diferentes estrategias y técnicas como: Técnicas basadas en Hoeffding Bound las cuales han permitido desarrollar las técnicas de Very Fast Machine Learning (VFML) y han sido aplicadas a una extensión de técnicas tradicionales denominándolas Very Fast K-Means (VFKM) y Very Fast Decision Tree (VFDT) por la capacidad de análisis.

Asimismo, da a conocer las nuevas tendencias a las cuales se está orientando la minería de flujo de datos, y que aplica a nuestra investigación como es la minería de flujo de datos de sensores, la cual nace del “aumento de la potencia computacional de los nodos de sensores conectados de forma inalámbrica” (Mohamed, 2012) y que “han dado lugar a esta nueva dirección de investigación en el área de minería de flujo de datos” (Mohamed, 2012).

Handling Big Data Stream Analytics using SAMOA Framework - A Practical Experience.:
(Prasad & Agarwal, 2014)

(Prasad & Agarwal, 2014), hacen mención a dos posibilidades para mejorar el análisis de flujos de datos. El primer se detalla como el tipo de entorno de procesamiento que no requiere ser distribuido y cuyo proceso es realizado por una sola máquina, considerando que requiere mayor tiempo de procesamiento y respuesta, sin embargo, no es capaz del manejo de Big Data. Y el segundo tipo se refiere al procesamiento distribuido mediante el uso de clusters lo que permite paralelizar las tareas e incrementar la velocidad procesamiento.

Estos conceptos presentados permiten orientar nuestra investigación al uso de ciertas de herramientas específicas considerando las particularidades propias del proyecto.

Otro punto importante que se trata este artículo, es una vista objetiva y práctica de una herramienta de análisis de flujos de datos denominada Scalable Advanced Massive Online Analysis (SAMOA), la misma que presenta una arquitectura de alto nivel y se basa en el procesamiento distribuido.

Minería de datos sobre streams de redes sociales, una herramienta al servicio de la Bibliotecología.: (Jaramillo, Cardona, & Fernández, 2015)

Este artículo da a conocer el estado del arte con respecto a algoritmos de minería de datos sobre streams que se han originado en redes sociales. Además, presenta una revisión de las técnicas de minerías de datos presentes en dos marcos de trabajo más relevantes como son MOA y SAMOA. Las técnicas que ambas manejan son las de Clasificación y las de Clustering y que ya han sido aplicadas anteriormente para minería de opinión y análisis de sentimientos sobre Twitter, en sistemas de tiempo real que lee tuits, detecta cambios y halla los términos cuya frecuencia ha cambiado, entre otras aplicaciones.

1.4 SISTEMAS DE MONITORIZACIÓN

Los sistemas de monitorización tienen como objeto, “mantener el control e informar el estado de variables medidas y ejecutar acciones correctivas cuando estas sean necesarias” (Abarca, 2008). Así mismo, el uso de tecnología informática ha permitido tener un óptimo control automático, alcanzar procesamientos de datos que anteriormente eran muy difíciles de alcanzar.

Un sistema de monitorización busca conjugar componentes tanto físicos (dispositivos electrónicos o mecánicos) como “abstractos” o intangibles (software embebido) y medios de comunicación de datos que logren integrarse entre sí, y que puedan ejecutar operaciones sin la intervención humana o a su vez reducirla al mínimo facilitando su supervisión desde cualquier lugar y en todo momento.

Sin duda, los sistemas y dispositivos utilizados para la monitorización y control son fundamentales en un proceso industrial, comercial o de producción; pero otro componente muy importante, es el medio que se emplea para la transmitir los datos que estos sistemas captan y generan.

En los últimos años, el avance tecnológico ha obligado que otros sectores estén a la par con la evolución tecnológica, y que gracias a esta evolución ha permitido converger en un concepto de gran trascendencia denominado el Internet de la Cosas. Por tanto, esta tecnología aborda la sincronización y conexión simultánea de varios dispositivos usando la internet, creando un medio adecuado para estos sistemas de monitorización y, además, dotándolos de mejores servicios.

1.4.1 Sistemas basados en IoT, Wsn Y Data Stream Mining

Al considerar la palabra Internet del concepto de IoT, es fácil hacerla relación con servicios, con protocolos y principalmente con su medio de conexión, es decir, el uso de cable o de forma inalámbrica. Hoy en día, el uso de redes inalámbricas ya no solo se limita a un hogar, a un puesto de trabajo o un aeropuerto; sino más bien, ha ido ganando espacios en todo tipo de lugares como parques, centros comerciales inclusive ciudades enteras y que son aprovechadas por varios dispositivos para transmitir información.

Por este motivo, los sistemas de monitoreo actuales sumados con el concepto de IoT y aprovechando la capacidad de conexión inalámbrica con la que podemos disponer, han llevado a conformar redes inalámbricas de sensores (Wireless Sensor Network) con las que permiten obtener información desde diferentes lugares y de cierta manera reduciendo costos por la infraestructura de red mediante cable; así también, este medio permite obtener todos los beneficios de internet y la capacidad de tener sistemas de monitoreo inteligentes y oportunos es decir sistemas de IoT basados en una WSN.

1.4.2 Estudios realizados

Predictive Analytics for Complex IoT Data Streams.: (Akbar, Khan, Carrez, & Moessner, 2017)

Este artículo relaciona dos conceptos importantes para investigación propuesta. El primer concepto implica al análisis predictivo de flujo de datos el en el que se propone un algoritmo de adaptativo de predicción denominado Adaptive Moving Window Regression (AMWR). El segundo concepto involucra al Internet de las Cosas (IoT), que se refiere a la interconexión de cualquier objeto tecnológico a través de la internet. Al disponer de dispositivos tecnológicos que se comuniquen entre sí y nos ofrezcan datos y detalles sobre su funcionamiento, ambos conceptos convergen en el término dato. Es así que, este artículo muestra una arquitectura orientada al análisis predictivo con la particularidad que implica a la obtención de datos desde dispositivos IoT.

Internet of Things Greenhouse Monitoring and Automation System.: (Doknić, 2014)

El presente artículo detalla la aplicación del concepto IoT en ambientes controlados y con una orientación a la agricultura y la industrialización. Aquí se menciona la recolección de datos mediante el uso de sensores y el monitoreo de los mismos a través de un software.

Otro punto que es de relevancia de este artículo es el detalle de ciertos pasos que permite entender el proceso que sigue este sistema de monitoreo y como lo interrelaciona con el IoT. A su vez, presenta ciertos factores abióticos que están siendo medidos y enviados a través de los sensores hacia la red y plantea también una arquitectura que puede ser adaptada a las necesidades de esta investigación.

1.5 APACHE SPARK STREAMING

Es un motor de procesamiento para grandes cantidades de datos (Big Data), el cual permite generar modelos para flujo de datos (streaming), lenguaje estructurado de consulta (SQL), machine

learning y para el procesamiento de grafos. Esta herramienta, permite desarrollar aplicaciones para flujos de datos escalables y tolerantes a fallos y que a su vez soporta diferentes lenguajes de programación como Java, Python y Scala.

Apache Spark comprende un conjunto de librerías para analítica compleja y que permite combinar cada una de ellas en una misma aplicación y a su vez, tiene la capacidad de ser ejecutado en varios sistemas.

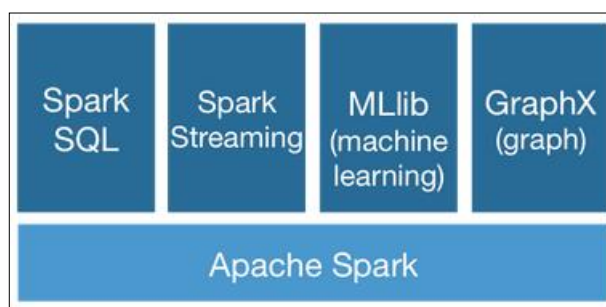


Figura 1 Arquitectura de Spark Core
Fuente: (Apache Software Foundation, 2019)

1.6 LENGUAJE DE PROGRAMACIÓN SCALA

Este lenguaje combina la programación orientada a objetos con la programación funcional y un potente lenguaje de programación. Una característica que denota a este lenguaje, es que mediante los tipos tático que maneja Scala, permite crear aplicaciones complejas evitando caer en errores.

La codificación mediante este lenguaje ofrece un incremento en el rendimiento de las aplicaciones, así como también, permite disponer de ciertas características tanto avanzadas como funcionales.

Otra característica notable en este lenguaje, es que es una entrega libre (open-source) y que brindan excelentes funcionalidades para el análisis de datos.

1.7 METODOLOGÍA TRADICIONAL: MODELO EN “V”

El Modelo en “V” “es considerado una variante del modelo en cascada, debido a que ofrece un enfoque sistemático y secuencial para el desarrollo de software” (Pressman, 2010), este modelo toma su nombre de la letra “V” ya que, la letra se le considera en dos partes, la primera parte es la que desciende por el lado izquierdo de “V”, el cual establece los conceptos y requerimientos básicos del sistema, su arquitectura y un diseño detallado, hasta llegar a la realización del código.

Una vez terminada la codificación y obtenido un software ejecutable, se inicia con la segunda parte, la cual asciende por el lado derecho de la “V”; está implica la realización de pruebas unitarias, integración, verificación y validación del sistema desarrollado.

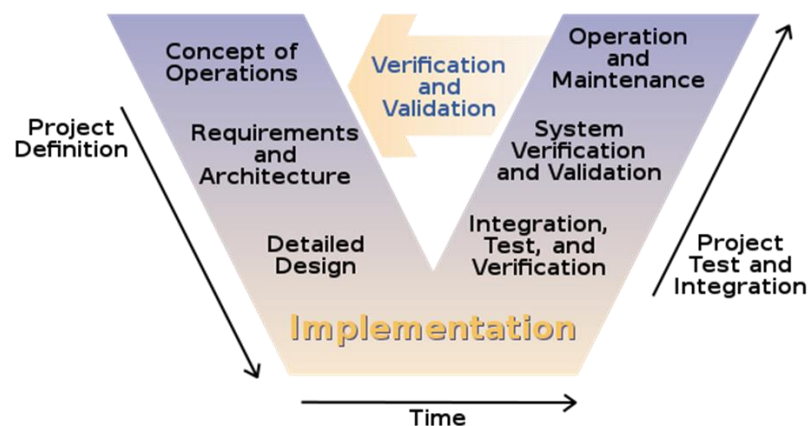


Figura 2 Metodología de desarrollo “Modelo en V”

Fuente: (Perez, Berreteaga, Ruiz, Urkidi, & Perez, 2006)

El modelo en “V” comprende un conjunto de etapas definidas o acciones, que tienen un enfoque sistemático y secuencial, que relacionar la validación y verificación con las primeras acciones de ingeniería.

Concepto de la Operación:

Describe de manera general los requerimientos necesarios, las características operativas que el sistema debe cumplir sin hacer referencia explícita a las funciones del sistema, su hardware o pantallas.

Esta etapa proporciona una visión a largo plazo del alcance y las capacidades que dispondrá el sistema.

Requerimientos y Arquitectura:

Los requerimientos conllevan a la comprensión de lo que el cliente desea que el sistema (hardware y software) realice y muestra cómo va a interactuar el sistema con el o los usuarios finales. Los requerimientos son importantes, ya que permiten “diseñar y construir un sistema que satisfaga las necesidades del cliente” (Pressman, 2010).

Para validar los requerimientos obtenidos, es necesario realizar un modelado de requerimientos mediante el empleo de texto y gráficos que permitan visualizar y entender fácilmente el funcionamiento del sistema y también poder identificar posibles inconsistencias y corregirlas oportunamente.

La arquitectura se refiere a la estructura general del sistema, a los componentes que lo integran y la forma de cómo éstos interactúan. Según Shaw y Garlan (Shaw & Garlan, 1995), para el diseño de la arquitectura se debe considerar:

- Propiedades Estructurales: Agrupamiento e interacción de los módulos unos con otros del sistema.
- Propiedades Extra funcionales: Cumple con los requerimientos funcionales y no funcionales establecidos.

- Familias de Sistemas Relacionados: Capacidad de reutilizar su arquitectura o que su diseño sea repetible.

Diseño Detallado:

Brinda una descripción detallada de la arquitectura del sistema, las estructuras de datos, interfaces y componentes que se necesitan para ser implementado. El diseño permite modelar el sistema o producto que se va a desarrollar y en la cual se ven reflejados los requerimientos establecidos inicialmente y se definen los lineamientos y atributos de calidad.

Implementación:

Los modelos, diseños y diagramas se materializan traduciéndolos a código fuente y en configuraciones de los dispositivos que integrarán el sistema.

Integración, Prueba y Verificación:

Se realiza la integración todos los componentes que conforman el sistema. Se generan las pruebas correspondientes de integración, pruebas unitarias y se verifica cada módulo del sistema realizando pruebas de fallos de inserciones, de alimentación o fallos de conexión.

Verificación y Validación del Sistema:

Se comprueba que se haya seguido una correcta implementación, coherencia cumplimiento de las especificaciones técnicas y requerimientos respectivos.

Operación y Mantenimiento:

Describe detalladamente como se realiza la operación del sistema, sus capacidades y las actividades que se pueden realizar.

También establece el mantenimiento que se debe realizar a los componentes que lo conforman.

CAPÍTULO II

DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DEL SISTEMA

2.1 CONCEPTO DE OPERACIÓN DEL SISTEMA

2.1.1 Perspectiva del Sistema

El sistema SIS-1 será un producto diseñado para realizar el análisis y la minería de flujos de datos obtenidos de una WSN y que podrá desplegar la visualización del comportamiento de la data en tiempo real de los eventos o factores que se estén suscitando en un ambiente controlado.

El SIS-1 estará conformado por sensores especializados de temperatura, humedad, rayos uv, intensidad lumínica y de dióxido de carbono, dispositivos electrónicos de control, medios de comunicaciones y software para el análisis y procesamiento de datos, los cuales serán clasificados en cuatro componentes detallados en la Tabla 4.

Tabla 4

Componentes del sistema

Código	Componentes
CTE-1	Componente de Sensores (Toma de datos del medioambiente)
CTE-2	Componente de Adquisición de datos (Toma de datos desde la WSN)
CTE-3	Componente de Análisis y Almacenamiento
CTE-4	Componente de Visualización

2.1.2 Funcionalidad del Sistema

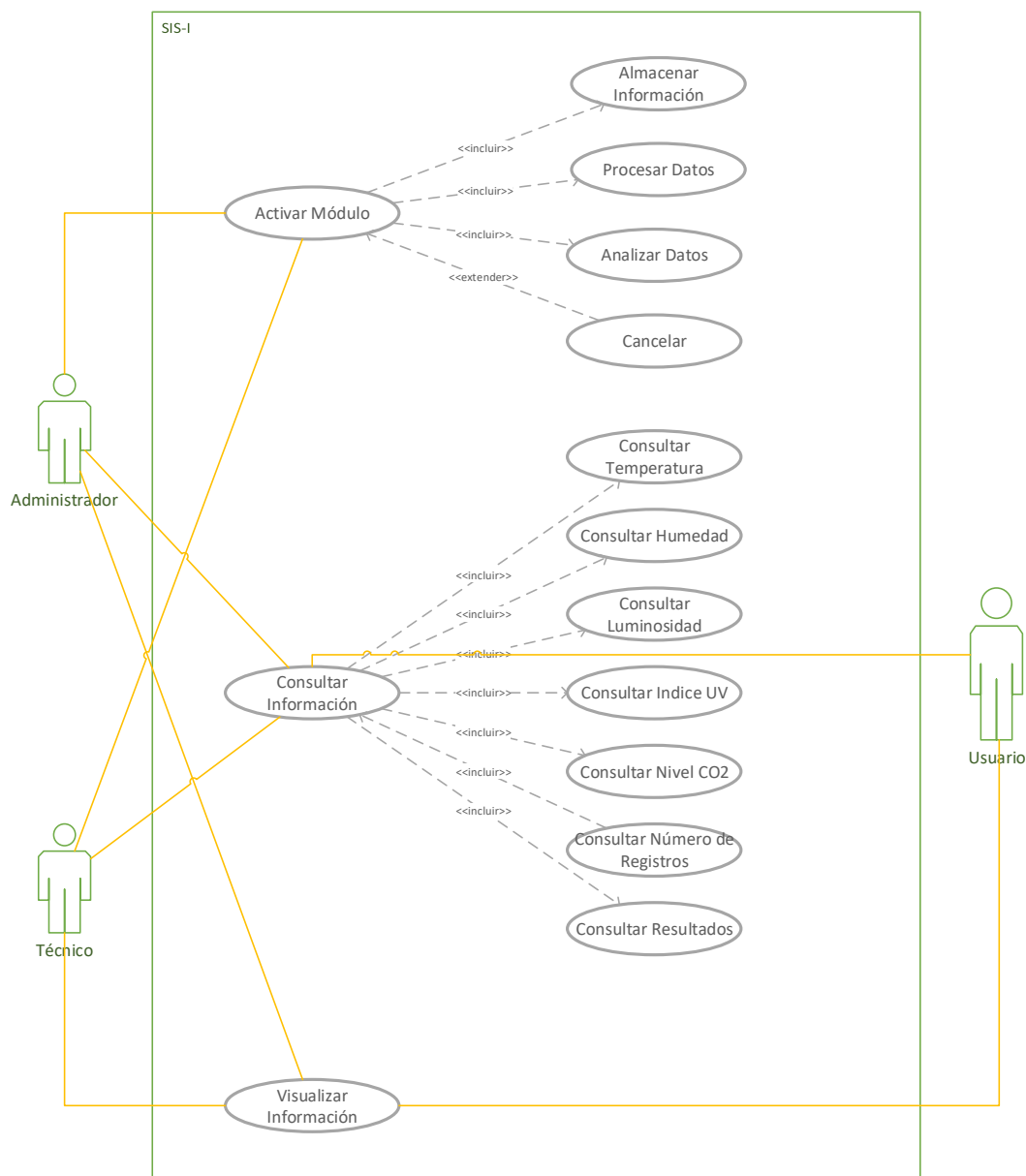


Figura 3 Diagrama de caso de uso del SIS-I

El adoptará la siguiente funcionalidad:

- Los sensores toman continuamente el estado del medioambiente donde se encuentra la plantación.
- Los componentes con los sensores se conectarán mediante una WSN.

- c) La Raspberry Pi se comunica mediante conexión serial y activa al Arduino para iniciar la obtención del flujo de datos desde los sensores hacia la Raspberry para su futuro almacenamiento y análisis.
- d) Se despliega por consola los datos obtenidos de los sensores con su número de captura, simultáneamente son almacenados y analizados.
- e) Se muestra la información necesaria de manera sencilla y comprensible al usuario final.

2.1.3 Características de los usuarios

Tabla 5

Características de los usuarios

Características	Descripción
Tipo de usuario	Administrador
Formación	Superior
Actividades	Control y gestión del sistema en general e invernadero
Tipo de usuario	Profesional en el Área
Formación	Superior
Actividades	Análisis y evaluación de los datos
Tipo de usuario	Estudiante
Formación	Superior
Actividades	Toma de métricas con respecto a las flores

2.1.4 Restricciones del sistema

- a) La interfaz está diseñada para ser usada con conexión remota o internet
- b) El lenguaje y tecnologías de uso: Scala, Python, Arduino, Spark Streaming, Raspbian y SQL
- c) El componente de análisis deberá atender consultas concurrentemente
- d) El diseño deberá ser de implementación sencilla

e) El SIS-1 debe ser de bajo costo

2.1.5 Suposiciones y dependencias

a) Los dispositivos electrónicos están operantes y son estables.

b) Los sistemas operativos empleados son estables.

c) Los dispositivos toleran estar encendidos de forma permanente y continua por varios meses.

d) El material del que están compuestos los componentes de sensores y adquisición de datos toleran los cambios de los factores medioambientales del entorno durante su operación.

e) Se depende de una conexión eléctrica de forma permanente y continua.

f) Se depende de una red para la transmisión de datos de forma permanente y continua.

2.2 REQUERIMIENTOS Y ARQUITECTURA DEL SISTEMA

2.2.1 Requisitos funcionales

2.2.1.1 Requisito funcional 1 (RF-1)

- **Activar Módulo:** los usuarios deberán activar el módulo de sensores para iniciar con el proceso de captura y generación del flujo de datos.
 - El sistema podrá ser activado a través del servidor por cualquier usuario dependiendo su nivel de accesibilidad.

2.2.1.2 Requisito funcional 2 (RF-2)

- **Obtener Datos:** el sistema deberá obtener el flujo de datos de la WSN desde el servidor (CTE-2).
 - El sistema de obtención de datos iniciara con la ejecución del RF-1 activado por cualquier usuario dependiendo su nivel de accesibilidad.

2.2.1.3 Requisito funcional 3 (RF-3)

- **Analizar los datos:** el sistema deberá analizar el flujo de datos que están siendo obtenidos desde la WSN.
 - La función de análisis del flujo de datos iniciara automáticamente cuando se ejecute el RF-1.

2.2.1.4 Requisito funcional 4 (RF-4)

- **Almacenar la Información:** el sistema deberá almacenar los datos generados en una base de datos.
 - El sistema almacenará automáticamente los datos generados una vez se ejecute el RF-1, RF-2 y RF-3.

2.2.1.5 Requisito funcional 5 (RF-5)

- **Consultar Información:** el sistema permitirá a los usuarios acceder a la información que está siendo almacenada y analizada del flujo de datos.
 - **Consultar Temperatura:** Muestra valores los registrados de Temperatura (Máxima, Mínima, Promedio) por día y semana.
 - **Consultar Humedad:** Muestra valores los registrados de Humedad (Máxima, Mínima, Promedio) por día y semana.
 - **Consultar intensidad Lumínica:** Muestra valores los registrados de intensidad Lumínica (Máxima, Mínima, Promedio) por día y semana.
 - **Consultar índice UV:** Muestra valores los registrados de del índice UV (Máxima, Mínima, Promedio) por día y semana.

- **Consultar nivel de CO₂**: Muestra valores los registrados del nivel de CO₂ (Máxima, Mínima, Promedio) por día y semana.
- **Consultar Número de Registros**: Muestra el número de capturas realizadas (totales, por día o por semana).
- **Consultar resultado Análisis**: Muestra información relevante producto del análisis generado en base a los valores medidos.

2.2.1.6 Requisito funcional 6 (RF-6)

- **Visualizar la Información**: el sistema permitirá visualizar los datos generados desde los sensores, así como desde la base de datos.
 - El sistema mostrará los datos generados una vez se ejecute el RF-1, RF-2 RF-3 y RF-4.

2.2.2 Requisitos no funcionales

2.2.2.1 Requisito no funcional 1 (RN-1)

- **Rendimiento del Sistema**: garantizar el funcionamiento continuo y conectividad a la red para la captura y envío de los datos y su para que el análisis y el flujo de datos no se vean afectados.
 - El sistema deberá disponer energía y red en lugar, y de ser posible, disponer de un UPS.

2.2.2.2 Requisito no funcional 2 (RN-2)

- **Seguridad del Sistema**: garantizar la integridad, confiabilidad y la seguridad de los datos que estas siendo recolectados, así como también la protección del hardware.
 - El sistema debe ser colocado en un lugar con cubierta que no afecte a la toma de datos, donde este protegido de factores como polvo y derrames de líquidos.

2.2.2.3 Requisito no funcional 3 (RN-3)

- **Fiabilidad del Sistema:** el sistema debe ser usable, intuitivo y de fácil instalación e interpretación.
 - El sistema debe aplicar la estructura de componentes para su fácil integración, ensamblaje, uso e interpretación.

2.2.2.4 Requisito no funcional 4 (RN-4)

- **Disponibilidad del Sistema:** el sistema debe estar disponible en todo momento a lo largo del todo el periodo de cultivo o ciclo de cultivo.
 - El ciclo de cultivo que debe estar disponible el sistema es de 3 meses durante 24 horas al día.

2.2.2.5 Requisito no funcional 5 (RN-5)

- **Mantenibilidad del Sistema:** el sistema dispondrá de la documentación necesaria para realizar las operaciones de mantenimiento tanto de su hardware como de su software.
 - El sistema podrá ser sujeto a inspección y mantenimiento solo por personal técnico especializado en el equipo.

2.2.2.6 Requisito no funcional 6 (RN-6)

- **Portabilidad del Sistema:** el sistema será instalado bajo la plataforma Raspbian en conjunto con el uso de programas Arduino, Scala, Apache Spark y Python.

2.2.3 Arquitectura del sistema

El sistema SIS-I se conforman de varios elementos y que se ven detallados figura 4. Esta arquitectura permite apreciar de manera general la interacción de todos los factores.

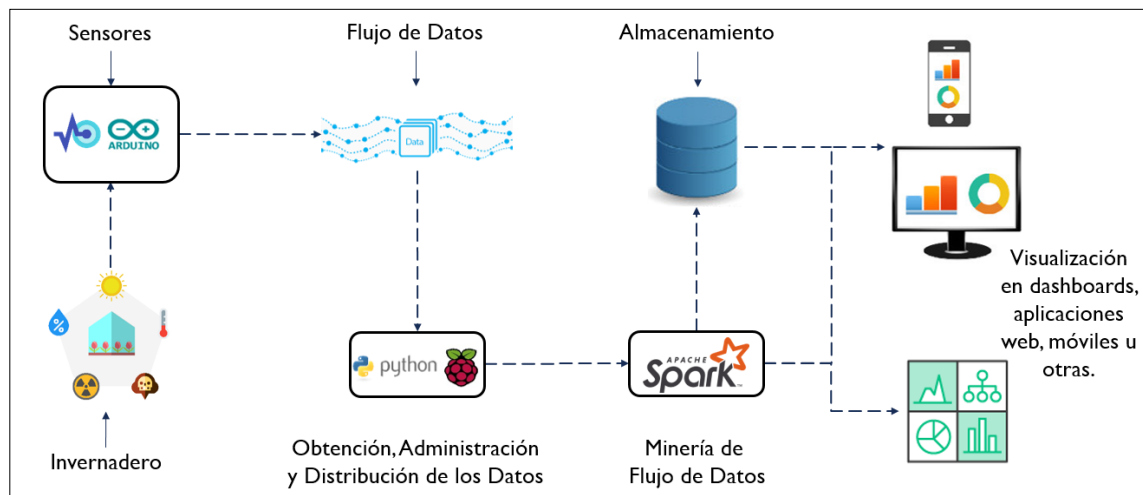


Figura 4 Arquitectura del sistema

Mientras que en la figura 5 se muestra una arquitectura de los componentes que conformarán tanto la parte de hardware como de software.:

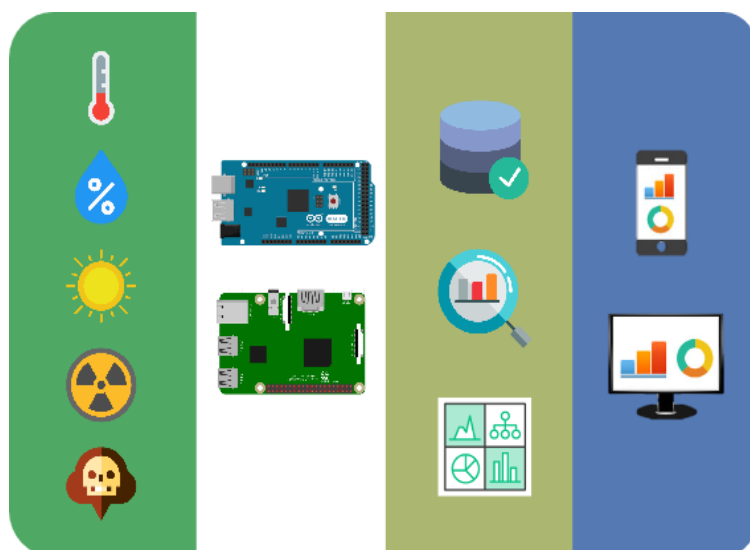


Figura 5 Componentes hardware-software

2.3 DISEÑO DETALLADO

2.3.1 Diseño de hardware

a) Componentes de Sensores CTE-1:

Aquí se encuentran los diferentes sensores, los mismos que estarán acoplados a una baquelita, formando un módulo para la captura de datos. Estos serán los encargados de capturar los datos de los factores medioambientales del ambiente controlado.

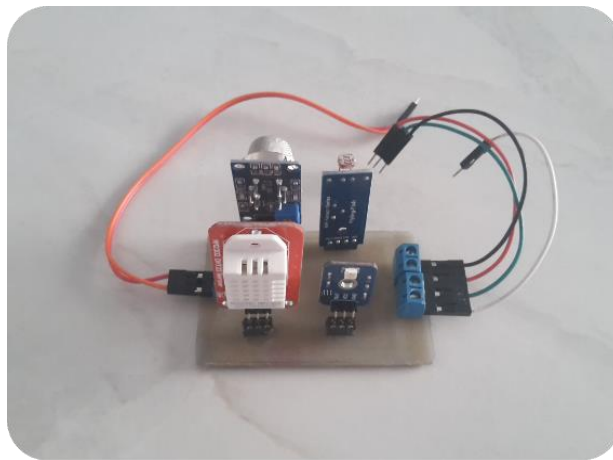


Figura 6 Componente CTE-1

Este módulo estará conformado de cinco sensores:

1. Sensor de temperatura
2. Sensor de humedad
3. Sensor de luminosidad
4. Sensor de radiación uv
5. Sensor de calidad del aire

b) Componente de Adquisición de datos CTE-2

Este componente dispondrá de un Arduino Mega y una Raspberry Pi 3 conectadas entre sí mediante puerto serial USB. Estos dispositivos serán los encargados de recolectar los datos

provenientes de los sensores de la WSN y del análisis/almacenamiento respectivamente. Cada uno ejecutará su propia programación, considerando que la Raspberry será el dispositivo en el cual se almacenará la información y ejecutará los comando que activan al Arduino para iniciar la toma de datos.



Figure 7 Componente CTE-2

2.3.2 Diseño de software

a) Componente de Análisis CTE-3

En este componente se realizará el análisis se lo realiza mediante el uso de herramientas especializadas, las misma que tendrán como insumo los datos que están siendo capturados por el componente CTE-1 y CTE-2.

b) Componente de Visualización CTE-4

Este componente será la encargada de mostrar de una manera comprensible toda la información procesada en CTE-3 y a la cual se podrá acceder desde diferentes medios como dashboards, monitores, tablets o navegadores y dispondrá de esta información cuando se lo requiera.

2.4 IMPLEMENTACIÓN

2.4.1 Diseño de la red de sensores (WSN)

La funcionalidad del sistema establece que se diseñe una WSN para la transmisión de datos. La red WSN se conforma de dos nodos de sensores, un servidor principal en el cual se realiza la obtención, procesamiento y almacenamiento de los datos, la misma que está conectada a una estación base para realizar el Data Stream Mining y permitirá visualizar los datos mediante la conexión de monitores, smartphones, tablets o dashboards.

La conexión se realizará través del protocolo TCP-IP. En este caso los sockets utilizados para el obtener la data son el puerto 7777 para el flujo de datos de entrenamiento y el puerto 9999 para el flujo de datos de testeo. De la misma manera, la estación base estará conectada a la red y será configurada para la recepción de los datos en los mencionados puertos.

La WSN propuesta para el sistema SIS-I es la siguiente:

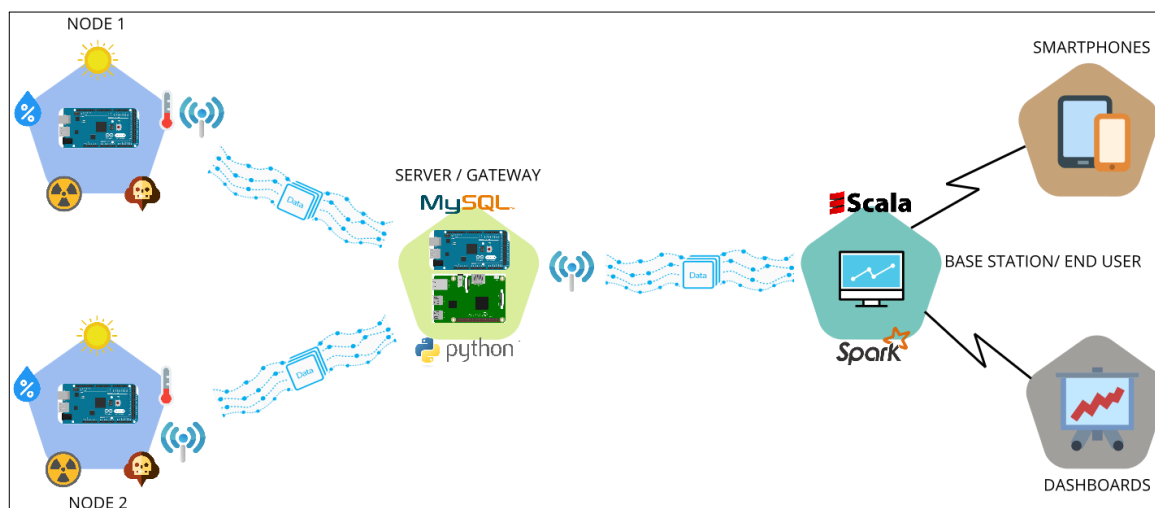


Figure 8 Diseño de la Wireless Sensor Network

2.4.2 Diseño y desarrollo de la base de datos

El RF-2 establece que el sistema SIS-1 pueda almacenar los datos generados provenientes de los sensores. Por lo tanto, la base de datos aplicable para el sistema SIS-1, considerando el uso de herramientas opensource, se ha considerado la base de datos MariaDB.

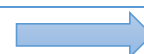
MariaDB es un gestor de bases de datos que ofrece opciones para modificar, crear, eliminar o generar consultas de los datos que hayan sido almacenados ésta. Una característica principal para optar por este gestor, es debido a que “MariaDB posee mejoras sobre su antecesor MySQL, principalmente en mecanismos de almacenamiento, mejoras en la velocidad y rendimiento y facilidad de uso” (Nerion Networks, 2019).

Una vez definido el gestor, se verificó los requerimientos descritos en el documento IEEE-830 para el desarrollo del sistema SIS-1 y en el cuál se determinó nueve atributos que formarán parte la base de datos y que se encuentran descritos en la tabla 6:

Tabla 6
Atributos para la base de datos

Identificador	Nombre	Descripción
id	Código de registro	Identificador del número de captura almacenada en la base de datos
node	Numero de nodo	Número que identifica el nodo del cual provino los datos que fueron almacenados
date	Hora y fecha	Año, mes, día con horas, minutos y segundos en la cual se almacenos los datos
temperature	Temperatura	Temperatura del ambiente
humidity	Humedad	Humedad del ambiente
brightness	Luminosidad	Indicador del nivel de luminosidad

CONTINÚA



uv_factor	Índice UV	Indicador del índice de radiación UV generado por el sol
air_quality	Calidad del aire	Indicador del nivel de la calidad del aire en el ambiente
others	Otros	Observaciones adicionales o importantes que se produzca en cada nodo

Una vez realizado el modelado la base de datos se obtiene el siguiente diseño de la entidad:

DATOS_SENSORES			
<u>id</u>	int	<pk>	not null
node	int		not null
date	datetime		not null
temperature	decimal(6,2)		not null
humidity	decimal(6,2)		not null
brightness	int		not null
uv_factor	int		not null
air_quality	int		not null
others	varchar(500)		null

Figura 9 Datos Sensores

Luego de generar el modelo físico, se obtuvo el código “.sql” para la creación de la base de datos en el servidor RAS-I.

```

/*=====*/
/* DBMS name:      MySQL 5.0                               */
/* Created on:     1/9/2019 7:22:05 AM                       */
/*=====*/

drop table if exists DATOS_SENSORES;

/*=====*/
/* Table: DATOS_SENSORES                                    */
/*=====*/
create table DATOS_SENSORES
(
  ID          int not null,
  NODE       int not null,
  DATE       datetime not null,
  TEMPERATURE decimal(6,2) not null,
  HUMIDITY   decimal(6,2) not null,
  BRIGHTNESS int not null,
  UV_FACTOR  int not null,
  AIR_QUALITY int not null,
  OTHERS     varchar(500),
  primary key (ID)
);

alter table DATOS_SENSORES comment 'Esta tabla almacena todos los datos provenientes de una red ';

```

Figura 10 Script de creación de la tabla datos_sensores

2.4.3 Diseño y desarrollo del software del nodo de sensores ARD-I (Arduino)

La perspectiva del sistema establece que el SIS-1 posee sensores especializados encargados de capturar factores ambientales y que se mencionan en el RF-1 y RF-5. Estos factores corresponden a la medida de temperatura, humedad, luminosidad, índice UV y de calidad del aire.

El CTE-1 es el encargado de realizar esta función, pero requiere que sus componentes sean programados mediante la interfaz de Arduino IDE con su propio lenguaje de programación. Para esto, se verificó los requerimientos descritos en el documento IEEE-830 para el desarrollo del sistema SIS-1 y se determiné el diagrama de flujo (ver figura 11) para el sistema que controlara el CTE-1 y en el cuál se determinó nueve atributos (ver tabla 7) que formarán parte de la programación del ARD-I.

El diagrama de flujo del ARD-I, muestra los procesos y subprocesos que va a cumplir el sistema embebido en la placa arduino y será la encargada de la captura de datos:

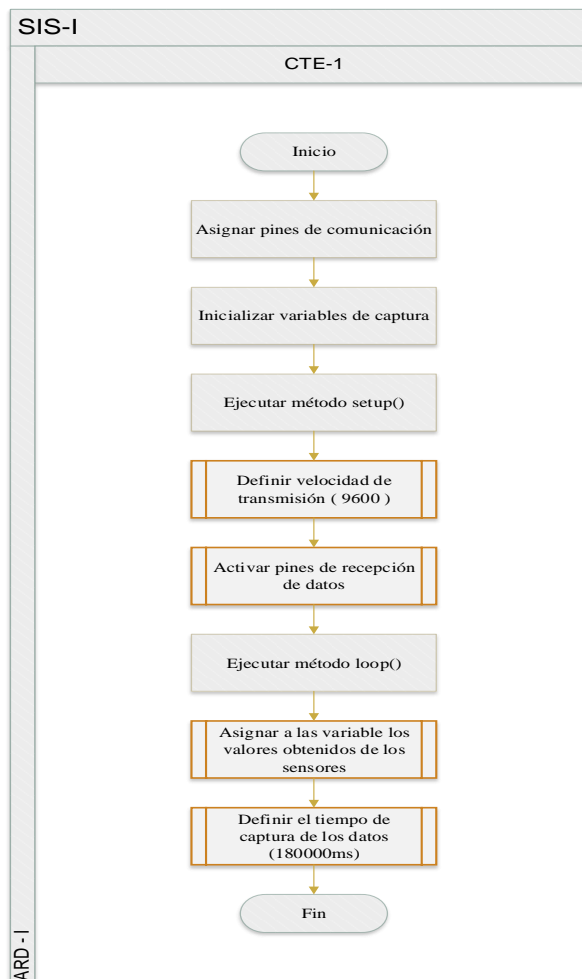


Figura 11 Diagrama de flujo del ARD-I

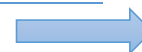
Una vez establecido el diagrama de flujo del sistema ARD-I, se muestra la descripción de los atributos y métodos que intervienen en el proceso para su futura codificación:

Tabla 7

Atributos de sistema ARD-I

Variables	Nombre	Descripción
temperature	Temperatura	Temperatura del ambiente
humidity	Humedad	Humedad del ambiente

CONTINÚA



bright	Luminosidad	Indicador del nivel de luminosidad
factorUv	Índice UV	Indicador del índice de radiación UV generado por el sol
airQuality	Calidad del aire	Indicador del nivel de la calidad del aire en el ambiente
sensorBright	Sensor serie MH	Variable de comunicación del sensor de luminosidad mediante entrada analógica
sensorUv	Sensor de UV	Variable de comunicación del sensor de radiación UV mediante entrada analógica
sensorAir	Sensor MQ-135	Variable de comunicación del sensor de calidad del aire mediante entrada analógica
sensorDht	Sensor DHT-22	Variable de configuración de la entrada digital y del modelo del sensor

Tabla 8
Métodos empleados para el ARD-I

Métodos	Nombre	Descripción
setup()	Configurar	Método que activa las entradas analógicas y la comunicación serial con el dispositivo arduino
loop()	Repetir	Método que genera la captura de datos desde los sensores continuamente y de acuerdo a una latencia especificada por el usuario

Luego de obtener la información necesaria en el diagrama de flujo y conocer los atributos se generó el respectivo algoritmo y su codificación:

Algoritmo del Sistema ARD-I que controla el CTE-1

Las variables utilizadas se encuentran especificadas en la tabla 8

1. Inicio
2. Asignar a la variable **sensorDht** los valores (52,22)
(**52** es el pin para transmitir los datos y **22** es la serie del sensor)
3. Asignar a la variable **sensorBright** el pin de salida análoga **A0**
(**A0** es el pin de transmisión de datos tipo analógica)
4. Asignar a la variable **sensorUv** el pin de salida análoga **A1**
(**A1** es el pin de transmisión de datos tipo analógica)
5. Asignar a la variable **sensorAir** el pin de salida análoga **A2**
(**A0** es el pin de transmisión de datos tipo analógica)
6. Declarar variables **temperatura**, **humidity**, **bright**, **factorUv** y **airQuality** y asignarles el valor de 0
7. Ejecutar el método **setup()** para la configuración de velocidad de transmisión y la activación de los pines
(La velocidad de transmisión será de **9600**)
8. Ejecutar el método **loop()** el cual iniciará la captura continua de datos y asignará a las variables **temperatura**, **humidity**, **bright**, **factorUv** y **airQuality** los valores correspondientes de cada sensor.
9. Definir la latencia de captura de datos
(La latencia de captura debe ser igual a **180000**)

```

#include <HT.h> //libreria para el uso del sensor DHT22
DHT sensorDht(52,22); //Variable para el Sensor DHT22 para medir la temperatura y la humedad, el numero 52 INDICA la salida digital y el numero 22 es el modelo del sensor
const int sensorBright = A0; //Variable para el Sensor Serie MH Flying Fish para medir la intensidad lumínica y se comunica mediante la salida analógica A0
const int sensorUV = A1; //Variable para el Sensor para medir la intensidad de radiación UV y se comunica mediante la salida analógica A1
const int sensorAir = A2; //Variable para Sensor MQ 135 para medir la calidad del aire y se comunica mediante la salida analógica A2

//Variables para guardar los datos provenientes de los sensores
float temperature = 0;
float humidity = 0;
int bright = 0;
int factorUV = 0;
int airQuality = 0;

void setup() { //Método para activar los pines de comunicación
  Serial.begin(9600); //Definición de la velocidad de transmisión
  pinMode(sensorBright, INPUT);
  pinMode(sensorUV, INPUT);
  pinMode(sensorAir, INPUT);
}

void loop() { //Método para iniciar la captura continua de los datos
  temperature = sensorDht.readTemperature();
  humidity = sensorDht.readHumidity();
  bright = analogRead(sensorBright);
  factorUV = analogRead(sensorUV);
  airQuality = analogRead(sensorAir);

  Serial.print(temperature);
  Serial.print(",");
  Serial.print(humidity);
  Serial.print(",");
  Serial.print(bright);
  Serial.print(",");
  Serial.print(factorUV);
  Serial.print(",");
  Serial.print(airQuality);
  Serial.println("");
  delay(10000); //Definición del tiempo de espera entre cada captura de datos 30seg (10000ms)
}

```

Figura 12 Codificación del programa ARD-I

2.4.4 Diseño y desarrollo del software del servidor RAS-I (Raspberry/Python)

La perspectiva del sistema establece que el SIS-1 menciona el uso de software que permita la recolección de los datos desde la WSN. Este se encuentra como parte del CTE-2 y que se menciona en el RF-2.

El RAS-I es un sistema desarrollado en lenguaje Python que se ejecuta sobre el sistema operativo Raspbian del dispositivo Raspberry Pi. Para su desarrollo, se verificó los requerimientos descritos en el documento IEEE-830 del sistema SIS-I y las salidas generadas en el sistema ARD-I del cual se obtendrán los datos.

El diagrama de flujo de la figura 13 muestra los procesos y subprocesos que va a cumplir el sistema RAS-I ejecutado en el servidor (dispositivo Raspberry Pi) y será la encargada de la obtención de datos de la WSN.

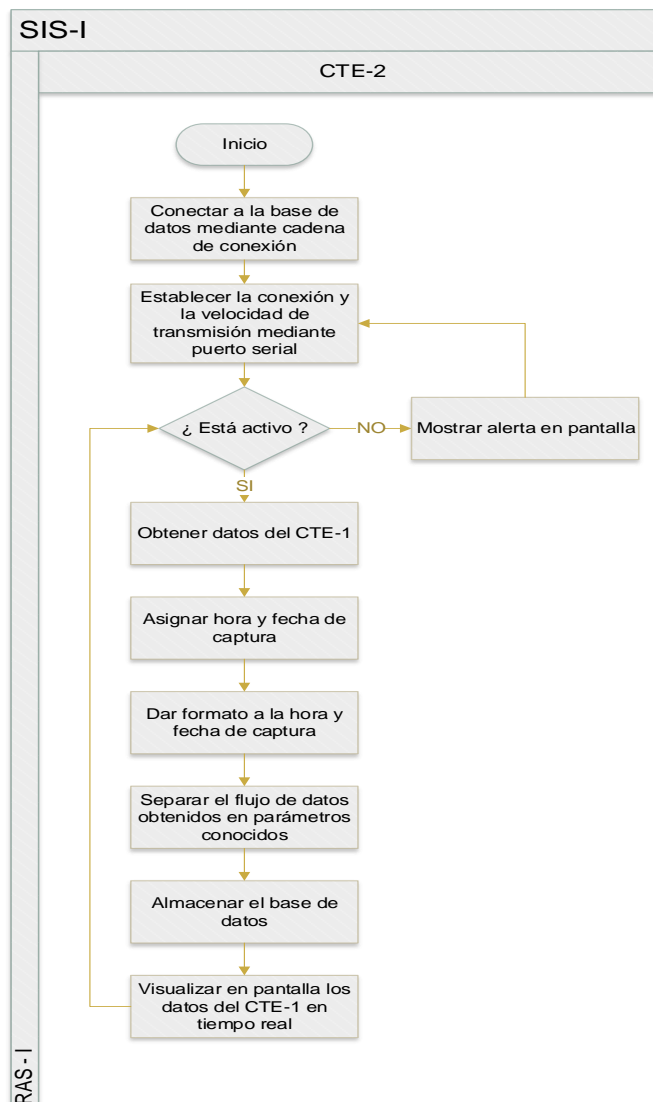


Figure 13 Procesos y subprocesos del sistema RAS-I

Una vez establecido el diagrama de flujo del sistema RAS-I, se muestra la descripción de los atributos que intervienen en el proceso para su futura codificación:

Tabla 9
Atributos del sistema RAS-I

Variables	Nombre	Descripción
node	Nodo	Número que identifica al nodo de donde provienen los datos
values	Valores	Cadena de datos obtenida desde los sensores
temperature	Temperatura	Temperatura del ambiente
humidity	Humedad	Humedad del ambiente
bright	Luminosidad	Indicador del nivel de luminosidad
factorUv	Índice UV	Indicador del índice de radiación UV generado por el sol
airQ	Calidad del aire	Indicador del nivel de la calidad del aire en el ambiente
Others	Otros	Observaciones adicionales o importantes que se produzca en cada nodo

Luego de obtener la información necesaria en el diagrama de flujo y conocer los atributos se generó el respectivo algoritmo y su codificación:

Algoritmo del Sistema RAS-I que controla el CTE-2

Las variables utilizadas se encuentran especificadas en la tabla 9

1. Inicio
2. Crear la **cadena de conexión** para la base de datos
3. Declarar las variables **node**, **values**, **temperature**, **humidity**, **bright**, **factorUv**, **airQ** y **Others**

4. Asignar a las variables **node**, **values**, **temperature**, **humidity**, **bright**, **factorUv**, **airQ** y **Others** el valor de string “ ”
5. Establecer conexión con el CTE-1
6. Asignar a la variable **value** la cadena de strings (flujo de datos) provenientes del CTE-1 (**value** contiene una cadena de caracteres con los valores de cada sensor)
7. Crear una expresión regular para separar cada dato de la variable **value** (se separa cada dato identificado por la terminación de una coma)
8. Asignar a las variables **node**, **values**, **temperature**, **humidity**, **bright**, **factorUv**, **airQ** y **Others** el valor de correspondiente de cada sensor
9. Generar el **query** con las variables **temperature**, **humidity**, **bright**, **factorUv**, **airQ** y **others**
10. Ejecutar el **query** para almacenar en la base de datos.
11. Mostrar en pantalla los valores de número de captura, fecha y hora, temperatura, humedad, índice de luminosidad, índice de radiación uv y calidad del aire.
12. Asignar a las variables **node**, **values**, **temperature**, **humidity**, **bright**, **factorUv**, **airQ** y **Others** el valor de string “”

```

import serial          #Permite el uso del puerto serial
import time
import mysql.connector
import decimal         #Libreria para la transformación de string a decimal

#Parametros de conexión con MariaDB (MySQL)
config_mysql = {
    'user': 'root',
    'password': 'root',
    'host': 'localhost',
    'database': 'GreenHouse',
}

conector = mysql.connector.connect(**config_mysql)    # Conector con el servidor de base de datos

# Clase "cursor" necesaria para gestionar SQL
cursor = conector.cursor()

#Variables Globales de los sensores y cadena de caracteres
node=1          #Describe el numero de nodo de donde se genera el flujo de datos
values=""       #Almacena toda la cadena de datos generada por el CTE-1
temperature=""  #Almacena datos de temperatura en grados centigrados
humidity=""     #Almacena datos de humedad en porcentaje
bright=""       #Almacena datos de de intensidad li=luminica de 0 a 1024. 0 más claro 1024 más oscuro
factorUv=""     #Almacena datos de intensidad de radiación uv
airQ=""         #Almacena datos de calidad de aire
others="none"    #Almacena datos de características importantes como position, error, etc

#Establecimietno de la conexión serial entre la raspberry y arduino ('id', puerto)
arduino=serial.Serial('/dev/ttyACM0', baudrate=9600)

flag=0          #Indica el número de dato obtenido desde el CTE-1 (Componente 1)

while True:
    values=arduino.readline()          #Variable que obtiene el flujo de datos en una cadena desde el CTE-1
    date=time.strftime("%Y-%m-%d %H:%M:%S") #Formato de fecha y hora personalizado

    print ("Nro. Captured Data: ",flag)

    # Algoritmo desarrollado para separar cada parametro de la cadena obtenida sin comas
    i=0
    j=0
    for i in range(len(values)):
        if values[i]!=',':
            if j==0:
                temperature+=values[i]
            if j==1:
                humidity+=values[i]
            if j==2:
                bright+=values[i]
            if j==3:
                factorUv+=values[i]
            if j==4:
                airQ+=values[i]
            else:
                j+=1

    # Consulta SQL (Inserción de datos)
    query = ("INSERT INTO datos_sensores (id, date, node, temperature, humidity, brightness, uv_factor, air_quality, others) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)")

    # Ejecución de la consulta SQL. El comando usa 2 parametros: 1. La consulta Query 2. Variables que van a ser guardadas
    variables=(' ',date, node, decimal.Decimal(temperature), decimal.Decimal(humidity), bright, factorUv, airQ, others)
    cursor.execute(query,variables)

    print (date)
    print (temperature)
    temperature=""
    print (humidity)
    humidity=""
    print (bright)
    bright=""
    print (factorUv)
    factorUv=""
    print (airQ)
    airQ=""

    flag+=1

# Cierre de la conexión
conector.close()

```

Figura 14 Codificación del programa ARD-I

2.4.5 Diseño y desarrollo del software de análisis SPK-I (Apache Spark/ Scala)

El RF-3 establece que el sistema SIS-I, dispondrá de una funcionalidad para el análisis de datos, en la cual se ejecutarán algoritmos que permitan realizar un análisis predictivo en base a flujos de datos tanto de entrenamiento como para pruebas que generen y validen el modelo.

El sistema SPK-I está desarrollado bajo el lenguaje de programación Scala, ejecutado bajo el sistema operativo Windows 10. Para su desarrollo, se verificó los requisitos funcionales y no funcionales detallados en el IEEE-830 del sistema SIS-I.

El diagrama de flujo de la figura 15, se muestran los procesos y subprocesos que va a cumplir el sistema SPK-I:

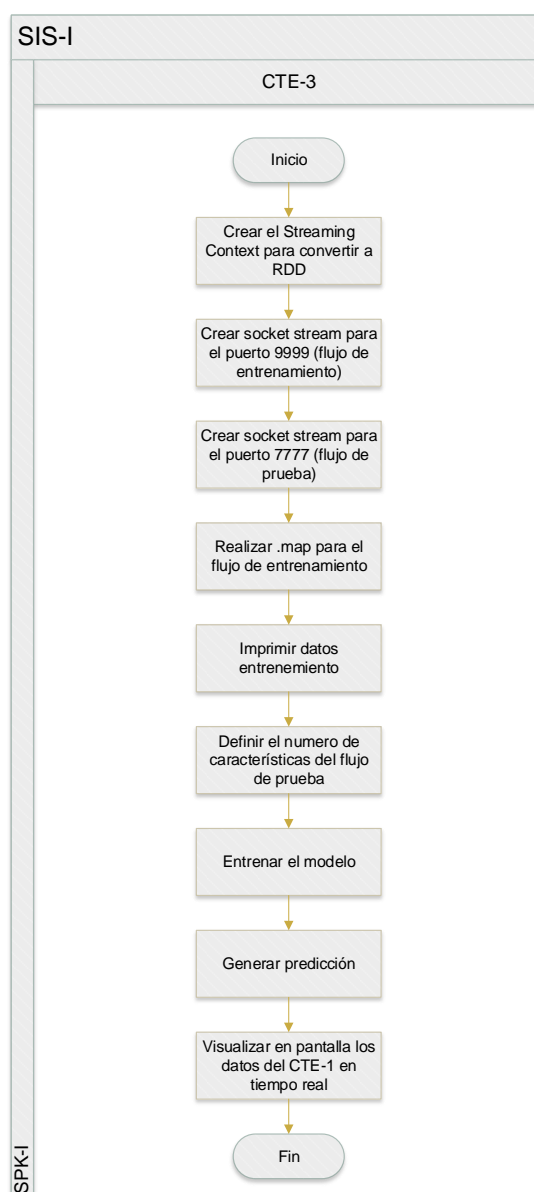


Figure 15 Procesos y subprocesos del sistema SPK-I:

Una vez establecido el diagrama de flujo del sistema SPK-I, se muestra la descripción de los atributos que intervienen en el proceso para su futura codificación:

Tabla 10
Atributos del sistema SPK-I

Variables	Nombre	Descripción
scc	Streaming Context	Crea el RDD respectivo con una ventana de un 1 segundo
trainingLines	Flujo de entrenamiento	Flujo de entrenamiento de entrada
testingLines	Flujo de prueba	Flujo de prueba de entrada
trainingData	Datos de entrenamiento	Datos de entrenamiento etiquetados para el uso del algoritmo de regresión en streaming
testData	Datos de prueba	Datos de prueba etiquetados para el uso del algoritmo de regresión en streaming
numFeatures	Número de características	Numero de características en los datos recibidos
model	Modelo	Inicio de construcción del modelo de regresión en streaming

Luego de obtener la información necesaria en el diagrama de flujo y conocer los atributos se generó el respectivo algoritmo y su codificación:

Algoritmo del Sistema SPK-I para el análisis:

Las variables utilizadas se encuentran especificadas en la tabla 10

1. Inicio
2. Crear el **contextStreaming** para inicializar con el sistema

3. Declarar las variables **trainingLines** y **testingLines** para captar el flujo de datos.
4. Asignar a la variable **trainingLines** el puerto **9999**
5. Asignar a la variable **testingLines** el puerto **7777**
6. Declarar la variable **trainingData** y asignar el valor etiquetado para el algoritmo de análisis
7. Declarar la variable **testData** y asignar el valor etiquetado para el algoritmo de análisis
8. Declarar la variable **numFeatures**.
9. Imprimir datos de prueba
10. Crear el modelo **model**
11. Entrenar el modelo
12. Realizar predicción
13. Imprimir predicción

```

package ec.com.streaming.twitter
import org.apache.spark.SparkConf

/** Example of using streaming linear regression with stochastic gradient descent. */
object StreamingRegression {

  def main(args: Array[String]) {

    // Crear el contexto con ventana de 1 segundo
    val ssc = new StreamingContext("local[*]", "StreamingRegression", Seconds(1))

    setupLogging()

    //Flujo de datos para entrenamiento
    val trainingLines = ssc.socketTextStream("127.0.0.1", 9999, StorageLevel.MEMORY_AND_DISK_SER)

    // Flujo de datos para testing
    val testingLines = ssc.socketTextStream("127.0.0.1", 7777, StorageLevel.MEMORY_AND_DISK_SER)

    val trainingData = trainingLines.map(LabeledPoint.parse)
    val testData = testingLines.map(LabeledPoint.parse)

    trainingData.print()

    //Número de características dentro del flujo de datos
    val numFeatures = 1

    //Definición de la técnica para DStream Mining
    val model = new StreamingLinearRegressionWithSGD().setInitialWeights(Vectors.zeros(numFeatures))
    model.algorithm.setIntercept(true)

    //Entrenamiento del Modelo
    model.trainOn(trainingData)

    //Generación de la predicción con los datos para Testing
    model.predictOnValues(testData.map(lp => (lp.label, lp.features))).print()

    // Kick it off
    ssc.checkpoint("C:/checkpoint/")
    ssc.start()
    ssc.awaitTermination()
  }
}

```

Figura 16 Codificación del programa SPK-I

2.5 INTEGRACIÓN, PRUEBA Y VERIFICACIÓN DE COMPONENTES

2.5.1 Integración

El sistema SIS-I consta de cuatro componentes, los cuales están integrados tanto a nivel de software como de hardware. La integración está basada conforme a la arquitectura descrita en el documento. La integración de los componentes se cumplirá de acuerdo al siguiente diagrama de flujo:

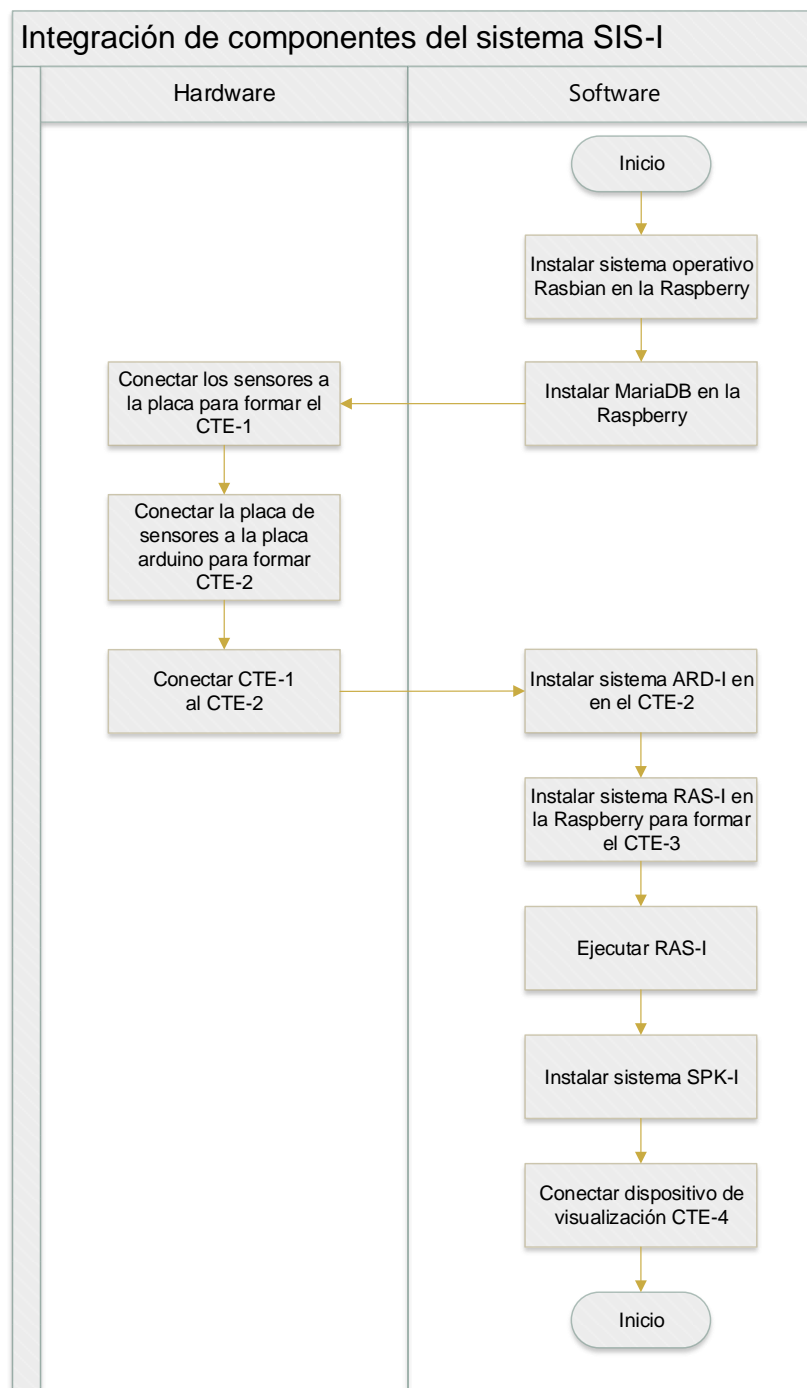


Figura 17 Flujograma Integración componentes

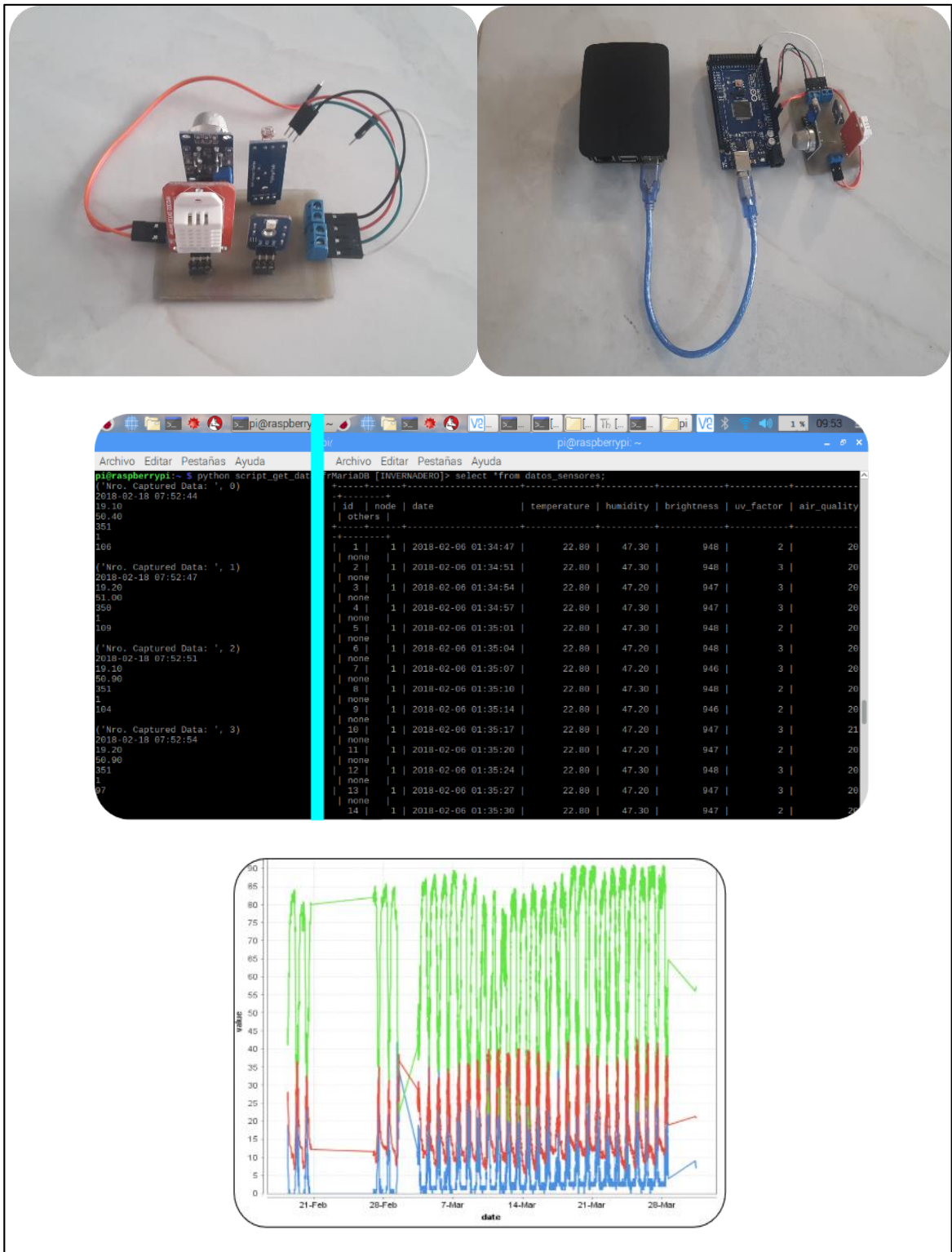


Figura 18. Integración de Componentes

2.5.2 Pruebas

Las pruebas aplicadas al sistema SIS-I, consisten comprobar el comportamiento operacional y el desempeño en conjunto del sistema, así como también, de cada uno de sus componentes individuales y funciones específicas que éste realiza, con la finalidad de verificar que el sistema cumpla con las especificaciones previamente establecidas.

2.5.2.1 Ítems A Probar

Componentes:

- CTE-1: Comprobar funcionamiento e integración de sensores, circuitos y software de control ARD-I.
- CTE-2: Comprobar funcionamiento de software de control RAS-I e integración con CTE-1.
- CTE-3: Comprobar funcionamiento de software SPK-I y software de almacenamiento.
- CTE-4: Comprobar funcionamiento y sincronización de visualización de los resultados de CTE-3.

Funciones:

- Captura automática de datos luego de producirse una apagado o reinicio.
- Captura de datos del ambiente con los sensores
- Obtención flujo de datos de los sensores
- Almacenamiento en la base de datos
- Conectividad de la red WSN
- Comportamiento de puertos/sockets para el envío de flujo de datos
- Visualización de resultados

2.5.2.2 Estrategia

En las diferentes pruebas se corroboró el cumplimiento de las especificaciones del SIS-I mediante el documento IEEE-830.

Para la prueba del componente CTE-1, se realizó la captura de datos de datos con cada sensor y el resultado fue comparado con valores medidos con otros dispositivos fiables. Se utilizó la hoja técnica (datasheet) de los sensores y placa arduino para comparar con el funcionamiento de los equipos y la unidad de medida de cada valor resultante.

Para la prueba del componente CTE-2, se utilizó datasheet de la raspberry para comparar con el funcionamiento del equipo, así como la estabilidad del sistema operativo, software instalado y conectividad.

Para la prueba del CTE-3, se realizó la obtención de datos con factores medioambientales cambiantes, y se ejecutó el software RAS-I para iniciar con las capturas. Con respecto al almacenamiento de datos, se verificó en la base de datos que existan las inserciones de las capturas realizadas, igualmente, se comparó el número total de registros almacenados con el número de capturas realizadas.

Para la prueba del CTE-4, se comparó los valores de los datos y resultados obtenidos con las gráficas generadas. Para la visualización del constante flujo de datos a través de los puertos, se utilizó la herramienta putty mediante conexión telnet.

2.5.2.3 Condiciones especiales

En la Tabla 11 se detallan ciertas consideraciones para que las pruebas pudieron ser suspendidas, repetidas o terminadas:

Tabla 11*Consideraciones especiales*

<i>Estado</i>	<i>Acontecimientos</i>
Suspendido	No existe conectividad con los dispositivos Los equipos sufren algún desperfecto
Repetido	Demora en tiempo de captura de datos establecido Existen ciertos datos que muestran posibles inconsistencias
Culminado	Los componentes no se pueden integrar Equipos sufran daños que no puedan ser reparados

CAPÍTULO III

CASO DE ESTUDIO:

“OBTENCIÓN, PROCESAMIENTO Y DATA STREAM MINING”

3.1 DESCRIPCIÓN GENERAL DEL CASO DE ESTUDIO

3.1.1 Resumen de la situación

Las redes de sensores y herramientas informáticas de análisis de datos, son herramientas de apoyo para la toma de decisiones, principalmente en la actividad florícola. Una estructura basada en una red de sensores automatizada que permita la recolección de datos meteorológicos en tiempo real, brindan un monitoreo constante que puede beneficiar en gran medida a la producción de flores en el Ecuador.

3.2 CARACTERÍSTICAS DEL ÁREA DE IMPLEMENTACIÓN

3.2.1 Ubicación

El sistema SIS-I fue instalado en el invernadero de la sede IASA - I de la Universidad de las Fuerzas Armadas - ESPE. La sede se localiza en la provincia de pichincha, cantón Rumiñahui, sector valle de los chillos en la Hacienda “El Prado” cuya descripción ecológica se describe en la tabla 12:

Tabla 12

Descripción del lugar del caso de estudio

Longitud:	78° 24' 44" E
Latitud:	0° 23' 20" S
Altitud:	2748 m s.n.m.
Región latitudinal:	Templada
Clasificación bioclimática:	Húmedo-Temperado



Figure 19 Ubicación IASA

3.2.2 Descripción del lugar

El invernadero comprende un área de 2000 m², en el cual se levanta una estructura de acero y recubierto en sus alrededores por plástico. Esta estructura corresponde al área en el cual estará instalado el sistema SIS-I y en donde se medirá los factores medioambientales que se producen al interior del mismo.

Una característica importante al interior del invernadero, radica en que es un ambiente controlado, ya que la estructura permite tener temperaturas diferentes a las experimentadas en el exterior ya que estas pueden ser aumentadas o reducidas conforme se requiera.

El invernadero aloja a cuatro variedades de flores: Wow, Sonrisa, Luxor y Night Fever (para el caso de estudio están consideradas la medición de treinta flores de cada especie).

En la figura 20 se muestra el lugar donde se instalará el sistema:



Figura 20 Invernadero

3.3 DISTRIBUCIÓN Y ORGANIZACIÓN DE LOS DISPOSITIVOS

El lugar de operación del sistema SIS-I ha sido colocado para que cubra las cuatro variedades que van a ser medidas.

En la figura 21 y figura 2 2se puede apreciar la ubicación del equipo en el área del invernadero y la colocación desde varios ángulos in situ:

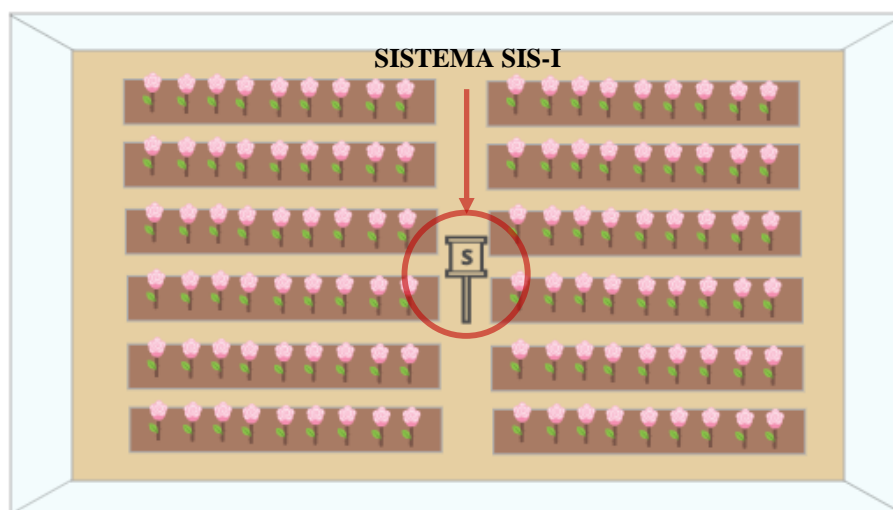


Figura 21 Ubicación equipo

El sistema se encuentra en el centro del invernadero, el cual captará datos meteorológicos que inciden en el entorno de crecimiento de las flores, y que, para el caso de estudio presentado, lleva el control para el desarrollo de cuatro variedades, treinta por cada una de ellas, dando un total de 120 flores.



Figura 22 Ubicación equipo invernadero

El sistema SIS-I, fue colocado en una estructura metálica semi cubierta con ranuras de gran tamaño con la finalidad que ningún elemento externo pueda influir en los datos que se van a tomar. De la misma manera, el componente CTE-1 se encuentra en el exterior para que los sensores capten los factores reales del lugar y tampoco se vean alterados por otro tipo de factores.

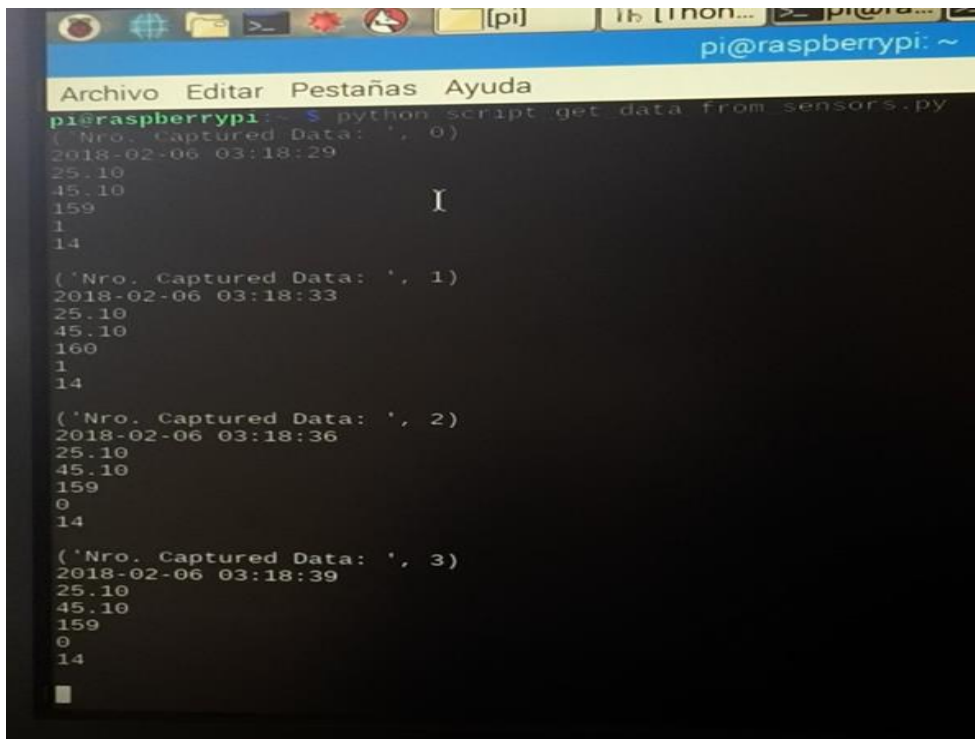
3.4 INICIO DE OPERACIÓN DEL SISTEMA SIS-I

3.4.1 Activación del sistema para la captura de datos

El sistema SIS-I inicia verificando que cada sensor y componente se encuentre correctamente conectado y tengan la alimentación eléctrica correspondiente. Una vez cumplido la verificación, se procede con el encendido del equipo y se verifica la conectividad del dispositivo a la red destinada para la recepción de datos denominada “invernadero”. La conexión a la red se realiza automáticamente debido a las configuraciones previas detalladas en el capítulo anterior.

Luego de cumplir con los pasos anteriores, se debe utilizar el comando que inicia con el proceso de captura de datos a través de la ejecución de un shell desarrollado en lenguaje Python con el nombre de “script_get_data_from_sensors.py” el cual activa e inicia con la captura y generación del flujo de datos.

En la figura 23 se visualiza el inicio de la toma de datos



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi: ~$ python script_get_data_from_sensors.py  
( 'Nro. Captured Data: ', 0)  
2018-02-06 03:18:29  
25.10  
45.10  
159  
1  
14  
  
( 'Nro. Captured Data: ', 1)  
2018-02-06 03:18:33  
25.10  
45.10  
160  
1  
14  
  
( 'Nro. Captured Data: ', 2)  
2018-02-06 03:18:36  
25.10  
45.10  
159  
0  
14  
  
( 'Nro. Captured Data: ', 3)  
2018-02-06 03:18:39  
25.10  
45.10  
159  
0  
14
```

Figura 23 Toma de datos

3.4.2 Procesamiento del flujo de datos

Tabla 13*Información de captura de datos*

Fecha y Hora de Captura	Tiempo de min	Datos esperados por día	Días empleados	Datos esperado por periodo	Datos obtenidos	Datos perdidos	Estimado días perdidos
2018-03-03 11:05	10	144	5	720	719	1	0.006
2018-03-03 11:05	3	480	57	27360	24634	2727	5.68

El proceso para la toma de datos y generación del flujo no solo se orienta al a simples capturas o envió de sus datos por una red. El proceso que conlleva una vez generado el flujo de datos, es obtener todo ese flujo que representa una cadena de caracteres en cualquier orden, cualquier formato, y que para este caso de estudio es una cadena de caracteres separada por comas que no especifica ningún parámetro de medida y que se muestra en la figura 24:

Cadena de caracteres generada en el flujo								
610,1,2018-03-06 00:09:36,12.50,84.20,1,960,38,none								
Identificación de los valores de la cadena								
610	1	2018-03-06 00:09:36	12.50	84.20	1	960	38	none
Id	Nodo	Hora y Fecha	Temperatura	Humedad	Factor Uv	Índice de Luz	Calidad de Aire	Observaciones
Expresión regular generada								
= (id, date, node, temperature, humidity, brightness, uv_factor, air_quality, others)								
Expresión regular realizada la transformación para almacenamiento en la base de datos								
=(' ',date, node, decimal.Decimal(temperature), decimal.Decimal(humidity), bright, factorUv, airQ, others)								

Figura 24 Cadena de caracteres generada en el flujo

Una vez generada la expresión regular, se genera el almacenamiento de los datos con el objetivo de realizar las respectivas pruebas, verificaciones y validaciones del sistema y posteriormente generar el análisis respectivo mediante técnicas de Data Stream Mining.

En la figura 25 se muestra los datos almacenados en CTE-2:

id	node	date	temperature	humidity	brightness	uv_factor	air_quality	others
610	1	2018-03-06 00:09:36	12.50	84.20	960	1	38	none
611	1	2018-03-06 00:19:37	12.50	82.00	959	1	38	none
612	1	2018-03-06 00:29:38	11.80	83.50	960	1	37	none
613	1	2018-03-06 00:39:38	11.20	86.40	959	1	38	none
614	1	2018-03-06 00:49:39	11.40	84.20	959	1	38	none
615	1	2018-03-06 00:59:39	10.90	87.20	961	1	38	none
616	1	2018-03-06 01:09:40	11.40	87.10	961	1	37	none
617	1	2018-03-06 01:19:41	11.60	87.00	958	1	37	none

Figura 25 Datos almacenados en CTE-2

Para el análisis los datos son enviados mediante el socket 9999 utilizando el protocolo TCP-IP, hacia la estación base donde. Los datos como puede apreciar en la figura 26, los datos son enviados para la simulación por un archivo de texto denominado “GreenHouseDataFlow.txt” apoyado en la herramienta “ncat.exe” y simultáneamente en la figura 27 se aprecia el flujo a través del puerto anteriormente descrito.

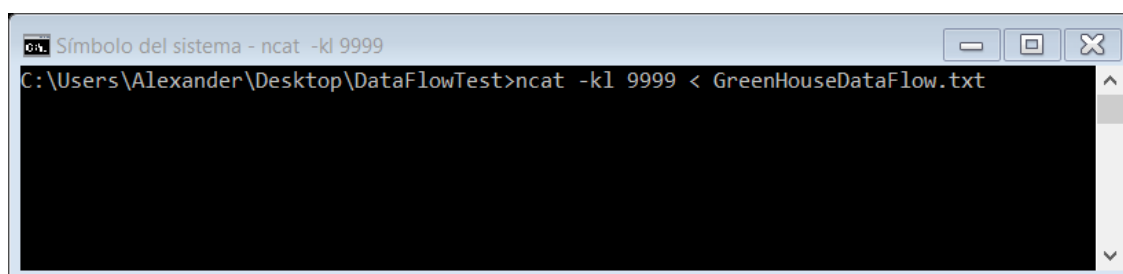


Figura 26 Datos enviados mediante socket

```

25083,1,"2018-06-07 14:35:56",32.90,30.90,17,17,19,"none"
25084,1,"2018-06-07 14:38:56",33.80,29.80,13,22,18,"none"
25085,1,"2018-06-07 14:41:56",32.10,29.10,15,18,19,"none"
25086,1,"2018-06-07 14:44:57",31.30,31.00,12,17,19,"none"
25087,1,"2018-06-07 14:47:57",32.20,33.40,16,18,20,"none"
25088,1,"2018-06-07 14:50:58",32.00,32.00,14,18,18,"none"
25089,1,"2018-06-07 14:53:58",32.00,28.40,16,18,19,"none"
25090,1,"2018-06-07 14:56:58",32.40,30.10,16,18,20,"none"
25091,1,"2018-06-07 14:59:59",32.40,32.60,12,18,19,"none"
25092,1,"2018-06-07 15:02:59",32.20,28.70,14,19,19,"none"
25093,1,"2018-06-07 15:06:00",31.30,30.70,15,19,19,"none"
25094,1,"2018-06-07 15:09:00",32.00,32.60,15,20,19,"none"
25095,1,"2018-06-07 15:12:00",32.30,31.60,13,20,20,"none"
25096,1,"2018-06-07 15:15:01",...

```

Figura 27 Flujo Generado

3.4.3 Data Stream Mining

El componente de Spark, denominado Spark Streaming, es la encargada de adquirir el gran conjunto de datos provenientes de los sensores, en este caso a través del socket 9999, y que, mediante la codificación de ciertos scripts controlarán cómo este conjunto de datos será transformado y servirá de salida para otros sistemas.

Es importante considerar que tenemos un flujo constante de datos proveniente de los sensores que son parte de un dispositivo IoT, que requiere ser procesado según como vaya ingresando cada dato y mantener actualizado los resultados una y otra vez, para lo cual su análisis se realiza a una pequeña sección a la vez y básicamente para siempre.

El uso de Spark Streaming como parte del SIS-I, permite aprovechar la capacidad de procesamiento de datos ya que esta herramienta apunta al paradigma del Internet de la Cosas.

Aquí se emplea un concepto denominado Resilient Distributed Datasets (RDD) cuyo significado indica que es un conjunto de datos distribuidos resilientes. Esto se refiere a que los conjuntos de datos pueden ser distribuidos y ejecutados en un cluster y que es resistente a fallas lo que se traduce a el grupo de datos pueda ser recuperado. Desde otro punto de vista, un RDD representa a un conjunto de datos que no puede caber en una sola máquina, pero lógicamente se refiere a un solo objeto al momento de codificar en el sistema.

Para el realizar el análisis de flujo de datos es necesario entender la manera global como Spark Streaming trabaja en el SIS-I y principalmente lo que se refiere y que se detalla en la figura 28:

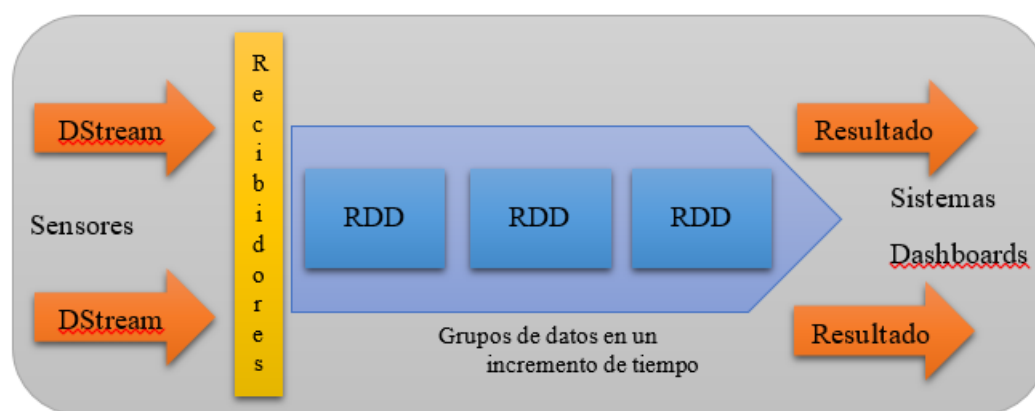


Figura 28 Spark Streaming

La implementación del RDD en el SIS-I inicia con su instanciación:

```
// Configuración de un Spark stream context denominado "GreenHouse" que se ejecuta localmente
val rdd = new StreamingContext("local[*]", "GreenHouse", Seconds(1))
```

Figura 29 Implementación del RDD.

En este caso la instanciación del RDD contempla tres argumentos (“**nodo master**”, “**nombre de la instancia**”, “**tamaño de la porción de datos**”). El Nodo Master, es el lugar donde se realizará el análisis, en el caso del código mostrado en la figura 29, se lo realiza localmente y el

asterisco (*) indica que se utilizará todos los procesadores que posea computador. El nombre de la instancia es el nombre que recibirá el RDD. El tamaño de porción de datos almacena los datos obtenidos en un lapso de tiempo definido, en este caso cada segundo.

La descripción de datos generados y obtenidos en el Streaming se encuentran detallados en la tabla 14:

Tabla 14

Descripción de datos generados y obtenidos

Batch	Promedio de Datos / Batch	Total de Datos Procesados	Total Rdd's generados	Tiempo total promedio empleado
1 seg	630	25953	42	41.195 seg

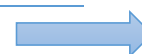
Una vez obtenido el Rdd se puede realizar dos acciones, la primera es transformar la data recibida y la segunda es realizar acciones sobre la data. Entre las operaciones más importantes que se pueden realizar tenemos las siguientes:

Tabla 15

Operaciones importantes en Spark Streaming

Operaciones		
Ítem	Nombre	Descripción
1	map	Aplicar alguna función a cada elemento en el conjunto de datos.
2	flatMap	Toma un Rdd con 'n' número de líneas de información y crea un nuevo Rdd con 'm' número de líneas de información
3	filter	Extrae información que se desea mantener para crear un rdd mucho más pequeño y más fácil de manejar

CONTINÚA



4	distinct	Muestra distintas filas de datos del Rdd
5	sample	Toma una muestra aleatoria de datos para realizar alguna experimentación
Acciones		
1	collect	Toma una instantánea de un Rdd y lo traduce una estructura de datos convencional
2	count	Indica el número de filas existente en el Rdd
3	countByValue	Cuenta los valores únicos existentes en el Rdd si disponen de datos Clave/Valor
4	take	Toma los primeros 'x' números de resultados definidos por el usuario
5	reduce	Combina elementos de un Rdd basados en la clave

Una vez considerado el tipo de datos que se va analizar, se requiere entrenar el modelo de regresión lineal con gradiente descendiente en un flujo de datos, y realizar las predicciones en otro flujo de datos. Las filas del flujo de datos deben estar etiquetadas de la siguiente manera:

$$(y, [x_1, x_2, x_3, \dots, \dots, x_n])$$

Donde n es el número de características. n debe ser el mismo para entrenamiento como para testing. A medida que siga ingresando flujos de datos para entrenamiento, el modelo se actualizará continuamente y de la misma manera a medida que ingrese un flujo de datos para testing, se observará las predicciones del modelo actual.

La figura 30 muestra el ingreso del flujo de datos en el sistema SIS-I, específicamente para el entrenamiento del modelo a través del puerto 9999 y para realizar el testing a través del puerto 7777 el cual se ve especificado en el siguiente extracto de código:

```
//Flujo de datos para entrenamiento
val trainingLines = ssc.socketTextStream("127.0.0.1", 9999, StorageLevel.MEMORY_AND_DISK_SER)
val trainingData = trainingLines.map(LabeledPoint.parse)

// Flujo de datos para testing
val testingLines = ssc.socketTextStream("127.0.0.1", 7777, StorageLevel.MEMORY_AND_DISK_SER)
val testData = testingLines.map(LabeledPoint.parse)
```

Figura 30 Ingreso del flujo de datos

```
//Número de características dentro del flujo de datos
val numFeatures = 1

//Definición de la técnica para DStream Mining
val model = new StreamingLinearRegressionWithSGD().setInitialWeights(Vectors.zeros(numFeatures))
model.algorithm.setIntercept(true)

//Entrenamiento del Modelo
model.trainOn(trainingData)

//Generación de la predicción con los datos para Testing
model.predictOnValues(testData.map(lp => (lp.label, lp.features))).print(20)
```

Figura 31 Definición de la técnica de análisis

El etiquetado para el modelo del SIS-I, se ha considera la temperatura y la humedad, para la cual realizará predicciones teniendo en cuenta que la temperatura se considerará como label (etiqueta) y la humedad será considerada como una feature (característica) expresada de la siguiente manera (*label*, [*feature*]), y ya en el flujo puede ser observada como se muestra en la figura 32:

```
Time: 1547726880000 ms
-----
(12.5, [84.2])
(12.5, [82.0])
(11.8, [83.5])
(11.2, [86.4])
(11.4, [84.2])
(10.9, [87.2])
(11.4, [87.1])
(11.6, [87.0])
(11.3, [85.9])
(11.1, [85.6])
(11.1, [86.8])
(11.3, [86.4])
(11.8, [84.1])
(12.1, [82.7])
(11.6, [82.7])
(11.1, [84.1])
(10.9, [84.4])
(10.5, [84.7])
(10.4, [83.2])
(9.4, [88.5])
...
```

Figura 32 Flujo de entrenamiento

Los datos que se muestran en la figura 32 son los datos de entrenamiento que están ingresando al modelo, y que una vez que ingrese el flujo de datos para testing genera predicciones como se expresa en la figura 33:

```

-----
Time: 1547726880000 ms
-----
(12.5, -5.1572617953210116E101)
(12.5, -5.022533920560182E101)
(11.8, -5.1143938351698385E101)
(11.2, -5.2919896700818406E101)
(11.4, -5.1572617953210116E101)
(10.9, -5.3409816245403234E101)
(11.4, -5.3348576302330124E101)
(11.6, -5.3287336359257027E101)
(11.3, -5.261369698545288E101)
(11.1, -5.2429977156233565E101)
(11.1, -5.3164856473110813E101)
(11.3, -5.2919896700818406E101)
(11.8, -5.1511378010137006E101)
(12.1, -5.065401880711355E101)
(11.6, -5.065401880711355E101)
(11.1, -5.1511378010137006E101)
(10.9, -5.169509783935632E101)
(10.5, -5.1878817668575634E101)
(10.4, -5.0960218522479075E101)
(9.4, -5.4205935505353585E101)
...

```

Figura 33 Flujo de testing aplicado el algoritmo

El algoritmo aplicado para la predicción de datos se denomina Regresión Lineal con Gradiente Descendiente Estocástica. Este, es un algoritmo iterativo de optimización y “es una de las técnicas supervisadas dentro del machine learning” (Jupudi, 2016), y determina la función de coste en el proceso de aprendizaje. Partiendo del de aprendizaje supervisado, este parte de un conjunto de datos considerados como datos de entrenamiento, y que de los cuales se van aprender características para realizar predicciones. Por lo tanto, este tipo de aprendizaje busca generar predicciones para un nuevo conjunto de datos. Es importante, para poder validar el modelo, aplicar un conjunto de datos denominado datos de prueba que permiten validar el modelo.

Otra característica importante para el uso de la técnica, es que permite actualizar pesos para obtener las predicciones basadas en cada ejemplo que ingresa y, “la característica particular de este

algoritmo es que permite ser aplicado cuando el conjunto de datos de entrenamiento es demasiado largo.” (Stanford University, 2018).

3.5 VISUALIZACIÓN DEL STREAMING DE DATOS

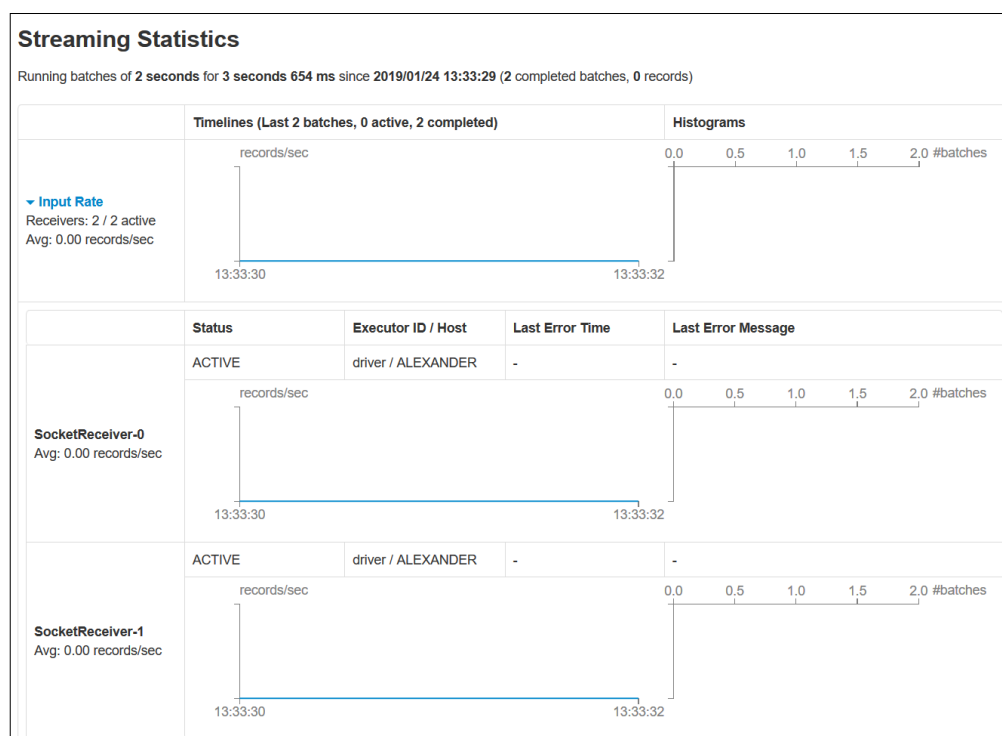


Figura 34 Inicio del proceso de streaming en Spark- Sockets sin datos

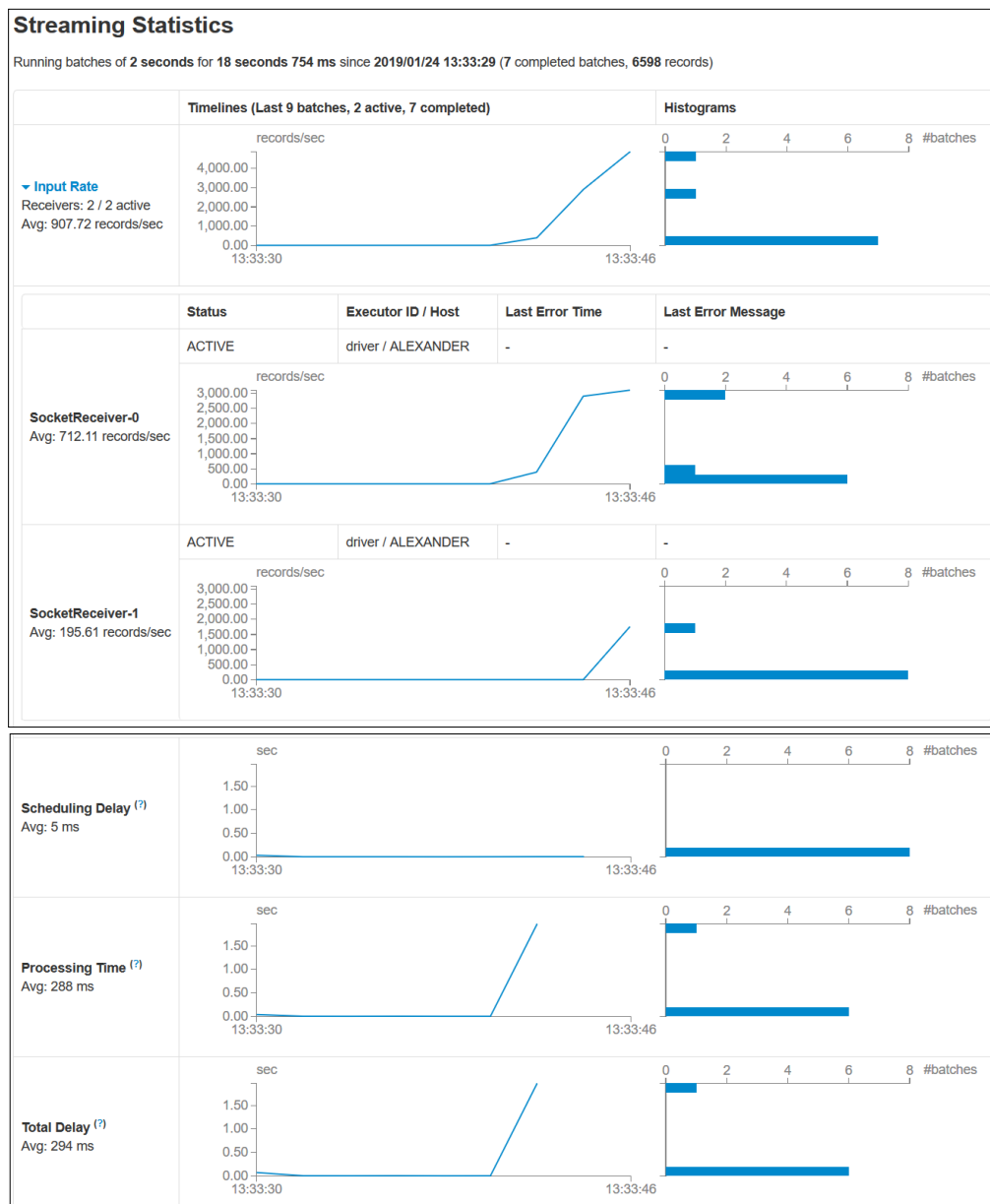


Figura 35 Recepción de Sockets y procesamiento de datos

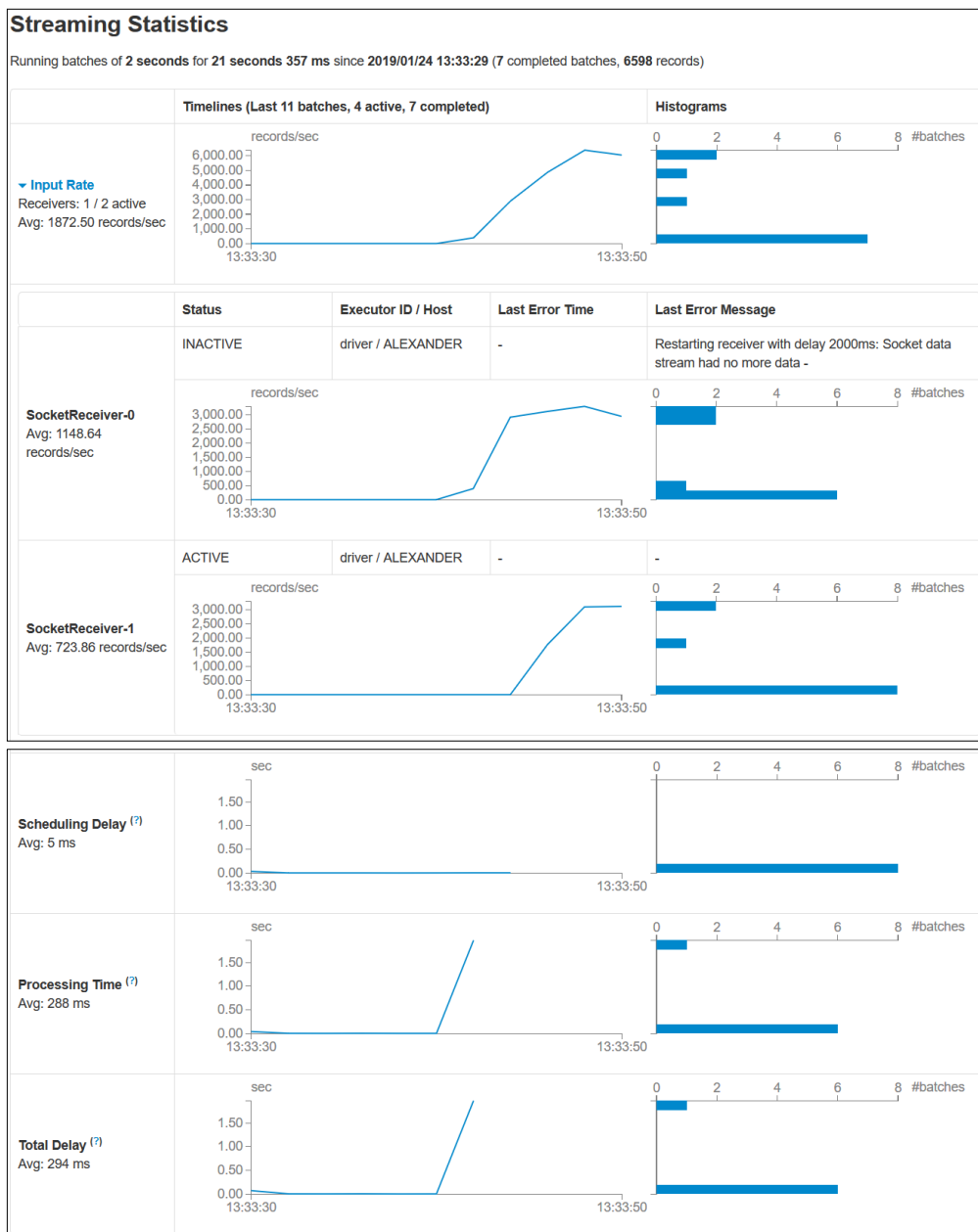


Figura 36 Procesamiento en tiempo real de datos

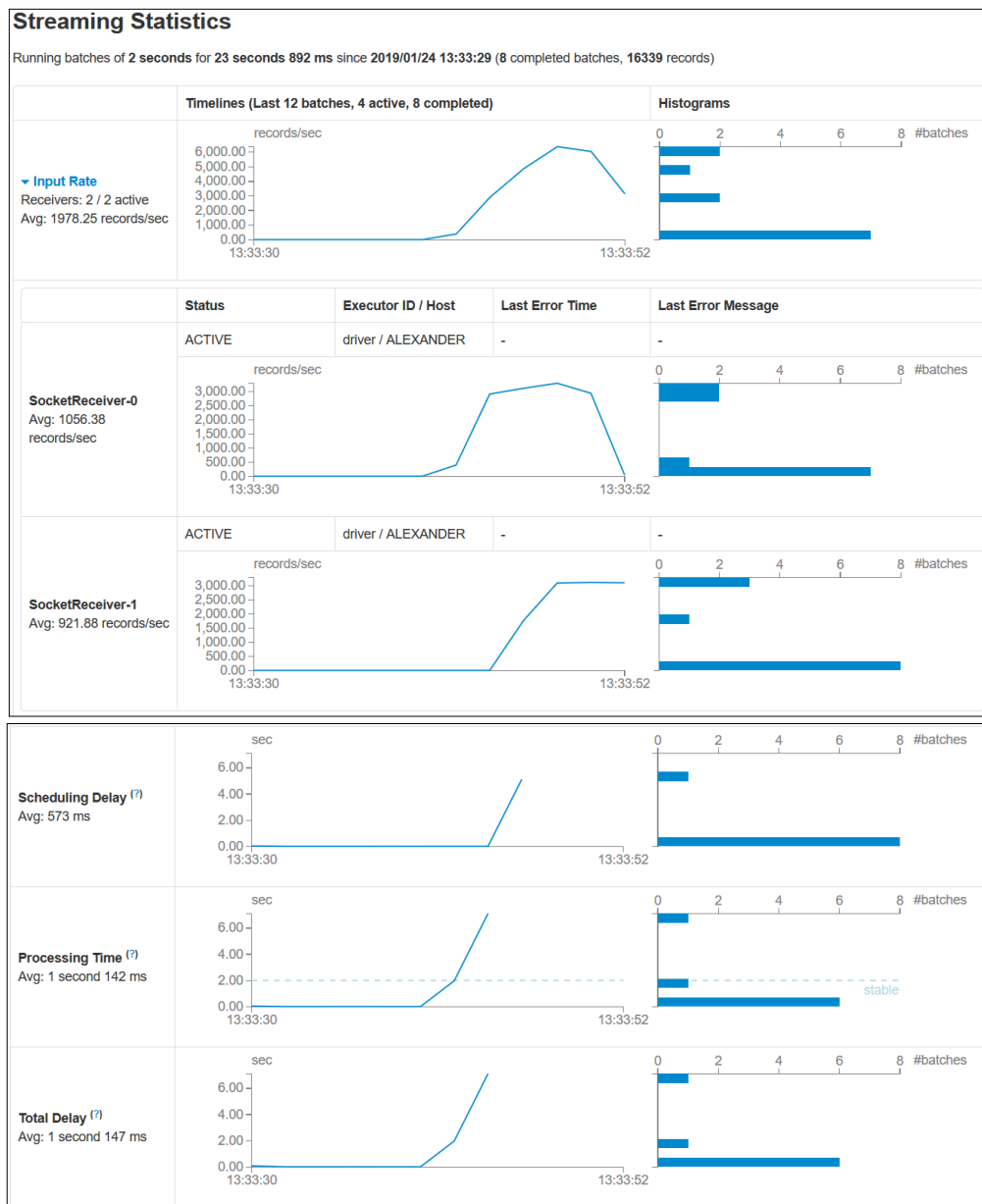


Figura 37 Finalización de procesamiento

3.6 VALIDACIÓN Y VERIFICACIÓN DEL SISTEMA SIS-I

La verificación “Busca comprobar que el sistema cumple con los requerimientos especificados (funcionales y no funcionales)” (Sommerville, 2011) mientras que la validación “busca comprobar que el software hace lo que el usuario espera.” (Cano, Calafate, Malumbres, & Manzoni, 2003) (Sommerville, 2011). Por lo tanto, para verificar y validar el sistema SIS-I realizó una revisión de los documentos involucrados en el desarrollo del sistema, y que se mencionan en la Tabla 16.

Tabla 16

Revisión de la documentación

Nro.	Documento
1	Documento de especificación de requerimientos IEEE-830
2	Datasheet de los sensores
3	Datasheet de las Raspberry Pi y Arduino

En el proceso de verificación y validación, ha sido revisado y comprobado los requerimientos funcionales y no funcionales en todas las etapas de desarrollo, donde se apreció que el sistema SIS-I cumple cada uno de los requerimientos y las especificaciones técnicas establecidas previamente durante todas las etapas de desarrollo.

Con respecto al análisis, de acuerdo a la tabla 16 de los requisitos para atender el paradigma de data stream, se ha revisado que cumpla con aquellas características descrita en la tabla 17:

Tabla 17*Requisitos del paradigma de Data Stream Mining*

Item	Requerimiento	Acción del SIS-I
1	Procesar un ejemplo a la vez e inspeccionarlo máximo una sola vez	Los datos son recibidos, transformados a un RDD y utilizados para el entrenamiento del modelo hasta salir del proceso
2	Usar una cantidad de memoria limitada	El algoritmo se enfoca en la optimización y permite ejecutarse en memoria tanto para acciones estadísticas como almacenamiento del modelo.
3	Trabajar en una cantidad de tiempo limitada	La acción de toma y análisis de datos se lo realiza en aproximadamente 41 segundos con un total de 25353 recibidos y procesados.
4	Estar listo para predecir en cualquier punto	El algoritmo de regresión lineal en streaming con descenso de gradiente estocástica permite evaluar y generar las predicciones al momento que ingresan los datos de entrenamiento y los datos de prueba.

3.7 MANTENIMIENTO DEL SISTEMA SIS-I

El software al ser un elemento importante dentro del proceso de análisis, requiere que reciba mantenimiento con la finalidad que sigan cubriendo la demanda que se requiera en el invernadero y cumplan con la funcionalidad o a su vez se les otorgue una mejor funcionalidad para su desempeño.

El mantenimiento se realiza mediante el personal técnico asignado y estipulado en el IEEE-830 de especificación de requerimientos del sistema SIS-I. Para este sistema debido a los componentes que maneja, necesito una atención tanto para la parte de software y principalmente para la de hardware. Para esto se ha determinado cierto puntos clave a los cuales se debe dar atención para el sistema SIS-I opere adecuadamente.

Tabla 18*Puntos clave para el mantenimiento del SIS-I*

Hardware	Software
Verificar componente CTE-1: <ul style="list-style-type: none"> • Sensores • Placa de circuitos 	Verificar componente CTE-3: <ul style="list-style-type: none"> • Funcionalidad del Sistema ARD-I • Funcionalidad del Sistema RAS-I • Funcionalidad del Sistema SPK-I
Verificar componente CTE-2: <ul style="list-style-type: none"> • Circuito Arduino • Raspberry Pi 	Verificar componente CTE-4: <ul style="list-style-type: none"> • Validar datos resultantes
Verificar fuente de alimentación y cableado	Verificar datos resultantes con datos con resultados esperados brindados por un experto en el área florícola

Estos puntos detallados anteriormente, responden a las partes importantes del sistema SIS-I, y que, en base a su desempeño pueden seguir operando de la misma manera o a su vez, se puede agregar nuevas funcionalidades. Por otro lado, si se requiere mejorar sustancialmente el sistema, se puede considerar un proceso de reingeniería.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

En el presente trabajo, se planteó un sistema denominado SIS-I, el cual está enfocado en el paradigma del Internet de la Cosas y permite la automatización en la toma de datos y sus análisis en tiempo real dentro de un entorno controlado (invernadero).

Se propuso una arquitectura básica de los dispositivos que intervienen, clasificándolos en cuatro componentes de acuerdo a su funcionalidad dentro del sistema y a su vez, facilitando su integración y su usabilidad para que éste pueda ser utilizado para el beneficio del sector florícola del Ecuador.

El conocimiento de tecnologías actuales, permitió implementar en el sistema SIS-I, técnicas de análisis en tiempo real mediante el uso del algoritmo de Regresión Lineal con Descenso de Gradiente Estocástica orientado a la Minería en Flujos de Datos o Data Stream Mining, logrando generar un modelo predictivo aplicable a las variables requeridas en el ámbito de la producción florícola.

El sistema desarrollado permite conjugar varias disciplinas y conocimientos técnicos, iniciando por el uso y configuración de dispositivos electrónicos, circuitos y diseño de una Wireless Sensor Network, luego pasando al uso e integración de varios lenguajes de programación para la adquisición, procesamiento, análisis y visualización de los resultados obtenidos y finalmente la consolidación e implementación de un sistema integrado autónomo basado en conceptos teórico y técnicos proporcionados por profesionales en el área de sistemas como de floricultura.

Mediante este sistema se busca brindar una alternativa de bajo costo, capaz de facilitar el proceso de medición de los factores medioambientales (abióticos) que intervienen en la producción de rosas brindando un monitoreo constante y objetivo de las mismo, y principalmente en tiempo real para una adecuada toma de decisiones.

4.2 RECOMENDACIONES

El sistema SIS-I propuesto en este trabajo dispone de componentes y de funcionalidades que a futuro pueden ser mejoradas. Por lo tanto, es importante tener en cuenta que este sistema está desarrollado bajo los lineamientos de una metodología tradicional denominada modelo en “V” que esta a su vez permite la mejora continua de este sistema.

Considerando lo anterior, es necesario atender los nuevos requerimientos que se presenten a futuro, con el objetivo de dotar de mejores dispositivos o mejores funcionalidades que satisfagan las necesidades del usuario final sin descuidar la premisa de bajo costo.

La tecnología es un ámbito que está en constante evolución, día a día nacen o se crean nuevos lenguajes o algoritmos que optimizan y disminuyen la velocidad de procesamiento de datos, por lo que es importante conocer estas tecnologías para que a futuro puedan ser implementadas para la mejora de este sistema o cualquiera de sus componentes.

Sin duda, los datos ganan mayor importancia en la sociedad, obligando a que las técnicas de extracción de conocimiento y sus algoritmos sean cada vez más efectivos, permitiendo a futuro que el sistema SIS-I pueda incorporar alguno de ellos, de tal manera que contraste la información generada para una mejor toma de decisiones.

Este trabajo representa un punto de partida hacia varios ámbitos, no solo la floricultura; por lo que, requiere una atención especial y un trabajo interdisciplinario que conjugue conocimientos

teóricos y prácticos no solo en sistemas o electrónica, sino al hecho de consolidar sistemas inteligentes que vayan de la mano del usuario, el hardware y el software, ya sea en ámbito educativo, militar, industrial, entre otras.

REFERENCIAS BIBLIOGRÁFICAS

- Abarca, P. (5 de Mayo de 2008). *Asociación de la Industria Eléctrica-Electrónica, AIE*. Obtenido de <http://www.aie.cl/files/file/comites/ca/abc/sistemas-de-control-automatico.pdf>
- Akyildiz, I. F., & Vuran, M. C. (2010). *Wireless Sensor Networks*. John Wiley & Sons Ltd.
- Apache Software Foundation. (24 de Enero de 2019). *Apache Spark*. Obtenido de <https://spark.apache.org/>
- Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2011). *Data Stream Mining*. Open Source Edition.
- Cano, J. C., Calafate, C., Malumbres, M., & Manzoni, P. (2003). *Redes Inalámbricas Ad Hoc como Tecnología Soporte para la Computación Ubicua*. 10.
- Evans, D., & Grupo de Soluciones Empresariales para Internet, I. C. (2011). *Internet of Things*.
- Jupudi, L. (2016). Stochastic Gradient Descent using Linear Regression with Python. *International Journal of Advanced Engineering Research and Applications (IJA-ERA)*.
- Nerion Networks. (05 de 01 de 2019). *Soporte Nerion*. Obtenido de Que es MariaDB y mejoras sobre MySQL: <https://www.nerion.es/soporte/tutoriales/que-es-mariadb-y-mejoras-sobre-mysql/>
- Perez, A., Berreteaga, O., Ruiz, A., Urkidi, A., & Perez, J. (2006). Una metodología para el desarrollo de hardware y software embebidos en sistemas críticos de seguridad. *Sistemas, cibernética e informática, III*.
- Pressman, R. (2010). *Ingeniería del Software: Un enfoque práctico*. México: McGraw-Hill.
- Reina, J. L. (1 de Enero de 2018). Modelos lineales.
- Shaw, M., & Garlan, D. (1995). *Formulations and Formalisms in Software Architecture*.

Sommerville, I. (2011). *Software Engineering*. Massachusetts: Pearson.

Stanford University. (22 de Octubre de 2018). *Supervised Learning*. Obtenido de CS229: Machine Learning: <http://cs229.stanford.edu/notes/cs229-notes1.pdf>

US Department of Transportation . (Octubre de 2005). *Intelligent Transportation Systems*. Obtenido de Clarus Concept of Operations: https://web.archive.org/web/20090705102900/http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/14158.htm

Vazhnov, A. (2016). *La Red de Todo*. Obtenido de SCRIBD: <https://es.scribd.com/read/301538618/La-Red-de-Todo-Internet-de-las-Cosas-y-el-Futuro-de-la-Economia-Conectada>