

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Tesis previa a la obtención del título de:  
INGENIERO E INGENIERA DE SISTEMAS**

**TEMA:**

**DESARROLLO DE UN SISTEMA DE MONITORIZACIÓN DE AMENAZAS FÍSICAS CON SISTEMAS EMBEBIDOS Y RED DE SENSORES (WNS), PARA EL DATA CENTER DEL LABORATORIO DE SERVIDORES DE LA CARRERA DE COMPUTACIÓN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA CAMPUS SUR, MEDIANTE UNA APLICACIÓN PARA ANDROID.**

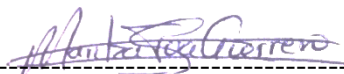
**AUTORES:  
MIGUEL ÁNGEL ROJAS BRAVO  
MARITZA SOFÍA TUA GUERRERO**

**TUTOR:  
MANUEL RAFAEL JAYA DUCHE**

**Quito, agosto de 2018**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros: MARITZA SOFÍA TUA GUERRERO, con documento de identificación N° 1721161899 y MIGUEL ÁNGEL ROJAS BRAVO, con documento de identificación N° 1714893664, manifestamos que voluntariamente otorgamos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “DESARROLLO DE UN SISTEMA DE MONITORIZACIÓN DE AMENAZAS FÍSICAS CON SISTEMAS EMBEBIDOS Y RED DE SENSORES (WNS), PARA EL DATA CENTER DEL LABORATORIO DE SERVIDORES DE LA CARRERA DE COMPUTACIÓN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA CAMPUS SUR, MEDIANTE UNA APLICACIÓN PARA ANDROID”, mismo que ha sido desarrollado para optar por el título de INGENIERO E INGENIERA DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



MARITZA SOFÍA  
TUA GUERRERO  
CC: 1721161899



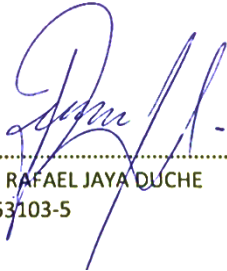
MIGUEL ÁNGEL  
ROJAS BRAVO  
CC: 1714893664

Quito, agosto de 2018

## DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR

Yo, Manuel Rafael Jaya Duche, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DESARROLLO DE UN SISTEMA DE MONITORIZACIÓN DE AMENAZAS FÍSICAS CON SISTEMAS EMBEBIDOS Y RED DE SENSORES (WNS), PARA EL DATA CENTER DEL LABORATORIO DE SERVIDORES DE LA CARRERA DE COMPUTACIÓN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA CAMPUS SUR, MEDIANTE UNA APLICACIÓN PARA ANDROID”, realizado por: Maritza Sofía Tua Guerrero y Miguel Ángel Rojas Bravo , obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto del 2018



.....  
MANUEL RAFAEL JAYA DUCHE  
CI: 171063103-5

## **DEDICATORIA**

A mi madre Rocío por sus consejos, apoyo incondicional, sus valores, su motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor y amistad, a mis hermanos por su ejemplo y valentía los amo Raquel, Daniel y Laksmi, a mi padre José por su incentivo en mis estudios y en el deporte, a mis sobrinas Anee y Raquelita por los bellos momentos compartidos, a mi amiga del alma Dianita, a mis amigos de toda la vida mi familia.

Todo este trabajo ha sido posible gracias a ellos.

Maritza Sofía Tua Guerrero

Gracias a mi madre Edy, por su dedicación, entrega, enseñanzas y muestras de confianza que siempre tiene conmigo, a mi padre Ángel, que juntos me ayudaron en mi formación como persona y profesional, a mis hermanos Martín y Darío que son mis mejores amigos en mi paso por esta vida y las siguientes, gracias a ti Óscar por ser mi apoyo incondicional, por ser mi modelo a seguir, por tus palabras de aliento y tus enseñanzas de vida que las aplico en mi día a día, a mis buenos amigos que siempre están a mi lado. Sería imposible llegar a este punto sin la ayuda de cada uno de ustedes. Simplemente un millón gracias

Miguel Ángel Rojas Bravo

## **AGRADECIMIENTOS**

A nuestra querida Universidad Politécnica Salesiana Campus Sur, por las enseñanzas morales y académicas.

A nuestro tutor Ing. Manuel Rafael Jaya Duche, por su guía y paciencia brindada en el desarrollo de nuestro proyecto.

A cada docente que nos impartió sus conocimientos y vivencias, preparándonos para la vida.

Miguel Ángel Rojas Bravo

Maritza Sofía Tua Guerrero

## ÍNDICE

INTRODUCCIÓN .....	1
Antecedentes .....	1
Planteamiento del problema .....	2
Justificación.....	2
Objetivos .....	3
Objetivo general .....	3
Objetivos específicos .....	3
Marco metodológico .....	3
CAPÍTULO 1 .....	4
MARCO TEÓRICO.....	4
Centro de procesamiento de Datos.....	4
1.1 Diseño .....	5
1.2 Estándar Uptime Institute en Data Center.....	6
1.3 Sistemas de Monitorización .....	6
1.4 Sistemas embebidos .....	7
1.5 Arduino .....	9
1.6 Raspberry .....	10
1.7 Red De Sensores (WSN).....	10
1.8 Hardware.....	11
1.8.1 Sensores .....	11
1.8.2 Sensor DHT22.....	12
1.8.3 Sensor de humo MQ – 2.....	13
1.8.4 Sensor Nivel de luz Resistor dependiente de la luz (LDR).....	15
1.8.5 Sensor de Vibración Piezoeléctrico .....	16
1.8.6 Sensor de Sonido FC 109.....	17
1.8.7 Sensor de presencia de infrarrojo pasivo (PIR).....	18
1.8.8 Sensor Detección de líquidos Higrómetro FC-28 .....	19
1.8.9 Sensor de Flujo de Aire de coeficiente de temperatura negativo (NTC) .....	20
1.9 Módulo GSM .....	21
1.10 Software .....	22
1.10.1 Procesador de Hipertexto (Php) .....	22
1.10.2 Arduino IDE.....	23

CAPÍTULO 2 .....	24
ANÁLISIS DE HARDWARE Y SOFTWARE.....	24
2.1 Análisis de Hardware .....	24
2.1.1 Análisis Sensores de temperatura y humedad DHT11 y DHT22 .....	25
2.1.1.1 Sensor DHT11 .....	25
2.1.1.2 Sensor DHT22 .....	25
2.1.2 Sensores de Humo.....	28
2.1.2.1 MQ- 2 .....	28
2.1.2.2 MQ- 135 .....	29
2.1.3 Nivel de luz .....	29
2.1.3.1 Resistor dependiente de la luz (LDR) .....	30
2.1.3.2 Luxómetro BH1750 .....	30
2.1.4 Sensor de Vibración .....	31
2.1.4.1 Piezoeléctrico .....	31
2.1.4.2 Sensor de vibración SW-420 .....	32
2.1.5 Sensor de Sonido.....	33
2.1.5.1 Sensor de Sonido FC 109.....	33
2.1.5.2 Sensor de Sonido y micrófono KY-038.....	33
2.1.6 Sensor Detección de líquidos .....	34
2.1.6.1 Detección de líquidos Higrómetro FC-28.....	34
2.1.7.1 Sensor PIR.....	36
2.1.7.2 Sensor magnético .....	36
2.1.7.3 Análisis de PIR y Sensor magnético .....	36
2.1.8 Sensor de Flujo de Aire de coeficiente de temperatura negativo.....	36
2.1.8.1 Termistor NTC (Negative Temperature Coefficient) .....	36
2.1.8.2 Termistor PTC (Positive Temperature Coefficient).....	36
2.1.9 Arduino .....	37
2.1.10 Raspberry .....	38
CAPÍTULO 3 .....	42
DESARROLLO Y DISEÑO .....	42
3.1 Sensores .....	42
3.1.1 Diagrama Sensor de Humo MQ2.....	44
3.1.2 Diagrama sensor para medir nivel de luz NTC GL5549.....	45
3.1.3 Diagrama sensor Vibración.....	46

3.1.4 Diagrama sensor de Sonido.....	46
3.1.5 Diagrama del sensor para detección de líquidos .....	47
3.1.6 Diagrama sensor PIR .....	48
3.1.7 Diagrama sensor Flujo de Aire .....	48
3.2 Módulos .....	49
3.2.1 Diagrama Módulo 1 .....	50
3.2.2 Diagrama Módulo 2 .....	52
3.2.3 Diagrama Módulo 3 .....	53
3.2.4 Diagrama Módulo 4 .....	54
3.2.5 Diagrama módulo 5.....	55
3.2.6 Programación del módulo GSM.....	57
3.2.7 Programación para realizar una llamada vía celular .....	57
3.2.8 Programación para el envío de datos mediante el formato JSON a la web .....	58
3.2.9 Diagrama del módulo 5, Raspberry Pi3 y GSM .....	58
3.2.9.1 Programación de Raspberry Pi .....	59
3.3 Software .....	60
3.3.1 Diagrama de bloques de hardware .....	60
3.3.2 Análisis de software .....	61
3.3.3 Arquitectura.....	61
3.3.4 Diagramas de caso de uso. ....	62
3.3.4.1 Actores .....	62
3.3.4.2 Diagrama de Casos de uso del sistema.....	62
3.3.4.3 Administrador .....	63
3.3.4.4 Invitado .....	63
3.3.5 Interfaces.....	64
3.3.6 Product BackLog.....	65
3.3.7 Sprint del proyecto .....	66
3.3.8 Tareas del Sprint .....	67
3.9 Lista de tareas de la iteración. ....	67
3.9.1 Pantalla de inicio .....	68
3.9.2 Gráfica lineal de los valores de un sensor .....	69
3.9.3 Pantalla de usuarios.....	70
3.10 Desarrollo aplicación Android .....	70
CAPÍTULO 4.....	72



4.1 Pruebas del entorno web .....	72
4.2 Pruebas de la aplicación en Android.....	75
4.3 Pruebas de toma de datos de los sensores desde Arduino ID .....	79
4.4 Resultado de pruebas de efectividad .....	79
4.4.1 Indicadores .....	80
4.4.2 Cálculo .....	80
4.5 Pruebas de diseño con respecto comunicación .....	81
4.5.1 Diseño con 4 bluetooth esclavos y 1 bluetooth receptor.....	81
Diseño con 4 bluetooth esclavos y 1 bluetooth receptor.....	81
4.5.2 Diseño 4 bluetooth esclavos y 4 bluetooth receptores .....	82
CONCLUSIONES .....	83
RECOMENDACIONES .....	84
LISTA DE REFERENCIAS .....	85

## ÍNDICE DE TABLAS

Tabla 1. Criterios.....	25
Tabla 2. Criterios y Opciones.....	26
Tabla 3. Tabla de valoraciones a utilizar .....	26
Tabla 4. Criterios vs Criterios .....	27
Tabla 5. Criterio Precisión Vs Opciones.....	27
Tabla 6. Criterio Rango de medición Vs Opciones.....	27
Tabla 7. Criterio Precio Vs Opciones .....	27
Tabla 8. Opciones vs Criterios .....	27
Tabla 9. Criterios vs. Criterios .....	37
Tabla 10. Precio comparado con diferentes tecnologías.....	37
Tabla 11. Desempeño comparado con diferentes tecnologías. ....	38
Tabla 12. Tecnología comparada con diferentes tecnologías del mercado.....	38
Tabla 13. Programación comparada con diferentes tecnologías del mercado .....	38
Tabla 14. Resultados de la tabla de criterios en factor a las opciones .....	38
Tabla 15. Criterios vs. Criterios .....	38
Tabla 16. Precio comparado con diferentes tecnologías.....	39
Tabla 17. Desempeño comparado con diferentes tecnologías. ....	39
Tabla 18. Tecnología comparada con diferentes tecnologías del mercado.....	39
Tabla 19. Compatibilidad comparada con las tecnologías del mercado. ....	39
Tabla 20. Resultados de criterios en factor a las opciones.....	39
Tabla 21. Criterios vs. Criterios .....	40
Tabla 22. Lenguaje de programación comparado con diferentes bases de datos.....	40
Tabla 23. Análisis de datos comparado con diferentes bases de datos. ....	40
Tabla 24. Tecnología usada comparada con las diferentes bases de datos. ....	40
Tabla 25. Comparativa entre la compatibilidad con herramientas de desarrollo y las diferentes bases de datos. ....	40
Tabla 26. Resultados de la tabla de criterios en factor a las opciones .....	41
Tabla 27. Módulos, ubicación de sensores y cálculos. ....	50
Tabla 28. Tabla de requisitos Scrum.....	65
Tabla 29. Tabla de requerimientos Sprint Backlog.....	67
Tabla 30. Tareas del Sprint .....	67
Tabla 31. Tareas de la iteración .....	67

Tabla 32. Prueba del requisito No. 1.....	72
Tabla 33. Prueba del requisito No. 2.....	73
Tabla 34. Prueba del requisito No. 3.....	73
Tabla 35. Prueba del requisito 4.....	74
Tabla 36. Prueba del requisito 5.....	74
Tabla 37. Prueba del requisito 6.....	75
Tabla 38. Prueba del requisito 1 para Android. ....	75
Tabla 39. Prueba del requisito 2 para Android ....	76
Tabla 40. Escala de valoración .....	79
Tabla 41. Tabla de indicadores .....	80
Tabla 42. Tabla de resultados y línea base.....	80

## ÍNDICE DE FIGURAS

Figura 1. Diseño Data Center.....	5
Figura 2. Historia y clasificación de los TIERS.....	6
Figura 3. Descripción de un sistema embebido .....	8
Figura 4. Componentes de un sistema embebido.....	9
Figura 5. Raspberry Pi 3 .....	10
Figura 6. Estructura básica de una red de sensores inalámbrica. ....	11
Figura 7. Esquema de funcionamiento de un sensor.....	12
Figura 8. Sensor de temperatura y humedad DHT22.....	13
Figura 9. Características sensor de temperatura y humedad DHT22.....	13
Figura 10. Sensor de gas MQ-2.....	14
Figura 11. Características sensor de humo MQ-2.....	14
Figura 12. Sensor de nivel de luz LDR.....	15
Figura 13. Características sensor de nivel de luz LDR .....	16
Figura 14. Sensor piezoeléctrico .....	16
Figura 15. Características sensor de vibración.....	17
Figura 16. Sensor de Señor de Sonido FC 109 y amplificador MAX9812 .....	17
Figura 17. Características sensor de vibración.....	18
Figura 18. Sensor de movimiento – presencia PIR.....	18
Figura 19. Características Sensor PIR.....	19
Figura 20. Sensor de Detección de líquidos HIGRÓMETRO FC-28.....	20
Figura 21. Características sensor Higrómetro FC-28.....	20
Figura 22. Sensor de Flujo de Aire .....	21
Figura 23. Características Sensor de Flujo de Aire NTC y PTC .....	21
Figura 24. Módulo GSM/SHIELD para Arduino .....	22
Figura 25. Configuración Módulo GSM/SHIELD para Arduino .....	22
Figura 26. Librería de Arduino para conexión con módulo GSM/SHIELD.....	24
Figura 27. Características DHT11 y DHT22 .....	26
Figura 28. Características comparativas sensor de humo MQ 2 y MQ 135 .....	29
Figura 29. Características Fotorresistencia LDR.....	30
Figura 30. Características Luxómetro BH1750 .....	31
Figura 31. Características del Piezoeléctrico. ....	32
Figura 32. Características del SW-420.....	32

Figura 33. Características Sensor de Sonido FC 109.....	33
Figura 34. Características Sensor de Sonido y micrófono KY-038.....	34
Figura 35. Características Detección de líquidos Higrómetro FC-28 y módulo detector de Líquidos, Sensor de agua.....	35
Figura 36. Características Sensor de Flujo de Aire NTC.....	37
Figura 37. Diagrama de temperatura y humedad DHT22.....	42
Figura 38. Código para inicializar y obtener datos del sensor DHT22.....	43
Figura 39 Librerías utilizadas. ....	43
Figura 40. Diagrama sensor de Humo MQ2 y conexión Arduino nano .....	44
Figura 41. Código sensor Humo - MQ2.....	44
Figura 42. Diagrama sensor de nivel de luz LDR GL5549 y conexión Arduino nano .....	45
Figura 43. Código sensor de Nivel de Luz LDR - GL5549.....	45
Figura 44. Diagrama de sensor Vibración - conexión Arduino nano .....	46
Figura 45. Código sensor de Vibración.....	46
Figura 46. Diagrama de sensor Sonido y conexión Arduino Mega.....	46
Figura 47. Código sensor de Sonido .....	47
Figura 48. Diagrama de sensor de detección de líquidos.....	47
Figura 49. Código para sensor para detección de líquidos. ....	47
Figura 50. Diagrama de sensor PIR – HC SR501 y conexión Arduino nano .....	48
Figura 51. Código para detectar Presencia sensor PIR .....	48
Figura 52. Diagrama de sensor Flujo de Aire y conexión Arduino nano .....	49
Figura 53. Código Flujo de aire .....	49
Figura 54. Diagrama de sensores DHT22, MQ2, PIR, BLUETOOTH HC-05 .....	50
Figura 55. Código módulo 1 .....	51
Figura 56. Diagrama de sensores DHT22, BLUETOOTH, HC-05 y del higrómetro. .....	52
Figura 57. Código de la programación módulo 2. ....	52
Figura 58. Diagrama de sensores, DHT22.....	53
Figura 59. Código Módulo 3.....	53
Figura 60. Diagrama de sensores, DHT22, LDR GL5549, PTC .....	54
Figura 61. Código Módulo 5.....	54
Figura 62. Diagrama de sensores, DHT22, FC 109, PIEZOELECTRICO, BLUETOOTH.....	55

Figura 63. Diagrama módulo 5 .....	56
Figura 64. Código para enviar mensaje en caso de existir alarma .....	57
Figura 65. Código para hacer una llamada en caso de existir alarma .....	57
Figura 66. Código para el envío de datos mediante el formato JSON a la web.....	58
Figura 67. Diagrama módulo 5, Raspberry Pi3 y GSM, DHT22, FC-109, Piezoeléctrico. ....	58
Figura 68. Comandos para instalación de componentes en Raspbian. ....	59
Figura 69. Comandos para comunicación entre Apache y PHP. ....	60
Figura 70. Diagrama de Bloques Hardware .....	60
Figura 71. Arquitectura del servidor web. ....	61
Figura 72. Diagrama de caso de uso del sistema usuario.....	62
Figura 73. Arquitectura del servidor web. ....	63
Figura 74. Diagrama de caso de uso usuario invitado .....	64
Figura 75. Esquema pantalla inicial .....	64
Figura 76. Mapa de navegación del entorno web .....	65
Figura 77. Pantalla inicial sistema monitoreo entorno web .....	68
Figura 78. Pantalla de visualización de variables ambientales del centro de datos ...	69
Figura 79. Gráfica lineal histórica de los valores tomados por un sensor en un determinado periodo de tiempo.....	69
Figura 80. Pantalla de usuarios registrados.....	70
Figura 81. Entorno gráfico de AppInventor 2 .....	71
Figura 82. Programación en AppInventor 2.....	71
Figura 83. Pantalla inicial aplicación para Android.....	76
Figura 84. Visualización gráfica de un módulo de detección de amenazas físicas....	77
Figura 85. Visualización de valores de los registros de un módulo de detección.....	78
Figura 86. Toma de datos de los diferentes sensores .....	79
Figura 87. Diseño con 4 bluetooth esclavos y 1 bluetooth receptor .....	81
Figura 88. Diseño 4 bluetooth esclavos y 4 bluetooth receptores.....	82

## **Resumen**

El prototipo desarrollado es una solución para monitorizar las amenazas físicas que se encuentran en el centro de procesamiento de datos de la Universidad Politécnica Salesiana, siendo posible verificar los valores desde una aplicación programada en Java y plataformas libres, para teléfonos inteligentes con sistema operativo Android.

Se compone de una red de sensores inalámbricos, un módulo coordinador y un mini computador Raspberry, que junto con la tecnología de las placas de Arduino, trabajan de manera armónica para la toma de datos, depuración y procesamiento de la información, para en lo posterior visualizarla en la aplicación móvil y en el entorno web del sistema de monitorización.

Con las cualidades del módulo GSM SHIELD para Arduino, se realizó alertas por medio de la red móvil, de envío de mensajes cortos de texto (SMS) y llamadas, para alertar al administrador de incidencias, siendo posible, tomar acciones preventivas y correctivas para evitar futuros sucesos.

El diseño realizado es una solución de bajo costo, en comparación a otros sistemas de monitorización propietarios que requiere pago de licencias y mantenimientos anuales, teniendo un ahorro a las empresas que requieren de un sistema similar.

Una vez implementado el prototipo, por su estructura autónoma, el administrador tiene la facilidad de monitorizar 24/7 las amenazas físicas dentro del centro de procesamiento de datos en cualquier instante, por medio de un dispositivo móvil o acceso al entorno web.

## **Abstract**

The developed prototype is a solution to monitoring the physical threats that be found in the data center of the Salesian Polytechnic University, being possible to visualize the values in an application programmed in Java and free platforms, for smartphones with Android operating system.

It consists of a network of wireless sensors, a coordinator module and a Raspberry minicomputer, which together with the technology of the Arduino boards, work harmoniously for data collection, filtering and processing of information, for later visualize it in the mobile application and in the web environment.

The qualities of the GSM SHIELD module for Arduino, alerts had made by means of the mobile network, sending short text messages (SMS) and calls, to alert the incident manager, being possible to take preventive and corrective actions to avoid future events.

The design is a low-cost solution, compared to other proprietary monitoring systems that require annual licensing and maintenance payments, with savings for companies that require a similar system.

Once the prototype is implemented, due to its autonomous structure, the administrator has the facility to monitor 24/7 the physical threats within the data center at any time, through a mobile device or access to the web environment.



## INTRODUCCIÓN

### **Antecedentes**

Con el avance de la tecnología, muchos equipos informáticos evolucionan, incrementando las tareas que realizan y acortando el tiempo de respuesta. En un centro de procesamiento de datos (data center) es primordial que los componentes eléctricos y electrónicos sean censados, para vigilar bajo qué condiciones ambientales y físicas se encuentran. La propuesta debe ser una solución versátil y autónoma para la detección de amenazas físicas y ambientales del centro de procesamiento de datos de la Universidad Politécnica Salesiana, el mismo que otorgará vigilancia en tiempo real en todo momento (24 horas / 7 días a la semana) para que el administrador tenga una intervención mínima en el control del mismo.

Para el desarrollo, se usó una metodología versátil que permita manejar procesos y hacerlos adaptables a las necesidades del proyecto, siendo accesible a posibles variaciones que se puedan tomar en cuenta.

El prototipo sistema debe constar de los siguientes componentes:

Componente de autenticación: Permite reconocer perfiles de usuario tales como Administrador o invitado, acceso al sistema.

Componente de visualización de reportes: Visualización mediante gráficas las variaciones ambientales.

Componente de administración: Permite configurar los rangos de la toma de información por parte de los sensores y remitente de mensajes y correo.

## **Planteamiento del problema**

El campus sur de la Universidad Politécnica Salesiana cuenta con el Data Center de la carrera de computación este se encuentra monitoreado por un sistema propietario de escritorio con una licencia de pago.

La detección de amenazas físicas como: temperatura, humedad, punto de rocío, nivel de luz, vibración, flujo de aire, nivel de sonido, aire, incendio, intrusión, son importantes para el buen funcionamiento del Data Center, el administrador debe monitorear estas amenazas y permanecer en su lugar de trabajo de forma permanente en el caso de que se presente algún incidente.

## **Justificación**

El Centro de Procesamiento de Datos (Data Center) es un espacio destinado a concentrar los recursos que sean necesarios para procesamiento de información de una organización, por ser equipos delicados y con un alto costo, se tiene que adecuarlos a ciertas condiciones ambientales, las mismas que se deben monitorear cada determinado periodo de tiempo.

Con el avance de la tecnología y el uso diario de los teléfonos inteligentes (Smartphone) se puede realizar múltiples tareas, como revisar noticias, conectarse a internet, tener acceso a redes sociales, una aplicación móvil para Android facilitaría la monitorización en tiempo real de las amenazas físicas existentes en el Data Center como: temperatura, humedad, punto de rocío, nivel de luz, vibración, flujo de aire, nivel de sonido, aire, incendio, intrusión de un data center, el administrador podrá en cualquier momento y desde cualquier lugar monitorizar el centro de procesamiento de datos en tiempo real.

## **Objetivos**

### **Objetivo general**

Desarrollar un sistema de monitorización de amenazas físicas como: temperatura, humedad, punto de rocío, nivel de luz, vibración, flujo de aire, nivel de sonido, aire, incendio, intrusión, con sistemas embebidos y red de sensores (WNS), para el Data Center del laboratorio de servidores de la carrera de computación de la Universidad Politécnica Salesiana Campus Sur, mediante una aplicación para Android.

### **Objetivos específicos**

Analizar e investigar las herramientas en hardware y software necesarias para la monitorización de las amenazas físicas.

Diseñar y configurar el prototipo para la monitorización de las amenazas físicas.

Implementar el prototipo para la monitorización de las amenazas físicas del Data Center.

Realizar pruebas de funcionamiento del prototipo.

### **Marco metodológico**

Para el desarrollo del prototipo se diseñó varios módulos distribuidos de forma estratégica, la comunicación inalámbrica entre módulos se realiza mediante una red de sensores, una vez que se obtuvieron los datos necesarios se envían al servidor y se los puede observar mediante la aplicación Web y Android.

Se tomó como referencia la metodología Scrum para la realización de las pruebas de la aplicación Android y la aplicación web, donde se estableció el Product Backlog, Sprint y pruebas de Sprint.

Con ayuda de la investigación de campo, se realizó las diferentes pruebas de conexión y validación de la toma de datos de las amenazas físicas del centro de datos.

# CAPÍTULO 1

## MARCO TEÓRICO

A continuación, se describe la información relacionada para el desarrollo del prototipo.

### **Centro de procesamiento de Datos**

Es la ubicación donde se concentran los recursos para el procesamiento de información en una organización. (SUPERNOVATEL, 2012)

Un centro de procesamiento de datos (Data Center) de la carrera de computación de la Universidad Politécnica Salesiana, por tener cobertura a nivel nacional, su infraestructura debe tener accesibilidad permanente a los recursos informáticos, esto dentro o fuera del campus, el mismo que tiene sistemas tales como: (SUPERNOVATEL, 2012)

- Energía.
- Aire acondicionado.
- Control de incendios.
- Intrusión (Control de acceso).

La monitorización de las amenazas ambientales es realizada por el administrador, teniendo un horario desde las 07.00 hasta las 21.00, fuera de este horario el centro de procesamiento de datos queda sin monitoreo y sin advertencias en tiempo real de amenazas.

Se tomó como prioridad tener sensores que controlen físicamente las siguientes variables:

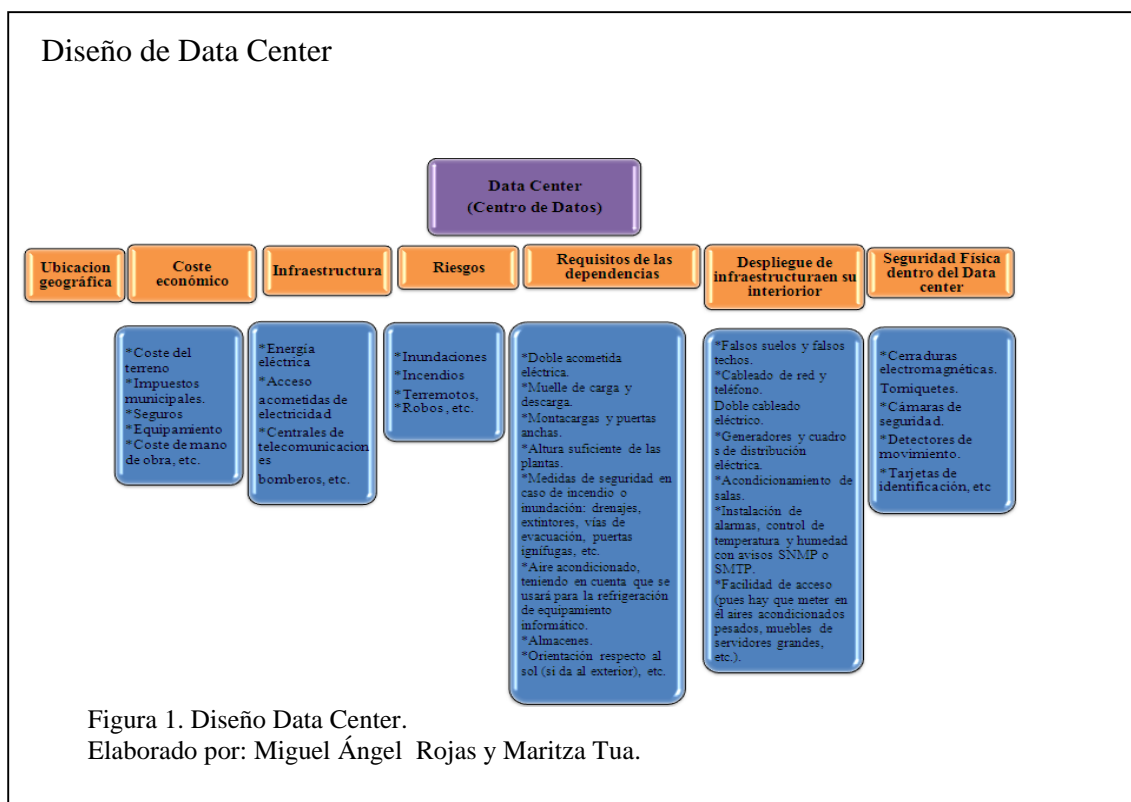
- Aire acondicionado.
- Humedad.
- Temperatura.
- Detección de fuga de líquido bajo el piso falso.

- Intrusión.
- Baja de energía.

### Criterios para valorar un centro de procesamiento de datos

La Asociación de la Industria de Telecomunicaciones (TIA), es la encargada de registrar estándares que salvaguarda: continuidad en el servicio, disponibilidad, solidez y seguridad operacional. El estándar demuestra los requerimientos mínimos de infraestructura en cuanto a telecomunicaciones se refiere, además que el Uptime Institute propone las especificaciones en cuanto a temas eléctricos, arquitectónicos, mecánicos y telecomunicaciones. Para el correcto funcionamiento del centro de procesamiento de datos, se cuenta con parámetros a seguir por parte de los fabricantes, normas y estándares. (SUPERNOVATEL, 2012)

### 1.1 Diseño



En la figura 1, se observa algunas de las características que debe tener un centro de procesamiento de datos por ejemplo la ubicación geográfica, la infraestructura, el costo económico, la seguridad física, etc.

### 1.2 Estándar Uptime Institute en Data Center

Es el encargado de crear y promover los conocimientos y lineamientos necesarios que debe cumplir un Data Center para garantizar su disponibilidad y continuidad.

En función al Uptime Institute hizo un recuento de la historia y clasificación de los TIERS. ( SUPERNOVATEL, 2012)

Historia y clasificación de los TIERS

TIERS	AÑO	CLASIFICACIÓN
I	Inicios 1960	Infraestructura básica
II	En 1970	Infraestructura con componentes redundantes
III	Finales de 1980	Infraestructura con mantenimiento simultaneo
IV	Mediados de 1990	Infraestructura tolerante a fallas

Figura 2. Historia y clasificación de los TIERS.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 1.3 Sistemas de Monitorización

Una parte de la administración de los sistemas informáticos es la monitorización y control, pensados en automatizarlos para obtener información de las variables del entorno en el que se desenvuelve para su posterior análisis. Se tiene en cuenta, que miles de actividades que realiza el hardware (parte tangible del equipo informático) se puede tomar datos mediante una red de sensores. Los sistemas pueden ser de dos categorías: (MINDMEISTER, 2013)

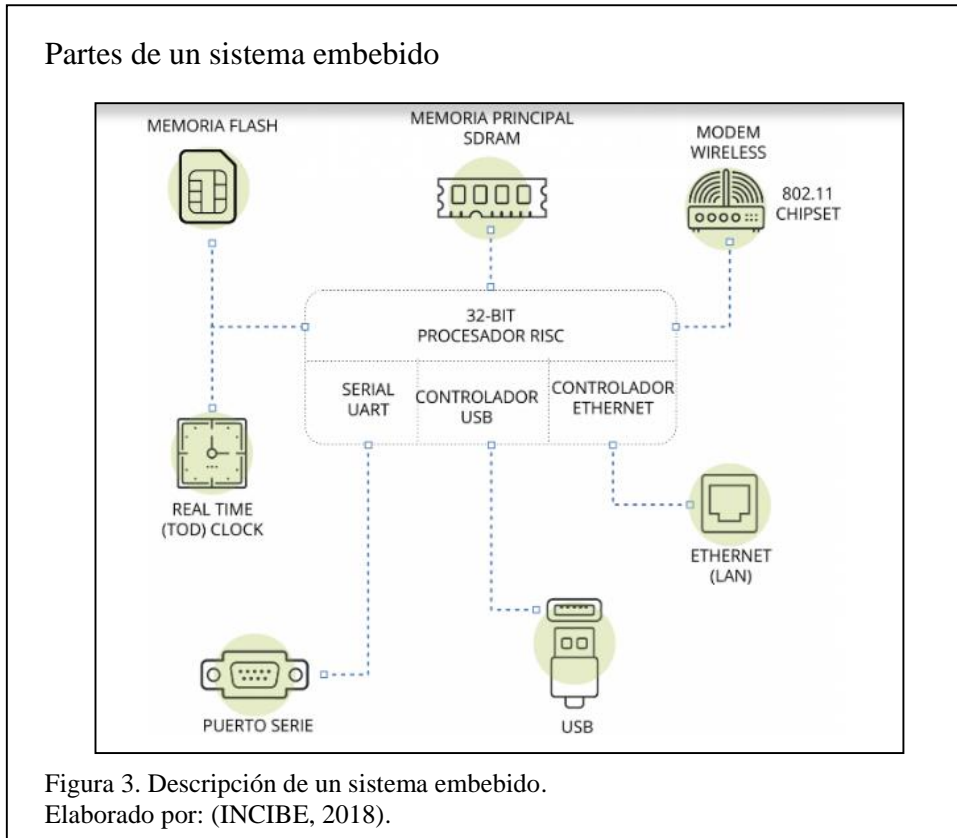
- Sistema de monitorización: Permite recopilar información por medio de la red de sensores de los valores obtenidos del medio, basándose en una interfaz de

usuario para posteriormente visualizarlos desde un teléfono inteligente, computador personal o un PDA (Asistente Digital Personal). Permite tener estadísticas, realizar consultas de históricos y gráficas

- Sistemas de control: Posterior a la recolección de datos, se pone en marcha acciones para contrarrestar cualquier anomalía detectada.

#### **1.4 Sistemas embebidos**

Son sistemas electrónicos fabricados para realizar procesamiento de datos. En la actualidad, los sistemas de control automático y de automatización se basan en sistemas embebidos, siendo necesarios para realizar tareas cotidianas de manera automática, estos con el desarrollo de la tecnología se hacen populares en el medio, siendo cada vez requeridos por empresas e industrias. Desarrolladores que realizan estas actividades, encuentran en los sistemas empotrados, integrados o embebidos, una solución a bajo costo para integrarlos en tareas diarias como automóviles, juguetes, etc. (SISBIB, 1999)



En la figura 3, se observa las principales partes que constan en un sistema embebido como la memoria flash (memoria de lectura y escritura), la memoria de acceso aleatorio de sincronía dinámica (SDRAM) que permite la escalabilidad pensado en usarlo como servidor o estación de trabajo, un módem inalámbrico para conexión a una red, un reloj en tiempo real para almacenar la hora en el sistema, un puerto de conexión serial transmisor receptor asíncrono universal (SERIAL UART) para periféricos, un controlador, el controlador de bus universal en serie (USB), un puerto Ethernet para conexión de red de área local (LAN). Todo esto controlado por un procesador que hace el papel de “cerebro” y coordinador.

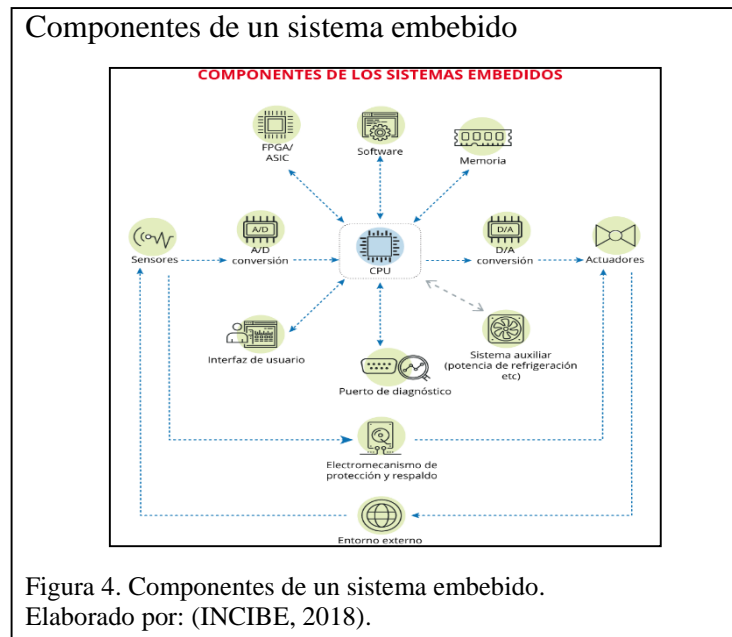
Siendo sistemas cerrados estos carecen de interfaz de usuario, los que se programan en lenguaje ensamblador. La parte central que funciona como cerebro se denomina microcontrolador. Algunas de las características principales son:



- Manejan licencia de software libre.
- Son fiables funcionando de manera constante e ininterrumpida.
- Son totalmente personalizables de acuerdo a la necesidad del usuario.

(Tecnología & Informática, 2018)

Las principales partes que constan en los sistemas embebidos son las siguientes:



En la figura 4, se tiene un esquema detallado de los componentes citados en la figura 3, adicional a esto, por medio de convertidores de señal analógica – digital se conecta a sensores y actuadores, dándole diferentes usos en ramas como: domótica, medicina, industria, control automático y de maquinaria, control industrial y automotriz, etc. Por sus puertos de conexión (LAN y Wireless) puede conectarse a internet y al entorno externo.

## 1.5 Arduino

Es una marca que se encarga del desarrollo de componentes electrónicos orientados a software de código abierto (libre), siempre apuntando a la flexibilidad y que sea de fácil manejo. Las placas poseen un microcontrolador (parte central de procesamiento),

puertos de entrada, salida, comunicación de datos. Por su versatilidad, sus principales aplicaciones son domótica, robótica, telemática, etc. Entre los modelos más usados están Arduino Nano (por su tamaño para aplicaciones que requieran módulos de espacio reducido, Arduino Lilipad usado en “tejidos inteligentes” y dispositivos vestibles (wearables), Arduino Mega para aplicativos que requieren desempeño, etc. (EcuRed, 2018)

### 1.6 Raspberry

Es un computador de placa única y reducida con la capacidad de realizar tareas procesos, caracterizado por ser de bajo costo comparado con un computador personal como se muestra en la figura 5. (Wikipedia, 2018)

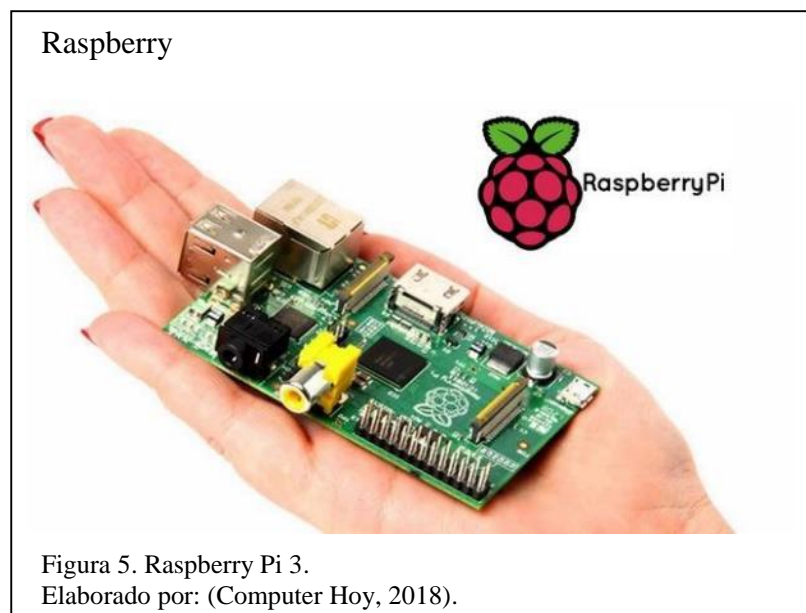
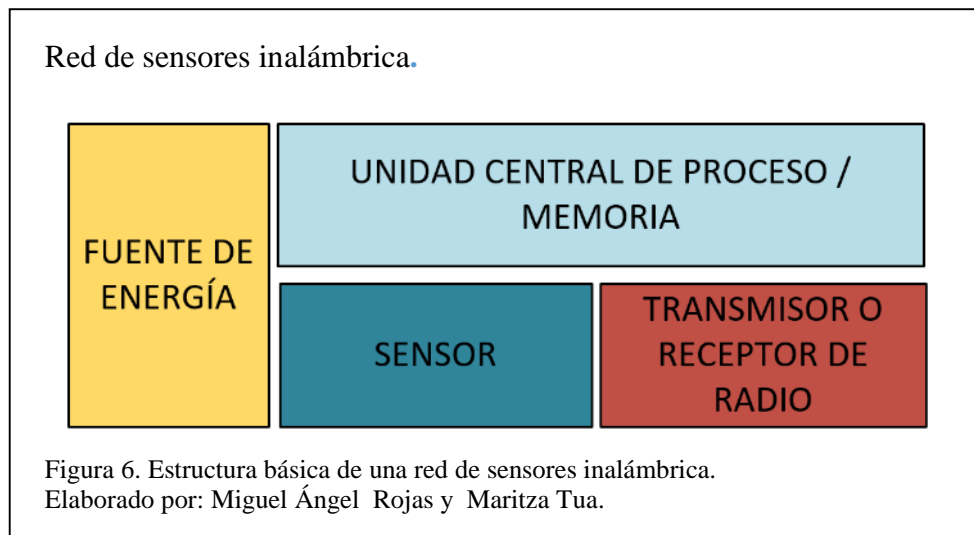


Figura 5. Raspberry Pi 3.  
Elaborado por: (Computer Hoy, 2018).

### 1.7 Red De Sensores (WSN)

Es una red basada en dispositivos de bajo costo y consumo energético, diseñada para captar información de su entorno para en lo posterior ser procesada, siendo importante el transmitirla por enlaces inalámbricos hasta un dispositivo central coordinador. Los sensores pueden ser fijos o móviles y permiten censar las diversas condiciones del

entorno como el sonido, temperatura, presión, etc., teniendo una amplia gama de aplicaciones en cualquier rama profesional. (Vera Romero, Barbosa Jaimes, & Pabón González, 2017)



En la figura 6, un esquema de una red de sensores inalámbrica, que está compuesta por una fuente de energía que es la encargada de proveer de energía al circuito, un módulo transmisor – emisor para el envío y recepción de los datos tomados por los sensores, un sensor para señal analógica o señal digital para la captura de datos del entorno, la unidad central de procesamiento que se encarga de orquestar, coordinar y ejecutar las acciones programadas.

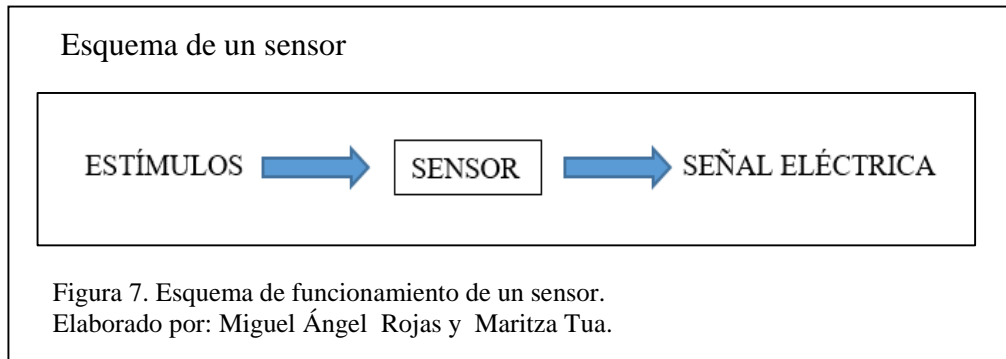
## 1.8 Hardware

A continuación, se detalla los componentes utilizados en el prototipo con sus características importantes.

### 1.8.1 Sensores

Se los conoce como entradas (input), los mismos que están creados para detectar acciones o estímulos que provienen del medio externo y responder de alguna manera,

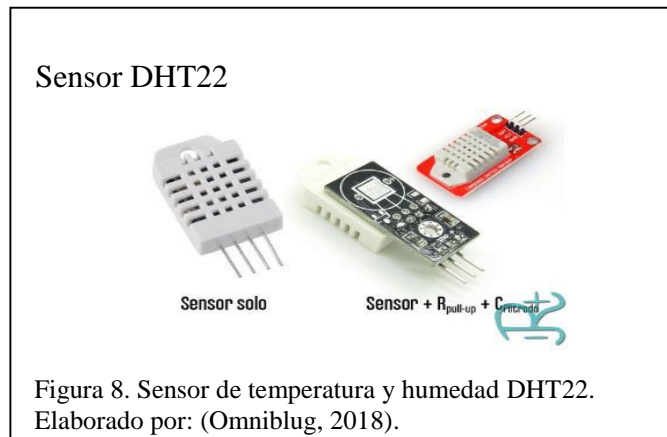
por medio de información medido en magnitudes físicas para que sean transformadas en señales eléctricas. (Wikipedia, 2018)



En la figura 7, esquema del funcionamiento de un sensor, el cual recibe un estímulo del medio, que al pasar por el sensor y transformarlo en una señal eléctrica. En lo posterior con un conversor de señal analógica – digital se toma el dato para procesarlo, sea en un cálculo o para visualizarlo.

### 1.8.2 Sensor DHT22

Sensor digital de temperatura y humedad, utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica), simple de usar, pero requiere sincronización cuidadosa para tomar datos, su inconveniente es que las lecturas que se pueden realizar serán mínimas cada 2 segundos, los valores de temperatura y humedad serán mostrados como números con una cifra decimal, se observa en la figura 8, la diferencia entre un sensor solo y con el módulo de adaptación para Arduino. (Electronilab, 20018)



Características:

Característica sensor DHT22

DHT22	
Alimentación	$3.3Vdc \leq Vcc \leq 6Vdc$
Señal de Salida	Digital
Rango de medida Temperatura	De $-40^{\circ}C$ a $80^{\circ}C$
Precisión Temperatura	$<\pm 0.5^{\circ}C$
Resolución Temperatura	$0.1^{\circ}C$
Rango de medida Humedad	De 0 a 100% RH
Precisión Humedad	2% RH
Resolución Humedad	0.1%RH
Tiempo de sensado	2s
Tamaño	14 x 18 x 5.5mm

Figura 9. Características sensor de temperatura y humedad DHT22.  
Elaborado por: Miguel Rojas y Maritza Tua.

En la figura 9, las características principales del sensor DHT22 como la alimentación, señal de salida, precisión que posee, las dimensiones, etc.

### 1.8.3 Sensor de humo MQ – 2

El MQ-2 Sensor de Gas tiene sensibilidad especial para medir concentraciones de gas en el aire, es sensible a, LPG, i-butano, propano, metano, alcohol, hidrógeno y humo. La resistencia del sensor cambia de acuerdo a la concentración del gas en el aire, se observa en la figura 10. (geekfactory, 2018)

Sensor MQ-2



Figura 10. Sensor de gas MQ-2.  
Elaborado por: (MECHATRONIC, 2018).

Características:

#### Características sensor MQ-2

MQ2	
Chip:	LM393, sensor de gas ZYMQ-2
Voltaje operativo:	DC 5V
Zona de detección:	300 a 10000 ppm (partes por millón)
Tiempo de respuesta:	$\leq 10s.$
Tiempo de recuperación:	$\leq 30s.$
Condiciones:	Temperatura ambiente: $-20^{\circ}C \sim 55^{\circ}C$
Resistencia de calentamiento:	$31\omega \pm 3\omega$
Corriente de calefacción:	$\leq 180mA$
Energía de calefacción:	$\leq 900mW$
Salida analoga:	0 ~ 5V (a mayor concentración, mayor voltaje)
Salida digital:	TTL
Peso	30 g
Dimensiones	3.2 x 2.7 x 2.2 cm

Figura 11. Características sensor de humo MQ-2.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 11, las características del sensor MQ-2, cuenta con un chip interno para detectar gas, un tiempo de respuesta óptimo menor a 10 segundos, trabaja a temperatura ambiental de 20 grados centígrados bajo cero hasta 55 grados centígrados.

#### 1.8.4 Sensor Nivel de luz Resistor dependiente de la luz (LDR)

Un LDR es un sensor es adecuado para medidas cuantitativas sobre el nivel de luz en exteriores e interiores, por ejemplo, encendiendo una luz, subiendo una persiana, o también orientando a un robot. (LLAMAS, 2015)

La relación entre la intensidad lumínica y el valor de la resistencia no es lineal. Su comportamiento es el siguiente:

- Más luz = menor resistencia eléctrica
- Menos luz = mayor resistencia eléctrica

Cuenta con un diámetro de 5 milímetros y comúnmente se lo conoce como fotocelda, se lo observa en la figura 12.



Características:



## Características sensor de nivel de luz LDR

LDR
Conformados por un semiconductor de alta resistencia
La resistencia varía entre $1M\Omega$ , cuando hay mucha oscuridad y $100\Omega$ con altas intensidades de luz
Tiempo de respuesta de 1 décima de segundo cuando varía la intensidad luminica.
Se puede encontrar en cámaras, medidores de luz, alarmas de seguridad, medidores de radio o los sistemas de alumbrado publico;

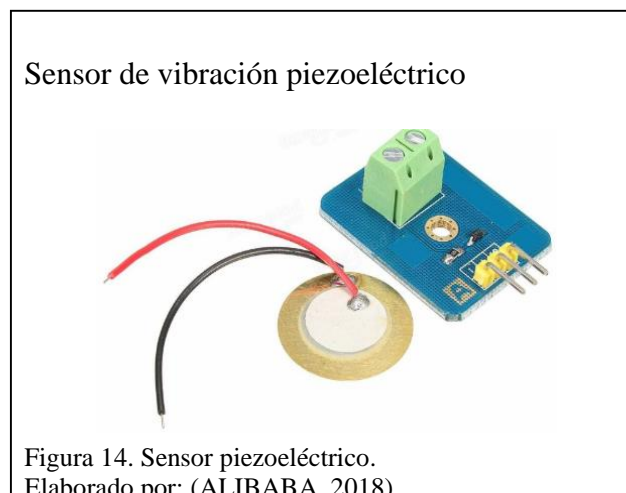
Figura 13. Características sensor de nivel de luz LDR.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 13, las características principales del sensor que tiene un resistor dependiente de la luz (LDR).

### 1.8.5 Sensor de Vibración Piezoeléctrico

Se utiliza para la detección de vibraciones. La placa piezocerámica responde a cambios de flexión, tacto, vibración y choque con un cambio en la salida analógica, se observa en la figura 14. A mayor grado de vibración, mayor es el cambio que se presenta a la salida, su diseño brinda facilidad de cambiar el sensor por otro si es necesario.

(ALIBABA, 2018)



En la figura 15, cuenta con características como señal de salida, entrada positiva y tierra, trabaja a 10 grados centígrados bajo cero hasta 70 grados centígrados de temperatura.



Características:

Características sensor de vibración

PIEZOELÉCTRICO
Frecuencia de Resonancia: $6.3 \pm 0.6$ KHz
Impedancia: $1000\Omega$ máx
Capacitancia: $10.0 \pm 30$ nF.
Material: Latón.
Longitud del cable: 50mm (cable tipo AWG32).

Figura 15. Características sensor de vibración .  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 1.8.6 Sensor de Sonido FC 109

Es un sensor pequeño, la señal es demasiado baja para ser registrada correctamente tiene un amplificador MAX9812 integrado, se observa en la figura 16. (LUIS LLAMAS, 2016)

Sensor de Sonido FC 109 y amplificador MAX9812



Figura 16. Sensor de Señor de Sonido FC 109 y amplificador MAX9812.  
Elaborado por: (maxelectronica, 2018).

Características:

Características sensor de Sonido FC 109

FC 109
Alimentación: 3.3 a 5 V DC
Permite detectar la intensidad de sonido del entorno
Incorpora un amplificador de audio MAX9812
Posee un orificio de instalación para facilitar su uso y montaje
Construido con un LED Indicador de alimentación
Conexión de 3 hilos
Dimensiones: 26 x 12 x 10

Figura 17. Características sensor de vibración.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 17, las características del sensor de sonido FC-109, permite detectar la intensidad de sonido del medio, posee un amplificador de audio y de fácil montaje.

### 1.8.7 Sensor de presencia de infrarrojo pasivo (PIR)

Sensor electrónico, que por medio de luz infrarroja detecta movimiento en un rango de aproximadamente 3 metros en campo abierto, escogido en proyectos y prototipos por su tamaño reducido, su precio frente a otros de iguales características, por trabajar con baja potencia y de fácil implementación, se observa en la figura 18. (Macho, 2017)



Características:

PIR
Sensor piroeléctrico (Pasivo) infrarrojo (También llamado PIR)
El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización
Rango de detección: 3 m a 7 m, ajustable mediante trimmer (Sx)
Lente fresnel de 19 zonas, ángulo < 100°
Salida activa alta a 3.3 V

Figura 19. Características Sensor PIR.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 19, las características del sensor de infrarrojo pasivo (PIR), entre los principales consumos energéticos menores a 50 micro amperios, fácil de usar, tamaño reducido

#### **1.8.8 Sensor Detección de líquidos Higrómetro FC-28**

Módulo para censar la humedad del suelo, se observa en la figura 20, mediante la conductividad entre las dos placas recubiertas con una capa de material conductor, de existir humedad, se crea un puente de una placa a la otra, siendo detectada por un amplificador operacional LM393, encargado de transformar a un valor analógico y listo para ser leído por Arduino, se observa en la figura 21, las principales características del sensor. (LUISLLAMAS, 2016)



Figura 20. Sensor de Detección de líquidos HIGRÓMETRO FC-28.  
Elaborado por: (Llama, 2017).

Características:

Características sensor de Detección de líquidos	
HIGRÓMETRO FC-28	
Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire	
Un suelo ligeramente húmedo daría valores típicos de 600-700.	
Un suelo seco tendrá valores de 800-1023.	
La salida digital	
Alimentación 5V	

Figura 21. Características sensor Higrómetro FC-28.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 1.8.9 Sensor de Flujo de Aire de coeficiente de temperatura negativo (NTC)

Los termistores de coeficiente de temperatura negativo (NTC), al aumentar la temperatura, aumentará también la concentración de portadores, por lo que la resistencia será menor, de ahí que el coeficiente sea negativo, el flujo es proporcional al enfriamiento, en la figura 22. (Wordpress, 2010)



Figura 22. Sensor de Flujo de Aire.  
Elaborado por: (luisllamas, 2018).

Características:

Características Termistor NTC y PTC
<b>NTC:</b> Aumenta la temperatura y disminuye la resistencia, termistor de coeficiente negativo.
<b>PTC:</b> Cuando aumenta la temperatura también aumenta la resistencia, termistor de coeficiente positivo.

Figura 23. Características Sensor de Flujo de Aire NTC y PTC.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 23, las principales características del termistor de coeficiente negativo (NTC) y el de coeficiente positivo (PTC).

### 1.9 Módulo GSM

Módulo shield GSM se utiliza para que las placas Arduino se conecten con el exterior, con los servicios de una operadora celular por medio de una SIM CARD (Módulo de Identificación de Suscriptor), simulando las funciones de llamar, enviar mensajes de texto corto (SMS) y acceder a internet por medio de una conexión GPRS (Servicio General de Paquetes vía Radio), en la figura 24 se observa la estructura. (HETPRO, 2015)

Módulo SIM 900



Figura 24. Módulo GSM/SHIELD para Arduino.  
Elaborado por: (prometec).

### Configuración Módulo SIM 900

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8); // 10 y 11 para el Arduino Mega. Configura el puerto serie para el SIM900.
```

Figura 25. Configuración Módulo GSM/SHIELD para Arduino.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 25, se observa la configuración para el puerto serial 10 y 11 con el módulo Arduino Mega, y el 7 y 8 para Arduino Nano.

## 1.10 Software

Para el desarrollo del sistema de monitorización, herramientas de desarrollo que sean de licencia de código abierto.

### 1.10.1 Procesador de Hipertexto (Php)

Es un lenguaje de programación que usa código abierto, muy usado para realizar desarrollo de entornos web y se puede añadir en el contenido lenguaje de marcas de hipertexto (HTML). (Lapiente, 2013)

### **1.10.2 Arduino IDE.**

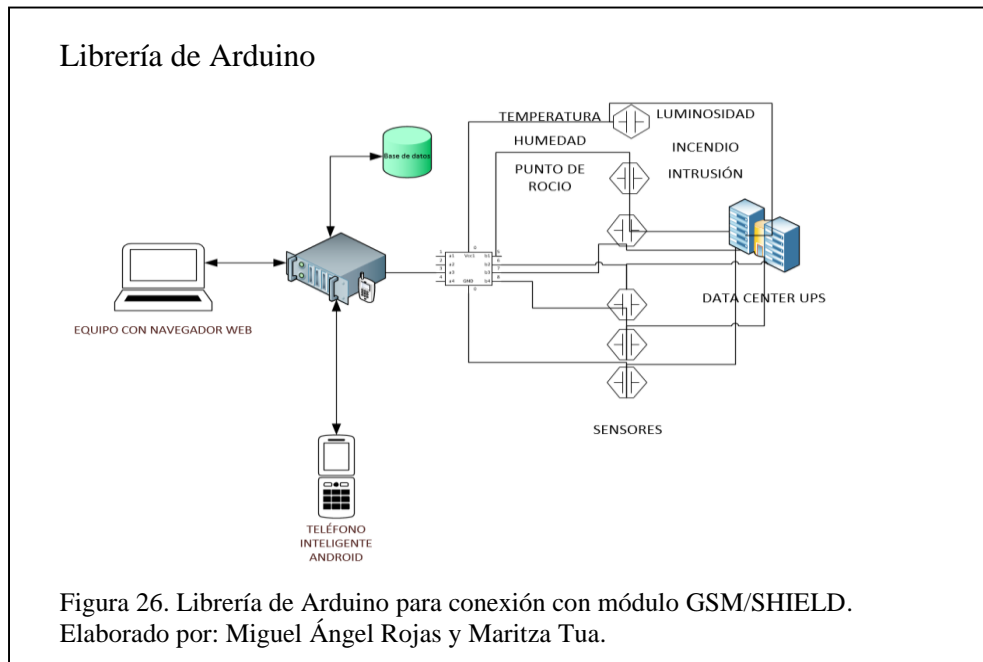
Es un entorno de desarrollo y programación para las placas Arduino, se basa en entorno Processing (código de programación basado en Java). (ARDUINO.cl, 2018)

## CAPÍTULO 2

### ANÁLISIS DE HARDWARE Y SOFTWARE

Se analiza el hardware y software utilizado en el desarrollo del prototipo.

#### 2.1 Análisis de Hardware



En la figura 26, el diagrama muestra el diagrama utilizado para el ensamblaje del prototipo, teniendo como periféricos un computador con navegador web y acceso a internet y un teléfono inteligente con sistema operativo Android. Se conectan al prototipo que toma datos de las variables ambientales del centro de procesamiento de datos por medio de la red de sensores, los concentra en el módulo orquestador, los almacena en la base de datos, y los visualiza en el entorno web y en la aplicación móvil.

Para realizar el análisis de hardware se toma en cuenta cualidades como precio, compatibilidad con diferentes tecnologías, rangos de cobertura, precisión, entre otros.



La ponderación va desde 10 que es una calificación alta, hasta 0,20 la calificación baja, se lo puede observar en la tabla 1.

#### Criterios

RESPUESTAS	VALOR
Igualmente importante	1
Significativamente importante	5
Excesivamente importante	10
Significativamente menos importante	0.2
Excesivamente menos importante	0.1

Tabla 1. Criterios.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### **2.1.1 Análisis Sensores de temperatura y humedad DHT11 y DHT22**

Se muestra la diferencia entre el sensor DHT11 y DHT22.

#### ***2.2.1.1 Sensor DHT11***

Sensor de temperatura y humedad, provee una salida de datos digital por lo que asegura buena calidad y una fiabilidad al transcurrir el tiempo, contiene un microcontrolador de 8 bits integrado para lectura de enteros, por lo tanto, no entrega datos con decimales, su tiempo de lecturas serán mínimo cada 2 segundos, capacidad de transmitir la señal hasta 20 metros de distancia, sensor de bajo costo, fácil manejo en el mundo de Arduino. (LUIS LLAMAS, 2016)

#### ***2.2.1.2 Sensor DHT22***

DHT22 nace como una evolución del DHT11, con mayores rangos de temperatura y humedad, y más precisión, los valores para temperatura y humedad serán números con una cifra decimal, como se observa en la figura 27. (LUIS LLAMAS, 2016)

Características:

Parámetro	DHT11	DHT22
Alimentación	$3V \leq 5$	$3.3V_{dc} \leq V_{cc} \leq 6V_{dc}$
Señal de Salida	Digital	Digital
Rango de medida Temperatura	De 0 a 50 °C	De -40°C a 80 °C
Precisión Temperatura	$\pm 2$ °C	$<\pm 0.5$ °C
Resolución Temperatura	0.1°C	0.1°C
Rango de medida Humedad	De 20% a 90% RH	De 0 a 100% RH
Precisión Humedad	4% RH	2% RH
Resolución Humedad	1%RH	0.1%RH
Tiempo de sensado	1s	2s
Tamaño	12 x 15.5 x 5.5mm	14 x 18 x 5.5mm

Figura 27. Características DHT11 y DHT22.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En las siguientes tablas se explica los cálculos que se realizaron para el análisis de factibilidad de hardware y software

Tabla 2. Criterios y Opciones

CRITERIOS	OPCIONES
PRESICIÓN	DHT11DHT22
RANGO DE MEDICIÓN	
PRECIO	

Nota: Criterios según opciones.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 3. Tabla de valoraciones a utilizar

RESPUESTAS	VALOR
Igualmente importante	1
Significativamente importante	5
Excesivamente importante	10
Significativamente menos importante	0.2
Excesivamente menos importante	0.1

Nota: Tabla de valoraciones según criterios personales.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 4. Criterios vs Criterios

#	A	B	C	D	E	F
0	CRITERIOS VS CRITERIOS	PRECISIÓN	RANGO DE MEDICIÓN	PRECIO	SUMA	PONDERACIÓN
1	PRECISIÓN		1	10	11	0,495
2	RANGO DE MEDICIÓN	1		10	11	0,495
4	PRECIO	0,1	0,1		0,2	0,009
				Σ	22,2	1,000

Nota: Valoraciones a cada criterio según Tabla 4; Ponderación (E1/ ΣE)

Elaborado por: Miguel Ángel Rojas y Maritza Tua

Tabla 5. Criterio Precisión Vs Opciones

#	A	B	C	D	E
0	PRECISIÓN	DHT11	DHT22	SUMA	PONDERACIÓN
1	DHT11		0,1	0,1	0,010
2	DHT22	10		10	0,990
				Σ	10,1
					1

Nota: Tabla de comparación (fila, columna), donde se toma en cuenta el primer criterio Precisión.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 6. Criterio Rango de medición Vs Opciones

#	A	B	C	D	E
0	RANGO DE MEDICIÓN	DHT11	DHT22	SUMA	PONDERACIÓN
1	DHT11		0,1	0,1	0,010
2	DHT22	10		10	0,990
				Σ	10,1
					1

Nota: Tabla de comparación (fila, columna), donde se toma en cuenta el segundo criterio Rango de medición.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 7. Criterio Precio Vs Opciones

#	A	B	C	D	E
0	PRECIO	DHT11	DHT22	SUMA	PONDERACIÓN
1	DHT11		0,1	0,1	0,010
2	DHT22	10		10	0,990
				Σ	10,1
					1

Nota: Tabla de comparación (fila, columna), donde se toma en cuenta el último criterio Precio.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 8. Opciones vs Criterios

#	A	B	C	D	E	F	G
0	CRITERIOS VS OPCIONES	PRECISIÓN	RANGO DE MEDICIÓN	PRECIO	SUMA	PONDERACIÓN	PORCENTAJE
1	DHT11	0,0049	0,0049	0,0001	0,01	0,010	1%
2	DHT22	0,4906	0,4906	0,0089	0,99	0,990	99%
				Σ	1	1	100%

Nota: Se realizan los cálculos necesarios para obtener entre Opciones vs Criterios.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

A continuación, se explica el cálculo de la tabla 8, este proceso se lo realiza en todas

las tablas para obtener el análisis de cada componente instalado en el prototipo.

- $PRECISIÓN\ DHT11\ Y\ DHT22\ (Tabla8(B1) = Tabla\ 4(F1) * Tabla5(E1);$

$$Tabla8(B2) = Tabla4(F1) * Tabla5(E2)).$$

- RANGO DE MEDICIÓN DHT11 Y DHT22 ( $\text{Tabla8(C1)} = \text{Tabla4(F2)} * \text{Tabla6(E1)}$ ;  $\text{Tabla8(C2)} = \text{Tabla4(F2)} * \text{Tabla6(E2)}$ ).
- PRECIO DHT11 Y DHT22 ( $\text{Tabla8(D1)} = \text{Tabla4(F3)} * \text{Tabla7(E1)}$ ;  $\text{Tabla8(D2)} = \text{Tabla4(F3)} * \text{Tabla7(E2)}$ ).
- SUMA horizontal ( $\text{Tabla8(E1)} = \text{Tabla8(B1)} + \text{Tabla8(C1)} + \text{Tabla8(D1)}$ ;  $\text{Tabla8(E2)} = \text{Tabla8(B2)} + \text{Tabla8(C2)} + \text{Tabla8(D2)}$ ).
- PONDERACIÓN ( $\text{Tabla8(F1)} = \text{Tabla8(E1)} / 1$  ;  $\text{Tabla8(F2)} = \text{Tabla8(E2)} / 1$ ).

### **Análisis de Priorización sensores DHT11 y DHT22**

Realizados los análisis de las opciones y criterios expuestos (Tabla 1), en la Tabla 7 se visualiza que el sensor DHT22 es el más factible con el 99% para ser aplicado en el prototipo.

#### **2.1.2 Sensores de Humo**

Son dispositivos construidos para detectar humo, ocasionado por un incendio.

##### **2.1.2.1 MQ- 2**

Como se observa en la figura 28, el módulo MQ-2, tiene como elemento principal el conocido sensor de gas MQ2, el cual es usado principalmente en la detección de humo, gas licuado de petróleo (GLP), gas inflamable butano, gas metano, hidrógeno, propano, alcohol, entre otros. Tiene un gran alcance de detección y es de respuesta rápida, alta sensibilidad y estabilidad. Detecta concentraciones de 300 a 10000ppm. (eekFACTORY, 2018)

### 2.1.2.2 MQ- 135

Se utilizan en equipos de control de calidad del aire para edificios y oficinas, son adecuados para la detección de NH<sub>3</sub>, NO<sub>x</sub>, alcohol, benceno, humo, dióxido de carbono (CO<sub>2</sub>), etc., las características se observan en la tabla 28. (tecnopura, 2016)

Características:

Características MQ 2 y MQ 135		
Parámetros	MQ 2	MQ-135
Chip:	LM393, sensor de gas ZYMQ-2	LM393
Voltaje operativo:	DC 5V	DC 5V
Zona de detección:	300 a 10000 ppm (partes por millón)	10 a 10000 ppm (partes por millón)
Tiempo de respuesta:	≤10s.	≤10s.
Tiempo de recuperación:	≤30s.	≤30s.
Condiciones:	Temperatura ambiente: -20°C ~ 55°C	Temperatura ambiente: -20°C ~ 55°C
Resistencia de calentamiento:	31Ω ± 3Ω	31Ω ± 3Ω
Corriente de calefacción:	≤180mA	≤180mA
Energía de calefacción:	≤900mW	≤900mW
Salida analoga:	0 ~ 5V (a mayor concentración, mayor voltaje)	0 ~ 5V (a mayor concentración, mayor voltaje)
Salida digital:	TTL	TTL
Peso	30 g	30 g
Dimensiones	3.2 x 2.7 x 2.2 cm	3.2 x 2.7 x 2.2 cm

Figura 28. Características comparativas sensor de humo MQ 2 y MQ 135.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### Análisis de Priorización sensores MQ2 y MQ 135

Se observa en la figura 28, el estudio de 2 sensores de la familia MQ que se encargan de la detección de humo, los dos tienen características que se aplican con el prototipo, MQ2 dispone información como tutoriales, stock y precio, se eligió este sensor por afinidad y facilidad a la hora de tomar datos con Arduino.

### 2.1.3 Nivel de luz

Se define como el flujo luminoso que incide sobre una superficie, medido en lúmenes (LUX). (Científicas, 2016)

### 2.1.3.1 Resistor dependiente de la luz (LDR)

Es un componente cuya resistencia varía sensiblemente con la cantidad de luz percibida, las características se muestran en la figura 29. La relación entre la intensidad lumínica y el valor de la resistencia no es lineal. Su comportamiento es el siguiente:

- Más luz = menor resistencia eléctrica
- Menos luz = mayor resistencia eléctrica (LLAMAS, 2015)

### 2.1.3.2 Luxómetro BH1750

El módulo BH1750 es un sensor de luz, que a diferencia del NTC es digital y entrega valores de medición en Lux (lumen /m<sup>2</sup>cxz) que es la unidad de medida estándar para el nivel de iluminación (iluminancia), estas características se muestran en la figura 31. Tiene alta precisión y un rango ente 1 – 65535 lx el cual es configurable. (naylampmechatronics, 2017)

Características:

LDR
Conformados por un semiconductor de alta resistencia
La resistencia varia entre 1MΩ, cuando hay mucha oscuridad y 100Ω con altas intensidades de luz
Tiempo de respuesta de 1 décima de segundo cuando varia la intensidad lumínica.
Se puede encontrar en cámaras, medidores de luz, alarmas de seguridad, medidores de radio o los sistemas de alumbrado publico;

Figura 29. Características Fotorresistencia LDR.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

## Características Luxómetro BH1750

LUXÓMETRO BH1750
Interfaz digital a través de bus I2C con capacidad de seleccionar entre 2 direcciones
Respuesta espectral similar a la del ojo humano
Realiza mediciones de luz y convierte el resultado a una palabra digital
Amplio rango de medición 1-65535 lux
Modo de bajo consumo de energía
Rechazo de ruido a 50/60 Hz
Baja dependencia de la medición contra la fuente de luz: halógeno, led, incandescente, luz de día, etc.
Es posible seleccionar dos direcciones de esclavo (I2C).
La influencia del espectro infrarrojo es baja.
Voltaje 3.3v-5v
Librería BH1750

Figura 30. Características Luxómetro BH1750.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### **Análisis de Priorización Fotorresistencia LDR y Luxómetro BH1750**

En la figura 29 y 30, se muestra la fotorresistencia LDR y al Luxómetro se encargan de la detección del nivel de luz, tienen características que se aplican con el proyecto, la fotorresistencia es más común y sencilla, por afinidad y bajo costo se elige a la fotorresistencia LDR.

#### **2.1.4 Sensor de Vibración**

Sirve para detectar movimiento ocasionado por vibraciones.

##### **2.1.4.1 Piezoeléctrico**

Se utiliza para la detección de vibraciones, se observa en la figura 31, las características principales del componente. La placa piezocerámica responde a cambios de flexión, tacto, vibración y choque con un cambio en la salida analógica. (ALIBABA, 2018)

### 2.1.4.2 Sensor de vibración SW-420

El SW-420 detecta las vibraciones a partir de un umbral, el cual se puede regular con el potenciómetro que incluye, se observa en la figura 32, las características principales del componente citado. Consta de un módulo del sensor SW-420 y un comparador LM393 produciendo una señal TTL cuando se activa. Se utiliza en alarmas para motos y coches, detector de terremotos, etc. (Nextia Fenix, 2018)

Características:

Características del Piezoeléctrico

PIEZOELÉCTRICO	
Frecuencia de Resonancia:	6.3 ± 0.6KHz
Impedancia:	1000Ω máx
Capacitancia:	10.0±30 nF.
Material:	Latón.
Longitud del cable:	50mm (cable tipo AWG32).

Figura 31. Características del Piezoeléctrico.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Características SW-420

SW-420	
Comparador	LM393
Voltaje de operación	3.3V/5.0V
Corriente de operación (LM393)	15mA
Salida digital	· Nivel alto: 1
	· Nivel bajo: 0
Agujero	Instalar en cualquier superficie
Indicador de alimentación de tensión	Incluido
LED Señal indicadora	Incluido

Figura 32. Características del SW-420.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.



## **Análisis del Piezoeléctrico y SW-420**

Estos dos sensores son comerciales existe información de cada uno de ellos, tomando como componente idóneo para el prototipo el Piezoeléctrico, porque la toma de datos es más sencilla, se tiene mayor compatibilidad con librerías de Arduino y su costo es bajo.

### **2.1.5 Sensor de Sonido**

Componente fabricado para detectar los niveles de presión de sonido, la unidad de medición es en decibeles (dB).

#### **2.1.5.1 Sensor de Sonido FC 109**

Es un sensor pequeño, la señal es demasiado baja para ser registrada correctamente tiene un amplificador MAX9812 integrado, en la figura 33, se encuentran las características del componente. (LUIS LLAMAS, 2016)

#### **2.1.5.2 Sensor de Sonido y micrófono KY-038**

El uso habitual de este tipo de sensor no amplificado es emplear la salida digital para detectar el sonido cuando este supera un cierto umbral, regulado a través de un potenciómetro ubicado en la placa, en la figura 34, se encuentran las características del componente. (LUIS LLAMAS, 2016)

Características:

FC 109
Alimentación: 3.3 a 5 V DC
Permite detectar la intensidad de sonido del entorno
Incorpora un amplificador de audio MAX9812
Posee un orificio de instalación para facilitar su uso y montaje
Construido con un LED Indicador de alimentación
Conexión de 3 hilos
Dimensiones: 26 x 12 x10 mm

Figura 33. Características Sensor de Sonido FC 109.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### Características Sensor de Sonido KY-038

KY-038
Chip principal: LM393
Voltaje de trabajo: 4-6VDC
Indicador de señal de salida: Luz de indicación de encendido; Luz de salida del comparador
Señal de salida de un solo canal
Sensitividad ajustada por potenciómetro
La señal de salida efectiva es a bajo nivel (Activa en BAJO)
Cuando detecta sonido, la salida es activa en BAJO y el LED correspondiente enciende
desplegar una alarma de sonido y luz trabajando con un sensor fotosensitivo.
La distancia máxima de inducción es de 0,5 metros.

Figura 34. Características Sensor de Sonido y micrófono KY-038.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

## **Análisis de Sonido FC 109 y KY-038**

El análisis realizado, depende de la necesidad de detectar sonido o tener una medida, en cuyo caso se escoge el FC109 con su módulo amplificado MAX9812 este módulo amplificador es adaptable al prototipo.

### **2.1.6 Sensor Detección de líquidos**

Componentes construidos para mostrar las fugas de fluidos en maquinaria que trabaja con refrigerante.

#### ***2.1.6.1 Detección de líquidos Higrómetro FC-28***

Sensor fácil de implementar, permite medir la humedad presente en el suelo.  
(LUISLLAMAS, 2016)

Características:

HIGRÓMETRO FC-28
Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire
Un suelo ligeramente húmedo daría valores típicos de 600-700.
Un suelo seco tendrá valores de 800-1023.
La salida digital
Alimentación 5V

MODULO DETECTOR DE LIQUIDOS, SENSOR DE AGUA
Alimentación: 3 - 5V
Consumo: menor que 20mA
Detección analógica
Area de detección 40 x 16mm
Temperatura de trabajo: 10 - 30°C
Peso: 3.5g
Tamaño: 62 x 20 x 8mm.

Figura 35. Características Detección de líquidos Higrómetro FC-28 y módulo detector de Líquidos, Sensor de agua.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### **Análisis higrómetro fc-28**

Para el análisis del higrómetro FC-28, se observan las características en la figura 35, por sus especificaciones técnicas es el que se acopla al prototipo, para entregar los datos de manera digital.

#### **2.1.7 Sensor de Presencia**

Componente electrónico compuesto de sensores y detectores que responden a eventos como puede ser movimiento de una masa.

### ***2.1.7.1 Sensor PIR***

Sensor para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por ejemplo, usados en juguetes, aplicaciones domóticas o sistemas de seguridad. (Macho, 2017)

### ***2.1.7.2 Sensor magnético***

Detector sencillo y efectivo. Consta de dos partes:

- Encapsulado con red switch
- Encapsulado con imán

### ***2.1.7.3 Análisis de PIR y Sensor magnético***

Estos dos sensores son comerciales existe información de cada uno de ellos, realizando el respectivo análisis en base a las especificaciones técnicas que se observa en la figura 36, el sensor PIR cumple con los requerimientos para la implementación.

### **2.1.8 Sensor de Flujo de Aire de coeficiente de temperatura negativo**

Un termistor es un sensor de temperatura por resistencia. Su funcionamiento se basa en la variación de la resistividad que presenta un semiconductor con la temperatura.

#### ***2.1.8.1 Termistor NTC (Negative Temperature Coefficient)***

Coeficiente de temperatura negativo.

#### ***2.1.8.2 Termistor PTC (Positive Temperature Coefficient)***

Coeficiente de temperatura positivo (también llamado posistor).

Características:

Características Sensor de Flujo de Aire NTC	
NTC y PTC	
NTC: Aumenta la temperatura y disminuye la resistencia, termistor de coeficiente negativo.	
PTC: Cuando aumenta la temperatura también aumenta la resistencia, termistor de coeficiente positivo.	

Figura 36. Características Sensor de Flujo de Aire NTC.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### Análisis NTC Y PTC

Luego de analizar las características de los termistores (figura 36), se escoge el termistor NTC, por afinidad, el flujo es proporcional al enfriamiento y se aplica con el prototipo.

#### 2.1.9 Arduino

En las siguientes tablas se explica los cálculos que se realizaron para llegar a concluir la decisión de utilizar los componentes para el prototipo, tomando en cuenta criterios que sirven para comparar entre diferentes tecnologías, marcas y modelos.

Tabla 9. Criterios vs. Criterios

CRITERIOS VS CRITERIOS	PRECIO	DESEMPEÑO	TECNOLOGÍA	PROGRAMACIÓN	SUMA	PONDERACION
PRECIO		10	5	10	25	0,492
DESEMPEÑO	0,10		10,00	10	20,10	0,396
TECNOLOGÍA	0,20	0,10		5	5,30	0,104
PROGRAMACIÓN	0,10	0,10	0,20		0,40	0,008
					50,8	1

Nota: En la tabla 9 se realizó la comparativa entre precio, desempeño, tecnología y compatibilidad, siendo el precio el criterio de ponderación.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 10. Precio comparado con diferentes tecnologías.

PRECIO	PICS	ARDUINO			SUMA	PONDERACION
PICS		0,1000			0,1	0,010
ARDUINO	10,0000				10,00	0,990
					10,10	1,000

Nota: En la tabla 10, se realiza la comparativa entre diferentes tecnologías del mercado en factor del precio.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 11. Desempeño comparado con diferentes tecnologías.

DESEMPEÑO	PICS	ARDUINO			SUMA	PONDERACION
PICS		10,0000			10	0,990
ARDUINO	0,1000				0,10	0,010
					10,10	1,000

Nota: En la tabla 11, se realizó la comparativa entre diferentes tecnologías del mercado en factor al desempeño que tiene en procesamiento de información.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 12. Tecnología comparada con diferentes tecnologías del mercado.

DESEMPEÑO	PICS	ARDUINO			SUMA	PONDERACION
PICS		10,0000			10	0,990
ARDUINO	0,1000				0,10	0,010
					10,10	1,000

Nota: En la tabla 12, se realizó la comparativa de las tecnologías que ofrece el mercado en factor a la tecnología que emplea.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 13. Programación comparada con diferentes tecnologías del mercado

PROGRAMACIÓN	PICS	ARDUINO			SUMA	PONDERACION
PICS		0,1000			0,1	0,010
ARDUINO	10,0000				10,00	0,990
					10,10	1,000

Nota: En la tabla 13, se realizó la comparativa entre diferentes tecnologías del mercado en factor a la facilidad de dar las instrucciones y comandos (programación).

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 14. Resultados de la tabla de criterios en factor a las opciones

CRITERIOS VS OPCIONES	PRECIO	DESEMPEÑO	TECNOLOGÍA	PROGRAMACIÓN	SUMA	PONDERACION
PICS	0,0049	0,3918	0,0010	0,0001	0,39286271	0,395
ARDUINO	0,4873	0,0039	0,1033	0,0078	0,60	0,605
					1,00	1,000

Nota: En la tabla 14, Los resultados de las tecnologías del mercado en factor a las opciones que fueron investigadas, tomándose como opción a la placa Arduino.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

## 2.1.10 Raspberry

Tabla 15. Criterios vs. Criterios

CRITERIOS VS CRITERIOS	PRECIO	DESEMPEÑO	TECNOLOGÍA	COMPATIBILIDAD	SUMA	PONDERACION
PRECIO		10	5	10	25	0,545
DESEMPEÑO	0,10		0,20	10	10,30	0,224
TECNOLOGÍA	0,20	5,00		5	10,20	0,222
COMPATIBILIDAD	0,10	0,10	0,20		0,40	0,009
					45,9	1

Nota: En la tabla 15, se realizó la comparativa entre precio, desempeño, tecnología y compatibilidad, siendo el precio el criterio de ponderación.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 16. Precio comparado con diferentes tecnologías.

PRECIO	RASPERRY PI 3	HUMMING BOARD	BANANA PI	BEAGLE BONE	SUMA	PONDERACION
RASPERRY PI 3		10	10	10	30	0,591
HUMMING BOARD	0,10		0,20	0,1	0,40	0,008
BANANA PI	0,10	5,00		0,2	5,30	0,104
BEAGLE BONE	0,10	10,00	5,00		15,10	0,297
					50,8	1

Nota: En la tabla 16, se realizó la comparativa entre diferentes tecnologías del mercado en factor del precio.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 17. Desempeño comparado con diferentes tecnologías.

DESEMPEÑO	RASPERRY PI 3	HUMMING BOARD	BANANA PI	BEAGLE BONE	SUMA	PONDERACION
RASPERRY PI 3		10	10	10	30	0,495
HUMMING BOARD	0,10		0,10	0,1	0,30	0,005
BANANA PI	0,10	10,00		0,1	10,20	0,168
BEAGLE BONE	0,10	10,00	10,00		20,10	0,332
					60,6	1

Nota: En la tabla 17, se realizó la comparativa entre diferentes tecnologías del mercado en factor al desempeño que tiene en procesamiento de información.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 18. Tecnología comparada con diferentes tecnologías del mercado.

TECNOLOGIA	RASPERRY PI 3	HUMMING BOARD	BANANA PI	BEAGLE BONE	SUMA	PONDERACION
RASPERRY PI 3		10	10	10	30	0,654
HUMMING BOARD	0,10		5,00	5	10,10	0,220
BANANA PI	0,10	0,20		5	5,30	0,115
BEAGLE BONE	0,10	0,20	0,20		0,50	0,011
					45,9	1

Nota: En la tabla 18, se realiza la comparativa de las tecnologías que ofrece el mercado en factor a la tecnología que emplea.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 19. Compatibilidad comparada con las tecnologías del mercado.

COMPATIBILIDAD	RASPERRY PI 3	HUMMING BOARD	BANANA PI	BEAGLE BONE	SUMA	PONDERACION
RASPERRY PI 3		10	10	10	30	0,654
HUMMING BOARD	0,10		5,00	5	10,10	0,220
BANANA PI	0,10	0,20		5	5,30	0,115
BEAGLE BONE	0,10	0,20	0,20		0,50	0,011
					45,9	1

Nota: En la tabla 19, se realiza la comparativa entre las tecnologías en factor a la compatibilidad con componentes del medio.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 20. Resultados de criterios en factor a las opciones

CRITERIOS VS OPCIONES	PRECIO	DESEMPEÑO	TECNOLOGIA	COMPATIBILIDAD	SUMA	PONDERACION
RASPERRY PI 3	0,32165097	0,11108954	0,145243282	0,006	0,262	0,386
HUMMING BOARD	0,00	0,00	0,05	0,002	0,06	0,083
BANANA PI	0,06	0,04	0,03	0,001	0,12	0,179
BEAGLE BONE	0,16	0,07	0,00	0,000	0,24	0,352
					0,678	1

Nota: En la tabla 20, los resultados de las tecnologías del mercado en factor a las opciones que se comercializan en el mercado, tomándose como opción a Raspberry Pi 3 Modelo B.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

## 2.2 Análisis software

Tabla 21. Criterios vs. Criterios

CRITERIOS VS CRITERIOS	LENGUAJE	ANÁLISIS	TECNOLOGÍA	COMPATIBILIDAD	SUMA	PONDERACION
LENGUAJE		10	5	10	25	0,545
ANÁLISIS	0,10		0,20	10	10,30	0,224
TECNOLOGÍA	0,20	5,00		5	10,20	0,222
COMPATIBILIDAD	0,1	0,1	0,2		0,4	0,009
					45,9	1,000

Nota: En la tabla 21, se realiza la comparativa entre los criterios lenguaje de programación y consulta, análisis de datos, tecnología y compatibilidad, siendo el lenguaje de programación el criterio importante.

Elaborado por: Miguel Ángel Rojas y Maritza Tua

Tabla 22. Lenguaje de programación comparado con diferentes bases de datos.

LENGUAJE	MARIA DB	POSTGRESQL	SQL SERVER	ORACLE	SUMA	PONDERACION
MARIA DB		10	10	10	30	0,591
POSTGRESQL	0,10		0,20	0,1	0,40	0,008
SQL SERVER	0,10	5,00		0,2	5,30	0,104
ORACLE	0,1	10	5		15,1	0,297
					50,8	1,000

Nota: En la tabla 22, se realiza la comparativa entre el lenguaje de programación con las bases de datos existentes.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 23. Análisis de datos comparado con diferentes bases de datos.

ANALISIS	MARIA DB	POSTGRESQL	SQL SERVER	ORACLE	SUMA	PONDERACION
MARIA DB		10	10	10	30	0,495
POSTGRESQL	0,10		0,10	0,1	0,30	0,005
SQL SERVER	0,10	10,00		0,1	10,20	0,168
ORACLE	0,1	10	10		20,1	0,332
					60,6	1,000

Nota: En la tabla 23, se observa la comparativa entre la velocidad de análisis de datos de las bases de datos existentes.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 24. Tecnología usada comparada con las diferentes bases de datos.

TECNOLOGIA	MARIA DB	POSTGRESQL	SQL SERVER	ORACLE	SUMA	PONDERACION
MARIA DB		10	10	10	30	0,654
POSTGRESQL	0,10		5,00	5	10,10	0,220
SQL SERVER	0,10	0,20		5	5,30	0,115
ORACLE	0,1	0,2	0,2		0,5	0,011
					45,9	1,000

Nota: En la tabla 24, se observa la comparativa entre la tecnología usada de las bases de datos existentes.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 25. Comparativa entre la compatibilidad con herramientas de desarrollo y las diferentes bases de datos.

COMPATIBILIDAD	MARIA DB	POSTGRESQL	SQL SERVER	ORACLE	SUMA	PONDERACION
MARIA DB		10	10	10	30	0,654
POSTGRESQL	0,10		5,00	5	10,10	0,220
SQL SERVER	0,10	0,20		5	5,30	0,115
ORACLE	0,1	0,2	0,2		0,5	0,011
					45,9	1,000

Nota: En la tabla 25, se observa la comparativa entre la compatibilidad que posee con herramientas de desarrollo con las bases de datos existentes.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.



Tabla 26. Resultados de la tabla de criterios en factor a las opciones

CRITERIOS VS OPCIONES	LENGUAJE	ANALISIS	TECNOLOGIA	COMPATIBILIDAD	SUMA	PONDERACION
MARIA DB	0,32165097	0,11108954	0,145243282	0,005695815	0,262028638	0,386
POSTGRESQL	0,00	0,00	0,05	0,001917591	0,06	0,083
SQL SERVER	0,06	0,04	0,03	0,001006261	0,12	0,179
ORACLE	0,161897655	0,074429992	0,002420721	0,00	0,238843299	0,352
					0,67834903	1,000

Nota: En la tabla 26, para los resultados de las bases de datos existentes en factor a las opciones del mercado, tomando como opción a la base de datos MariaDB.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

## CAPÍTULO 3

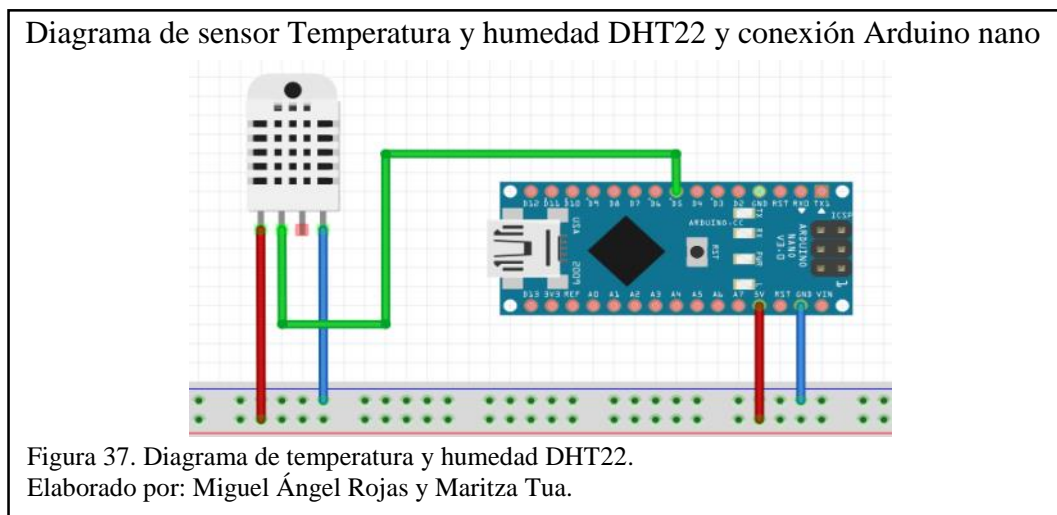
### DESARROLLO Y DISEÑO

A continuación, se muestra los diagramas y códigos implementados en el desarrollo del prototipo.

#### 3.1 Sensores

Un sensor es un dispositivo capaz de detectar acciones o estímulos, a continuación, se detallan los diagramas de los utilizados en el prototipo en la figura 37 y el código a implementarse en el prototipo (figura 38).

#### Diagrama Sensor de temperatura y humedad DHT22



## Código de la programación del sensor de temperatura y humedad DHT22

Código para inicializar y obtener datos del sensor DHT22

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN      5          // Pin which is connected to the DHT sensor.
#define DHTTYPE     DHT22     // DHT 22 (AM2302)
DHT_Unified dht(DHTPIN, DHTTYPE);
#include <SoftwareSerial.h>
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  dht.begin();
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  dht.humidity().getSensor(&sensor);
}

void loop() {
  sensors_event_t event; //evento
  dht.temperature().getEvent(&event); //obtenemos valor almacenado
  Serial.print("Temperatura Módulo 1: ");
  Serial.print(event.temperature); //impresion valor temperatura
  Serial.println(" °C");

  dht.humidity().getEvent(&event);
  var1 = event.relative_humidity; //impresion valor humedad
  Serial.print("Humedad Módulo 1: ");
  Serial.print(var1);
}
```

Figura 38. Código para inicializar y obtener datos del sensor DHT22.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Se detallan dos partes para la obtención de datos del sensor DHT22.

Parte 1: Se instalan las librerías necesarias (figura 39):

### Librerías utilizadas

```
Adafruit_Sensor.h
#include <DHT.h>
#include <DHT_U.h>
#include <SoftwareSerial.h> //librería simular puerto virtual
```

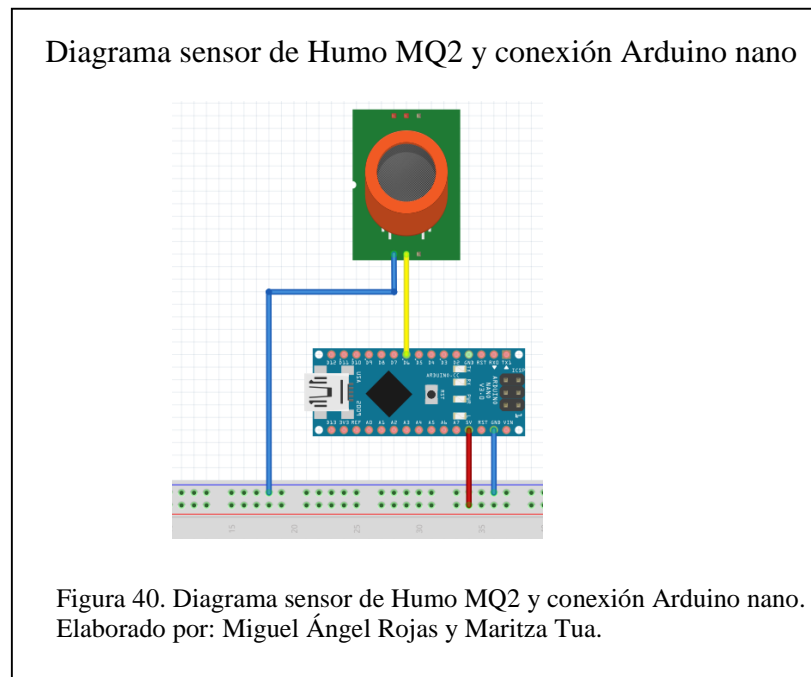
Figura 39 Librerías utilizadas.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Se define el tipo de sensor con el que se implementará el prototipo, (#define DHTTYPE DHT22), cada una de las librerías mencionadas son fundamentales para la

lectura correcta del sensor, seleccionar el pin al que se conecta el sensor al Arduino nano es igual de importante, se inicia la comunicación Serial, como empieza en begin (9600) y por último se inicializa el sensor, se puede observar en la figura 39.

Parte 2: Se crea un evento donde se almacenarán los datos obtenidos, por último, se carga el programa, en la figura 39, se explica parte del código.

### 3.1.1 Diagrama Sensor de Humo MQ2



### Código de la programación para utilizar el sensor de Humo MQ2

Código sensor Humo - MQ2

```
sensor = digitalRead(humo);  
Serial.print("Sensor HUMO: ");  
Serial.println(sensor);
```

Figura 41. Código sensor Humo - MQ2.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Se realiza la obtención de datos digital de 0 y 1 del sensor de humo esto quiere decir que hay o no humo detectado, ver figura 41.

### 3.1.2 Diagrama sensor para medir nivel de luz NTC GL5549

Diagrama sensor de Nivel de Luz LDR GL5549 y conexión Arduino nano

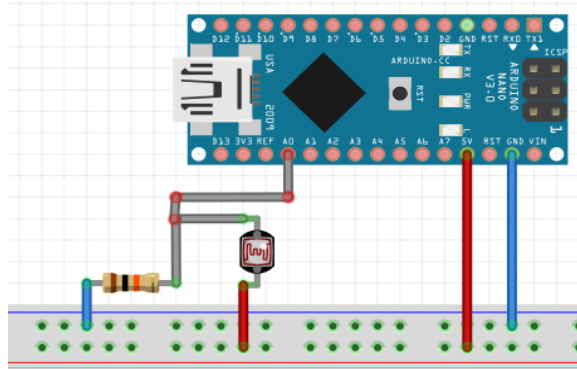


Figura 42. Diagrama sensor de nivel de luz LDR GL5549 y conexión Arduino nano.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 42, se observa la conexión de un módulo Arduino, a una resistencia y a la fotocelda. Se realizan las conexiones, según corresponde y se implementa una resistencia adicional. En la figura 43 se observa la instancia de los sensores, por medio de Arduino IDE.

### Código de la programación para medir el nivel de luz NTC GL5549

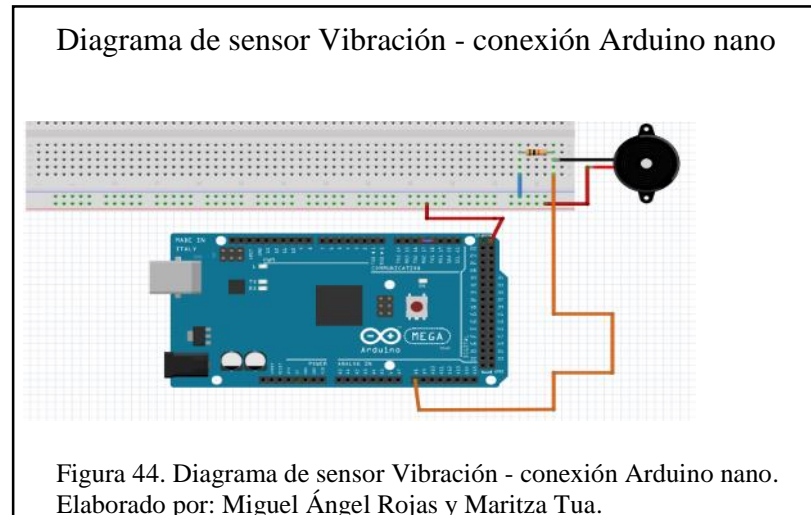
Código sensor de Nivel de Luz LDR - GL5549

```
sensor = analogRead(0);  
Serial.print("Sensor Luz: ");  
Serial.println(sensor);
```

Figura 43. Código sensor de Nivel de Luz LDR - GL5549.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.1.3 Diagrama sensor Vibración

En la figura 44, se observa la conexión con el sensor de vibración al módulo Arduino, con la ayuda de una resistencia, en la figura 45 se encuentra el código para extraer el valor del sensor, que lo toma del entorno.



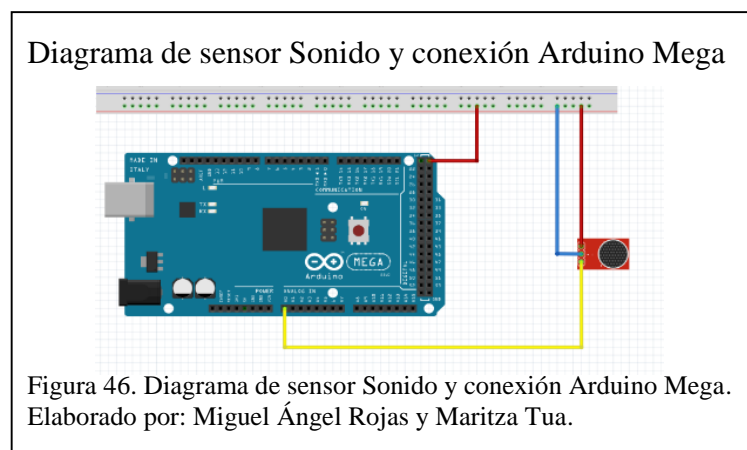
### Código de la programación para el sensor de vibración

Código sensor de Vibración

```
sensor = analogRead(vibma); //lectura analogica de vibración
Serial.print("Sensor de sonido : ");
Serial.println(sensor);
```

Figura 45. Código sensor de Vibración.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.1.4 Diagrama sensor de Sonido



En la figura 46, la conexión del sensor de sonido a la placa Arduino, además en la figura 47, el código para extraer el dato del sensor.

### Código de la programación para el sensor de sonido

Código sensor de Sonido

```
sensor = analogRead(sonidoma); //lectura analogica de sonido
Serial.print("Sensor de sonido : ");
Serial.println(sensor);
```

Figura 47. Código sensor de Sonido.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.1.5 Diagrama del sensor para detección de líquidos

En la figura 48, se tiene la conexión del sensor de líquidos con la placa Arduino, y en la figura 49, se observa las sentencias de programación para obtener el valor.

Diagrama de sensor para detección de líquidos y conexión Arduino nano

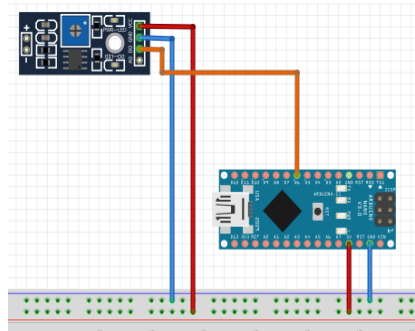


Figura 48. Diagrama de sensor de detección de líquidos.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### Código de la programación del sensor para detección de líquidos

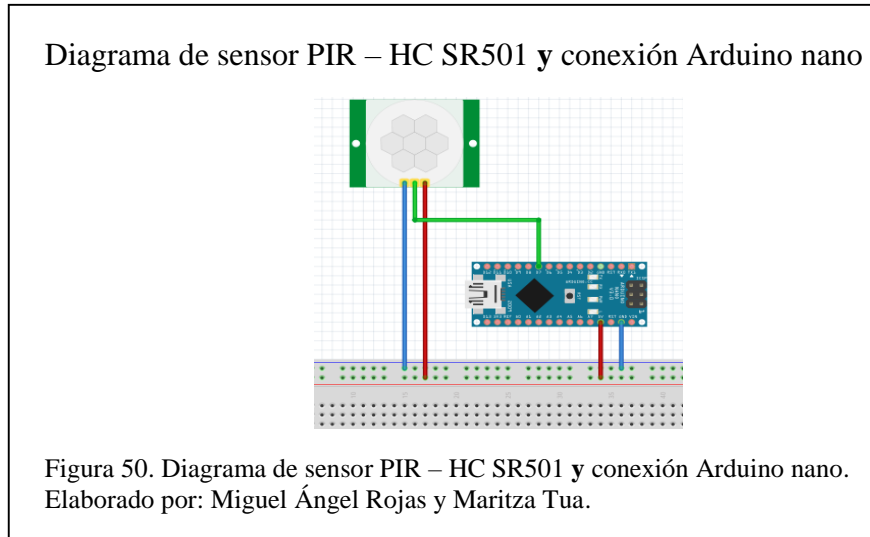
Código para el sensor para detección de líquidos

```
sensor = digitalRead(liq);
Serial.print("Sensor Liquido: ");
Serial.println(sensor);
```

Figura 49. Código para sensor para detección de líquidos.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.1.6 Diagrama sensor PIR

En la figura 50, se observa el sensor de presencia conectado a la placa Arduino, y en la figura 51, las líneas de código para la lectura de la señal digital.



### Código de la programación para el sensor PIR (Presencia)

Código para detectar presencia (sensor PIR)

```
sensor1 = digitalRead(pir);  
Serial.print("Sensor PIR: ");  
Serial.println(sensor1);
```

Figura 51. Código para detectar Presencia sensor PIR.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.1.7 Diagrama sensor Flujo de Aire

En la figura 52, se observa el diagrama del sensor que realiza la medición del flujo de aire, apoyado por una resistencia se conecta con la placa Arduino. En la figura 53, se encuentra el código que se usó para captar el valor.



Diagrama de sensor Flujo de Aire y conexión Arduino nano

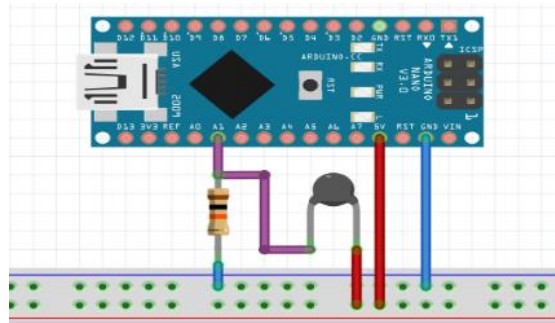


Figura 52. Diagrama de sensor Flujo de Aire y conexión Arduino nano. Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### Código de la programación para el sensor de flujo de aire

#### Código Flujo de aire

```
sensor1 = analogRead(1);  
Serial.print("Sensor Flujo de Aire: ");  
Serial.println(sensor1);
```

Figura 53. Código Flujo de aire. Elaborado por: Maritza Tua y Miguel Ángel Rojas.

### 3.2 Módulos

En el desarrollo del prototipo para la detección de amenazas físicas se diseñaron varios módulos distribuidos de forma estratégica en el espacio físico del centro de datos, según requerimiento del administrador. Para la comunicación entre módulos, se lo realiza por medio de dispositivos bluetooth, que envían a un módulo coordinador maestro que se encarga de orquestar los datos y procesarlos.

Tabla 27. Módulos, ubicación de sensores y cálculos.

MODULOS	UBICACIÓN ESTRATEGICA	SENSORES			CALCULO
1	Superior Puerta	DHT22 Temperatura y Humedad	MQ2 Humo	PIR Presencia	Punto de Rocío
2	Atrás Suelo	DHT22 Temperatura y Humedad	HIGROMETRO FC-28 Detector de Líquidos	-----	Punto de Rocío
3	Atrás superior	DHT22 Temperatura y Humedad	----	-----	Punto de Rocío
4	Adelante Rack	DHT22 Temperatura y Humedad	LDR GL5549 Nivel de Luz	PTC Flujo de Aire	Punto de Rocío
5	Atrás Rack	DHT22 Temperatura y Humedad	FC 109 Sonido	PIEZO ELECTRICO Vibración	Punto de Rocío

Nota: En la tabla 27, se realizó el cálculo del Punto de Rocío, el que se obtiene mediante la fórmula:

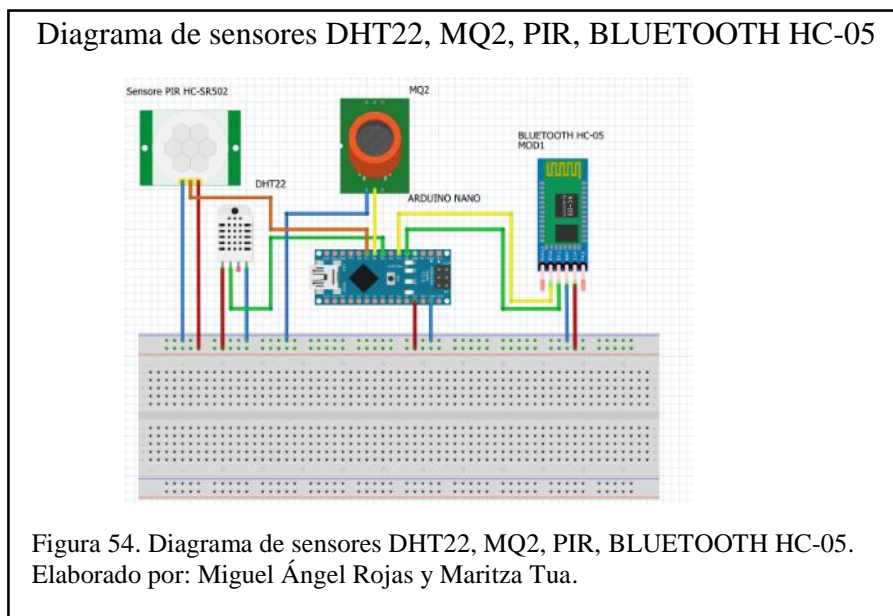
$P_r$  = Punto de rocío;  $T$  = Temperatura en grados;  $H$  = Humedad relativa;

$$P_r = \sqrt[8]{\frac{H}{100}} (112 + 0.9 * T) + (0.1 * T) = 112.$$

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.2.1 Diagrama Módulo 1

En la figura 54, el diagrama con las conexiones de los sensores de temperatura y humedad DHT22, presencia (PIR), sensor de humo MQ2, bluetooth HC-05, con la placa Arduino.



## Código de la programación del módulo 1

En la figura 55, se observa el código codificado en Arduino IDE, destacando el envío de la trama por medio del módulo bluetooth esclavo al módulo bluetooth maestro, para diferenciar se realizó un algoritmo con cabecera y cola, para seccionar en donde empieza la trama de envío del dato de un sensor y donde termina.

### Código para el envío y recepción de la trama.

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN      5           // Pin de entrada
//#define DHTTYPE    DHT11     // DHT 11
#define DHTTYPE    DHT22     // DHT 22 (AM2302)
//#define DHTTYPE    DHT21     // DHT 21 (AM2301)
DHT_Unified dht(DHTPIN, DHTTYPE);

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX
int adc = 0;
int humo = 6;
int pir = 7;
int sensor = 0;
int sensor1 = 0;
int var = 0;
int var1 = 0;

void loop() {

  sensors_event_t event;
  dht.temperature().getEvent(&event);
  Serial.print("Temperatura Módulo 1: ");
  Serial.print(event.temperature);
  Serial.println(" °C");
  var = event.temperature;
  mySerial.print("/"); // parámetro de trama de inicio de toma de datos
  mySerial.print(var); // dato temperatura
  mySerial.println('|'); //parámetro de trama de de fin de toma de datos
  delay(10);
  // así se toman los dato /datotemperatura|

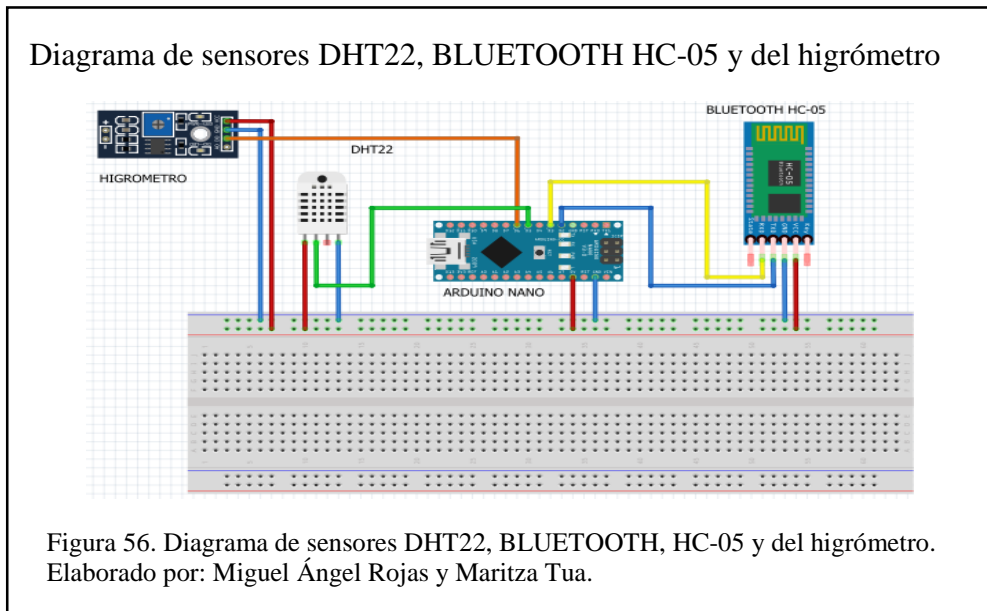
  dht.humidity().getEvent(&event);
  var1 = event.relative_humidity;
  Serial.print("Humedad Módulo 1: ");
  Serial.print(var1);
  Serial.println("%"); // parámetro de trama de inicio de toma de datos
  mySerial.print('*');
  mySerial.print(event.relative_humidity);
  mySerial.println('+');
  delay(10);

  float rocio,r1,r2,r3;
  int r4; //variabe de almacenamiento valor punto de rocio
  r1=var1/100; //calculo punto de rocio
```

Trama asignada a  
temperatura y  
humedad (/ , |, %, +)

Figura 55. Código módulo 1.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.2.2 Diagrama Módulo 2



En la figura 56, se observa el diagrama de conexiones del sensor de temperatura y humedad, el detector de humedad, el bluetooth a la placa Arduino. También en la figura 57, el código con la trama de envío y recepción de datos antes descrita.

### Código de la programación del módulo 2

```
Código módulo 2

pinMode(liq, INPUT_PULLUP);//sensor liquido como entrada
//pinMode(pir, INPUT_PULLUP);

//Serial.println("Goodnight moon!");
Serial.println("UNIVERSIDAD POLITECNICA SALESIANA");
Serial.println("MODULO 2");
Serial.println("DATOS");
Serial.println("TEMPERATURA");
Serial.println("HUMEDAD");
Serial.println("ROCIO");
Serial.println("LIQUIDO");
//Serial.println("PIR");

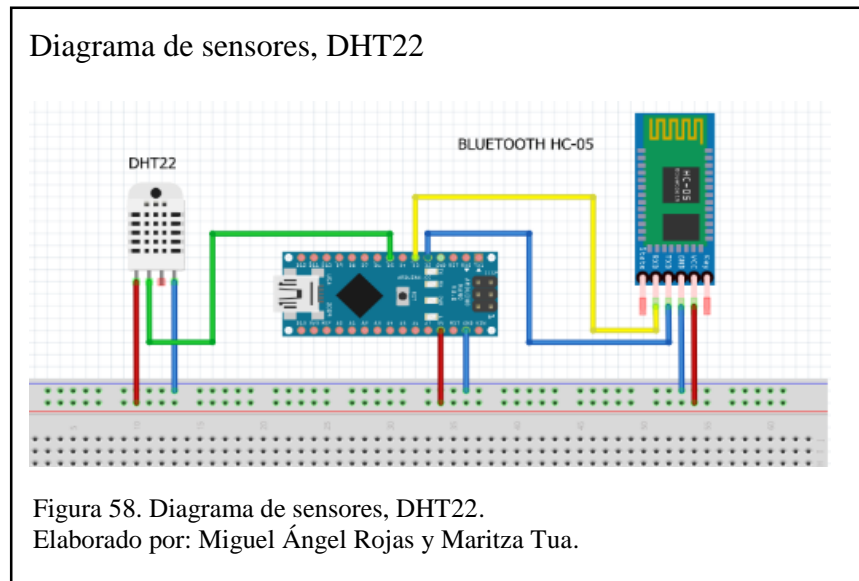
// set the data rate for the SoftwareSerial port
mySerial.begin(9600);//velocidad en baudios
//mySerial.println("Hello, world?");
}

sensor = digitalRead(liq);//lectura digital sensor liquido
Serial.print("Sensor liquido: ");
Serial.println(sensor);
mySerial.print("-");//envio del valor del sensor liquido
if (sensor == 0) mySerial.print("22"); //hay
if (sensor == 1) mySerial.print("44"); //no hay
mySerial.println('%');
delay(10);
```

Trama asignada  
detección de líquidos  
(-, %)

Figura 57. Código de la programación módulo 2. Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.2.3 Diagrama Módulo 3



### Código de la programación del Módulo 3

Código módulo 3

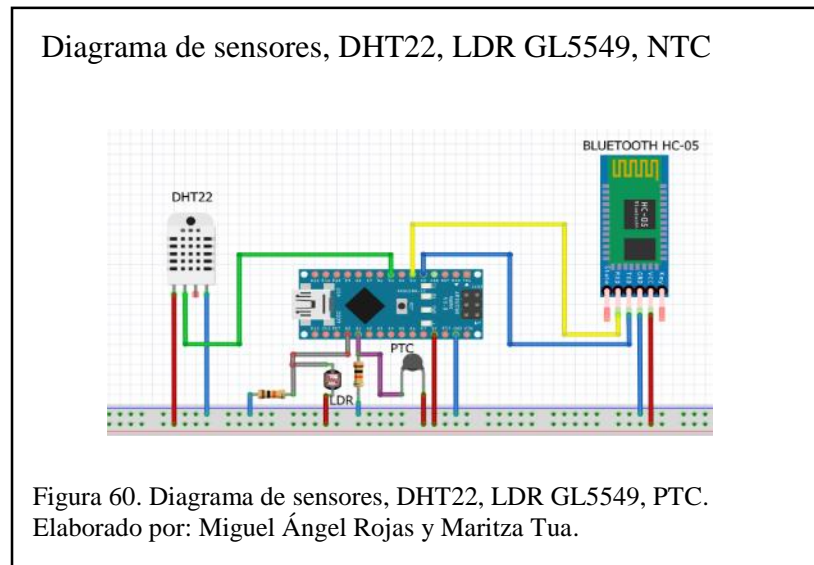
```
float rocio, r1, r2, r3;//punto de rocio
int r4;
r1 = var1 / 100;
r2 = 1 / 8;
r3 = pow(r1, r2);
r4 = r3 * (110 + var) - 110;
Serial.print("Punto de Rocio Modulo 3: ");
Serial.print(r4);
Serial.println("%");
mySerial.print('=');
mySerial.print(r4);//envio del punto de rocio
mySerial.println('?');
```

Trama asignada a punto de rocío (% ,=,?)

Figura 59. Código Módulo 3.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 58, el diagrama de conexiones del módulo tres del prototipo, del sensor de temperatura y humedad y bluetooth conectado a la placa Arduino. En la figura 59, el código del algoritmo de envío y recepción de datos por los sensores.

### 3.2.4 Diagrama Módulo 4



### Código de la programación del módulo 4

Código módulo 3

```
sensor = analogRead(0); //lectura analoga del sensor de luz LDR
Serial.print("Sensor Luz: ");
Serial.println(sensor);

mySerial.print('-');
mySerial.print(sensor);//envio del valor LDR
mySerial.println('*');
delay(10);

sensor1 = analogRead(1); // lectura analoga del sensor NTC
Serial.print("Sensor Flujo de Aire: ");
Serial.println(sensor1);
mySerial.print('*');
mySerial.print(sensor1);//envio del valor NTC
mySerial.println('*');
delay(10);

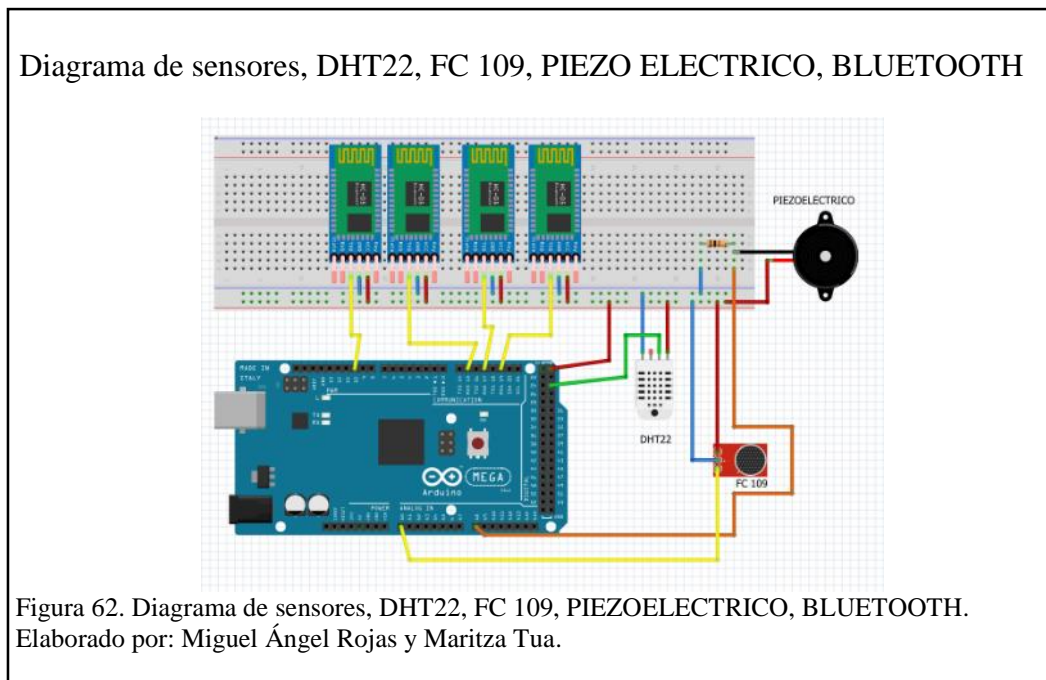
}
```

Trama asignada sensor de luz LDR y flujo de aire para el flujo de aire NTC (-, %, #,)

Figura 61. Código Módulo 5.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 60, está el diagrama de conexiones del cuarto módulo, los sensores conectados a la placa Arduino, y en la figura 61, el algoritmo elaborado para el envío y transmisión de datos de los módulos esclavos al módulo maestro o coordinador.

### 3.2.5 Diagrama módulo 5



En la figura 62, el diagrama de conexiones del módulo maestro conectado con los sensores y en la figura 63, el código escrito con el algoritmo creado, para realizar el envío y recepción de los datos.

## Código de la programación del módulo 5

### Diagrama módulo 5

```
MASTER_FFFun Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
MASTER_FFFun
envio();
//Serial.println("");
if ((contador > 150) && (contador < 300))
{
  //Serial2
  if (Serial2.available()) //modulo
  {
    incomingByte2 = Serial2.read();
    //Serial.println(incomingByte1);
    if ((incomingByte2 == '/') || (incomingByte2 == '*') || (incomingByte2 == '-') || (incomingByte2 == '#') || (incomingByte2 == '='))
    {
      if (incomingByte2 == '/')xbee2 = 1;//cierra con |
      if (incomingByte2 == '*')xbee2 = 2;//cierra con +
      if (incomingByte2 == '-')xbee2 = 3;//cierra con ?
      if (incomingByte2 == '#')xbee2 = 4;//cierra con %
      if (incomingByte2 == '=')xbee2 = 5;//cierra con )
    }
    started2 = true;
    index2 = 0;
    msg2[index2] = '\0'; // Throw away any incomplete packet
  }
  //End the message when the '>' symbol is received
  else if ((incomingByte2 == '|') || (incomingByte2 == '+') || (incomingByte2 == '%') || (incomingByte2 == '?') || (incomingByte2 == '?'))
  {
    ended2 = true;
  }
}

MASTER_FFFun Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
MASTER_FFFun
index2 = 0;
msg2[index2] = '\0'; // Throw away any incomplete packet
//End the message when the '>' symbol is received
else if ((incomingByte2 == '|') || (incomingByte2 == '+') || (incomingByte2 == '%') || (incomingByte2 == '?') || (incomingByte2 == '?'))
{
  ended2 = true;
  // break; // Done reading - exit from while loop!
}
//Read the message!
else
{
  if (index2 < 3) // Make sure there is room
  {
    msg2[index2] = incomingByte2; // Add char to array
    index2++;
    msg2[index2] = '\0'; // Add NULL to end
  }
}
if (started2 && ended2)
{
  int valueXbee2 = atoi(msg2);
  //Serial.print("*****");
  //Serial.println(valueXbee2); //Only for debuggingmkdfjda
  index2 = 0;
}

MASTER_FFFun Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
MASTER_FFFun
if (Serial.available()) //modulo
{
  incomingByte4 = Serial.read();
  //Serial.println(incomingByte1);
  if ((incomingByte4 == '/') || (incomingByte4 == '*') || (incomingByte4 == '-') || (incomingByte4 == '#') || (incomingByte4 == '='))
  {
    if (incomingByte4 == '/')xbee4 = 1;//cierra con |
    if (incomingByte4 == '*')xbee4 = 2;//cierra con +
    if (incomingByte4 == '-')xbee4 = 3;//cierra con ?
    if (incomingByte4 == '#')xbee4 = 4;//cierra con %
    if (incomingByte4 == '=')xbee4 = 5;//cierra con )
  }
  started4 = true;
  index4 = 0;
  msg4[index4] = '\0'; // Throw away any incomplete packet
}
//End the message when the '>' symbol is received
else if ((incomingByte4 == '|') || (incomingByte4 == '+') || (incomingByte4 == '%') || (incomingByte4 == '?') || (incomingByte4 == '?'))
{
  ended4 = true;
  // break; // Done reading - exit from while loop!
}
//Read the message!
else
{
  if (index4 < 3) // Make sure there is room
```

Figura 63. Diagrama módulo 5.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.



### 3.2.6 Programación del módulo GSM

#### Código para enviar mensaje en caso de existir alarma

```
void mensaje_sms()
{
  //mediante comando at para envio de sms
  Serial.println("Enviando SMS ...");
  //Serial.println("AT+CMGF=1\r"); //Comando AT para mandar un SMS
  mySerial.print("AT+CMGF=1\r");

  delay(1000);
  //0987599704
  mySerial.println("AT+CMGS=\"0987599704\"");
  delay(1000);
  mySerial.println(Mensaje); // Texto del SMS
  delay(100);
  mySerial.println((char)26); //Comando de finalizacion ^2
  delay(5000);
  mySerial.println();
  delay(10000);
  Serial.println("Enviado ...");

  llamar();
}

void llamar()
{
  //envio de comandos at para llamada
  Serial.println(F("Realizando llamada..."));
}
```

Figura 64. Código para enviar mensaje en caso de existir alarma.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 64, el código de programación del módulo GSM, para iniciar una llamada y en la figura 65, enviar un mensaje corto de texto (SMS).

### 3.2.7 Programación para realizar una llamada vía celular

#### Código para hacer una llamada en caso de existir alarma

```
void llamar()
{
  //envio de comandos at para llamada
  Serial.println(F("Realizando llamada..."));
  mySerial.println("ATD0987599704;"); //Comando AT para realizar una llamada
  delay(15000); // Espera 20 segundos mientras realiza la llamada
  mySerial.println("ATH"); // Cuelga la llamada
  delay(1000);
  Serial.println("Llamada finalizada");
  aux_gsm = 0;
  cont_gsm = 0;
}

void envio() {
  //envio de datos mediante el formato json a la web
  Serial.print("{\"humml\":");
  Serial.print(humeml);
}
```

Figura 65. Código para hacer una llamada en caso de existir alarma.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.2.8 Programación para el envío de datos mediante el formato JSON a la web

Código de Envío de datos mediante el formato JSON a la web

```
void envio() {  
  //envío de datos mediante el formato json a la web  
  Serial.print("{\"humml\":\"");  
  Serial.print(humeml);  
  Serial.print(",");  
  Serial.print("{\"tempml\":\"");  
  Serial.print(tempml );  
  
  Serial.print(",");  
  Serial.print("{\"roml\":\"");  
  Serial.print(rocioml );  
  
  Serial.print(",");  
  Serial.print("{\"humoml\":\"");  
  Serial.print(humoml );  
  
  Serial.print(",");  
  Serial.print("{\"pir\":\"");  
  Serial.print(pirml );  
}
```

Figura 66. Código para el envío de datos mediante el formato JSON a la web. Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 66, el código para el envío y recepción de datos, en formato JSON.

### 3.2.9 Diagrama del módulo 5, Raspberry Pi3 y GSM

Diagrama del módulo 5, Raspberry Pi3 y GSM, DHT22, FC 109, Piezoeléctrico

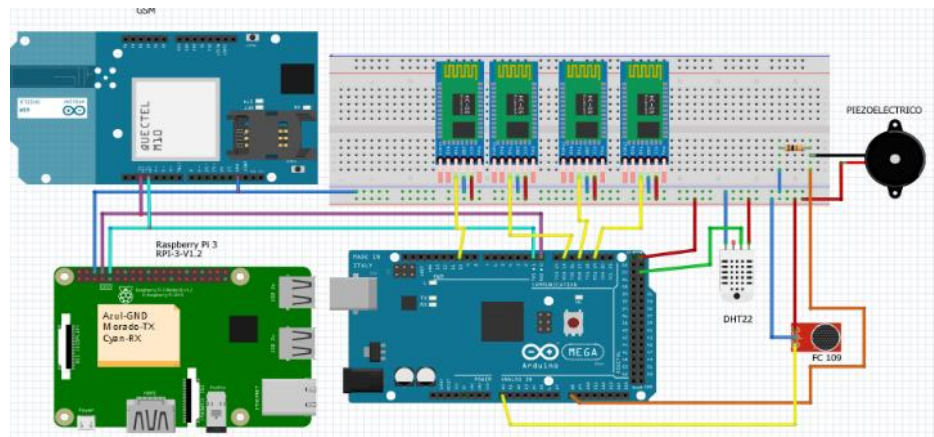


Figura 67. Diagrama módulo 5, Raspberry Pi3 y GSM, DHT22, FC-109, Piezoeléctrico. Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 67, el diagrama del prototipo con los componentes Arduino, módulo Shield GSM, los bluetooth, etc.

### 3.2.9.1 Programación de Raspberry Pi

Para iniciar los servicios de Raspberry Pi, se debe instalar algunos programas como los que se describen a continuación

1. En la página web oficial (<https://www.raspberrypi.org/downloads/raspbian/>) se procede a la descarga de la última versión de Raspbian, sistema operativo oficial de Raspberry.
2. El instalador de Raspbian debe guardarse en una tarjeta de memoria SD que sea mayor a 4GB (GigaBytes) de capacidad.
3. Se enciende la Raspberry y se instala el sistema operativo, siguiendo los pasos que muestra en pantalla.
4. Para instalar Mysql Server con la sentencia `sudo apt-get install mysql-server`
5. Se actualiza el repositorio de las librerías `sudo nano /etc/apt/source.list`
6. Se realiza la instalación de PHP, con los comandos listados en la figura 68.

Comandos para instalación de componentes en Raspbian.

```
sudo apt-get install php-curl
sudo apt-get install php-gd
sudo apt-get install php-imap
sudo apt-get install php-json
sudo apt-get install php-mcrypt
sudo apt-get install php-opcache
sudo apt-get install php-xmlrpc
sudo apt-get install php-mysql
```

Figura 68. Comandos para instalación de componentes en Raspbian.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

7. Se instala apache 2 `sudo apt-get install apache2`

8. Para la comunicación entre apache y php, como se observa en la figura 69.

**Comandos para comunicación entre Apache y PHP**

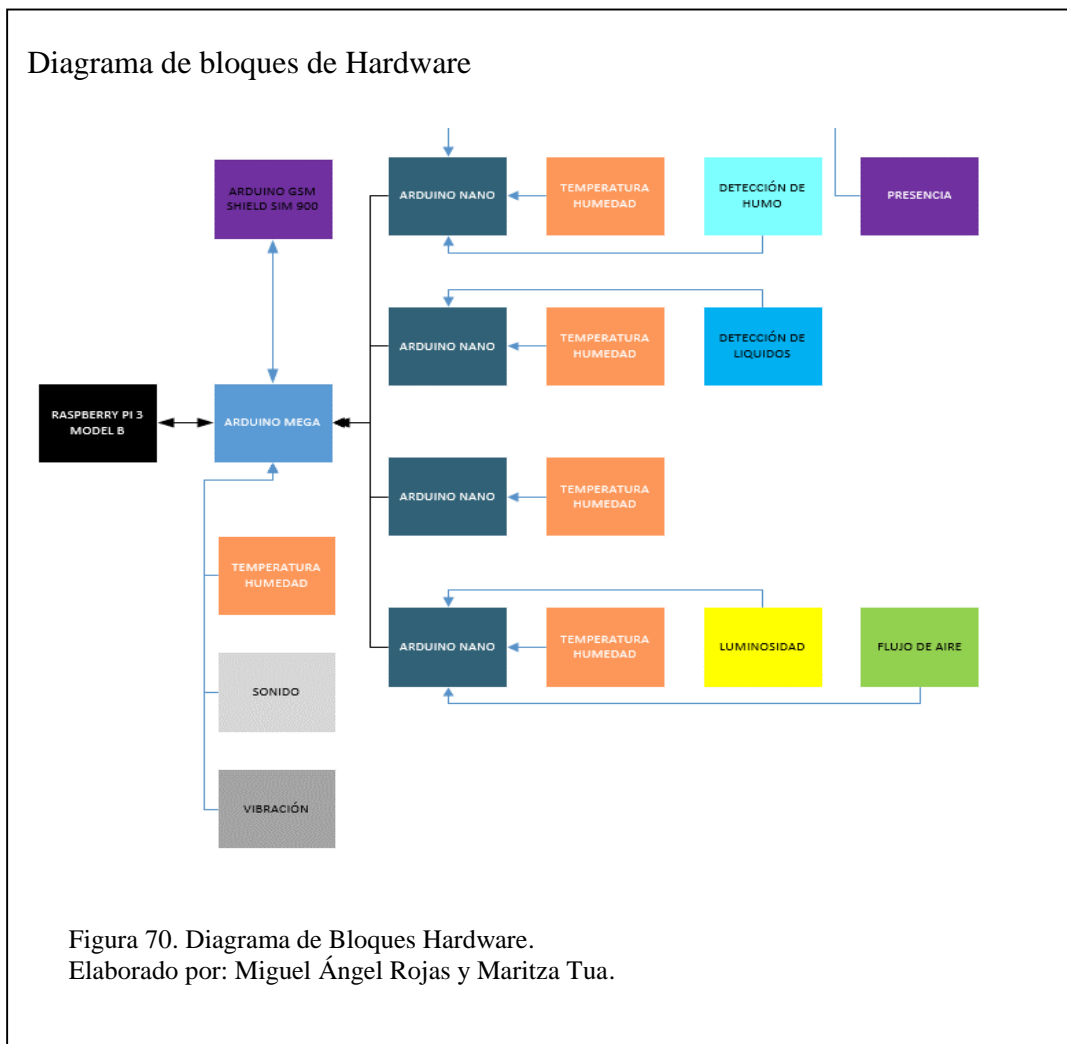
```
sudo apt-get update  
sudo service apache2 start  
sudo service mysql start
```

Figura 69. Comandos para comunicación entre Apache y PHP.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.3 Software

Es la parte intangible de los componentes de un equipo informático, compuesto de una serie de sentencias para controlar y ejecutar tareas con la parte tangible (hardware).

#### 3.3.1 Diagrama de bloques de hardware



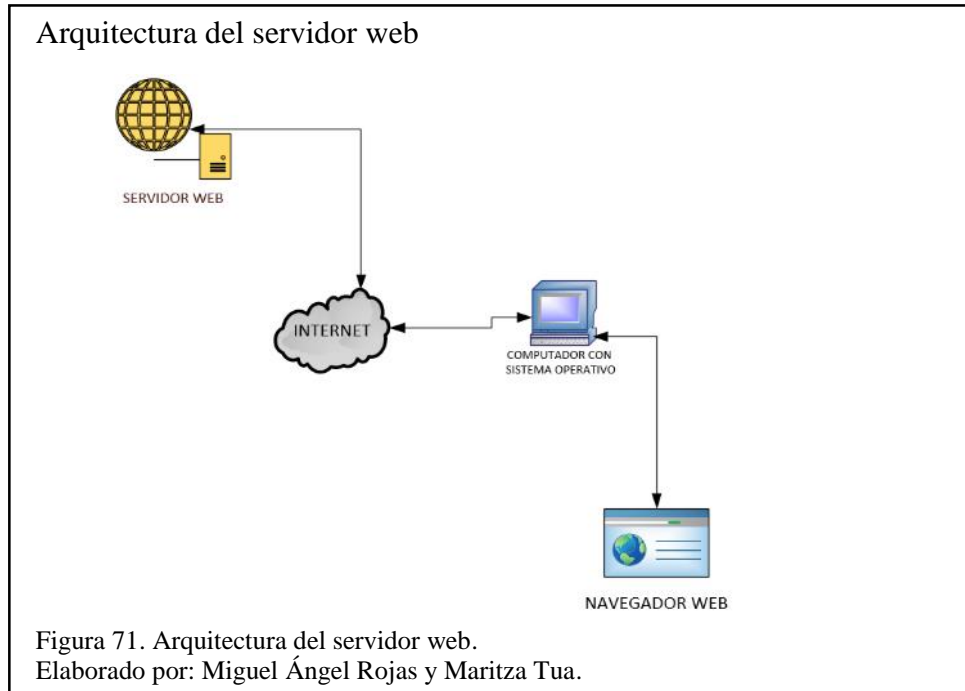
En la figura 70, el diagrama de bloques del prototipo, en donde los sensores se conectan a las placas Arduino, estas a su vez se conecta al módulo maestro, que cumple el rol de coordinador (orquestador) por medio de un enlace inalámbrico (bluetooth), procesa la información, y posteriormente enviarlos a la Raspberry (servidor).

### 3.3.2 Análisis de software

Consta de un ambiente web programado en procesador de hipertexto (PHP) y lenguaje de marcas de hipertexto (HTML), además de la aplicación móvil para dispositivos Android programada en Java y Scheme.

### 3.3.3 Arquitectura

En la figura 71, se observa el esquema de la arquitectura del servidor web, el cual por medio de internet envía los datos para que sean visualizados desde un navegador web, desde cualquier lugar.



### 3.3.4 Diagramas de caso de uso.

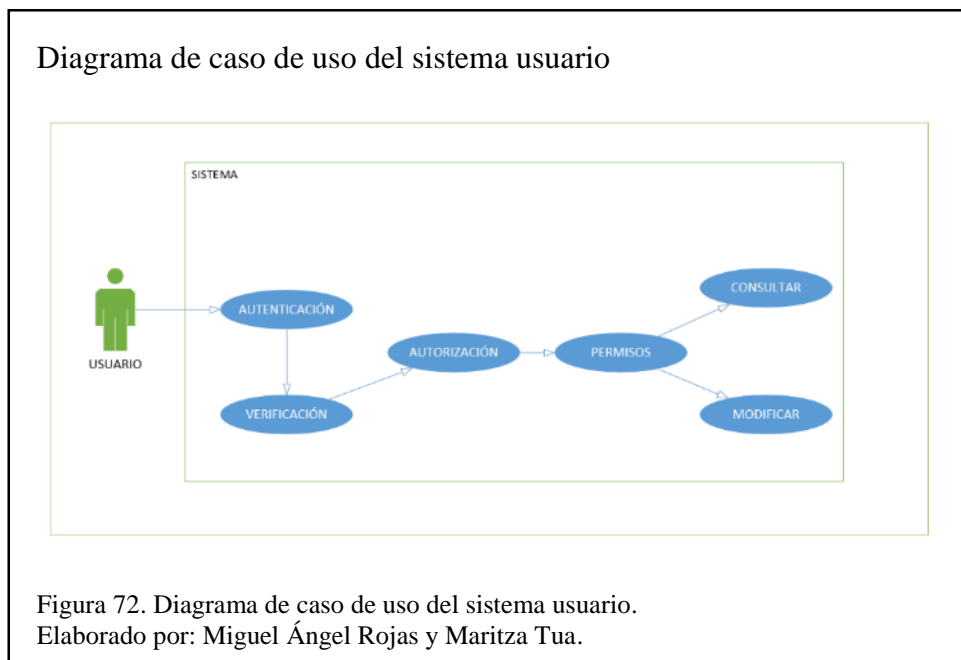
Dentro del Lenguaje de Modelado Único (UML) un diagrama de caso de uso describe el comportamiento de un actor dentro de un sistema. Se utilizan para dar a conocer los requisitos del sistema y ver las reacciones ante eventos propios o de su ambiente.

#### 3.3.4.1 Actores

- Administrador: actor que posee permisos especiales e información sin restricción.
- Usuario o invitado: actor con restricciones y permisos de visualización.

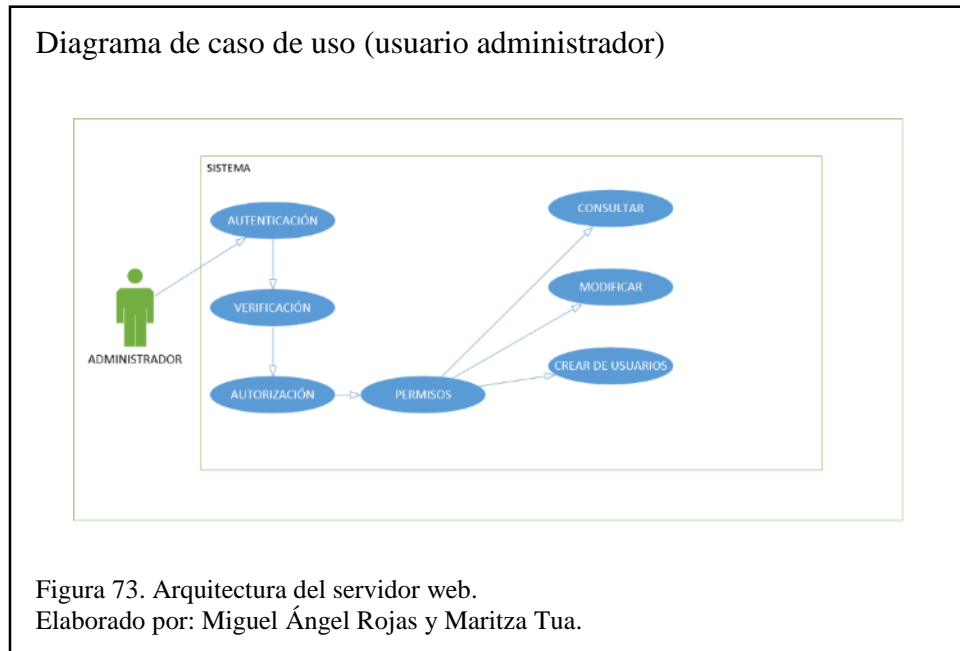
#### 3.3.4.2 Diagrama de Casos de uso del sistema

Es parte del lenguaje de modelado unificado (UML), es una representación gráfica del comportamiento y la comunicación de un sistema y la interacción con el usuario y otros sistemas, en la figura 72, se observa el rol del usuario frente al sistema.



### 3.3.4.3 Administrador

El rol del administrador tiene privilegios frente a otros usuarios como crear, modificar, eliminar y visualizar información. (Ver figura 73).



### 3.3.4.4 Invitado

El diagrama de caso de uso del usuario invitado, tiene restricciones con respecto al administrador, siendo la consulta de datos su única actividad en el sistema. (Ver figura 74).

### Diagrama de caso de uso (usuario invitado)

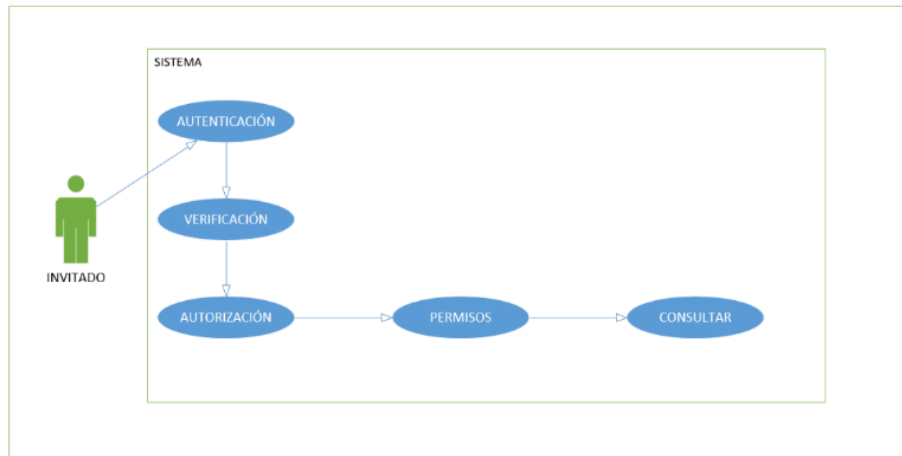


Figura 74. Diagrama de caso de uso usuario invitado.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.3.5 Interfaces

La interfaz de usuario es el medio de comunicación entre un equipo (computador, maquina o dispositivo) y el usuario, son intuitivas y amigables (de fácil comprensión).

### Esquema pantalla inicial

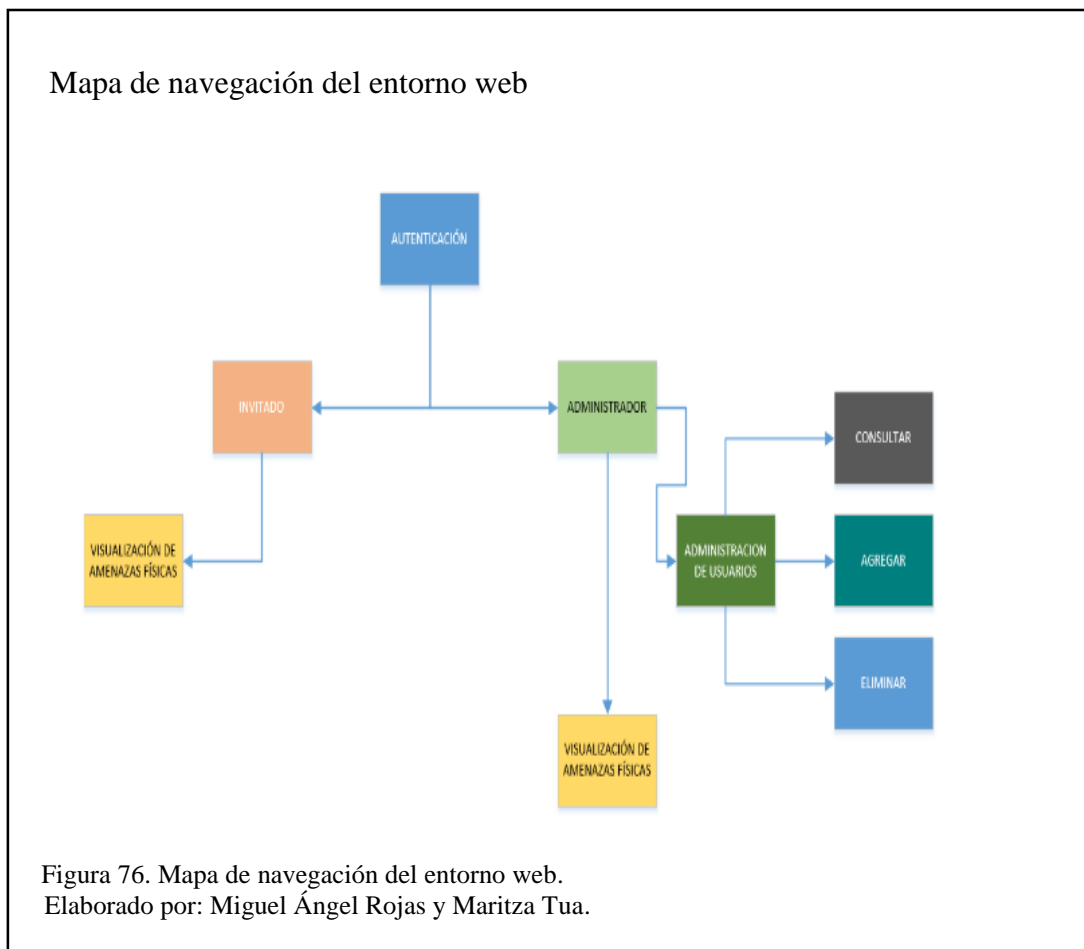


Figura 75. Esquema pantalla inicial.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.



En la figura 75, se observa la distribución esquemática del entorno web, distribuido de la siguiente manera:

- Espacio 1. Ubicación del logotipo y nombre del sistema.
- Espacio 2. Espacio para el menú de navegación del sistema
- Espacio 3. Ubicación de los indicadores del estado de variables físicas.



En la figura 76, se observa el mapa de navegación del entorno web, con los roles, actividades y privilegios antes descritos.

### 3.3.6 Product BackLog

Tomando en cuenta los requerimientos del administrador del centro de procesamiento de datos, se realizó una lista de requisitos que se exponen a continuación.

Tabla 28. Tabla de requisitos Scrum

REQUISITO 1	DISEÑO INTERFAZ DE USUARIO	PRIORIDAD: ALTA
La interfaz inicial de usuario debe ser intuitivo y gráfico		
REQUISITO 2	CONTROL DE ACCESO	PRIORIDAD:ALTA
El sistema de monitorización debe tener un control de acceso a usuarios por medio de autenticación de usuario y contraseña		
REQUISITO 3	USUARIO ADMINISTRADOR	PRIORIDAD: ALTA
Los roles del usuario administrador serán: Añadir usuarios Eliminar usuario Visualización de usuarios		
REQUISITO 4	GENERACIÓN DE REPORTE	PRIORIDAD: ALTA
El sistema de monitorización debe generar reportes diarios		
REQUISITO 5	ACCESO DESDE UNA APLICACIÓN PARA ANDROID	PRIORIDAD: ALTA
Los datos visualizados en la interfaz web deben ser visualizados desde una aplicación para un dispositivo movil Android		
REQUISITO 6	ALERTAS VÍA SMS Y LLAMADA CELULAR	PRIORIDAD: MEDIA
En el caso de ocurrir una incidencia, se notificará via mensaje corto de texto (SMS) al dispositivo móvil, además de una llamada telefónica.		

Nota: En la tabla 28, se observa la tabla de requisitos expuestos por el administrador del centro de procesamiento de datos.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.3.7 Sprint del proyecto

Pasando por las fases de análisis, diseño y desarrollo, tomando en cuenta los requerimientos, el Sprint BackLog (producto que se desarrolla en base del Product BackLog) se convierte en un producto listo para ser usado y funcional.

Tabla 29. Tabla de requerimientos Sprint Backlog

#	REQUISITOS	PRIORIDAD
Requisito 1	Diseño interfaz de usuario	Alta
Requisito 2	Control de acceso	Alta
Requisito 3	Usuario administrador	Alta
Requisito 4	Generación de reportes	Alta
Requisito 5	Acceso desde una aplicación para Android	Alta
Requisito 6	Alertas vía sms y llamada de celular	Media

Nota: En la tabla 29, se observa la lista de requisitos del administrador y la prioridad dentro del proyecto.

Elaborado por: Miguel Ángel Rojas y Maritza.

### 3.3.8 Tareas del Sprint

Tabla 30. Tareas del Sprint

TAREA	RESPONSABLE	TIEMPO ESTIMADO
Diseño interfaz de usuario	Maritza Tua y Miguel Rojas	12 horas
Control de acceso	Maritza Tua y Miguel Rojas	8 horas
Usuario administrador	Maritza Tua y Miguel Rojas	12 horas
Generación de reportes	Maritza Tua y Miguel Rojas	12 horas
Acceso desde una aplicación para Android	Maritza Tua y Miguel Rojas	12 horas
Alertas vía sms y llamada de celular	Maritza Tua y Miguel Rojas	8 horas
Pruebas	Maritza Tua y Miguel Rojas	4 horas
HORAS TOTALES		68 horas

Nota. En la tabla 30, se define los responsables de los requisitos, tareas que se cumplen en tiempo determinado.

Elaborado por: Miguel Ángel Rojas y Maritza.

### 3.9 Lista de tareas de la iteración.

Tabla 31. Tareas de la iteración

TAREA	TIPO	ESTADO FINAL
Diseño interfaz de usuario	Codificación	Terminado
Control de acceso	Prototipo	Terminado
Usuario administrador	Codificación	Terminado
Generación de reportes	Codificación	Terminado
Acceso desde una aplicación para Android	Prototipo	Terminado
Alertas vía sms y llamada de celular	Prototipo	Terminado
Pruebas	Pruebas	Terminado

Nota. En la tabla 31, se expone las tareas realizadas y el estado final.

Elaborado por: Miguel Ángel Rojas y Maritza.

### 3.9.1 Pantalla de inicio

La pantalla de inicio de la interfaz web del prototipo (ver figura 77), tiene el control de acceso por medio de un inicio de sesión, el que valida por medio de la consulta a la base de datos, si el usuario registrado tiene el perfil de administrador o de invitado.



Cuando es un usuario administrador, se despliega un menú en la parte izquierda de la interfaz web, permitiendo administrar el sistema por medio de consultas, modificación y eliminación de usuarios, además de visualizar los valores de las variables físicas. (ver figura 78).

## Pantalla de visualización de variables ambientales

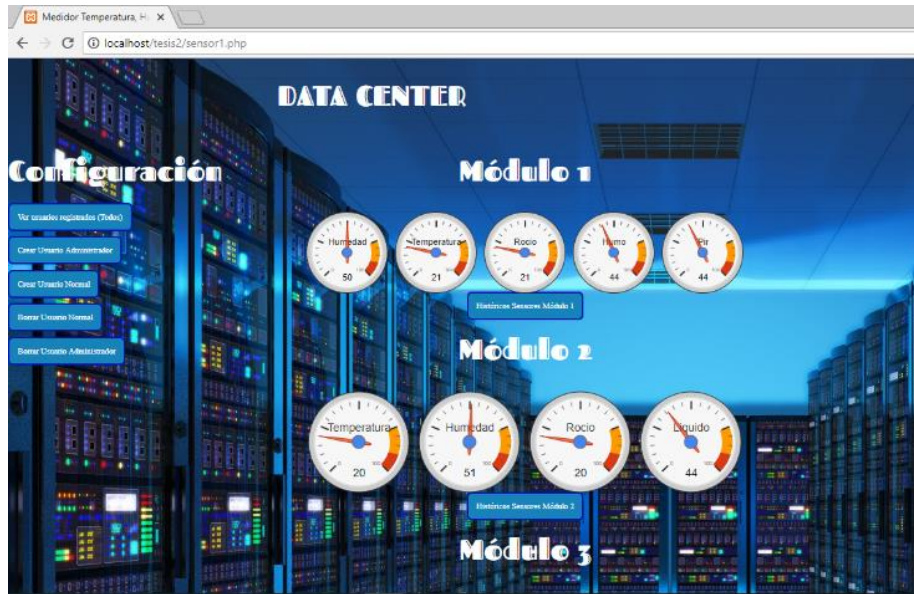


Figura 78. Pantalla de visualización de variables ambientales del centro de datos.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

### 3.9.2 Gráfica lineal de los valores de un sensor

#### Gráfica lineal de los valores de un sensor

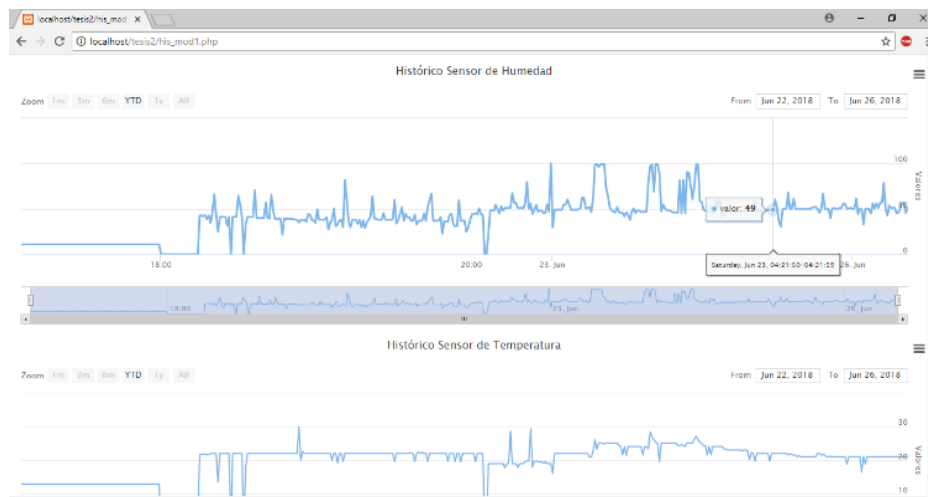


Figura 79. Gráfica lineal histórica de los valores tomados por un sensor en un determinado periodo de tiempo.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 79, se muestra los datos obtenidos por cada módulo instalado.

### 3.9.3 Pantalla de usuarios

Pantalla de usuarios

**Usuarios Normales**

ID	Usuario	Contraseña	Fecha de registro
6			2018-06-14 22:26:31

El Total de Registros es de: 1

**Usuarios Administradores**

ID	Usuario	Contraseña	Fecha de registro
2	mary	1234	2018-06-14 18:24:26
3	buznahim	1234	2018-06-14 18:24:37
9	cris	cris	2018-06-22 19:17:37
10	admin	admin	2018-06-22 19:26:20

El Total de Registros es de: 4

[Regresar](#)

Figura 80. Pantalla de usuarios registrados.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 80, se observa los usuarios registrados en el sistema, con atributos especiales dependiendo el tipo de permiso que se otorgó (administrador o invitado).

### 3.10 Desarrollo aplicación Android

Para el desarrollo de la aplicación en Android, se escogió una plataforma que sea de fácil manejo, fácil comprensión, amigables con el programador y de código abierto (software libre), teniendo a la aplicación en entorno web APPINVENTOR 2, teniendo a Java y Scheme como lenguaje de programación, se observa en la imagen 81, el entorno de desarrollo de la aplicación.

## Entorno gráfico de APPINVENTOR2

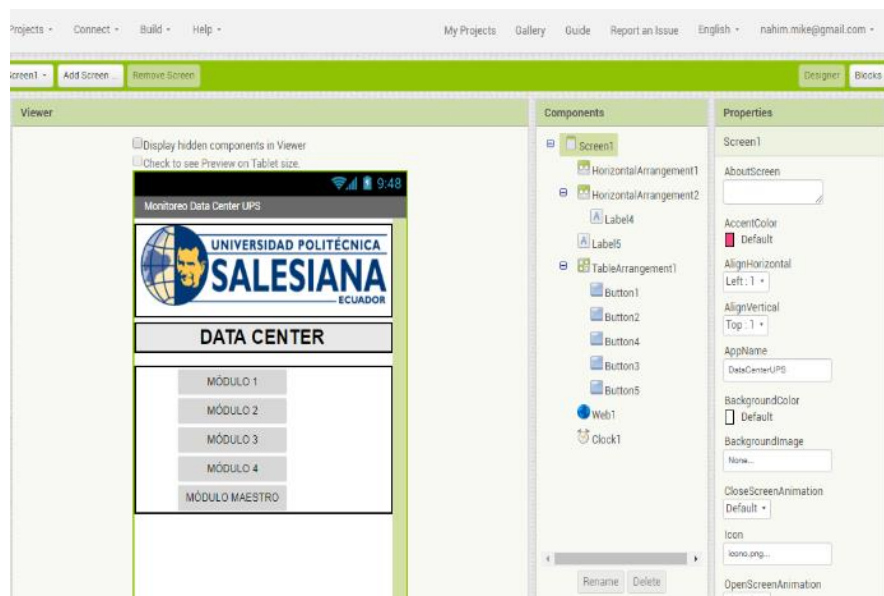


Figura 81. Entorno gráfico de AppInventor 2.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

En la figura 81, se visualiza la programación en bloques, muy semejante a un ensamblaje de piezas de LEGO, se puede agregar controles y eventos.

## Programación de AppInventor

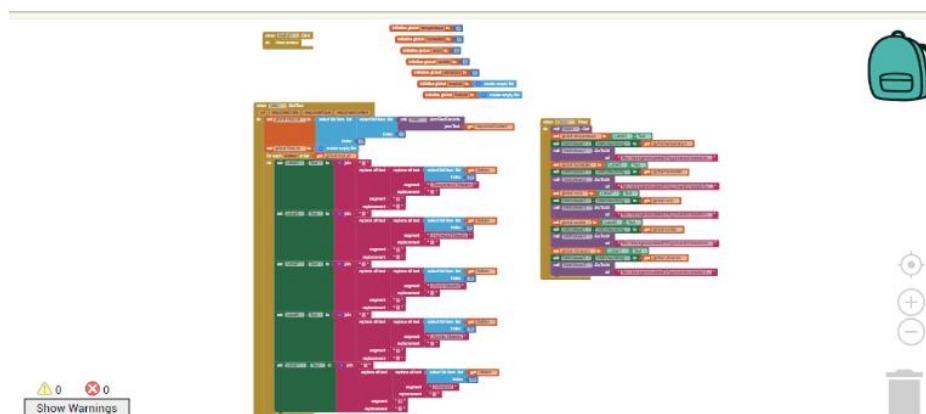


Figura 82. Programación en AppInventor 2.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

## CAPÍTULO 4

Se muestra las pruebas realizadas durante la verificación del funcionamiento del prototipo.

### 4.1 Pruebas del entorno web

Se procedió a realizar las pruebas de funcionalidad de la aplicación web, en base a los requerimientos del Sprint Backlog, para la detección oportuna de fallas.

Tabla 32. Prueba del requisito No. 1

<b>REQUISITO 1</b>	<b>DISEÑO INTERFAZ DE USUARIO</b>
La interfaz inicial de usuario debe ser intuitivo y gráfico	
<b>Desarrollo:</b>	Se procedió a ejecutar el archivo Index.html para acceder con el usuario registrado.
<b>Condicionamiento:</b>	El usuario a ingresar debe estar registrado.
<b>Procedimiento:</b>	Inicio de la interfaz web del sistema Ingreso de información campos usuario y contraseña Interacción con los termómetros digitales Visualización de valores de condiciones ambientales
<b>Resultado:</b>	Usuario interactuó fácilmente con el entorno web

Nota. En la tabla 32, se observa la solución al requisito 1, teniendo una interacción entre el usuario y el ambiente web.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.



Tabla 33. Prueba del requisito No. 2

<b>REQUISITO 2</b>	<b>CONTROL DE ACCESO</b>
El sistema de monitorización debe tener un control de acceso a usuarios por medio de autenticación de usuario y contraseña	
<b>Desarrollo:</b>	Se Procedió a ejecutar el achivo Index.html para autenticarse con los datos del usuario registrado.
<b>Condicionamiento:</b>	El usuario a ingresar debe estar registrado.
<b>Procedimiento:</b>	Inicio de la interfaz web del sistema Ingreso de información campos usuario y contraseña Interacción con los termómetros digitales Visualización de valores de condiciones ambientales
<b>Resultado:</b>	Accedió al entorno web solo con el usuario registrado

Nota. En la tabla 33, se expone el requisito 2, la interacción con el módulo de autenticación del sistema.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 34. Prueba del requisito No. 3

<b>REQUISITO 3</b>	<b>USUARIO ADMINISTRADOR</b>
Los roles del usuario administrador serán: Añadir usuarios Eliminar usuario Visualización de usuarios	
<b>Desarrollo:</b>	Autenticación con usuario con permisos de administrador
<b>Condicionamiento:</b>	El usuario a ingresar debe estar registrado y con permisos de administrador
<b>Procedimiento:</b>	Creación de usuario con permisos de administrador Visualización de usuarios registrados. Aumento de usuarios al sistema. Eliminación de usuarios del sistema. Ingreso de información campos usuario y contraseña Interacción con los termómetros digitales Visualización de valores de condiciones ambientales
<b>Resultado:</b>	Accedió al entorno web solo con el usuario registrado como administrador

Nota. En la tabla 34, se observa la interacción del usuario con el módulo de registro de nuevos usuarios.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 35. Prueba del requisito 4

<b>REQUISITO 4</b>	<b>GENERACIÓN DE REPORTE</b>
El sistema de monitorización debe generar reportes diarios	
<b>Desarrollo:</b>	Se toman datos y guarda por un periodo de tiempo
<b>Condicionamiento:</b>	El usuario a ingresar debe estar registrado y con permisos de administrador
<b>Procedimiento:</b>	Visualización en el módulo gráfico de valores representador por líneas
<b>Resultado:</b>	Visualización de reportes

Nota. En la tabla 35, se expone la solución al requisito 4, la generación oportuna de reportes para tener un control de las amenazas físicas.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 36. Prueba del requisito 5

<b>REQUISITO 5</b>	<b>ACCESO DESDE UNA APLICACIÓN PARA ANDROID</b>
Los datos visualizados en la interfaz web deben ser visualizados desde una aplicación para un dispositivo movil Android	
<b>Desarrollo:</b>	Acceso a la visualización de amenazas ambientales por medio de un dispositivo Android
<b>Condicionamiento:</b>	El usuario debe tener un smartphone con android
<b>Procedimiento:</b>	Instalación del aplicativo desarrollado para ver las amenazas ambientales en el teléfono inteligente
<b>Resultado:</b>	Consulta de los valores de amenazas ambientales desde Smartphone con Android

Nota. En la tabla 36, se tiene el resultado del requisito 5. Con la implementación de la aplicación para Android, se visualiza en tiempo real los valores.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 37. Prueba del requisito 6

<b>REQUISITO 6</b>	<b>ALERTAS VÍA SMS Y LLAMADA CELULAR</b>
En el caso de ocurrir una incidencia, se notificará vía mensaje corto de texto (SMS) al dispositivo móvil, además de una llamada telefónica.	
<b>Desarrollo:</b>	Envío de alertas de cambios de variables ambientales
<b>Condicionamiento:</b>	El administrador debe tener un teléfono con recepción de llamadas y mensajes cortos de texto (SMS)
<b>Procedimiento:</b>	Programación del Módulo Arduino con el número del administrador del centro de datos
<b>Resultado:</b>	Usuario recibió alertas en su teléfono (vía llamada celular y mensaje corto de texto SMS)

Nota. En la tabla 37, se observa el cumplimiento del requisito 6, implementando el módulo GSM Shield para envío de mensajes de texto y llamadas al número registrado del administrador.

Elaborado por: Miguel Ángel Rojas y Maritza Tua.

#### 4.2 Pruebas de la aplicación en Android

Se realizaron pruebas de funcionamiento de la aplicación en Android para ver la funcionalidad, posibles errores y controles. Las pruebas se las realizan en un teléfono inteligente con sistema operativo Android 7.0

Tabla 38. Prueba del requisito 1 para Android.

<b>REQUISITO 1</b>	<b>VISUALIZACIÓN DE ALERTAS</b>
Se tienen que visualizar las alertas de amenazas ambientales en tiempo real	
<b>Desarrollo:</b>	Visualizar las alertas de amenazas ambientales
<b>Condicionamiento:</b>	Tener activo paquete de conexión a internet o wifi
<b>Procedimiento:</b>	Instalación y apertura de la aplicación
<b>Resultado:</b>	Se visualizó la información en tiempo real

Nota. En la tabla 38, se obtiene el resultado del requisito 1 para el aplicativo Android, en las pruebas se observa los datos en tiempo real.

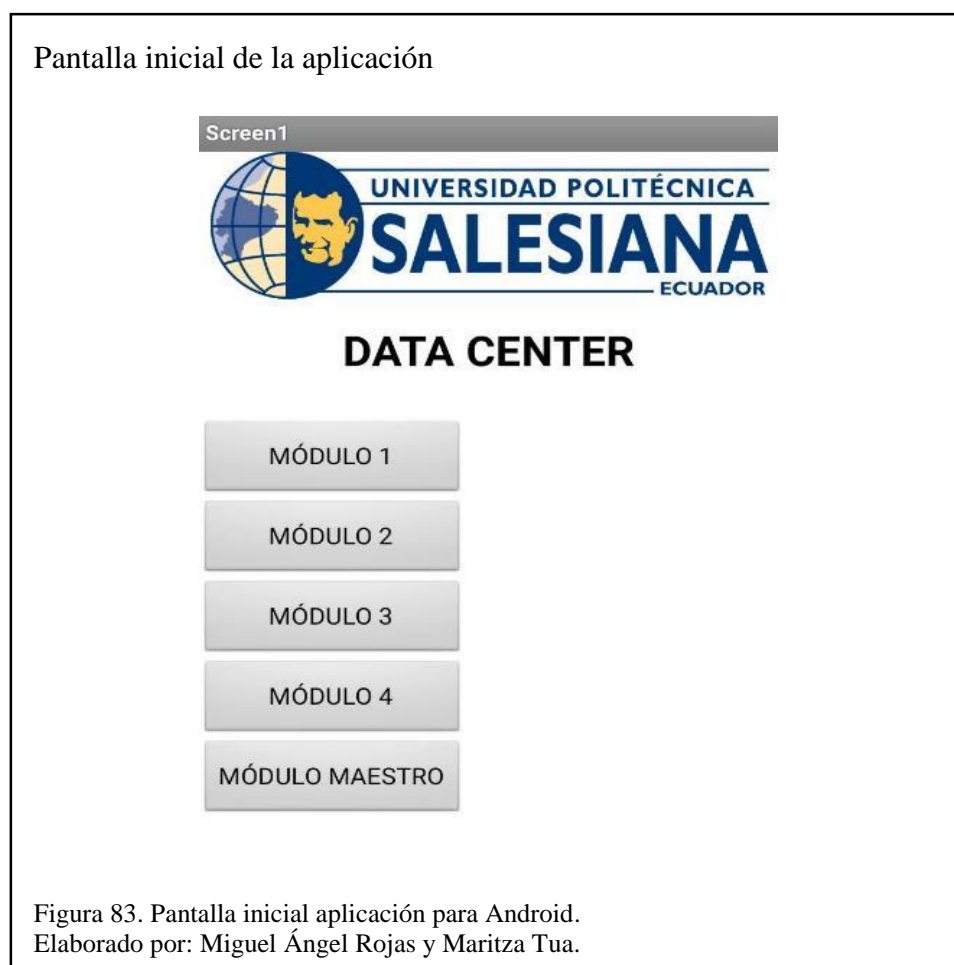
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Tabla 39. Prueba del requisito 2 para Android

<b>REQUISITO 2</b>	<b>DISEÑO DE FÁCIL MANEJO</b>
La interacción con el usuario debe ser amigable y de fácil comprensión	
<b>Desarrollo:</b>	Interacción con la aplicación de Android
<b>Condicionamiento:</b>	Tener activo paquete de conexión a internet o wifi
<b>Procedimiento:</b>	Instalación y apertura de la aplicación
<b>Resultado:</b>	Usuario interactuó fácilmente con la aplicación de monitorización

Nota. En la tabla 39, el resultado del requisito 2 para el aplicativo Android, interacción y manejo fácil, por tener una interfaz simplificada e intuitiva.

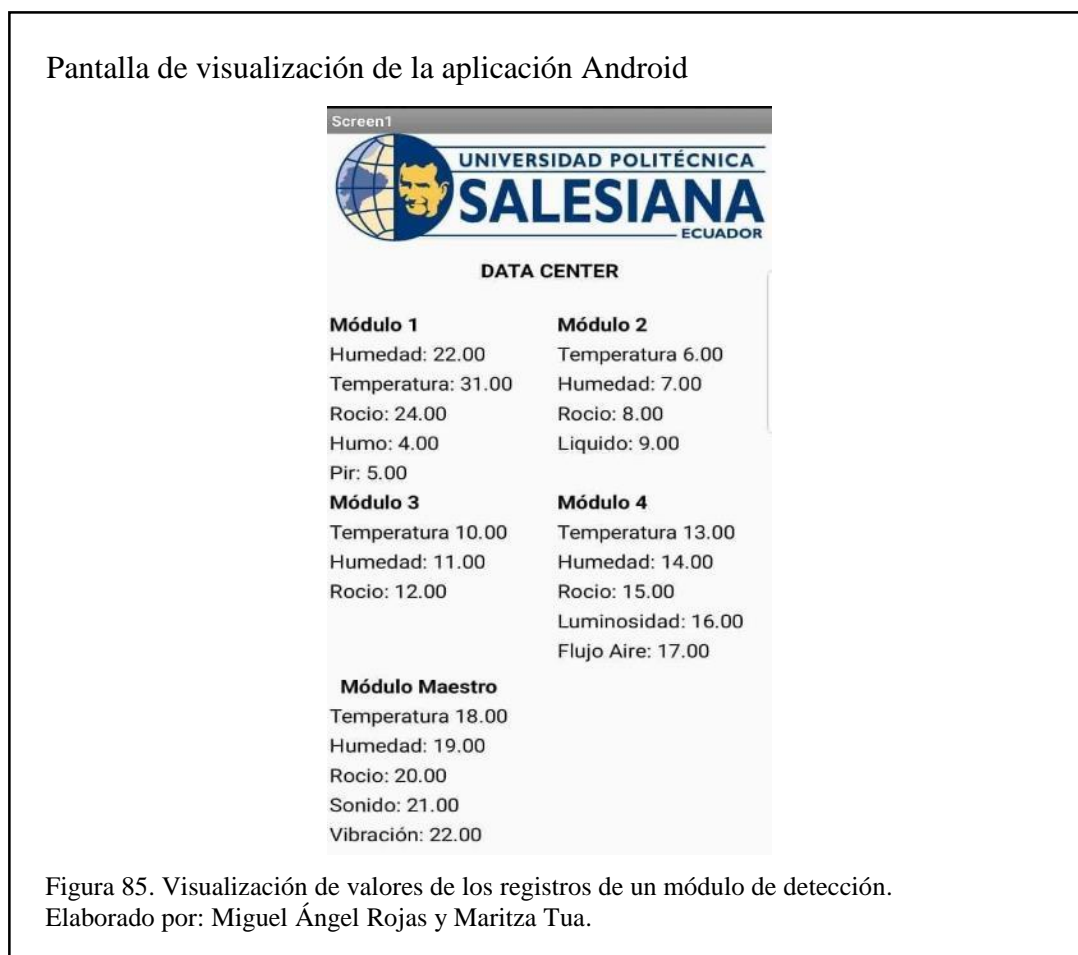
Elaborado por: Miguel Ángel Rojas y Maritza Tua.



En la figura 83, se tiene la interfaz de la aplicación para un teléfono inteligente Android, es fácil de usar e intuitiva. Permite observar la temperatura por niveles gracias a que los módulos están distribuidos de manera estratégica en el centro de procesamiento de datos.

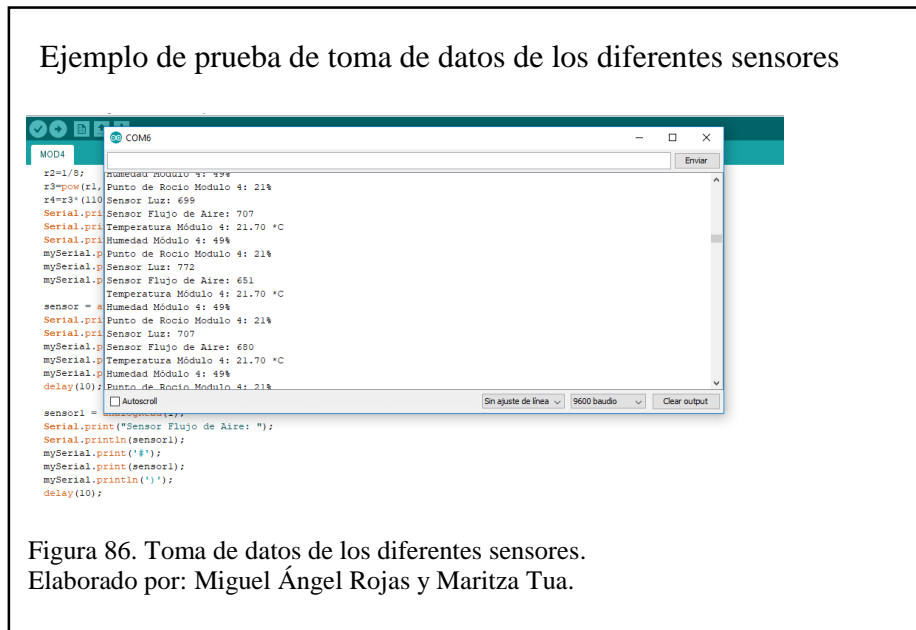


En la figura 84, se observa la interfaz de la aplicación para Android, desplegando de manera gráfica los datos de las amenazas físicas que se registran en el módulo maestro (nivel medio).



En la figura 85, se observa la información en valores de las amenazas físicas registradas en el centro de procesamiento de datos.

### 4.3 Pruebas de toma de datos de los sensores desde Arduino ID



La figura 86, muestra un ejemplo de la toma inicial de datos de los diferentes sensores que se implementaran, este proceso se lo realiza en los 5 módulos.

### 4.4 Resultado de pruebas de efectividad

Una vez que se realizan todos los requerimientos solicitados por el administrador del centro de procesamiento, se procede a verificar la efectividad del sistema y la aplicación Android mediante indicadores establecidos.

Tabla 40. Escala de valoración

PORCENTAJE DE CUMPLIMIENTO DE LA META		OBSERVACIÓN
98-100	Se cumple plenamente	Si el cumplimiento de la meta es igual o inferior al 90% obligatoriamente debe especificar un plan de mejoramiento para alcanzar la meta propuesta
75-97	Se cumple en alto grado	
60-74	Se cumple aceptablemente	
40-59	Se cumple insatisfactoriamente	
Menor a 40	No se cumple	

Nota: Porcentajes de cumplimiento de las metas establecidas.  
Elaborado por: Miguel Ángel Rojas y Maritza Tua.

#### 4.4.1 Indicadores

Se establecen los siguientes indicadores:

- Pruebas del Sistema IP privada.
- Pruebas con IP institucional del sistema (página web).
- Pruebas de funcionamiento de sensores.
- Prueba de conectividad IP institucional de la Aplicación en Android.
- Pruebas de Aplicación Android IP privada.

#### 4.4.2 Cálculo

Se proceden a realizar todos los cálculos necesarios para obtener efectividad de prototipo

Tabla 41. Tabla de indicadores

TABLA DE INDICADORES		
INDICADORES	%	OBSERVACIONES
Pruebas del Sistema ip privada	100	El sistema funciona según requerimientos solicitados por Administrador del Centro de Porcesamiento de Datos
Pruebas con ip institucional del sistema (página web)	98	Restricciones en la red (perdida de cobertura)
Pruebas de funcionamiento de sensores	100	Sensores funcionan correctamente
Prueba de conectividad ip	98	Se pierde conexión ip institucional con muchas restricciones
Pruebas de Aplicación Android ip privada	100	Se realiza todas las conexiones a una ip privada y la aplicación funciona corectamente
SUMA	496	
TOTAL %	99,2	

Nota: En la Tabla 41 se muestra la asignación de porcentaje según Tabla 40. Elaborado por: Miguel Ángel Rojas y Maritza Tua.

Como se observa en la tabla 41, se asigna a la falta de conectividad por ip institucional el 2% de pérdida de efectividad.

Tabla 42. Tabla de resultados y línea base.

MEDICIÓN	LÍNEA BASE	RESULTADOS
RESULTADO	100	99,2
META		100,0
% EFECTIVIDAD		99,2%

Nota: En la tabla de resultados, las metas siempre serán el 100% ya que son los objetivos establecidos. Elaborado por: Miguel Ángel Rojas y Maritza Tua.

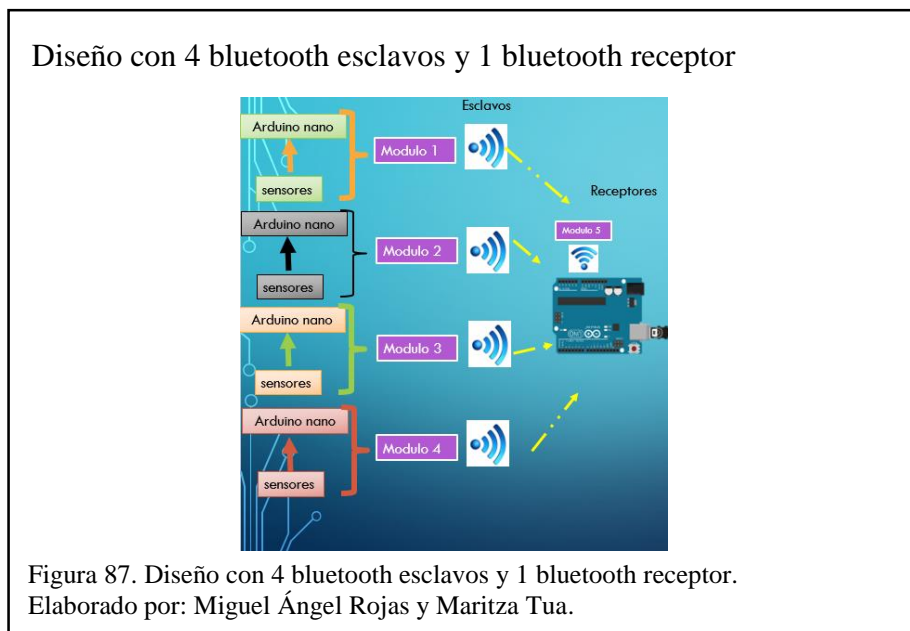


La efectividad del prototipo está basada mediante indicadores y objetivos (metas) establecidos, dando el 99,2% de cumplimiento como se observa en la Tabla 42.

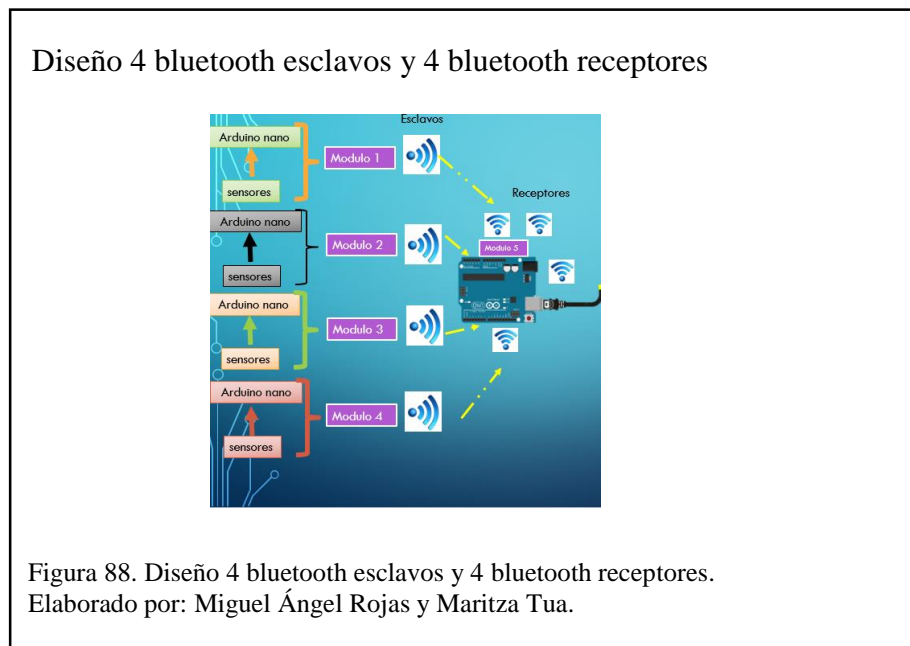
#### 4.5 Pruebas de diseño con respecto comunicación

Se muestra el diseño inicial a utilizar y el diseño que se utilizó para la comunicación de datos.

##### 4.5.1 Diseño con 4 bluetooth esclavos y 1 bluetooth receptor



#### 4.5.2 Diseño 4 bluetooth esclavos y 4 bluetooth receptores



Al observar un retardo de comunicación entre los 4 módulos esclavos con el módulo 5 que contiene 1 bluetooth receptor como se visualiza en la figura 87, se opta por un diseño que requiere un bluetooth esclavo en cada módulo y cuatro maestros en el módulo 5, ver figura 88.

## CONCLUSIONES

- Se diseñó, configuró e implementó el prototipo mediante una red de sensores ubicados en diferentes niveles del centro de procesamiento de datos obteniendo una efectividad del 99,2%, de acuerdo a los resultados obtenidos.
- Una vez que se realizaron las pruebas con el diseño inicial de los módulos que contienen a los diferentes sensores, se produjo retardos superiores a 10 segundos en la recepción de datos, ante lo cual, se desarrolló un nuevo diseño que requiere un bluetooth esclavo en cada módulo y cuatro maestros en el módulo 5, disminuyendo así el retardo.
- El prototipo implementado por su estructura autónoma, reduce en un 50% la intervención del personal a cargo, brindando al administrador la facilidad de monitorizar 24/7 las amenazas físicas dentro del centro de procesamiento de datos en cualquier instante, por medio de un dispositivo móvil o acceso al entorno web, con esto, al ser construido por medio de herramientas de software libre, no se realiza el pago de licenciamientos anuales, por conceptos de software propietario.

## **RECOMENDACIONES**

- Raspberry siendo un mini computador de bajo precio y debido a su flexibilidad para actuar como un servidor web, es efectiva en proyectos de bajo costo.
- Se recomienda utilizar los sensores implementados en este prototipo, por las facilidades que prestan, por la amplia bibliografía existente y fácil configuración.
- Se recomienda utilizar plataformas de software libre para evitar el pago de licencias de software propietario.

## LISTA DE REFERENCIAS

- SUPERNOVATEL*. (8 de 8 de 2012). Recuperado el 16 de 4 de 2018, de SUPERNOVATEL: <https://es.scribd.com/document/102391071/Data-Center-Estandares-de-Diseno-y-Funcionamiento>
- ABC Tecnología*. (21 de 7 de 2013). Obtenido de <http://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>
- ALIBABA. (16 de 2 de 2018). *aliexpress*. Obtenido de <https://es.aliexpress.com/item/Arduino-piezoelectric-ceramic-vibration-sensor-vibration-sensor-electronic-building-blocks/32666692766.html>
- ARDUINO.cl. (2018). *ARDUINO.cl*. Obtenido de <http://arduino.cl/que-es-arduino/>
- Betty. (14 de 2 de 2016). *Metodología I UCV*. Recuperado el 15 de 4 de 2018, de Metodología I UCV: <http://metodologia1ucv.blogspot.com/2016/02/tema-4-tipos-de-investigacion.html>
- CERTSI\_. (08 de 02 de 2018). *CERTSI\_*. Obtenido de <https://www.certsi.es/blog/introduccion-los-sistemas-embebidos>
- Científicas, C. S. (2016). Obtenido de <http://www.icv.csic.es/prevencion/Documentos/breves/FREMAP/iluminacion.pdf>
- Computer Hoy*. (2018). Obtenido de <https://computerhoy.com/noticias/hardware/raspberry-pi-que-modelo-me-compro-23811>
- concepto definicion.DE*. (22 de 3 de 2015). Obtenido de <http://concepto definicion.de/arduino/>
- CZX. (1).
- EcuRed. (27 de 7 de 2018). Obtenido de <https://www.ecured.cu/Microcontrolador>
- eekFACTORY. (2018). Obtenido de <https://www.geekfactory.mx/tienda/sensores/mq-2-sensor-de-gas/>
- Electronilab. (20018). *Electronilab*. Obtenido de <https://electronilab.co/tienda/sensor-de-temperatura-y-humedad-dht22/>
- Gámez, R. (10 de 7 de 2012). *vembu*. Recuperado el 14 de 3 de 2018, de vembu: <http://blog.aodbc.es/2012/07/10/clasificacion-tier-en-el-datacenter-el-estandar-ansitia-942/>
- geekfactory*. (2018). Obtenido de <https://www.geekfactory.mx/tienda/sensores/mq-2-sensor-de-gas/>
- grupo cofitel*. (1 de 2 de 2014). Recuperado el 20 de 3 de 2018, de <https://www.c3comunicaciones.es/data-center-el-estandar-tia-942/>

HETPRO. (13 de 10 de 2015). *Eduardo Lara*. Obtenido de <https://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>

INCIBE. (8 de 2 de 2018). *CERTSI*. Obtenido de <https://www.certsi.es/blog/introduccion-los-sistemas-embebidos>

Joa, C. (10 de 8 de 2015). *SlideShare*. Obtenido de SlideShare.

Lapuente, M. J. (12 de 8 de 2013). Obtenido de [http://www.hipertexto.info/documentos/lenguajes\\_h.htm](http://www.hipertexto.info/documentos/lenguajes_h.htm)

Llama, L. (2017). *luisllama*. Obtenido de <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

LLAMAS, L. (16 de 3 de 2015). *LUIS LLAMAS*. Obtenido de <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoreistencia-ldr/>

LUIS LLAMAS. (11 de 10 de 2016). *LUIS LLAMAS*. Obtenido de <https://www.luisllamas.es/medir-sonido-con-arduino-y-microfono-amplificado-max9812/>

LUIS LLAMAS. (29 de 3 de 2016). *LUIS LLAMAS*. Obtenido de <https://www.luisllamas.es/arduino-dht11-dht22/>

LUIS LLAMAS. (29 de 3 de 2016). *LUIS LLAMAS*. Obtenido de <https://www.luisllamas.es/arduino-dht11-dht22/>

LUIS LLAMAS. (9 de 10 de 2016). *LUIS LLAMAS*. Obtenido de <https://www.luisllamas.es/detectar-sonido-con-arduino-y-microfono-ky-038/>

*LUISLLAMA*. (2018). Obtenido de <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoreistencia-ldr/>

*LUISLLAMAS*. (19 de 1 de 2016). *LUISLLAMAS*. Obtenido de <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

*luisllamas*. (2018). Obtenido de <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

Macho, J. C. (2017). *Prometec*. Obtenido de <https://www.prometec.net/sensor-pir/>

Martin, S. (21 de 8 de 2017). *Pandorafms*. Recuperado el 26 de 3 de 2018, de <https://blog.pandorafms.org/es/monitorizacion-de-sistemas/>

*maxelectronica*. (2018). Obtenido de <http://www.maxelectronica.cl/sonido/250-sensor-de-sonido-analogico-microfono-max9812-modelo-fc109.html>

*MECHATRONIC*. (2018). Obtenido de [https://naylampmechatronics.com/blog/42\\_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html](https://naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html)

Merino, J. P. (2016). *definicion.de*. Obtenido de [definicion.de: https://definicion.de/smartphone/](https://definicion.de/smartphone/)

- MINDMEISTER. (13 de 9 de 2013). Obtenido de <https://www.mindmeister.com/es/326622411/hardware-se-refiere-a-todas-las-partes-tangibles-de-un-sistema-informatico-partes-fisicas>
- naylampmechatronics*. (2017). Obtenido de [https://naylampmechatronics.com/blog/44\\_Tutorial-m%C3%B3dulo-sensor-de-luz-BH1750.html](https://naylampmechatronics.com/blog/44_Tutorial-m%C3%B3dulo-sensor-de-luz-BH1750.html)
- Nextia Fenix. (2018). Obtenido de <https://www.nextiafenix.com/producto/sensor-vibracion-sw-420/>
- Omniblug. (24 de 7 de 2018). *Omniblug*. Obtenido de Omniblug: <http://www.omniblug.com/sensor-temperatura-humedad-DHT11-DHT22.html>
- Oña Salazar, D. M., & Vaca Mármol, B. A. (abril de 2015). *Repositorio Digital Universidad Politécnica Salesiana*. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/10117>
- prometec*. (s.f.). Obtenido de <https://www.prometec.net/gprs-llamar-enviar-sms/>
- SemanticWebBuilder*. (s.f.). Obtenido de [http://www.semanticwebbuilder.org.mx/es\\_mx/swb/Sistemas\\_Embebidos\\_Innovando\\_hacia\\_los\\_Sistemas\\_Inteligentes\\_](http://www.semanticwebbuilder.org.mx/es_mx/swb/Sistemas_Embebidos_Innovando_hacia_los_Sistemas_Inteligentes_)
- SISBIB. (1 de 2 de 1999). Obtenido de [http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v02\\_n1/produccion.htm](http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v02_n1/produccion.htm)
- SUPERNOVATEL*. (8 de 8 de 2012). Recuperado el 17 de 4 de 2018, de SUPERNOVATEL: <https://es.scribd.com/document/102391071/Data-Center-Estandares-de-Diseno-y-Funcionamiento>
- Tecnología & Informática. (2018). *Tecnología & Informática*. Obtenido de <https://tecnologia-informatica.com/tipos-licencias-software-libre-comercial/>
- tecnopura*. (2016). Obtenido de <http://www.tecnopura.com/producto/modulo-sensor-de-gas-mq-3-para-deteccion-de-alcohol-con-arduino/>
- tesisdeinvestig*. (31 de 5 de 2011). Obtenido de <http://tesisdeinvestig.blogspot.com/2011/05/tipos-de-investigacion.html>
- ventasdeseguridad*. (3 de 8 de 2007). Obtenido de *ventasdeseguridad*: <http://www.ventasdeseguridad.com/2007080347/articulos/analisis-tecnologico/el-estandar-tia-942.html>
- Vera Romero, C. A., Barbosa Jaimes, J. E., & Pabón González, D. (3 de 9 de 2017). *redalyc*. Obtenido de <http://www.redalyc.org/pdf/849/84954626002.pdf>
- Wikipedia. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Red\\_de\\_sensores](https://es.wikipedia.org/wiki/Red_de_sensores)
- wikipedia*. (3 de 2 de 2018). Obtenido de *wikipedia*: [https://es.wikipedia.org/wiki/Centro\\_de\\_procesamiento\\_de\\_datos#Consumo](https://es.wikipedia.org/wiki/Centro_de_procesamiento_de_datos#Consumo)