



**Instituto Tecnológico de Tuxtla Gutiérrez**

**Comisión Federal de Electricidad (Gerencia Regional de Transmisión Sureste)**

**Título del proyecto**

**MONITOREO Y CONTROL DE ALARMAS DEL CENTRO DE OPERACIÓN DE FIBRA  
ÓPTICA DE LA GERENCIA REGIONAL DE CFE TRANSMISIÓN SURESTE**

**Presenta: Alexander Jiménez Moreno**

**Carrera: Ingeniería Electrónica**

**Asesor interno:** Ing. Vicente León Orozco

**Asesor externo:** Ing. Guillermo Iván Huerta Franco

Subgerencia de Comunicaciones, Depto. Conexión de Servicios GRTSE

# Índice

Índice de tablas .....	4
Índice de ilustraciones.....	4
<b>Capítulo 1 Generalidades .....</b>	<b>6</b>
Introducción.....	6
<b>1.2 Antecedentes .....</b>	<b>8</b>
1.2.1 Empresa.....	8
1.2.2 Visión general de la CFE .....	8
1.2.3 Fundación.....	8
1.2.4 CFE – Organigrama.....	9
1.2.5 Gerencia Regional de Transmisión Sureste.....	10
1.2.6 COREFO GRTSE .....	11
1.2.7 Ubicación geográfica.....	11
1.3 Datos técnicos.....	12
1.4 Topología de la red regional del COREFO Sureste .....	12
1.5 Problemática .....	13
1.6 Objetivo de proyecto.....	13
1.6.1 Objetivo general.....	13
1.6.2 Objetivos específicos .....	14
1.7 Hipótesis .....	14
1.8 Justificación.....	14
<b>Capítulo 2 Marco teórico.....</b>	<b>16</b>
2.1 Importancia de monitoreo de condiciones ambientales en salas.....	16
2.1.1 Especificaciones/ normas de salas de equipos TIA/EIA 942 .....	16
2.2 Importancia de detección de gases en salas de fuerza .....	17
2.2.1 Especificaciones/normas de hidrogeno ISO 10156.....	17
2.3 Historia del Arduino .....	18
2.4 Protocolos de red .....	19
2.4.1 Protocolo SNMP.....	19
<b>Capítulo 3 Marco referencial.....</b>	<b>22</b>
3.1 Arduino mega .....	22
3.2 Sensores .....	23
3.2.1 Sensor de temperatura y humedad DHT22.....	23
3.2.2 Sensor movimiento PIR .....	24

3.2.3 Sensor de hidrogeno gas MQ-8.....	26
3.2.4 Pantalla LCD 16x2 .....	27
3.2.5 Adaptador I2C.....	28
3.3 Shield Ethernet W5100 .....	29
3.4 Plataforma ZABBIX.....	30
3.5 Mensajería Telegram.....	31
<b>Capítulo 4 Diseño y desarrollo (simulación) .....</b>	<b>32</b>
4.1 Etapa 1 Sensor de temperatura y humedad .....	32
4.2 Etapa 2 Sensor de hidrogeno .....	33
4.3 Etapa 3 Sensor de movimiento .....	34
4.4 Etapa 4 Montaje de sistema .....	35
4.5 Apartado de Alarmas .....	41
4.5.1 Parte 1.....	41
4.5.1 Parte 2.....	42
<b>Capítulo 5 Desarrollo del proyecto (físico).....</b>	<b>44</b>
5.1 Etapa 1 Sensores .....	44
5.2 Etapa 2 Programación de tarjeta ethernet .....	45
5.2.1 Desarrollo página web .....	49
5.3 Etapa 3 Protocolo de red SNMP .....	51
5.3.1 Pruebas de plataforma ZABBIX .....	52
5.3.1 Notificación Telegram .....	56
5.4 Etapa 4 Programación de control de actuadores.....	58
5.4.1 Sensor de hidrogeno .....	58
5.4.2 Sensor de movimiento .....	60
5.5 Etapa 5 Pruebas y resultados obtenidos .....	61
<b>Conclusión .....</b>	<b>62</b>
<b>Bibliografía .....</b>	<b>64</b>
<b>Anexos .....</b>	<b>66</b>

## Índice de tablas

Tabla 1 Pruebas y datos del primer censado del nuevo sistema .....	44
Tabla 2 Pruebas y datos del segundo censado vía intranet.....	50
Tabla 3 OID´S de los sensores DHT22.....	51
Tabla 4 OID´S de los sensores MQ-8 y PIR .....	51
Tabla 5 Análisis costo vs beneficio.....	63
Tabla 6 Programa de trabajo (Cronograma) .....	66

## Índice de ilustraciones

Ilustración 1. Organigrama CFE.....	9
Ilustración 2. Organigrama DT - GRTSE .....	10
Ilustración 3. COREFO Sureste .....	11
Ilustración 4. Sala de equipos del Hotel Tuxtla.....	12
Ilustración 5. Red SDH GRT Sureste.....	12
Ilustración 6. Conexiones de sistema anterior .....	15
Ilustración 7 Funcionalidad con RFSIM900 (anterior) .....	15
Ilustración 8. Salas de equipos TIA/EIA 942.....	16
Ilustración 9 peligros de sobrepasar niveles de Hidrogeno .....	17
Ilustración 10 Creadores de tarjeta Arduino.....	18
Ilustración 11. Tipos de Arduino existentes.....	18
Ilustración 12. Diagramas de protocolos de red.....	19
Ilustración 13. Estructura MIB del protocolo SNMP para equipos de Teleprotecciones.....	20
Ilustración 14. Arduino mega 2560 R3.....	22
Ilustración 15. Conexión de DHT22.....	23
Ilustración 16. Conexión de PIR.....	24
Ilustración 17. Conexión de MQ8.....	26
Ilustración 18. Conexión de LCD 16x2.....	27
Ilustración 19. Ejemplo de conexión y funcionamiento.....	32
Ilustración 20. Diagrama de conexión de simulación.....	32
Ilustración 21. Ejemplo de conexión y funcionamiento.....	33
Ilustración 22. Diagrama de conexión de simulación.....	33
Ilustración 23. Ejemplo de conexión y funcionamiento.....	34
Ilustración 24. Diagrama de conexión de simulación.....	34
Ilustración 25. Simulación de sistema completo.....	35
Ilustración 26. Sensor dht22 #1.....	36
Ilustración 27. Sensor dht22 #2.....	37
Ilustración 28. Sensor dht22 #3.....	37
Ilustración 29. Sensor dht22 #4.....	37
Ilustración 30. Sensor MQ-8 .....	38

Ilustración 31. Función de encendido de tarea 1.....	38
Ilustración 32. Función de apagado de tarea 1 .....	39
Ilustración 33. Función de apagado de tarea 2 .....	40
Ilustración 34. Función de tarea 2 con valor de sensor "1" .....	40
Ilustración 35. Función de tarea 2.....	41
Ilustración 36. Función de led verde.....	42
Ilustración 37. Demostración de alarma con extractor encendido.....	42
Ilustración 38. Demostración de alarma con extractor apagado.....	42
Ilustración 39. Sensor de temperatura 1 sobre el límite establecido.....	43
Ilustración 40. Sensor de temperatura 2 debajo del límite establecido .....	43
Ilustración 41. sensor DHT22 1 en operación.....	44
Ilustración 42. Mensaje de presentación del sistema .....	44
Ilustración 43. sensor DHT22 3 en operación.....	44
Ilustración 44. sensor DHT22 2 en operación.....	44
Ilustración 45. sensor MQ-8 en operación.....	44
Ilustración 46. sensor DHT22 4 en operación.....	44
Ilustración 47. instalación y pruebas de alimentación de Shield Ethernet.....	46
Ilustración 48. Primeras comunicaciones de Arduino y Shield Ethernet .....	47
Ilustración 49. valores de pines análogos de Arduino en Intranet.....	47
Ilustración 50. instalación de tarjeta Ethernet en el sistema.....	48
Ilustración 51. Primeras pruebas de web (página en proceso de diseño).....	48
Ilustración 52. Valor de los sensores.....	48
Ilustración 53. Diseño página web .....	49
Ilustración 54. Diseño de página web .....	49
Ilustración 55. Diseño final de página web .....	50
Ilustración 56. Captura de MIB Browser .....	51
Ilustración 57. Datos SNMP Arduino .....	52
Ilustración 58. OID'S de los sensores y valores.....	52
Ilustración 59. Estatus de sistemas registrados en ZABBIX.....	53
Ilustración 60. Configuración del sistema de monitoreo y control de alarmas.....	53
Ilustración 61. Configuración de variables en la plataforma .....	54
Ilustración 62. Grafica sala de baterías.....	55
Ilustración 63. Grafica de sala de equipos .....	55
Ilustración 64. Configuración de triggers .....	56
Ilustración 65. Estructura del mensaje en la notificación .....	57
Ilustración 66. Aviso sensor de temperatura y humedad 2.....	57
Ilustración 67. Aviso sensor de hidrogeno.....	57
Ilustración 68. Diagrama de conexión de tarea 1 .....	58
Ilustración 69. Notificación de alarma de hidrogeno .....	59
Ilustración 70. Alerta de nivel alto de hidrogeno vía web.....	59
Ilustración 71. Notificación vía web de movimiento en el COREFO.....	60
Ilustración 72. Notificación Telegram de movimiento en el COREFO .....	60
Ilustración 73. Presentación del proyecto a personal operativo de la GRTSE .....	68
Ilustración 74. Demostración para programar Arduino .....	68

# Capítulo 1 Generalidades

## Introducción

Hoy en día, uno de los mayores avances tecnológicos en el área de la electrónica es la domótica, misma que tiene la finalidad de automatizar casi cualquier objeto, desde viviendas enteras hasta edificaciones colosales, aportando con esto servicios de monitoreo y control de gestión, tanto para el área energética, de seguridad, para el bienestar de las personas y para diversos sistemas de comunicaciones. Este tipo de sistemas pueden estar integrados por medio de redes interiores y exteriores de comunicaciones, ya sea cableadas (fibra óptica o cobre) o inalámbricas (tecnologías Wifi o Bluetooth), y cuyo control goza de cierta comodidad. Con lo anterior, la domótica se podría definir como: “la integración de la tecnología en el diseño inteligente de un recinto cerrado para la facilitación de tareas sin necesidad de contar con personal presente en el área de trabajo”.

Desde la construcción y puesta en operación del Centro Regional de Fibra Óptica, mejor conocido como COREFO y que forma parte de la Gerencia Regional de Transmisión Sureste (GRTSE), perteneciente al corporativo de la Empresa Productiva Subsidiaria CFE Transmisión, no contaba con un sistema domótico capaz de monitorear, vía Intranet, los cambios de niveles de temperatura, humedad, hidrogeno y hasta movimiento que se presentaban en sala de equipos del Hotel Tuxtla (área de alojamiento de sistemas comunicaciones), por tal razón, se genera el presente el reporte, mismo que tiene el objeto de identificar el programa de trabajo y actividades realizadas para alcanzar, a través de componentes electrónicos basados en la tecnología Arduino, el proyecto denominado: Monitoreo y Control de Alarmas del Centro de Operación Regional de Fibra Óptica Sureste.

Es necesario precisar que la GRTSE hacia uso de un sistema domótico capaz de enviar notificaciones vía telefonía celular (mensajes SMS), utilizando tecnología Arduino, pero que representaba un gasto adicional al proyecto dependiendo el número de veces que algún sensor se activará, representando con esto un costo mensual absorbido por la GRTSE y el mismo personal de la empresa.

Por lo anterior, se tomó la decisión de rediseñar el sistema y la innovación, con aporte del Instituto Tecnológico de Tuxtla Gutiérrez, para la identificación de los puntos con oportunidad de desarrollo y mejoras. Al igual de solucionar los obstáculos que presentaba el sistema anterior.

El nuevo sistema tiene como objetivo principal el envío de notificaciones de alarmas vía Intranet al sistema de monitoreo local por protocolo SNMP y a su vez, sin cargos adicionales, envío de mensajería a teléfonos celulares usando la aplicación Telegram.

En el capítulo 1 se explican datos generales de la empresa donde se desarrolla el proyecto, además de la problemática, los objetivos para solucionar el problema, la hipótesis planteada para el cumplimiento de los objetivos y la justificación del porqué se va a desarrollar el mismo.

Siguiendo con el desarrollo, en el capítulo 2, se presenta la investigación de las normas y reglas que se deben de cumplir en los sitios donde se desprenden gases inflamables y que forman parte de la sala de equipos de comunicaciones de cualquier empresa.

En el capítulo 3, se habla del marco referencial, donde se plasma la información de las especificaciones de los materiales que se utilizarán en el proyecto, así como conocer el funcionamiento de cada dispositivo.

A continuación, en el capítulo 4 se presenta la explicación y pruebas del nuevo sistema domótico en un ambiente de simulación por software para comprender y observar el comportamiento de los sensores y la tarjeta sin provocar daños. Este mecanismo de pruebas es necesario antes de la integración física de cada componente.

En el capítulo 5, se realiza la integración de todos los componentes, así como las pruebas de operación y puesta en servicio del sistema de monitoreo y control de alarmas en la sala de equipos del COREFO SURESTE.

Como puntos finales de este reporte, se presentan la conclusión obtenida por los resultados de las pruebas, y los anexos de información para la aclaración de dudas.

## **1.2 Antecedentes**

### **1.2.1 Empresa**

Comisión Federal de Electricidad, Empresa Productiva del Estado

Actualmente, Comisión Federal de Electricidad (CFE) es una Empresa Productiva del Estado, propiedad exclusiva del Gobierno Federal, con personalidad jurídica y patrimonio propio, que goza de autonomía técnica, operativa y de gestión, conforme a lo dispuesto en la Ley de la Comisión Federal de Electricidad.

El objetivo general de Comisión Federal de Electricidad tiene como fin el desarrollo de actividades empresariales, económicas, industriales y comerciales en términos de su objeto, generando valor económico y rentabilidad para el Estado Mexicano como su propietario.

En la ejecución de su objeto, Comisión Federal de Electricidad debe actuar de manera transparente, honesta, eficiente, con sentido de equidad, y responsabilidad social y ambiental, procurando el mejoramiento de la productividad con sustentabilidad para minimizar los costos de la industria eléctrica en beneficio de la población y contribuir con ello al desarrollo nacional.

Asimismo, Comisión Federal de Electricidad garantizará el acceso abierto a la Red Nacional de Transmisión y a las Redes Generales de Distribución, la operación eficiente del sector eléctrico y la competencia.

### **1.2.2 Visión general de la CFE**

*“Para 2025, CFE tiene la visión de ser una de las empresas líderes en el sector eléctrico y energético, con presencia internacional, fortaleza financiera, e ingresos adicionales por servicios relacionados con su capital intelectual e infraestructura física y comercial.”*

### **1.2.3 Fundación**

Fue fundada el 14 de agosto de 1937 por el Gobierno Federal y sus primeros proyectos se realizaron en Teloloapan, Guerrero; Pátzcuaro, Michoacán; Suchiate y Xía, en Oaxaca, y Ures y Altar, en Sonora. CFE abastece cerca de 26.9 millones de clientes e incorpora anualmente más de un millón. Desde octubre de 2009, se hace cargo de las operaciones de la compañía Luz y Fuerza del Centro. CFE es la empresa más grande del sector eléctrico de Latinoamérica. Asimismo es propietaria de la única central nucleoelectrica existente en el país, la Central Nuclear de Laguna Verde ubicada en el estado de Veracruz, misma que usa dos reactores de tipo BWR construidos por General Electric.



## 1.2.4 CFE – Organigrama

CFE Transmisión es una Empresa Productiva Subsidiaria que forma parte del corporativo de Comisión Federal de Electricidad, la cual tiene por objeto realizar las actividades necesarias para prestar el servicio público de transmisión de energía eléctrica, así como de llevar a cabo, entre otras actividades, el financiamiento, instalación, mantenimiento, gestión, operación y ampliación de la infraestructura necesaria para prestar el servicio público.



Ilustración 1. Organigrama CFE

Fuente: (CFE, 2019)

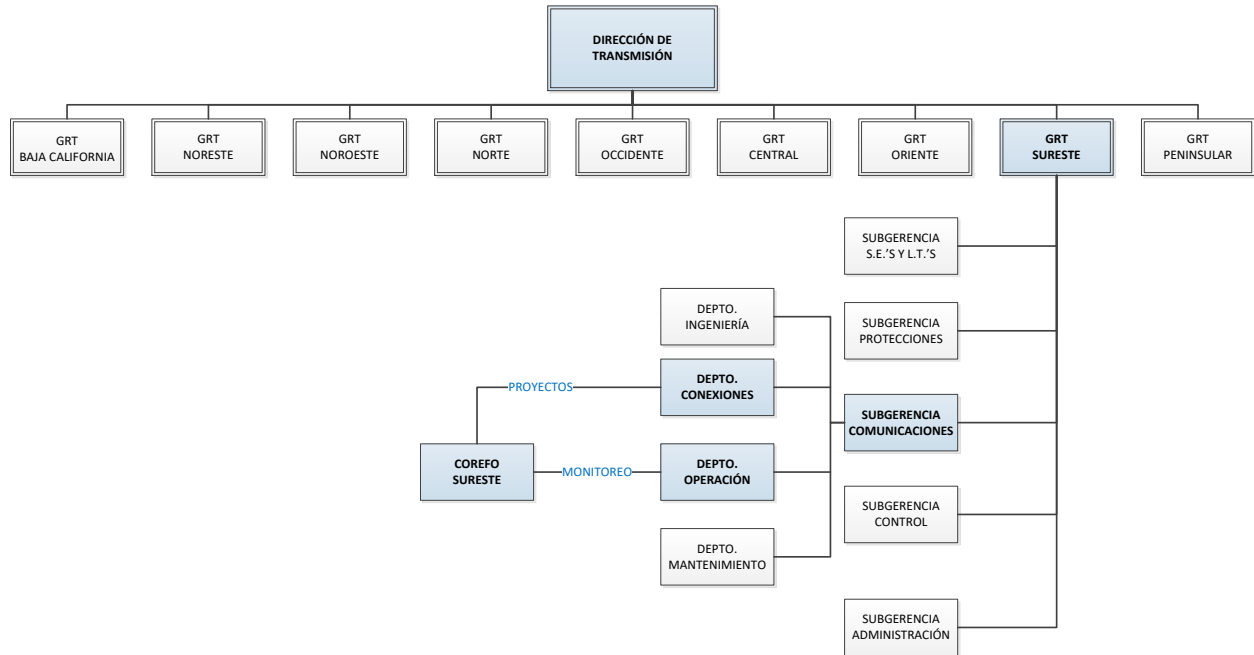


Ilustración 2. Organigrama DT - GRTSE

Fuente: Elaboración propia, 2019

### 1.2.5 Gerencia Regional de Transmisión Sureste

La Gerencia Regional de Transmisión Sureste (GRTSE) es una de las nueve gerencias que conforman la Dirección de Transmisión, y su ámbito de influencia abarca los Estados de Chiapas, Oaxaca, Tabasco y parte de Veracruz. La responsabilidad de la GRTSE es prestar el servicio público de transmisión de Energía Eléctrica, mediante la operación, mantenimiento, expansión y modernización de la Red Nacional de Transmisión, garantizando un acceso abierto y no indebidamente discriminatorio y cumpliendo con condiciones reguladas de disponibilidad, continuidad y eficiencia, para crear valor económico y rentabilidad para el Estado Mexicano.

Geográficamente está constituida por 5 Zonas de Transmisión, Villahermosa, Tuxtla, Tapachula, Istmo y Malpaso y una Zona de Operación Sureste.

## 1.2.6 COREFO GRTSE

Es el Centro de Operación Regional de Fibra Óptica de la Gerencia Regional de Tuxtla Gutiérrez y tiene bajo su responsabilidad el monitoreo de todos los servicios de comunicaciones que transitan en nodos SDH, MPLS y BG30.

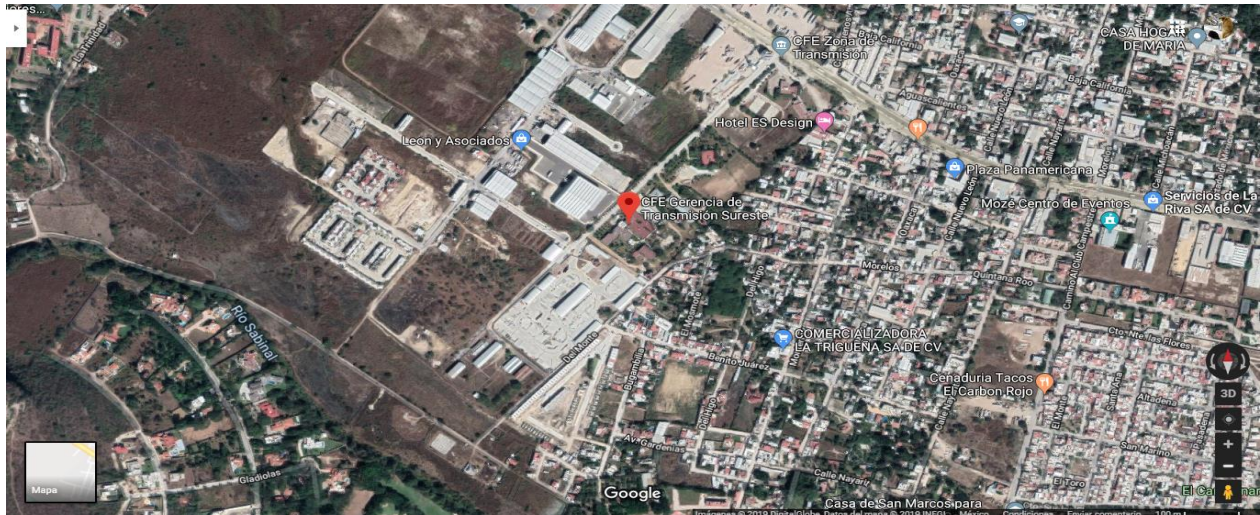


Ilustración 3. COREFO Sureste

Fuente: Imagen propia, 2019

## 1.2.7 Ubicación geográfica

Coordenadas: 16.761099, -93.190841. Mapa 50km:



Ubicación de residencia: Carretera Panamericana 4, plan de Ayala, 29020 Tuxtla Gutiérrez, chis.

### 1.3 Datos técnicos

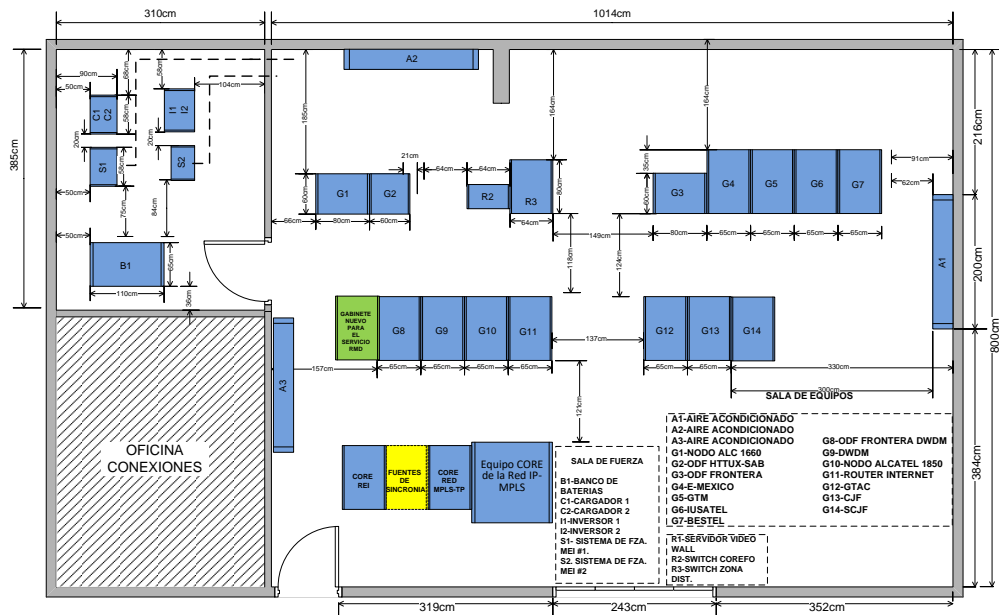


Ilustración 4. Sala de equipos del Hotel Tuxtla

Fuente: Elaboración propia, 2019

### 1.4 Topología de la red regional del COREFO Sureste

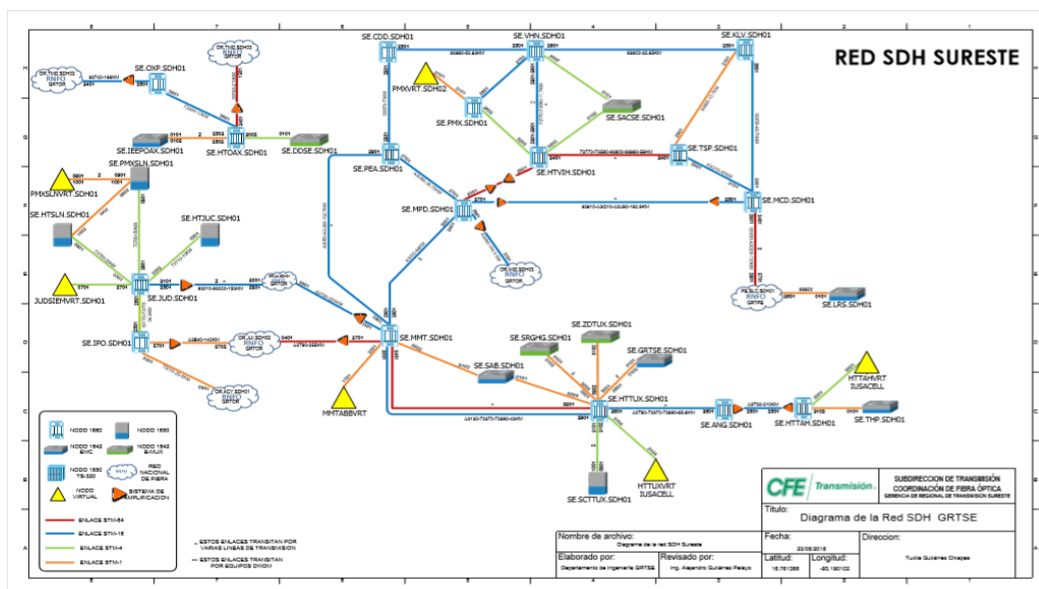


Ilustración 5. Red SDH GRT Sureste

Fuente: Elaboración propia, 2019

## 1.5 Problemática

El sistema de alarmas que se encuentra en operación se basa en la utilización de una tarjeta programable llamada Arduino de modelo mega, esta tarjeta se encuentra operando con 4 sensores DHT22 para el monitoreo de temperatura y humedad, un sensor PIR para detectar el movimiento, un sensor MQ-8 para el monitoreo de nivel de hidrogeno que se encuentra en una habitación de sistema de fuerza, y por ultimo una tarjeta GSM modelo RFSIM900 para poder mandar mensajes SMS a números celulares específicos.

Debido a lo anterior, el hacer uso del envío de notificación de alarmas vía mensajería de texto (SMS), a través de la tarjeta RFSIM900 de tecnología Arduino, implicaba un gasto mensual como parte de la funcionabilidad del proyecto que, para este momento, se encontraba fuera de operación con las siguientes limitaciones.

Varios de estas limitaciones son los siguientes:

- Los sensores solo se encuentran monitoreando las condiciones de manera constante sin realizar alguna otra acción.
- El sistema de aviso que alerta a las personas de alguna variación en los sensores se vuelve obsoleto, cuando este no cuenta con lo necesario para cumplir dicha función.
- No se cuenta con una base de datos que capture valores de los sensores para futuras graficas o tablas.
- Para observar el monitoreo, uno debe de estar de manera física en la misma sala.

## 1.6 Objetivo de proyecto

### 1.6.1 Objetivo general

El objetivo general del proyecto ***“Monitoreo y control de alarmas”*** es:

- Usando como base la tecnología Arduino, realizar la planeación, diseño y puesta en servicio de un sistema de alarmas que permita monitorear la temperatura, humedad y presencia de movimiento en la sala de equipos del COREFO, así como el control automático de extracción de gases cuando se superen los niveles de hidrógeno permitidos en la sala de baterías, atendiendo con esto las normas internacionales aplicables a un Site de Comunicaciones; TIA/EIA 942 e ISO 10156.

## 1.6.2 Objetivos específicos

- Evitar los gastos de telefonía celular utilizados anteriormente
- A través del módulo Ethernet Arduino, obtener los niveles de los sensores y enviarlos al sistema de monitoreo ZABBIX (protocolo SNMP).
- Enviar las notificaciones a los teléfonos celulares del personal responsable de la sala a través de la mensajería Telegram (sin costo).
- Monitorear en tiempo real a través de la intranet (página web).
- Control automático de extracción de gases.

## 1.7 Hipótesis

¿Es posible realizar la implementación de un sistema de monitoreo y control de alarmas, basado en tecnología Arduino, que notifique vía Red de Datos de la CFE (Intranet) cuando se exceden los límites de operación de los sensores utilizados en el proyecto con la finalidad de detectar de manera inmediata alguna situación anormal que cause algún evento perjudicial en los sistemas de Comunicaciones monitoreados de la Gerencia Regional de Transmisión Sureste?

## 1.8 Justificación

Al mes de enero del año 2019, en la sala de equipos del COREFO Sureste existía un sistema de monitoreo de alarmas y notificaciones remotas vía mensajería de texto por medio de tecnología de telefonía celular, sin embargo, este llevaba más de un año fuera de operación.

Las variables de monitoreo de ese sistema eran las siguientes:

- Temperatura
- Humedad
- Hidrógeno en sala de baterías
- Detección de movimiento



## Diseño usando RFSIM900 Arduino (anterior):

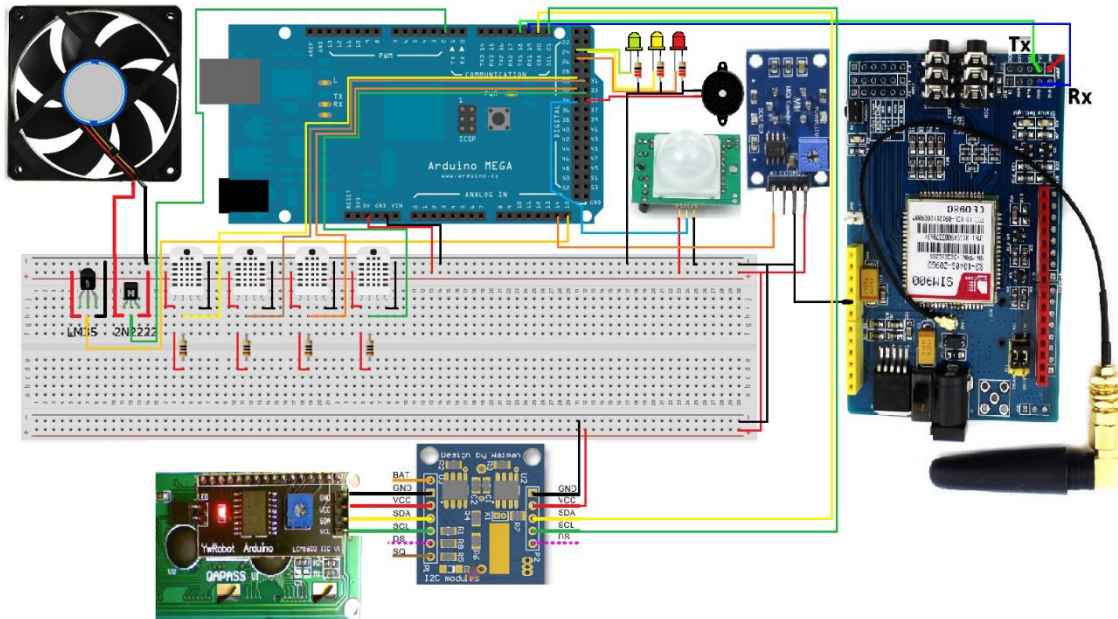


Ilustración 6. Conexiones de sistema anterior

Fuente: Elaboración propia, 2019

El servicio de notificación por mensajería SMS se activa debido alguna alteración en el valor medido de los sensores o falla en la conexión de alguno sensor, alertando a cada número telefónico de los operadores que se encuentren registrados en la programación de la tarjeta.



Ilustración 7 Funcionalidad con RFSIM900 (anterior)

Fuente: Elaboracion propia, 2019

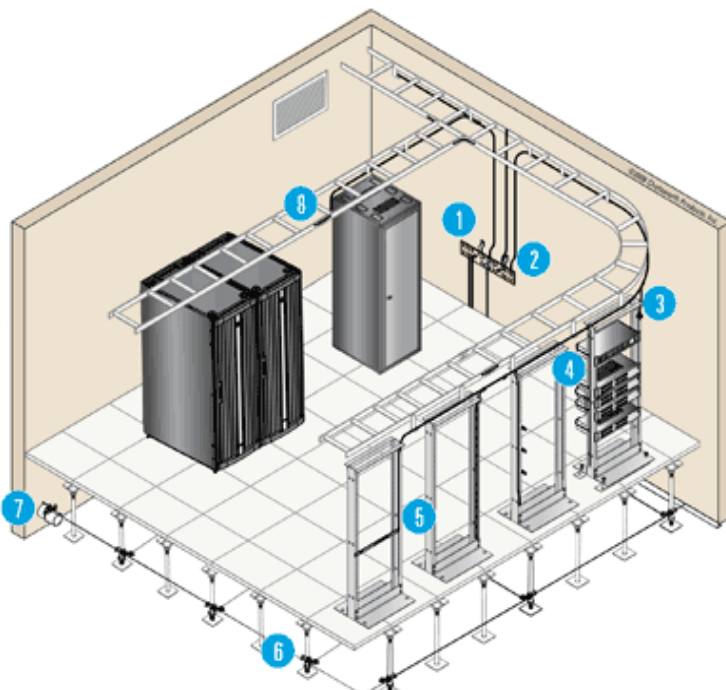
## Capítulo 2 Marco teórico

### 2.1 Importancia de monitoreo de condiciones ambientales en salas.

#### 2.1.1 Especificaciones/ normas de salas de equipos TIA/EIA 942

De acuerdo con las normas que aplican para las salas de datos se tiene que establecer límites de operación y condiciones atmosféricas aceptables para prolongar la vida útil de los equipos de datos, así como asegurar que no ocurrirán accidentes relacionados con líquidos o gases.

##### 1. Conexión a tierra de barras



2. Lengüetas de compresión y grifos

3. Agujero – bloque de terminales de tierra doble

4. Alambre y puentes de tierra

5. Abrazadera de pedestal con el conector de puesta a tierra

6. Abrazadera de tubo con conector de puesta a tierra

7. Kit de cinta de tierra para aterrizaje de cable Distribución de un Data Center, tomado de (Vicent, 2013). Es por lo anterior que hacemos

Ilustración 8. Salas de equipos TIA/EIA 942

Fuente (GRTS, 2019)

Referencia de la norma TIA/EIA-942.

TIA/EIA-942: Mantener la temperatura adecuada de forma estabilizada y controlada dentro de la sala de datos es una pauta fundamental del control ambiental que debe asegurar un DCIM, permitiendo el establecimiento y ejecución de una política claramente definida que contribuya a tener un Data Center robusto, confiable y durable. Actualmente en el mundo de IT existe una discusión sobre cual temperatura ideal para operar una sala de datos debido a las mejores prácticas recomendadas por el reconocido organismo ASHRAE (American Society of Heating, Refrigerating and Air conditioning Engineers)



En 2011, donde la entidad sugiere un rango de operación permitido más amplio según el tipo de sala de datos, es decir, más elevado que lo afirmado en su previa publicación en 2008, y más aún comparada contra la versión del 2004.

En 2004 la recomendación era entre 20°C y 25°C, en 2008 fue de 18°C y 27°C. Este último es el estándar de muchas salas de datos alrededor del mundo que utilizan como tendencia.

Las mediciones típicas de rendimiento y durabilidad en las fichas técnicas de los fabricantes están establecidas a temperatura estándar de 25°C para lograr un funcionamiento ideal en rendimiento y durabilidad, de acuerdo con la Ley de Arrhenius o también conocida como la regla de los 10 grados. Esta regla dice que la vida de un componente o material se reduce a la mitad por cada 10°C de aumento en la temperatura, e inversamente; por cada 10°C de disminución de temperatura, la vida útil de un semiconductor se duplicará. (Génesis Data, 2015)

## 2.2 Importancia de detección de gases en salas de fuerza

### 2.2.1 Especificaciones/normas de hidrogeno ISO 10156

De acuerdo con las normas que aplican para riesgos con baterías de ácido-plomo, como la batería **LPF12-180 180ah** que está instalada en la sala de fuerza del COREFO, se explica que en caso de que el sello de la batería se rompa, se puede salir gas hidrogeno y este tipo de gas es extremadamente explosivo. Por esta razón se utiliza la norma ISO 10156 que explica algunas consideraciones a tomar.

Norma ISO 10156. Las mezclas de hidrógeno – gas inerte son mezclas combustibles cuando su contenido de hidrógeno está por encima de ciertos niveles. Los valores correspondientes se han estandarizado en la norma ISO 10156. Se especifica que las mezclas de hidrógeno / nitrógeno con más de 5.7% y las de hidrógeno / helio o hidrógeno / argón con más de 2.9% de concentración H<sub>2</sub> son combustibles. (Abelló Linde, 2015).



Ilustración 9 peligros de sobrepasar niveles de Hidrogeno

Fuente: Internet

## 2.3 Historia del Arduino

Arduino es una tarjeta electrónica digital y además es un lenguaje de programación basado en C++ que es «open-source». En español se traduce como de «uso-libre». Su Hardware está construido por un microcontrolador de la familia AVR y es una de las tarjetas electrónicas más usadas para crear prototipos. Las instrucciones del lenguaje Arduino son muy fáciles de aprender y usar, incluso para personas con poco conocimiento de electrónica y/o programación. Es una herramienta muy utilizada por estudiantes y profesionales de sistemas embebidos. Dentro de las tarjetas Arduino más conocidas se encuentra el Arduino UNO R3.

Arduino Nació en el año 2005 el Instituto de Diseño Interactivo de Ivrea (Italia). Arduino apareció por la necesidad de contar con un dispositivo para utilizar en aulas que fuera de bajo coste. La idea original fue, fabricar una placa para uso interno de la escuela.

Sin embargo, el instituto se vio obligado a cerrar sus puertas precisamente en 2005. Ante la perspectiva de perder todo el proyecto Arduino en el proceso, se decidió liberarlo y abrirlo al público para que todo el mundo pudiese participar en la evolución del proyecto, proponer mejoras y sugerencias.

Los principales responsables de la idea y diseño de Arduino fueron Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe y Gianluca Martino.



Ilustración 10 Creadores de tarjeta Arduino

Fuente: (Arduino, 2019)

### 2.3.1 Evolución del Arduino

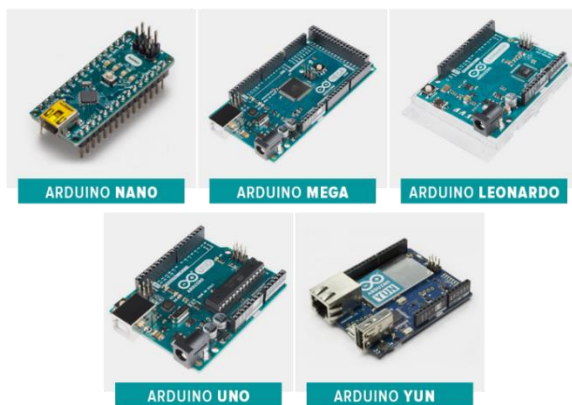


Ilustración 11. Tipos de Arduino existentes

Fuente: (Arduino, 2019)

Se han fabricado diferentes modelos de placas Arduino oficiales, cada una pensada con un propósito diferente y características variadas (como el tamaño físico, número de pines E/S, modelo del microcontrolador, etc). A pesar de las varias placas que existen todas pertenecen a la misma familia (microcontroladores AVR marca Atmel), esto significa que comparten la mayoría de sus características de software, como arquitectura, librerías y documentación.

## 2.4 Protocolos de red

Es el término que se emplea para denominar al conjunto de normas, reglas y pautas que sirven para guiar una conducta o acción. Red, por su parte, es una clase de estructura o sistema que cuenta con un patrón determinado.

El concepto de protocolo de red se utiliza en el contexto de la informática para nombrar a las normativas y los criterios que fijan cómo deben comunicarse los diversos componentes de un cierto sistema de interconexión. Esto quiere decir que, a través de este protocolo, los dispositivos que se conectan en red pueden intercambiar datos.

También conocido como protocolo de comunicación, el protocolo de red establece la semántica y la sintaxis del intercambio de información, algo que constituye un estándar. Las computadoras en red, de este modo, tienen que actuar de acuerdo con los parámetros y los criterios establecidos por el protocolo en cuestión para lograr comunicarse entre sí y para recuperar datos que, por algún motivo, no hayan llegado a destino.

### 2.4.1 Protocolo SNMP

#### Protocolo SNMP:

Uno de los protocolos más importantes que se puede utilizar en la recopilación de información de los dispositivos es SNMP.

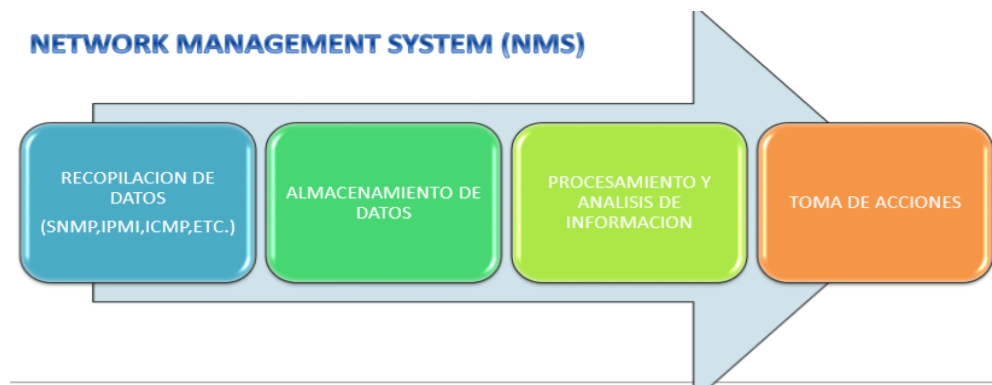


Ilustración 12. Diagramas de protocolos de red

Fuente: Elaboración propia, 2019

El protocolo SNMP perteneciente a la pila de protocolos de modelo OSI ubicado en la capa de aplicación, significa Protocolo Simple de Administración de Red o SNMP (del inglés Simple Network Management Protocol), facilita el intercambio de información de administración entre dispositivos de red. Permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

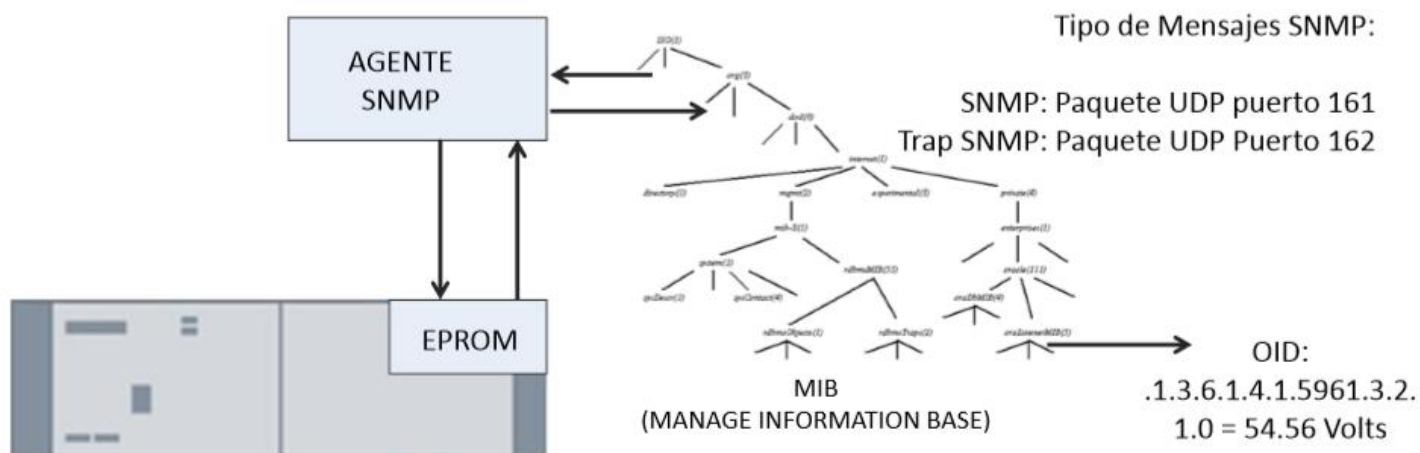


Ilustración 13. Estructura MIB del protocolo SNMP para equipos de Teleprotecciones

Fuente: Elaboración propia, 2019

Puntos para manejar por parte del Protocolo Simple de Administración de Red:

- Protocolo estándar de internet para administrar dispositivos en redes IP
- Más precisamente, es la manera estándar de monitorear hardware y software de cualquier fabricante desde Cisco a Juniper, desde Microsoft a Unix.
- Parte del protocolo de control de transmisiones / protocolo de internet (TCP / IP)
- Utiliza UDP como protocolo de transporte

El protocolo SNMP consta de una estructura de trabajo para realizar sus funciones como la que muestra a continuación:

- Componentes
  - Administrador de SNMP
  - Agente SNMP
  - Dispositivos administrados



Descripción de componentes SNMP:

- **Dispositivo administrado:** Puede ser cualquier dispositivo de red como:
  - Equipos de redes unificadas (switches, routers, firewalls, servidores, acces points, etc)
  - Sistemas de alimentación (sistemas de fza 48 VCD, 12 VCD, inversores)
  - Equipos de Teleproteccion y Powerlink (acuses RX-TX)
  - Puede ser cualquier servidor:
    - Físico o virtual

- cualquier SO: linux, solaris, windows, HP-UX
  - 
  - Cualquier dispositivo con una IP y un agente SNMP (MICROCONTROLADORES SNMP)
- **Agente SNMP:**
  - Windows SNMP
  - \*UX net-snmp
  - Cisco Cisco SNMP
  - OTROS Agent SBMP
- **Dispositivo administrado:** Cada fabricante de dispositivos, configura el agente SNMP en el dispositivo administrado para poder:
  - Recolecta información administrativa sobre su entorno local
  - Almacena y recupera información tal como se definió en la MIB
  - Indica cuando se produce un evento al administrador

El protocolo SNMP ha obtenido cambios de versiones de sí mismo con forme el pasar de los años:

Versiones de SNMP:

- **SNMPv1**
  - Ampliamente utilizado, protocolo de administración de red de facto
  - Disponible desde 1988
  - Autenticación mediante nombre de comunidad
- **SNMPv2**
  - Modificación de la versión 1
  - Mejoras en el desempeño, seguridad, y en las comunicaciones entre administradores
  - Incorporó el comando GetBulkRequest
    - Escasa seguridad
    - Nombre de comunidad pasado en texto plan
- **SNMPv3** Agrega mejoras en la seguridad y en la configuración remota para:
  - Cada entidad SNMP tiene un identificador
    - SNMPEngineID
  - La comunicación es posible únicamente si la entidad SNMP conoce la identidad de su par.
  - Especificación para USM Modelo de seguridad basado en usuario
    - NoAuthNoPriv
    - AuthNoPriv
    - AuthPriv
  - Protocolos de autenticación
    - MD5, SHA
  - Protocolos de privacidad

# Capítulo 3 Marco referencial

## 3.1 Arduino mega

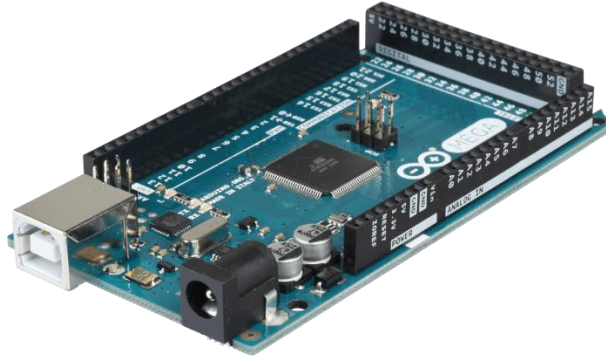


Ilustración 14. Arduino mega 2560 R3

Fuente: (Arduino, 2019)

Arduino Mega es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede comunicarse a un PC a través del puerto serial (conversión con USB) utilizando lenguajes como Flash, Processing, MaxMSP, etc. Las posibilidades de realizar desarrollos basados en Arduino tienen como límite la imaginación.

El Arduino Mega tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Arduino Mega incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa (9 hasta 12VDC). El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor utiliza un microcontrolador ATmega8U2 en vez del circuito integrado FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

Características:

- Microcontrolador ATmega2560.
- Voltaje de entrada de – 7-12V.
- 54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).



- 16 entradas análogas.
- 256k de memoria flash.
- Velocidad del reloj de 16Mhz.

## 3.2 Sensores

### 3.2.1 Sensor de temperatura y humedad DHT22



El DHT22 (AM2302) es un sensor digital de temperatura y humedad relativa de buen rendimiento y bajo costo. Integra 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 posee salida analógica). Utilizado en aplicaciones de control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.

Ilustración 15. Conexión de DHT22

Fuente: (NALBER, 2019)

Utilizar el sensor DHT22 con las plataformas Arduino/Raspberry Pi/Nodemcu es muy sencillo tanto a nivel de software como hardware. A nivel de software se dispone de librerías para Arduino con soporte para el protocolo "Single bus". En cuanto al hardware, solo es necesario conectar el pin VCC de alimentación a 3-5V, el pin GND a Tierra (0V) y el pin de datos a un pin digital en nuestro Arduino. Si se desea conectar varios sensores DHT22 a un mismo Arduino, cada sensor debe tener su propio pin de datos. Quizá la única desventaja del sensor es que sólo se puede obtener nuevos datos cada 2 segundos. Cada sensor es calibrado en fabrica para obtener unos coeficientes de calibración grabados en su memoria OTP, asegurando alta estabilidad y fiabilidad a lo largo del tiempo. El protocolo de comunicación entre el sensor y el microcontrolador emplea un único hilo o cable, la distancia máxima recomendable de longitud de cable es de 20m, de preferencia utilizar cable apantallado. Proteger el sensor de la luz directa del sol (radiación UV).

El DHT22 presenta mejores prestaciones respecto al sensor DHT11, como mejor resolución, mayor precisión y un empaque más robusto.

Especificaciones técnicas:

- Voltaje de Operación: 3V - 6V DC
- Rango de medición de temperatura: -40°C a 80 °C
- Precisión de medición de temperatura: <math>\pm 0.5^\circ\text{C}</math>
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: De 0 a 100% RH
- Precisión de medición de humedad: 2% RH

- Resolución Humedad: 0.1%RH
- Tiempo de sensado: 2s
- Interfaz digital: Single-bus (bidireccional)
- Modelo: AM2302
- Dimensiones: 20\*15\*8 mm
- Peso: 3 gr.
- Carcasa de plástico blanco

Pines:

- Alimentación: +5V (VCC)
- Datos (DATA)
- No Usado (NC)
- Tierra (GND)
- \*Recomendamos utilizar una resistencia de 4.7K Ohm en modo Pull-up, entre el pin de Datos y VCC

Límites establecidos para CFE:

- Límite mínimo temperatura: 18°C
- Límite máximo temperatura: 30°C
- Límite de operación estable de temperatura: 18°C – 30°C
- Límite mínimo de humedad: 20%
- Límite máximo de humedad: 60%
- Límite de operación estable de humedad: 20% - 60%

### 3.2.2 Sensor movimiento PIR

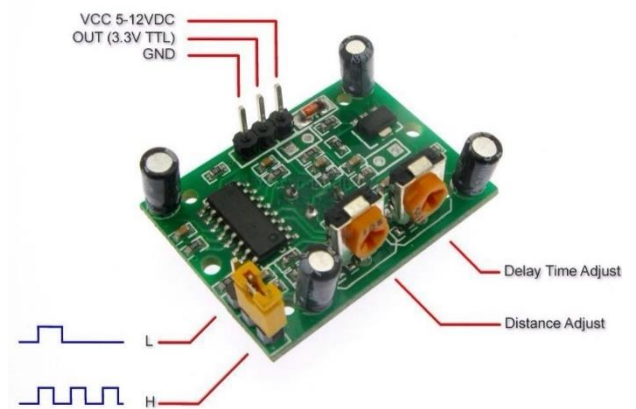


Ilustración 16. Conexión de PIR

Fuente: (Arduino, 2019)

Los detectores PIR (Passive Infrared) o Pasivo Infrarrojo, reaccionan sólo ante determinadas fuentes de energía tales como el calor del cuerpo humano o animales. Básicamente reciben la variación de las radiaciones infrarrojas del medio ambiente que cubre. Es llamado pasivo debido a que no emite radiaciones, sino que las recibe. Estos captan la presencia detectando la diferencia entre el calor emitido por el cuerpo humano y el espacio alrededor.



Su componente principal son los sensores piroeléctricos. Se trata de un componente electrónico diseñado para detectar cambios en la radiación infrarroja recibida. Generalmente dentro de su encapsulado incorporan un transistor de efecto de campo que amplifica la señal eléctrica que genera cuando se produce dicha variación de radiación recibida.

La información infrarroja llega al sensor piroeléctrico a través de una lente de fresnell que divide el área protegida en sectores. Se distribuyen lentes con diferentes características: gran angular, cortina, corredor, antimascotas, etc.

### Forma de un Lente Fresnell

La señal eléctrica que genera los sensores piroeléctricos cuando detecta un cambio es procesada por un circuito electrónico de control que activará un relé en el caso que la señal tenga ciertas características (amplitud, frecuencia, duración, etc)

### Interior de un Detector de Movimiento PIR



Cuando se instalan los sensores infrarrojos (PIR) y se lo energiza por primera vez, este comenzara a “acostumbrarse” a la radiación infrarroja del ambiente (todos los cuerpos que están a una temperatura superior a 0° Kelvin, emiten radiación infrarroja negativa; esta radiación infrarroja aumenta si aumenta la temperatura del cuerpo en cuestión. Es decir, que recibe la información infrarroja de una pared, el piso, los muebles, etc).

Una vez que se mantiene estable, si un intruso ingresa al recinto se experimentará un cambio en la radiación infrarroja del ambiente y el PIR dará una condición de alarma. Una vez realizada la calibración y ajuste del detector, el led de prueba deberá quedar apagado por norma de instalación.

Con objeto de lograr total confiabilidad, esta tecnología integra, además, un filtro especial de luz que elimina toda posibilidad de falsas detecciones causadas por la luz visible (rayos solares), así como circuitos especiales que dan mayor inmunidad a ondas de radio frecuencia, así es como funcionan los sensores de movimiento.

#### Características:

- Voltaje de alimentación: 4.5 – 20V
- Voltaje de salida: 3.3V
- Corriente en stand by: 50uA
- Retardo: 5-200S (ajustable)
- Ángulo de detección: 120 °
- Temperatura de operación: -15 – +70 grados

#### Límites establecidos en CFE:

- 20 segundos de activación

- Tiempo de encendido y apagado:
  - Viernes 4:30 pm – Lunes 7:30 am

### 3.2.3 Sensor de hidrogeno gas MQ-8



- 1 = Vcc
- 2 = AOUT
- 3 = DOUT
- 4 = GND

(bottom view)

Ilustración 17. Conexión de MQ8

Fuente: (Arduino, 2019)

Sensor de Gas MQ-8 MQ8 (H<sub>2</sub>) Hidrogeno. Ideal para detección del mismo gas. Tiene la sensibilidad ajustable para tener una lectura adecuada al sistema particular. El sensor tiene un tamaño pequeño y es muy práctico y sencillo de utilizar. El módulo tiene una salida digital a través de un comparador con umbral ajustable y una salida analógica que la puedes medir con cualquier microcontrolador o tarjeta de desarrollo con ADC

Características:

- Tiene alta sensibilidad y tiempo de respuesta rápido
- Desde 100 a 10000ppm.
- Voltaje de entrada 5 V.
- Temperatura de funcionamiento: -10 a 50 °C
- Consumo de potencia: menos de 750 mW
- Resistencia de carga: regulable
- Concentración de Oxígeno: 21%
- Dispositivo base sensor MQ8
- Pines 1-salida, 2 GND, 3-VCC

Límites establecidos en CFE:

- Límite mínimo: 0.0 %
- Límite máximo: 0.40%
- Límite para activación de alarma:  $\geq 0.50\%$

### 3.2.4 Pantalla LCD 16x2

#### Pantalla LCD 16x2:



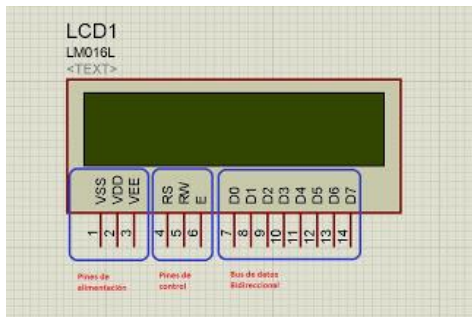
El LCD (Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está gobernado por un microcontrolador el cual dirige todo su funcionamiento.

En este caso se utilizará un LCD de 16x2, esto quiere decir que dispone de 2 filas de 16 caracteres cada una. Los píxeles de cada símbolo o carácter varían en función de cada modelo.

En la siguiente imagen de Proteus se puede observar la estructura de sus pines.

Los Pines se dividen en alimentación, pines de control y los pines del bus de datos bidireccional. Por lo general podemos encontrar además en su estructura los pines de Ánodo de led backlight y cátodo de led backlight.

Ilustración 18. Conexión de LCD 16x2



#### Pines de alimentación:

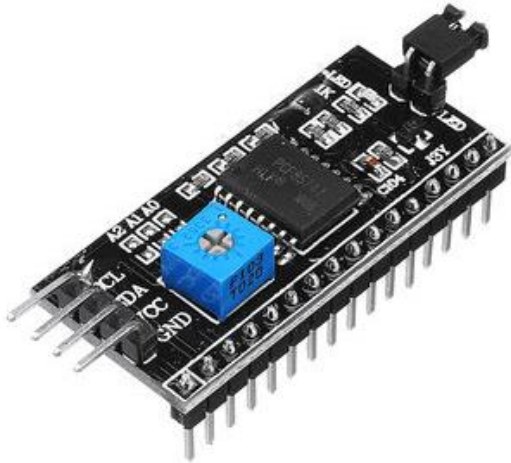
- Vss: Gnd
- Vdd: +5 voltios
- Vee: corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a Vdd.

Fuente: Elaboración propia, 2019

#### Pines de control:

- RS: Corresponde al pin de selección de registro de control de datos (0) o registro de datos (1). Es decir, el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de control/instrucción. y cuando RS es 1 el dato presente en el bus de datos pertenece a un registro de datos o un carácter.
- RW: Corresponde al pin de Escritura (0) o de Lectura (1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla.
- E: Corresponde al pin Enable o de habilitación. Si E (0) esto quiere decir que el LCD no está activado para recibir datos, pero si E (1) se encuentra activo y podemos escribir o leer desde el LCD.

### 3.2.5 Adaptador I2C



I2C es un puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA y SCL. Además, el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/s. También es conocido como IIC ó TWI – Two Wire Interface.

El protocolo I2C es uno de los más utilizados para comunicarse con sensores digitales, ya que a diferencia del puerto Serial, su arquitectura permite tener una confirmación de los datos recibidos, dentro de la misma trama, entre otras ventajas.

La conexión de tantos dispositivos al mismo bus es una de las principales ventajas. Además, si comparamos a I2C con otro protocolo serial, como Serial TTL, este incluye más bits en su trama de comunicación que permite enviar mensajes más completos y detallados.

Los mensajes que se envían mediante un puerto I2C, incluye además del byte de información, una dirección tanto del registro como del sensor. Para la información que se envía siempre existe una confirmación de recepción por parte del dispositivo. Por esta razón es bueno diferenciar a los distintos elementos involucrados en este tipo de comunicación.

#### I2C – ESQUEMA DE COMUNICACIÓN Y ELEMENTOS

Siempre que hablamos de una comunicación oral, se entiende que es entre dos o más personas. Como consecuencia podemos también indicar que en una comunicación digital existen distintos dispositivos o elementos. En el caso de I2C se diferencian dos elementos básicos, un MAESTRO y un ESCLAVO. La Figura-1, muestra una conexión típica de tres dispositivos, el bus consiste en dos líneas llamadas, Serial DAta – SDA y Serial CLock – SCL. Es decir, Datos Seriales y Reloj Serial. En particular al bus se le conectan dos resistencias en arreglo pull-up, de entre 2.2K y 10K.

**EI MAESTRO I2C** se encarga de controlar al cable de reloj, por sus siglas en inglés llamada SCL – Serial CLock. Además, el MAESTRO se encarga de iniciar y parar la comunicación. La información binaria serial se envía sólo por la línea o cable de datos seriales, en inglés se llama SDA – Serial DAta. Dos Maestros no pueden hacer uso de un mismo puerto I2C. Puede funcionar de dos maneras, como maestro-transmisor o maestro-receptor. Sus funciones principales son:

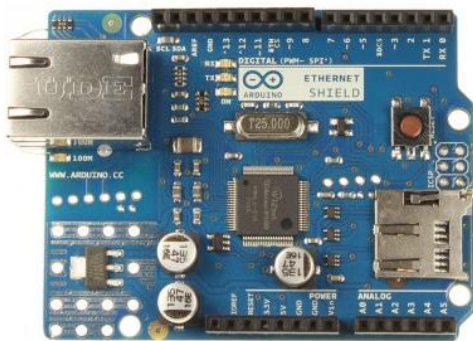
- Iniciar la comunicación – S
- Enviar 7 bits de dirección – ADDR

- Generar 1 bit de Lectura ó Escritura – R/W
- Enviar 8 bits de dirección de memoria
- Transmitir 8 bits de datos –
- Confirmar la recepción de datos – ACK – ACKnowledged
- Generar confirmación de No-recepción, NACK – No-ACKnowledged
- Finalizar la comunicación

**EI ESCLAVO I2C** generalmente suele ser un sensor. Este elemento suministra de la información de interés al MAESTRO. Puede actuar de dos formas: esclavo-transmisor ó esclavo-receptor. Un dispositivo I2C esclavo, no puede generar a la señal SCL. Sus funciones principales son:

Enviar información en paquetes de 8 bits y enviar confirmaciones de recepción, llamadas ACK

### 3.3 Shield Ethernet W5100



El Arduino Ethernet Shield permite a una placa Arduino conectarse a internet. Está basada en el chip ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para escribir programas que se conecten a internet usando la shield.

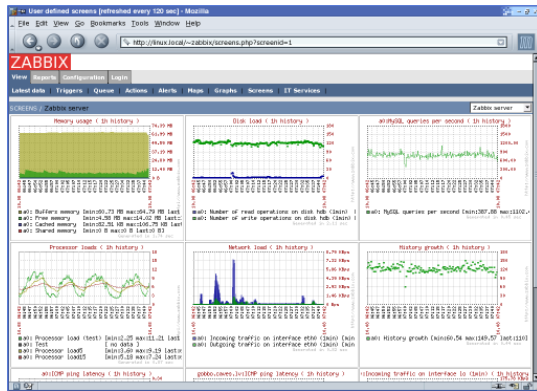
- Es compatible con el Arduino UNO y Arduino Mega.
- El shield provee un conector ethernet estándar RJ45 y un conector lector de tarjeta Micro SD
- El botón de reset en la shield resetea ambos, el W5100 y la placa Arduino.

El shield contiene un número de LEDs para información:

- PWR: indica que la placa y la shield están alimentadas
- LINK: indica la presencia de un enlace de red y parpadea cuando la shield envía o recibe datos
- FULLD: indica que la conexión de red es full duplex
- 100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- RX: parpadea cuando la shield recibe datos
- TX: parpadea cuando la shield envía datos
- COLL: parpadea cuando se detectan colisiones en la red



## 3.4 Plataforma ZABBIX



Zabbix es un Sistema de Monitorización de Redes creado por Alexei Vladishev. Está diseñado para monitorizar y registrar el estado de varios servicios de red, Servidores, y hardware de red.

Usa MySQL, PostgreSQL, SQLite, Oracle o IBM DB2 como base de datos. Su backend está escrito en C y el frontend web está escrito en PHP. Zabbix ofrece varias opciones de monitorización:

Chequeos simples que pueden verificar la disponibilidad y el nivel de respuesta de servicios estándar como SMTP o HTTP sin necesidad de instalar ningún software sobre el host monitorizado.

Un agente Zabbix puede también ser instalado sobre máquinas UNIX y Windows para monitorizar estadísticas como carga de CPU, utilización de red, espacio en disco, etc.

Como alternativa a instalar el agente sobre los host, Zabbix incluye soporte para monitorizar vía protocolos SNMP, TCP y ICMP, como también sobre IPMI, JMX, SSH, telnet y usando parámetros de configuración personalizados. Zabbix soporta una variedad de mecanismos de notificación en tiempo real, incluyendo XMPP.

Lanzado sobre los términos de la versión 2 de la GNU General Public License, Zabbix es Software Libre.

Zabbix fue iniciado como un proyecto interno de software en 1998. Después de 3 años, en 2001, este fue lanzado al público sobre GPL. Y tomo 3 años más hasta su primera versión estable, 1.0, que fue lanzada en 2004.

Funcionalidades:

- Alto rendimiento y alta capacidad (posibilidad de monitorizar cientos de miles de dispositivos)
- Auto descubrimiento de servidores y dispositivos de red
- Monitorización distribuida y una administración web centralizada
- Agentes nativos en múltiples plataformas
- Posibilidad de monitorización sin agentes
- Monitorización JMX
- Monitorización Web
- Configuración de permisos por usuarios y grupos
- Métricas SLA y ITIL
- Sistema flexible de notificación de eventos (Email, XMPP, etc)

## Desarrollo

Hoy en día Zabbix es desarrollado principalmente por una empresa dedicada a ello, Zabbix SIA.

Código fuente:

- Zabbix consiste en algunos módulos aislados:
- Servidor
- Agentes
- Frontend
- Proxy
- Java gateway

Mientras que el servidor, proxy y agentes están escritos en C, el frontend está implementado en PHP y Javascript.

El Java gateway, disponible desde Zabbix 2.0, está escrito en Java.

## 3.5 Mensajería Telegram



Telegram

Telegram Messenger es una aplicación de mensajería y VOIP desarrollada desde el año 2013 por los hermanos Nikolái y Pável Dúrov. Está enfocada en la mensajería instantánea, envío de varios archivos y comunicación en masa. El servicio lo administra una organización autofinanciada cuya sede principal opera en Dubái, Emiratos Árabes.

Entre sus funcionalidades principales están: conversaciones entre usuarios (como mensajes guardados, opción de reenvío, sincronización, alojamiento y archivado desde la nube), envío de archivos (hasta 1.5 GB, incluyendo documentos, multimedia y animaciones gráficas), gestión de contactos (adicionando la búsqueda global), encuestas, llamadas, canales de difusión, grupos, entre otros. Adicionalmente, los usuarios pueden desarrollar bots que pueden realizar otros servicios como pagos, juegos, moderación de grupos o automatización de tareas bajo inteligencia artificial.

Inicialmente el servicio fue empleado para teléfonos móviles (Android, iOS) y el año siguiente para multiplataforma (macOS, Windows, GNU/Linux, Firefox OS, navegadores web, y otros sistemas operativos). Las aplicaciones usan la interfaz de acceso gratuito, permitiendo a los desarrolladores crear clientes externos. Parte del software está bajo software libre —excepto el lado del servidor— y recibe el apoyo de la comunidad.

# Capítulo 4 Diseño y desarrollo (simulación)

## 4.1 Etapa 1 Sensor de temperatura y humedad

En esta etapa se realiza la conexión del Arduino mega con los sensores de temperatura y humedad DHT22 a través de las salidas 30, 31, 32 y 33 así como su conexión común a tierra. Se agregan imágenes de ejemplo para la conexión a Arduino y su funcionamiento.

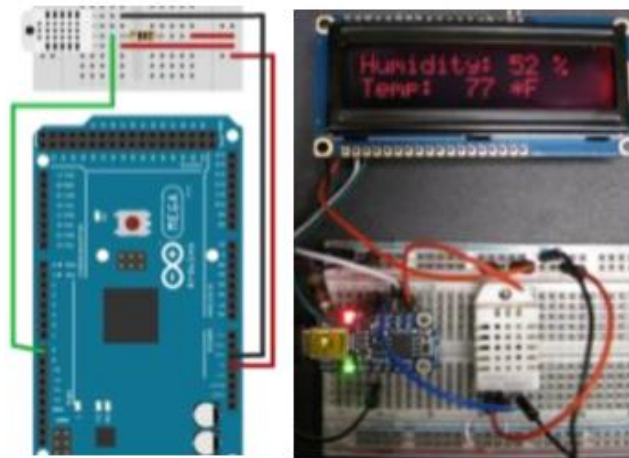


Ilustración 19. Ejemplo de conexión y funcionamiento

Fuente: Elaboración propia, 2019

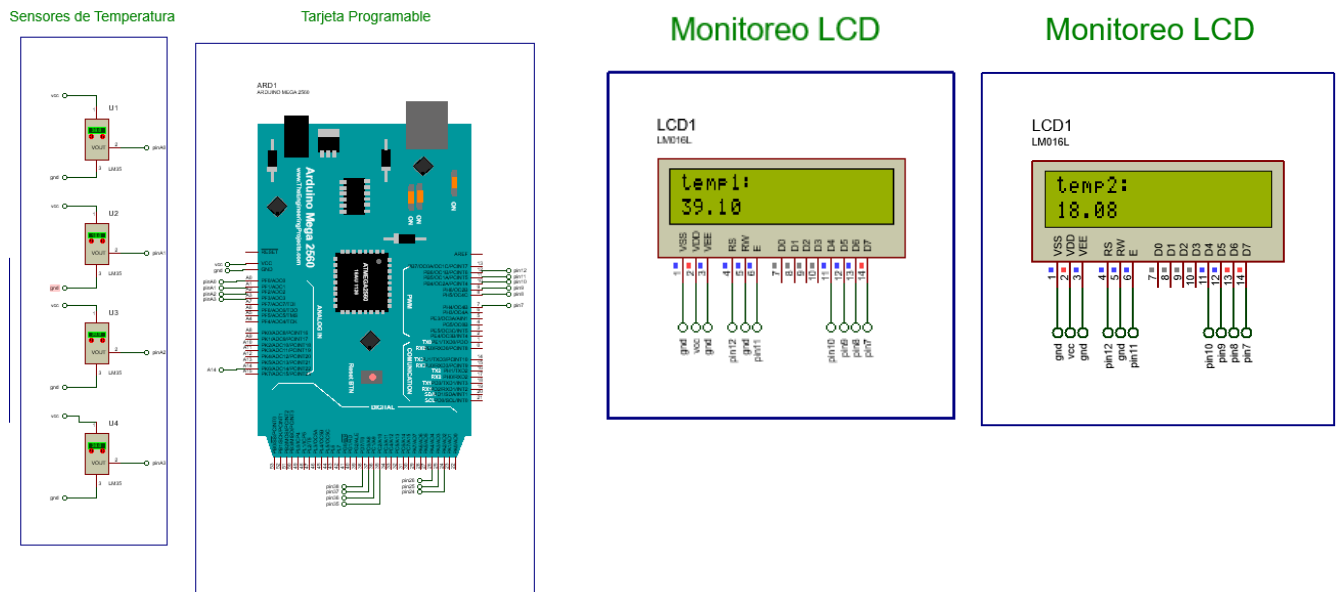


Ilustración 20. Diagrama de conexión de simulación

Fuente: Elaboración propia, 2019



En la primera etapa, se llevó acabo la programación de los sensores de temperatura, para verificar la forma de sintaxis del comando para el control y monitoreo del sensor, por precaución se realizó primero la prueba simulada, esto para evitar el riesgo.

## 4.2 Etapa 2 Sensor de hidrogeno

En esta etapa se realiza la conexión del Arduino mega con el sensor de hidrogeno a través de la salida A14, así como su conexión común a tierra. Se agregan imágenes de ejemplo para la conexión a Arduino y su funcionamiento.



Ilustración 21. Ejemplo de conexión y funcionamiento

Fuente: Elaboración propia, 2019

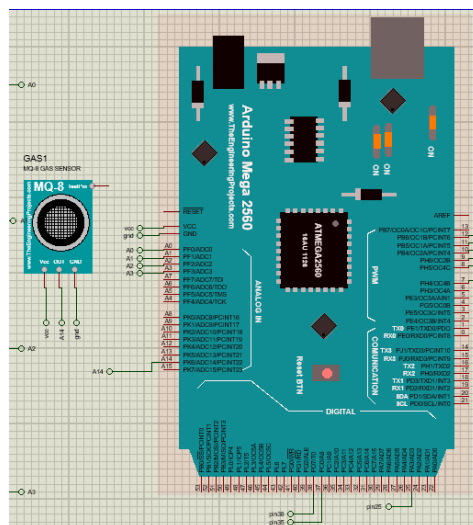


Ilustración 22. Diagrama de conexión de simulación

Fuente: Elaboración propia, 2019

Como segunda parte de la etapa 2, se programó los valores de límites del sensor de hidrogeno, para que de esta manera se pueda realizar la función de una alarma, la cual su función principal será de dar aviso cuando los valores registrados superen los límites establecidos y accionar mediante control automático el encendido de un extractor de gases que ya se encuentra en la sala de baterías.

### 4.3 Etapa 3 Sensor de movimiento

En esta etapa se realiza la conexión del Arduino mega con el sensor de movimiento a través de la salida 34, así como su conexión común a tierra. Se agregan imágenes de ejemplo para la conexión a Arduino y su funcionamiento.

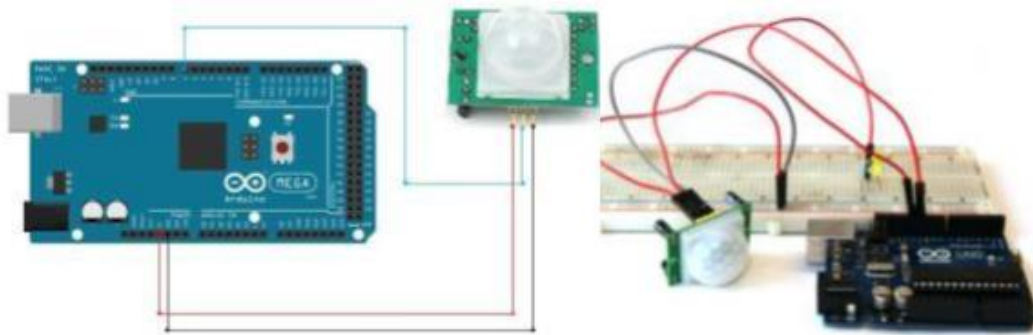


Ilustración 23. Ejemplo de conexión y funcionamiento

Fuente: Elaboración propia, 2019

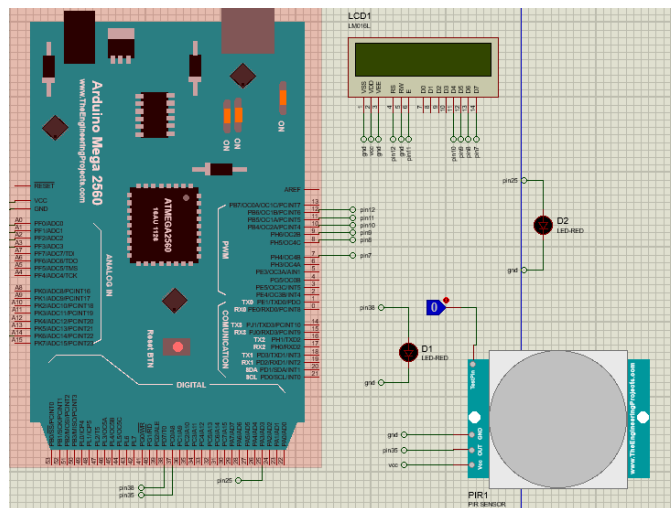


Ilustración 24. Diagrama de conexión de simulación

Fuente: Elaboración propia, 2019

Para la parte 2 de la etapa 3 se llevó a cabo la configuración en el código para la programación de tarea que el sensor debe de realizar.

La tarea asignada es la activación de una salida de la tarjeta Arduino, esta salida se puede utilizar para la activación de un objeto, en este caso se utilizara para activar un buzzer de alarma que alertara a los operadores cada vez que detecte movimiento dentro de la sala de equipos, el tiempo de operación del sensor PIR será solamente los fines de semana, del viernes a las 16 horas se encenderá automáticamente y se apagará hasta el día lunes a las 7 horas.

#### 4.4 Etapa 4 Montaje de sistema

En esta etapa ya se procedió a realizar el montaje de todos los sensores trabajando juntos, esto para ver el comportamiento que tienen y la respuesta de la tarjeta Arduino al realizar diferentes medidas al mismo tiempo, para después mostrarlas en una pantalla, como se mencionó anteriormente, ya dos sensores están programados para accionar dos salidas del Arduino, esto para poder controlar dos objetos que dependerán del monitoreo.

Claro que dicho control es de manera automática, la activación y apagado, lo realizara la propia tarjeta, sin necesidad de que haya personas.

A continuación, se procede a mostrar una ilustración simulada de cómo queda el montaje del sistema con todos los sensores trabajando:

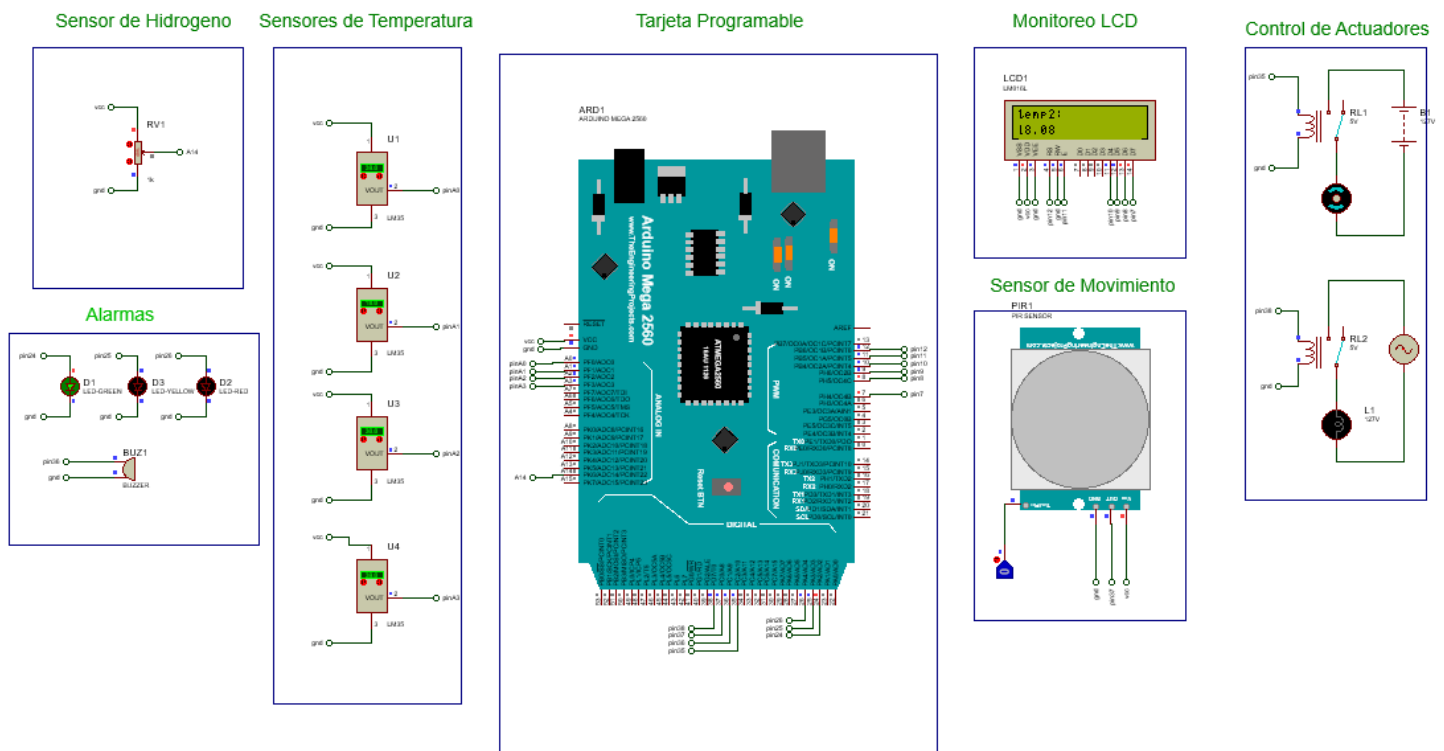


Ilustración 25. Simulación de sistema completo

Fuente: Elaboración propia, 2019

En esta parte, se procedió a utilizar el simulador, para poder observar de manera segura como opera el código ya establecido con todos los sensores trabajando al mismo tiempo, cabe decir que, en la simulación, el sistema se comporta de maravilla, responde como se desea, la recolección de los datos lo hace de manera estable.

Como se mencionó anteriormente, el sistema funciona correctamente y en lo que corresponde a la simulación, se va a dejar completado hasta esta parte, porque debido a las limitantes de software de simulación.

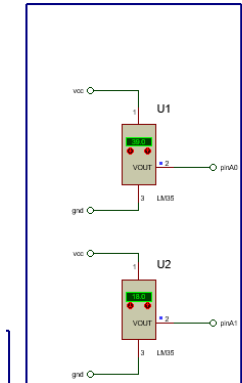
No se puede simular una tarjeta de red como la que utilizaremos para poder conectar nuestro sistema a una red local de intranet y así monitorear de manera remota los valores que el sistema nos enviara en tiempo real sin necesidad de estar físicamente en la misma oficina.

**Nota:** en la simulación se procedió a utilizar la función de default en sustitución de cableado para poder conectar los pines de cada componente, para que la simulación tenga una vista más limpia, estéticamente hablando, cada pin tiene un nombre y se conecta con los mismos pines que tengan el mismo nombre.

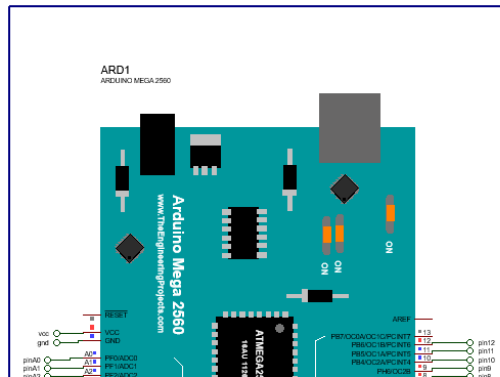
**Se procede a mostrar capturas de la simulación en funcionamiento para mayor comprensión:**

Monitoreo de los sensores DHT22 y el sensor MQ-8:

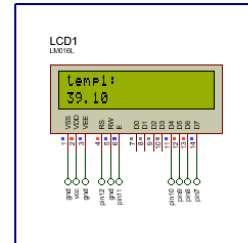
Sensores de Temperatura



Tarjeta Programable



Monitoreo LCD



Pantalla LCD muestra el valor Temp1 =39.10°C

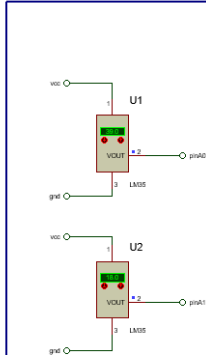
Sensor de Movimiento



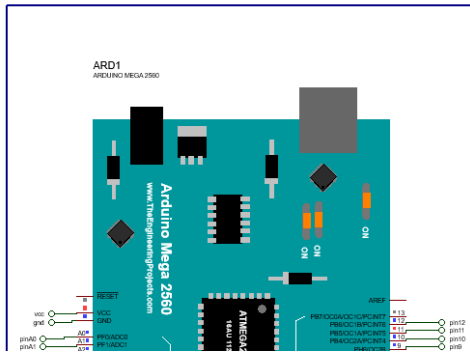
Ilustración 26. Sensor dht22 #1

Fuente: Elaboración propia, 2019

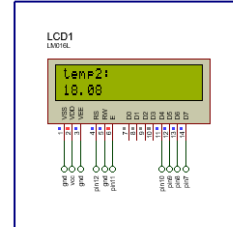
Sensores de Temperatura



Tarjeta Programable



Monitoreo LCD



Sensor de Movimiento

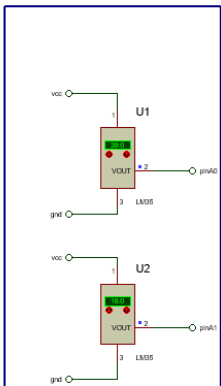


Pantalla LCD muestra el valor Temp2 =18.08°c

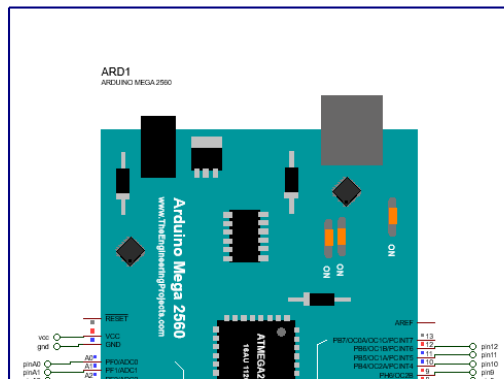
Ilustración 27. Sensor dht22 #2

Fuente: Elaboración propia, 2019

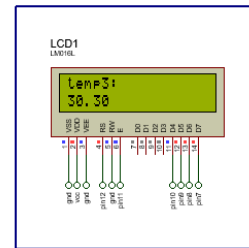
Sensores de Temperatura



Tarjeta Programable



Monitoreo LCD



Sensor de Movimiento

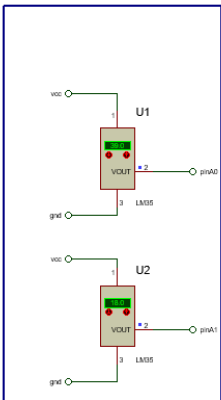


Pantalla LCD muestra el valor Temp3 =30.30°c

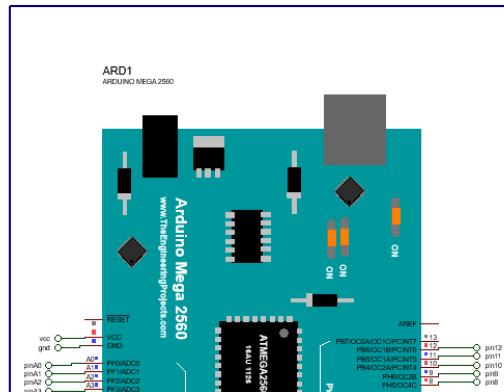
Ilustración 28. Sensor dht22 #3

Fuente: Elaboración propia, 2019

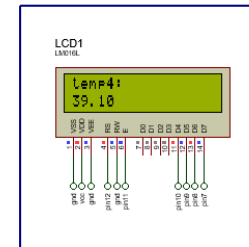
Sensores de Temperatura



Tarjeta Programable



Monitoreo LCD



Sensor de Movimiento

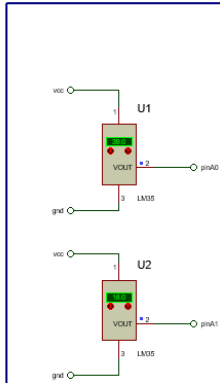


Pantalla LCD muestra el valor Temp4 =39.10°c

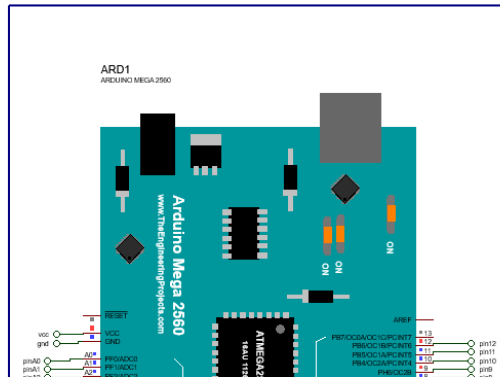
Ilustración 29. Sensor dht22 #4

Fuente: Elaboración propia, 2019

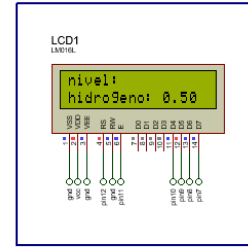
### Sensores de Temperatura



### Tarjeta Programable



### Monitoreo LCD



Pantalla LCD muestra el valor de hidrogeno = 0.50%

### Sensor de Movimiento



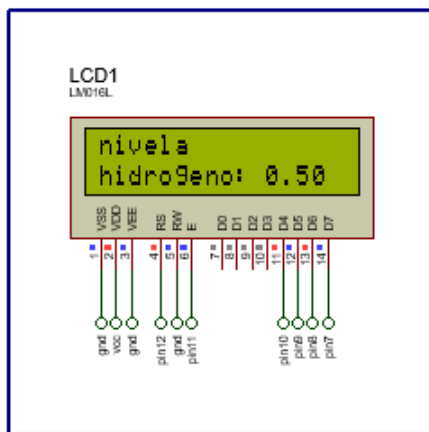
Ilustración 30. Sensor MQ-8

Fuente: Elaboración propia, 2019

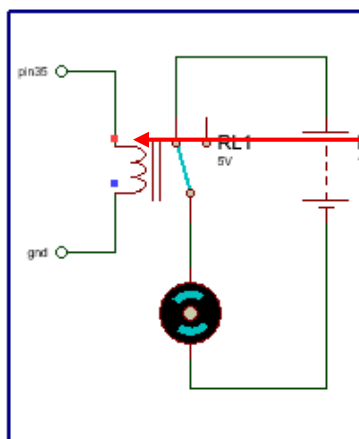
## Tarea 1 de Arduino:

Si el sensor MQ-8 registra el nivel de hidrogeno mayor a 0.5, se deberá prender la salida digital número 48 de la tarjeta Arduino, a continuación, la demostración ilustrada de la función de la tarea 1:

### Monitoreo LCD



### Control de Actuadores



Como se puede apreciar, en la pantalla LCD se logra ver que el nivel de hidrogeno esta elevado a 0.50 y la salida de la tarjeta fue accionada de manera correcta, y prendió el motor.

Ilustración 31. Función de encendido de tarea 1

Fuente: Elaboración propia, 2019

En la siguiente ilustración se observa que pasa cuando el nivel de hidrogeno está dentro de los límites establecidos

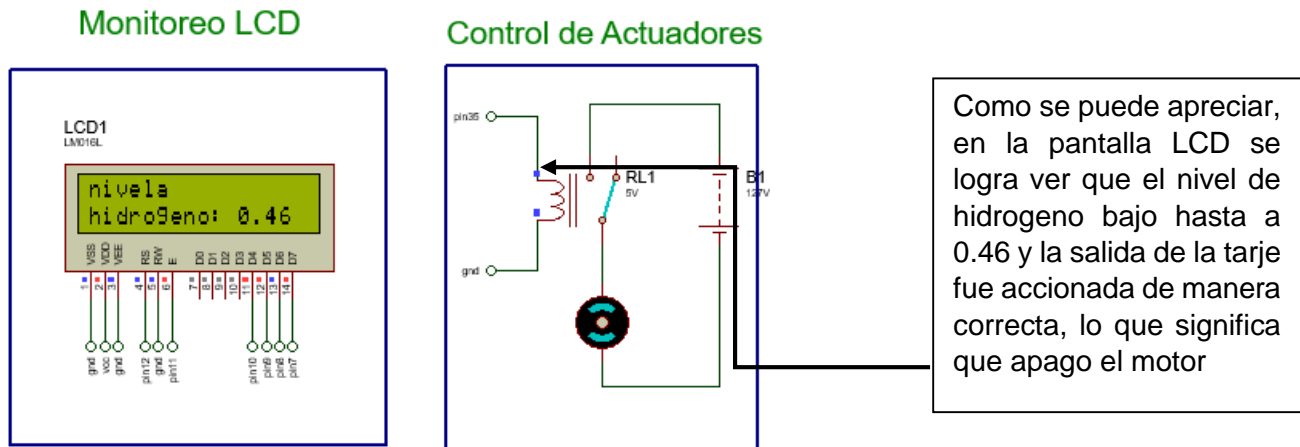


Ilustración 32. Función de apagado de tarea 1

Fuente: Elaboración propia, 2019

De manera automática la tarjeta Arduino acciona la salida dependiendo en qué situación se encuentre el nivel del sensor, en las dos imágenes se logra apreciar que la simulación representa el encendido o apagado de la salida con un punto de color, el punto vario su color dependiendo de lo indicado, color rojo si la salida fue encendida o de color azul si la salida fue apagada.

En cuanto a la función física, se llevó acabo con el accionamiento de un led, para probar si de verdad se accionada la salida de la forma que uno desea, y como fue previsto en la simulación, la tarea del sensor de hidrogeno se llevó acabo de manera correcta.

### Tarea 2 de Arduino:

Para la tarea 2 que se le programo a la tarjeta Arduino es la del accionamiento de una salida dependiendo de la situación en la que se encuentre el sensor de movimiento, esta tarea está diseñada para trabajar como un interruptor, esto para la activación de un segundo actuador, en el caso del sensor PIR, uno de los usos más cotidianos que se da es el encendido de lámparas de 127 V A.C, este caso se programara la salida como se desea, pero se le dio la funcion de accionar un buzzer de alarma, que sonara en cada momento que detecte movimiento el sensor PIR.

## Demostración ilustrada de la función de la tarjeta 2:

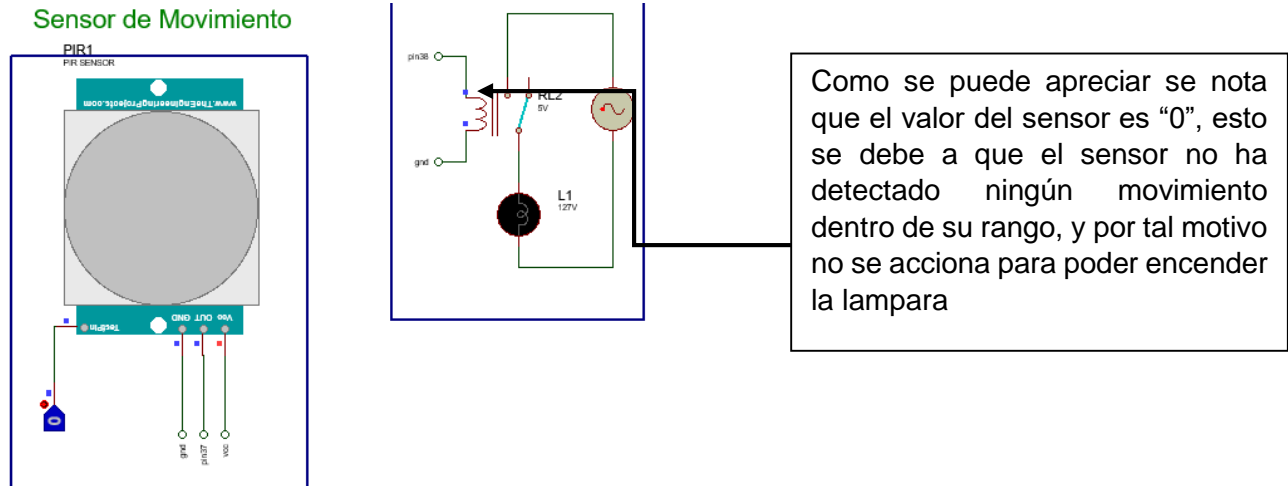


Ilustración 33. Función de apagado de tarea 2

Fuente: Elaboración propia, 2019

En la siguiente ilustración se ve el funcionamiento de la tarea 2 cuando el sensor PIR detecta un movimiento, en esta parte se demostrara con dos ilustraciones, por el motivo de que el valor del sensor solo tiene dos formas, la cual una es "0" y la otra es "1", y se menciona este dato por el hecho de que el sensor está configurado para mantener su valor de "1" durante 5 segundos después de detectar un movimiento, para que de esta manera se pueda mantener activo el actuador de manera autónoma, si la necesidad de que este una persona para apagarlo.

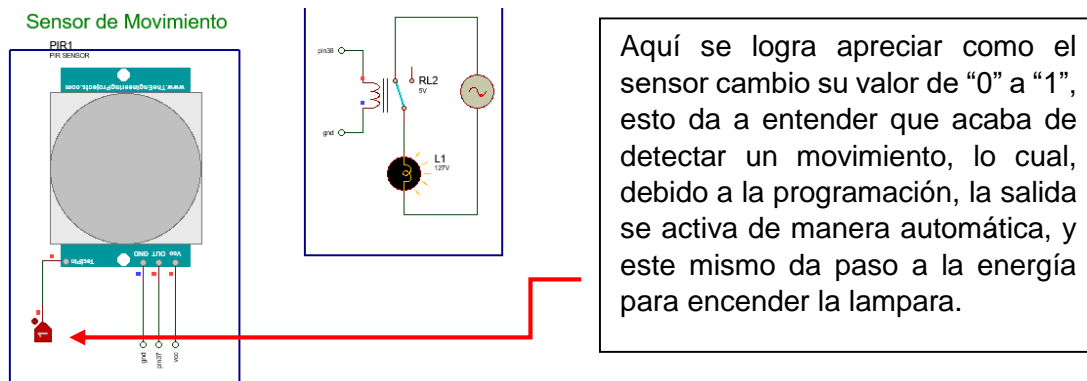


Ilustración 34. Función de tarea 2 con valor de sensor "1"

Fuente: Elaboración propia, 2019



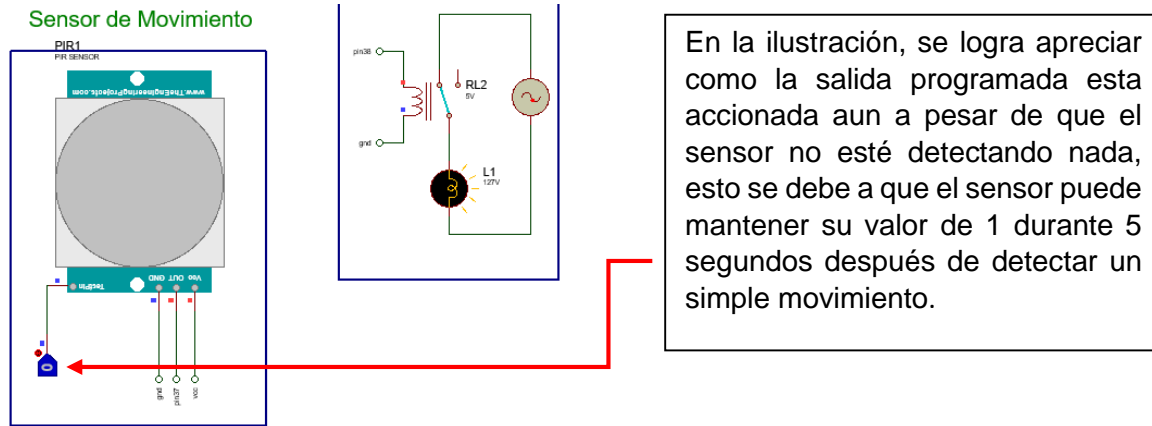


Ilustración 35. Función de tarea 2

Fuente: Elaboración propia, 2019

En este código se programó un tiempo de 5 segundos, pero solo es de prueba, porque realmente el tiempo en el código real tiene es más largo, esta función se puede modificar de manera digital modificando el código de programación o de manera física, ya que el sensor PIR tiene en su parte inferior dos perillas que una es de distancia y otra de tiempo, la segunda se puede aumentar o disminuir de manera manual para que el tiempo pueda mantener el valor del sensor.

Después de la programación de las dos tareas del código, se llevó a cabo una programación más clasificada como alarmas, esto para poder informar desde la LCD, si algo está sucediendo con el sistema, de igual forma, la parte de alarmas será implementada en la página de internet y vía protocolo de red SNMP.

## 4.5 Apartado de Alarmas

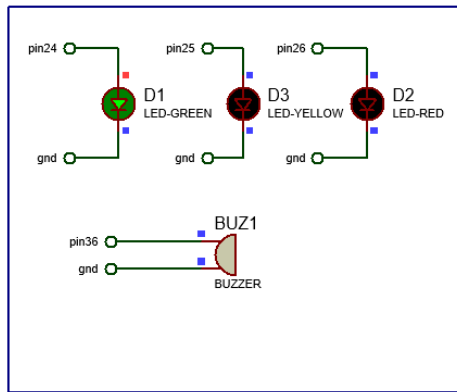
### 4.5.1 Parte 1

Las alarmas se deben de activar cada vez que los límites establecidos para los sensores son superados, el rango de los sensores de temperatura oscila entre 18°C como temperatura mínima, y 30°C como temperatura máxima, y el sensor de hidrogeno sus rangos son: nivel de hidrogeno normal es de 0% a 0.5% y el nivel de alerta para poder accionar el extractor es de 0.5% a 1.0%.

La parte de alarmas está dividida en 3 leds de colores, y una bocina pequeña (buzzer) para alertar de manera auditiva a las personas.

El funcionamiento de los tres leds es el siguiente:

## Alarmas



La función del led verde es estar encendido siempre que el sistema esté en funcionamiento, de esta manera cualquier persona sabrá que el sistema está trabajando de manera cotidiana.

Por lo tanto, dicho led no se debe de apagar mientras el sistema siga conectado, solamente se apagará si alguien llega a desconectar el sistema de su toma de corriente.

Ilustración 36. Función de led verde

Fuente: Elaboración propia, 2019

El led amarillo es el único que siempre se va a encender cada vez que el nivel del sensor de hidrogeno haya superado el rango establecido, de esta manera el sistema va a indicar que el extractor ha sido encendido y está trabajando de manera estable para poder expulsar el exceso de gas de hidrogeno que hay en la sala de baterías, de igual manera se apagará cuando el nivel regrese a su rango de estabilidad.

## Alarmas

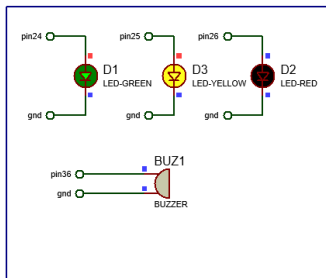


Ilustración 37. Demostración de alarma con extractor encendido

Fuente: Elaboración propia, 2019

## Alarmas

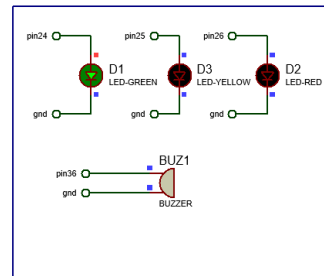


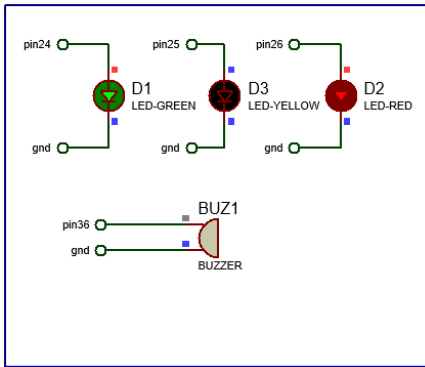
Ilustración 38. Demostración de alarma con extractor apagado

Fuente: Elaboración propia, 2019

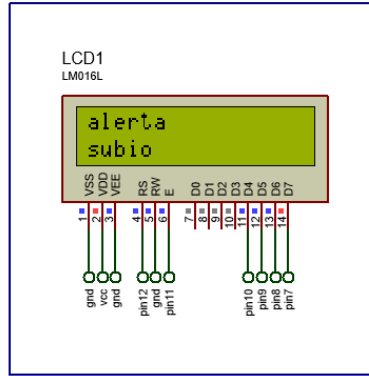
### 4.5.1 Parte 2

Dependiendo de los valores de los sensores de temperatura se debe de activar la alarma. el indicador que será activado es el led color rojo, a continuación, se mostraran las ilustraciones de dos sensores uno con valor arriba de los 40 °C y otro por debajo del valor de 18 °C, esto para poder demostrar cómo funciona esta última parte de la alarma.

## Alarmas



## Monitoreo LCD

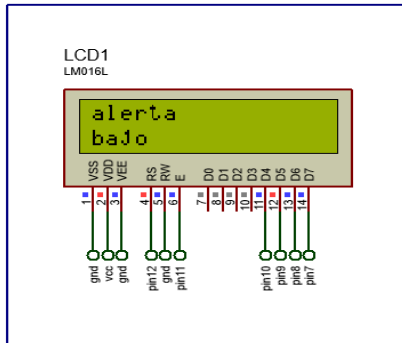


Como se puede apreciar en la ilustración, la tercera función de alarma trabaja de manera satisfactoria, de igual manera en la pantalla LCD se logra apreciar como su texto cambio, en lugar de mostrar el valor del sensor, nos muestra un mensaje de alerta con la palabra subió, que indica que el valor sobrepaso el límite.

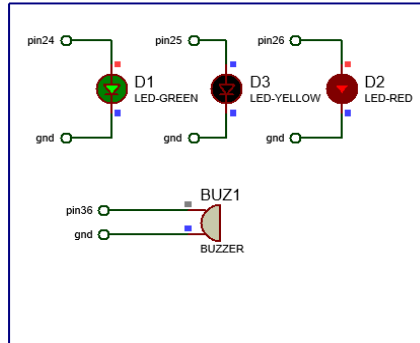
Ilustración 39. Sensor de temperatura 1 sobre el límite establecido

Fuente: Elaboración propia, 2019

## Monitoreo LCD



## Alarmas



Al igual que en la ilustración anterior, se puede apreciar que la función alarma trabaja de manera satisfactoria, y en la LCD se puede apreciar que el texto cambio a "alerta bajo" indicando que el nivel del sensor esta debajo del limite

Ilustración 40. Sensor de temperatura 2 debajo del límite establecido

Fuente: Elaboración propia, 2019

Dicho todo lo anterior, con esto se acaba la explicacion detallada del sistema de manera simulada, a continuacion seguira la demostracion del sistema motado fisicamente, dentro de esta explicacion se adentrara la explicacion de la pagina web que se desarrollo para **"el monitoreo del sistema de manera remota desde cualquier parte de la republica mexicana que tenga acceso a la intranet de CFE, mientras cumpla con ciertas restricciones establecida por dicha empresa."**

# Capítulo 5 Desarrollo del proyecto

## 5.1 Etapa 1 Sensores

### Primeras pruebas:

Como primeras pruebas después de quitar la tarjeta GMS, se procedió a cargar en al Arduino la programación de prueba que se utilizó en la simulación, esto para poder observar cómo operan los sensores y obtener los primeros datos de censado para el nuevo sistema.



Ilustración 42. Mensaje de presentación del sistema

Fuente: Elaboración propia, 2019



Ilustración 41. sensor DHT22 1 en operación

Fuente: Elaboración propia, 2019



Ilustración 44. sensor DHT22 2 en operación

Fuente: Elaboración propia, 2019



Ilustración 43. sensor DHT22 3 en operación

Fuente: Elaboración, propia



Ilustración 46. sensor DHT22 4 en operación

Fuente: Elaboración propia, 2019



Ilustración 45. sensor MQ-8 en operación

Fuente: Elaboración propia 2019

Tabla 1 Pruebas y datos del primer censado del nuevo sistema

<b>Variables</b>	<b>Registro Actual</b>	<b>Limites</b>	<b>¿Muestra datos en pantalla?</b>
<b>Temperatura 1</b>	26°C	18°C - 30°C	SI
<b>Temperatura 2</b>	20°C	18°C - 30°C	SI
<b>Temperatura 3</b>	21°C	18°C - 30°C	SI
<b>Temperatura 4</b>	22°C	18°C - 30°C	SI
<b>Humedad 1</b>	26%	20% - 60%	SI
<b>Humedad 2</b>	42%	20% - 60%	SI
<b>Humedad 3</b>	43%	20% - 60%	SI
<b>Humedad 4</b>	32%	20% - 60%	SI
<b>Nivel de Hidrógeno</b>	0.23%	0.0% - 1.0%	SI
<b>Movimiento</b>	ON/OFF	3m – 7m	NO

La obtención de los nuevos registros permite apreciar el funcionamiento en tiempo real el comportamiento que nos brinda cada sensor junto con la tarjeta Arduino, al igual que la activación de las alertas que se imprimen de forma correcta en la pantalla y la activación del buzzer.

Con esto se procedió a realizar arreglos menores como los límites y funcionamiento de los sensores, tiempo de lectura y activación de salida de actuadores.

## 5.2 Etapa 2 Programación de tarjeta ethernet

Una vez realizadas las primeras pruebas de operación de los sensores y adquirir la tarjeta Shield Ethernet W5100, se realizaron las siguientes pruebas con la tarjeta ya instalada fuera de su caja.

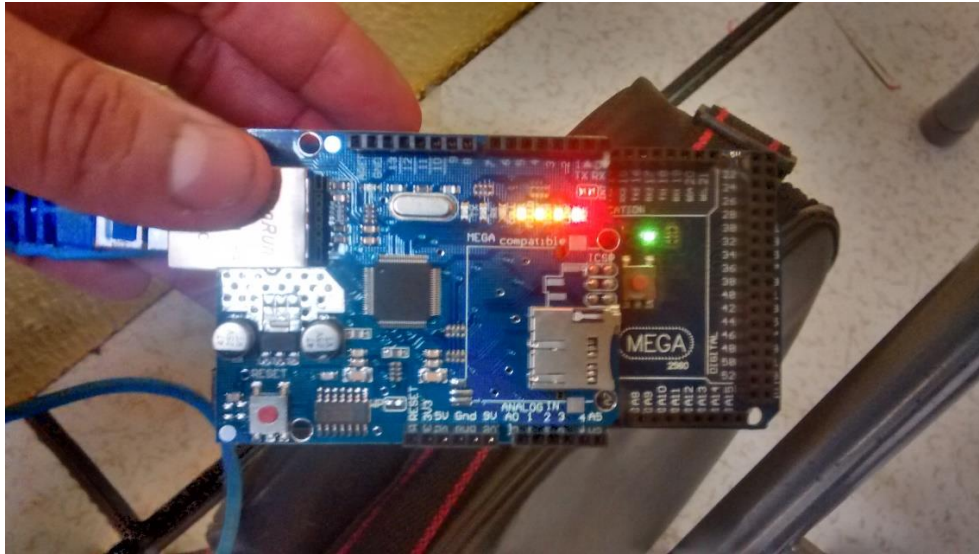


Ilustración 47. instalación y pruebas de alimentación de Shield Ethernet

Fuente: Imagen propia, 2019

Direcciones introducidas para obtener acceso a la red intranet de CFE:

- Dirección IP de sistema de alarmas: 10.27.3.31
- Mascara de subred: 255.255.255.192
- Dirección Gateway: 10.27.3.62
- Dirección DNS: 10.27.2.134
- Numero de puerto de servidor ethernet: 80
- Numero de puerto para protocolo SNMP: 161

Las primeras pruebas que se realizaron con la tarjeta ethernet, fue checar la comunicación vía web, utilizando los pines analógicos del propio Arduino para poder observar sus valores.

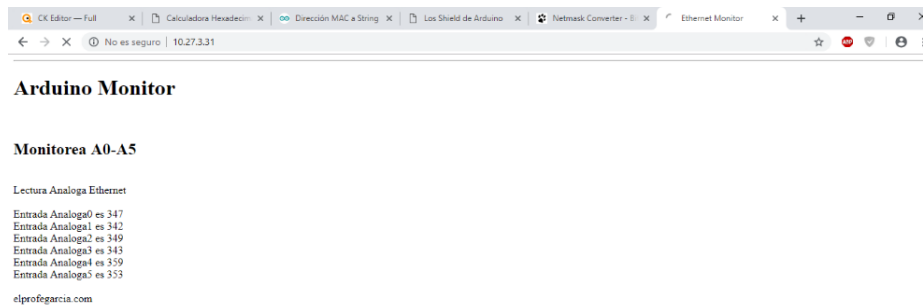


Ilustración 48. Primeras comunicaciones de Arduino y Shield Ethernet

Fuente: Elaboración propia, 2019

## Arduino Monitor

### Monitorea A0-A5

Lectura Analoga Ethernet

Entrada Analoga0 es 347  
Entrada Analoga1 es 342  
Entrada Analoga2 es 349  
Entrada Analoga3 es 343  
Entrada Analoga4 es 359  
Entrada Analoga5 es 353

Ilustración 49. valores de pines analógicos de Arduino en Intranet

Fuente: Elaboración propia, 2019

Comprobando que la tarjeta ethernet funciona correctamente, se llevo a cabo la instalacion de dicha tarjeta ya dentro de la caja de presentacion, para posteriormente realizar las primeras pruebas de impresion de valores de los sensores en la intranet.



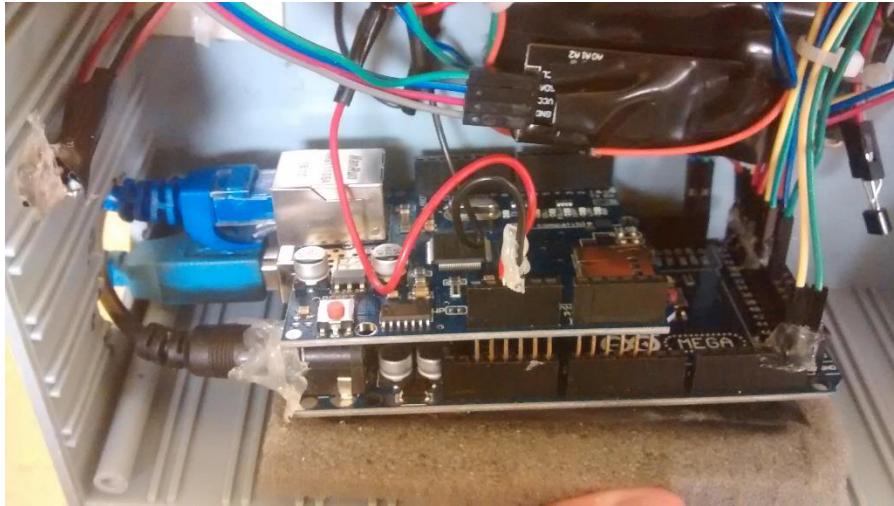


Ilustración 50. instalación de tarjeta Ethernet en el sistema

Fuente: imagen propia, 2019

Con los sensores ya conectados la respuesta de la tarjeta fue la siguiente:

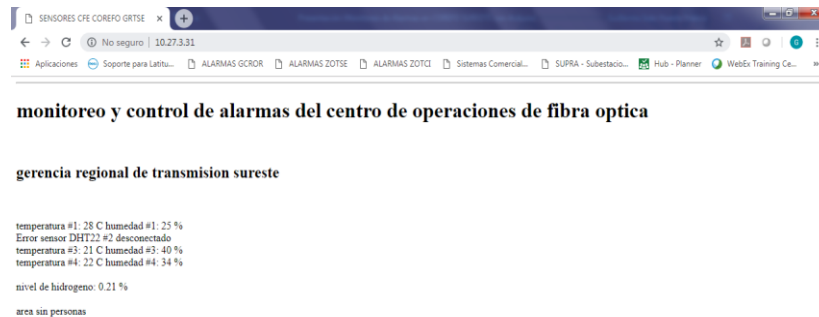


Ilustración 51. Primeras pruebas de web (página en proceso de diseño)

Fuente: Elaboración propia, 2019

```
temperatura #1: 28 C humedad #1: 25 %
Error sensor DHT22 #2 desconectado
temperatura #3: 21 C humedad #3: 40 %
temperatura #4: 22 C humedad #4: 34 %
```

nivel de hidrogeno: 0.21 %

area sin personas

Ilustración 52. Valor de los sensores

Fuente: Elaboración propia, 2019

## 5.2.1 Desarrollo página web

A continuación, se presenta el desarrollo del diseño de que se llevó la página web:

Fecha: 12 / 3 / 2019  
Hora: 12 : 48 : 31

### Monitoreo y Control de Alarmas del Centro de Operaciones de Fibra Optica

Gerencia Regional de Transmision Sureste

Tabla de Valores y Limites de Operacion				
Ubicacion	Sensor	Valor actual	Mínimo	Maximo
Sala de Baterias	Temperatura #1:	28 ° C	18	30
	Humedad #1:	27 %	25%	60%
Centro	Temperatura #2:	20 ° C	18	30
	Humedad #2:	43 %	25%	60%
alado clima LG	Temperatura #3:	22 ° C	18	30
	Humedad #3:	44 %	25%	60%
Acceso	Temperatura #4:	22 ° C	18	30
	Humedad #4:	37 %	25%	60%
Sala de Baterias	Nivel de Hidrogeno:	0.28 %	0.0%	1.0%
Acceso	Movimiento	area sin personas		

Departamento de Conexion de Servicios  
12:48 p. m.  
12/03/2019

Ilustración 53. Diseño página web

Fuente: Elaboración propia, 2019

En el desarrollo de la página se decidió hacer pruebas con controles manuales

Fecha: 22 / 3 / 2019  
Hora: 13 : 19 : 2

### Monitoreo y Control de Alarmas del Centro de Operaciones de Fibra Optica

Gerencia Regional de Transmision Sureste

Tabla de Valores y Limites de Operacion				
Ubicacion	Sensor	Valor actual	Mínimo	Maximo
Sala de Baterias	Temperatura #1:	27 ° C	18	30
	Humedad #1:	32 %	25%	60%
Centro	Temperatura #2:	21 ° C	18	30
	Humedad #2:	43 %	25%	60%
alado clima LG	Temperatura #3:	21 ° C	18	30
	Humedad #3:	48 %	25%	60%
Acceso	Temperatura #4:	23 ° C	18	30
	Humedad #4:	39 %	25%	60%
Sala de Baterias	Nivel de Hidrogeno:	0.21 %	0.0%	1.0%
Acceso	Movimiento	Persona Detectada		

Control Manual de Extractor

ON OFF

Estado del Extractor= OFF

Control de Encendido del Sensor de Movimiento

ON OFF

Estado del Sensor de Movimiento= OFF

Departamento de Conexion de Servicios  
01:19 p. m.  
22/03/2019

Ilustración 54. Diseño de página web

Fuente: Elaboración propia, 2019

Gerencia Regional de Transmisión Sureste  
Subgerencia de comunicaciones

### Monitoreo y control de alarmas en Sala de Equipos Hotel Tuxtla

Descripción (Temperatura)	Actual (°C)	Lim. Inferior (°C)	Lim. Superior (°C)
Sala de Baterías	28	<18	>30
Centro Sala de Equipos	20	<18	>30
Lateral Sala de Equipos	21	<18	>30
Acceso Sala de Equipos	23	<18	>30

Descripción (Humedad)	Actual (%)	Lim. Inferior (%)	Lim. Superior (%)
Sala de Baterías	25	<20	>60
Centro Sala de Equipos	38	<20	>60
Lateral Sala de Equipos	40	<20	>60
Acceso Sala de Equipos	32	<20	>60

Hidrógeno y Movimiento	Actual	Lim. Inferior	Lim. Superior
Nivel de hidrógeno en Sala de Baterías	0.29	<0.0	>1.0
Sensor de movimiento	Desactivado		

Última actualización: Fecha: 13 / 5 / 2019 Hora: 9 : 43 :46

Departamento de Conexión de Servicios GRITSE e Instituto Tecnológico de Tuxtla Gutiérrez

Ilustración 55. Diseño final de página web

Fuente: Elaboración propia, 2019

Tabla 2 Pruebas y datos del segundo censado vía intranet

Variables	Registro Actual	Limites	¿Muestra datos en página web?	¿Muestra datos en la pantalla?	¿Alerta web?
Temperatura 1	28°C	18°C - 30°C	SI	SI	SI
Temperatura 2	20°C	18°C - 30°C	SI	SI	SI
Temperatura 3	21°C	18°C - 30°C	SI	SI	SI
Temperatura 4	23°C	18°C - 30°C	SI	SI	SI
Humedad 1	25%	20% - 60%	SI	SI	SI
Humedad 2	38%	20% - 60%	SI	SI	SI
Humedad 3	40%	20% - 60%	SI	SI	SI
Humedad 4	32%	20% - 60%	SI	SI	SI
Nivel de Hidrógeno	0.29%	0.0% - 1.0%	SI	SI	SI
Movimiento	ON/OFF	3m – 7m	SI	NO	SI

## 5.3 Etapa 3 Protocolo de red SNMP

En la actualidad, el GRTSE utiliza una plataforma llamada ZABBIX para el monitoreo de alarmas por Ethernet de sus equipos de red por medio del protocolo de red SNMP, para un mejor manejo de información de cada equipo.

Con esto se llevó a cabo la activación del protocolo SNMP en la tarjeta Arduino, introduciendo una OID para cada sensor:

Tabla 3 OID'S de los sensores DHT22

Sensor	OID temperatura	OID humedad
DHT22 #1	1.3.6.1.4.1.36582.5.1.0	1.3.6.1.4.1.36582.5.1.1
DHT22 #2	1.3.6.1.4.1.36582.5.1.2	1.3.6.1.4.1.36582.5.1.3
DHT22 #3	1.3.6.1.4.1.36582.5.1.4	1.3.6.1.4.1.36582.5.1.5
DHT22 #4	1.3.6.1.4.1.36582.5.1.6	1.3.6.1.4.1.36582.5.1.7

Tabla 4 OID'S de los sensores MQ-8 y PIR

Sensor	OID
Hidrogeno	1.3.6.1.4.1.36582.5.1.8
Movimiento	1.3.6.1.4.1.36582.5.1.9

Con la activación del protocolo SNMP en el Arduino, se pudo llevar a cabo el reconocimiento de los valores de cada sensor por medio de cada OID, utilizando el software llamado **"iReasoning MIB Browser"**, dicho software sirve para comunicar un agente SNMP con un administrador SNMP.

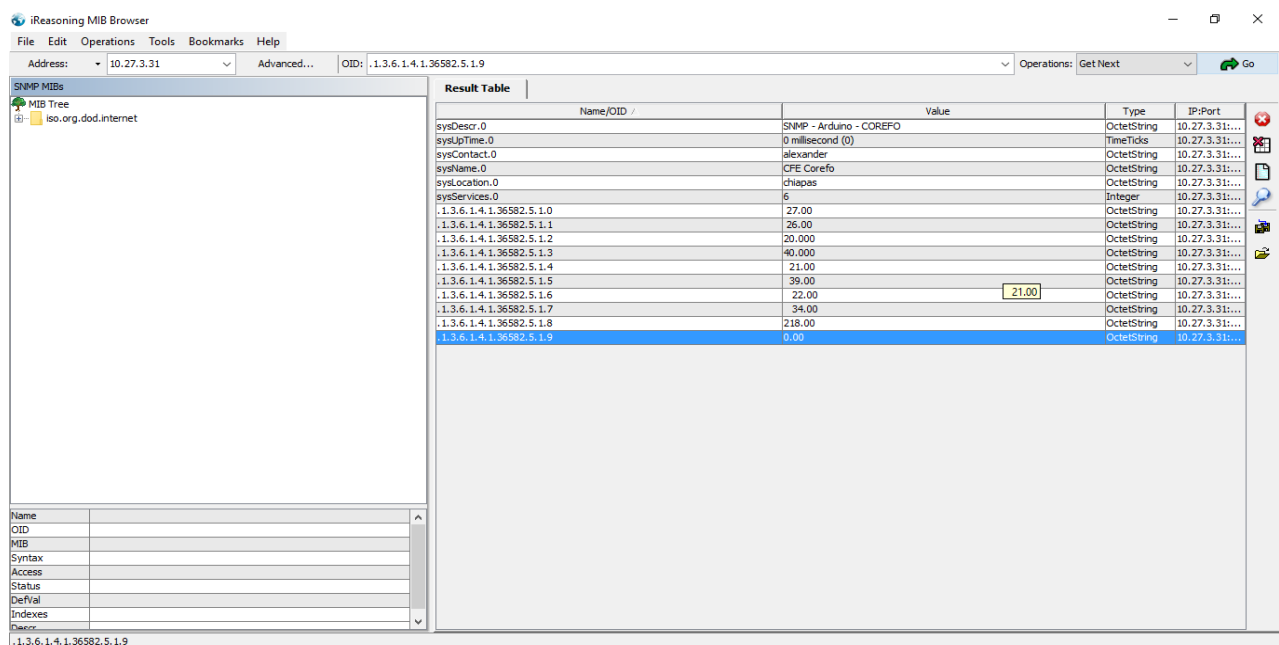


Ilustración 56. Captura de MIB Browser

Fuente: Elaboración propia, 2019

Result Table			
Name/OID /	Value	Type	IP:Port
sysDescr.0	SNMP - Arduino - COREFO	OctetString	10.27.3.31:...
sysUpTime.0	0 millisecond (0)	TimeTicks	10.27.3.31:...
sysContact.0	alexander	OctetString	10.27.3.31:...
sysName.0	CFE Corefo	OctetString	10.27.3.31:...
sysLocation.0	chiapas	OctetString	10.27.3.31:...
sysServices.0	6	Integer	10.27.3.31:...

Ilustración 57. Datos SNMP Arduino

Fuente: Elaboración propia, 2019

.1.3.6.1.4.1.36582.5.1.0	27.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.1	26.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.2	20.000	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.3	40.000	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.4	21.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.5	39.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.6	22.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.7	34.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.8	218.00	OctetString	10.27.3.31:...
.1.3.6.1.4.1.36582.5.1.9	0.00	OctetString	10.27.3.31:...

Ilustración 58. OID'S de los sensores y valores

Fuente: Elaboración propia, 2019

Con las OID'S activas y puestas en operación, el monitoreo se logra realizar de manera correcta concentrando toda la información en el puerto de red 161, puerto que es utilizado por el protocolo SNMP y por la plataforma ZABBIX para él envío de notificaciones de alarmas.

El sistema de alarmas consta de dos partes, la primera parte es desarrollada por el Arduino que se encarga de enviar los valores para las alarmas por tres formas diferentes, la primera es por medio de su pantalla LCD, la torreta de colores y el buzzer, la segunda es por su página web por el medio de intranet y la tercera forma es por medio del protocolo SNMP a través del puerto 161.

### 5.3.1 Pruebas de plataforma ZABBIX

La segunda parte del sistema de alarmas son el aviso de notificaciones, para realizar esta labor, es la plataforma ZABBIX que se encarga de recolectar toda la información que el Arduino manda a través del puerto 161 por medio de peticiones de tipo SNMP

Una vez que la información ya se encuentra registrada en el ZABBIX, los operadores de CFE adquieren un mayor control en cuanto al manejo de la información, debido a que gracias a la plataforma, se crea una base de datos para el registro del comportamiento de cada sensor del sistema, graficando cada sensor para una mayor visualización de los

niveles de las variables, y lo más importante, el envío de notificaciones ante cualquier variación por el medio de mensajería celular llamada “**TELEGRAM**” sin costo alguno.

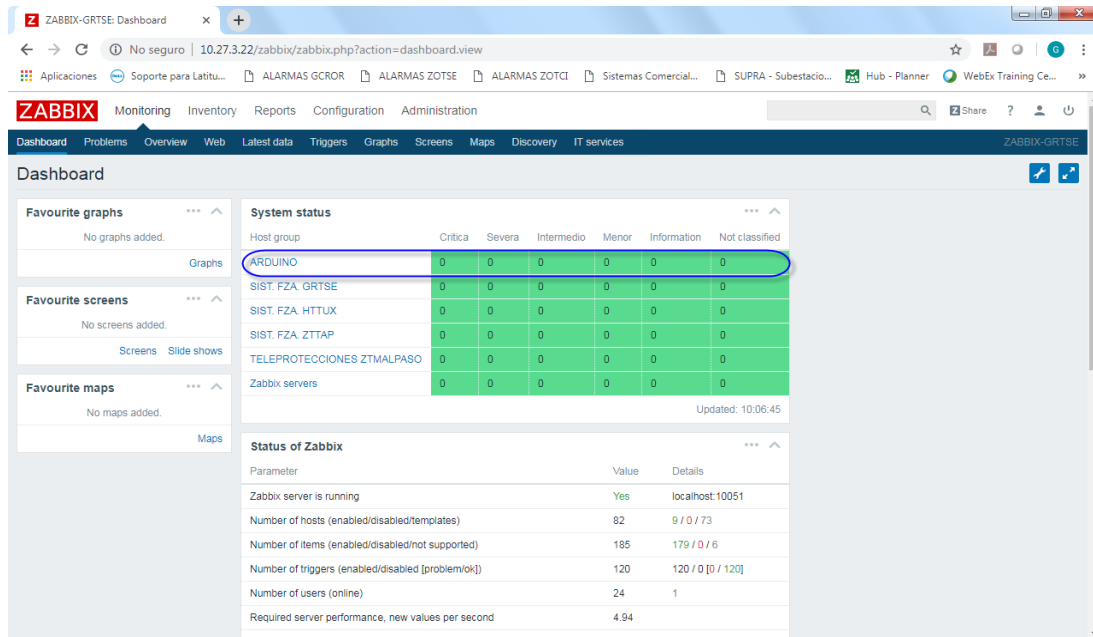


Ilustración 59. Estatus de sistemas registrados en ZABBIX

Fuente: Elaboración propia, 2019

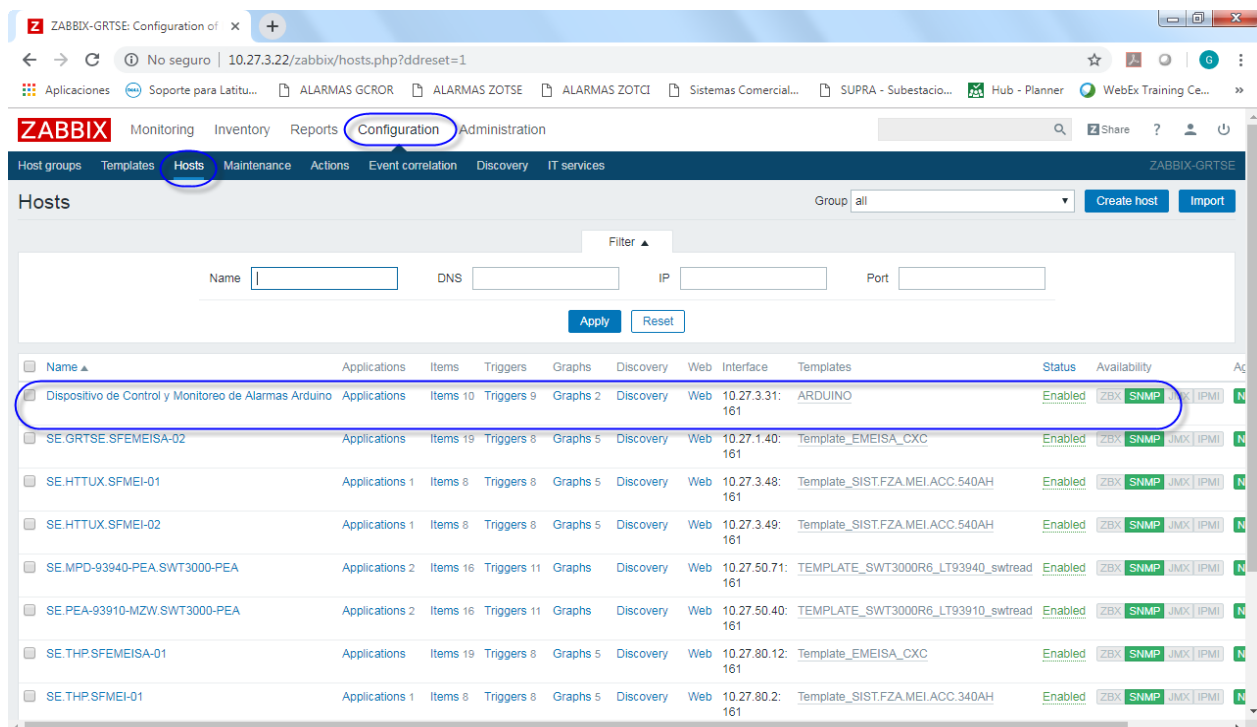


Ilustración 60. Configuración del sistema de monitoreo y control de alarmas

Fuente: Elaboración propia, 2019

Dentro de la configuración del ZABBIX el sistema se dio de alta con las siguientes características:

- Nombre: dispositivo de Control y Monitoreo de Alarmas Arduino
- Dirección IP: 10.27.3.31
- Numero de variables (ítems): 10

The screenshot shows the Zabbix web interface for configuring items. The browser address bar indicates the URL: 10.27.3.22/zabbix/items.php?filter\_set=1&hostid=10266. The Zabbix navigation menu is visible at the top, and the 'Items' section is active. A table lists 10 items, all of which are enabled and use SNMPv1 agents. The items are categorized by their keys: hydrogen, humidity, and temperature sensors. The table columns are: Name, Triggers, Key, Interval, History, Trends, Type, Applications, Status, and Info. The items are: ARDUINO: Hidrógeno Sala de Baterías, ARDUINO: Humedad Sala de Baterías, ARDUINO: Humedad Sala de Equipos 1, ARDUINO: Humedad Sala de Equipos 2, ARDUINO: Humedad Sala de Equipos 3, ARDUINO: Sensor de movimiento, ARDUINO: Temp. Sala de Baterías, ARDUINO: Temp. Sala de Equipos 1, ARDUINO: Temp. Sala de Equipos 2, and ARDUINO: Temp. Sala de Equipos 3. All items have a 30-second interval, 90-day history, and 365-day trends. The status for all items is 'Enabled'.

Name	Triggers	Key	Interval	History	Trends	Type	Applications	Status	Info
ARDUINO: Hidrógeno Sala de Baterías	Triggers 1	syshidrogeno	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Humedad Sala de Baterías	Triggers 1	syshumidity_1	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Humedad Sala de Equipos 1	Triggers 1	syshumidity_2	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Humedad Sala de Equipos 2	Triggers 1	syshumidity_3	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Humedad Sala de Equipos 3	Triggers 1	syshumidity_4	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Sensor de movimiento		sysmovimiento	30s	90d		SNMPv1 agent		Enabled	
ARDUINO: Temp. Sala de Baterías	Triggers 1	systemperature_1	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Temp. Sala de Equipos 1	Triggers 1	systemperature_2	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Temp. Sala de Equipos 2	Triggers 1	systemperature_3	30s	90d	365d	SNMPv1 agent		Enabled	
ARDUINO: Temp. Sala de Equipos 3	Triggers 1	systemperature_4	30s	90d	365d	SNMPv1 agent		Enabled	

Ilustración 61. Configuración de variables en la plataforma

Fuente: Elaboración propia, 2019

La plataforma agrupa las variables en la opción de "ítems", dentro de esta opción se configuro la plataforma de la siguiente manera:

- A 9 variables se activó su triggers, los triggers son las notificaciones.
- Se le dio un intervalo de llamado de 30 segundos a cada ítem para actualización de datos.
- El monitoreo se hace los 365 días del año.
- El registro de cada sensor será de manera constante desde el primer día de activación.
- El tipo de SNMP a leer es SNMP v1.
- Graficas de valores



La estructura de la base para la agrupación de datos y gráficas se llevó a cabo en dos partes para mayor comprensión de los operadores.

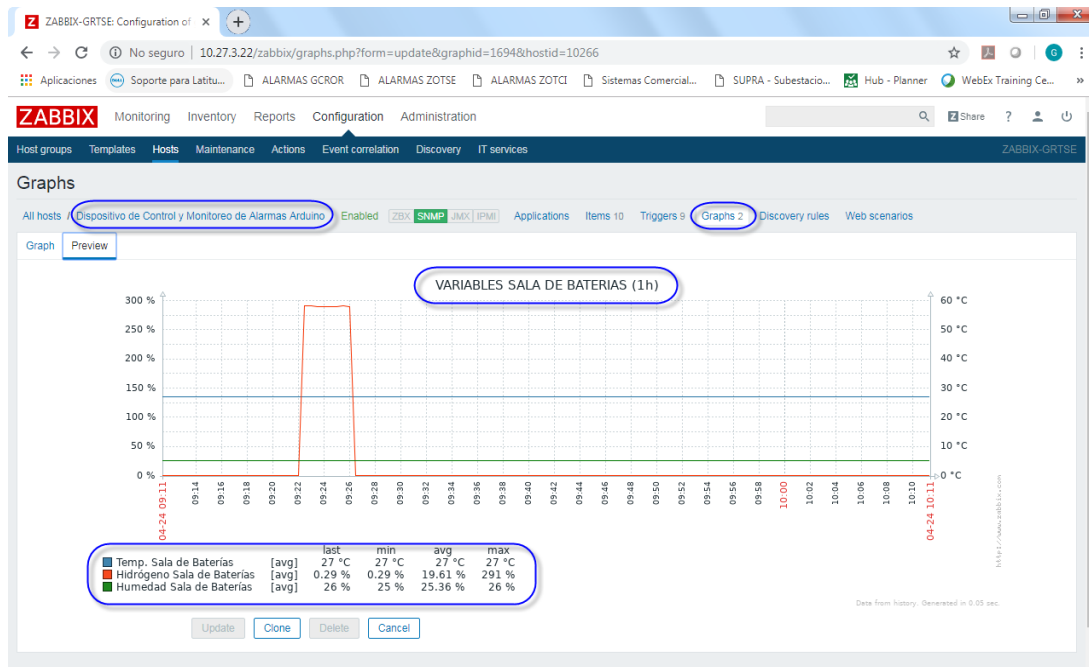


Ilustración 62. Grafica sala de baterías

Fuente: Elaboración propia, 2019

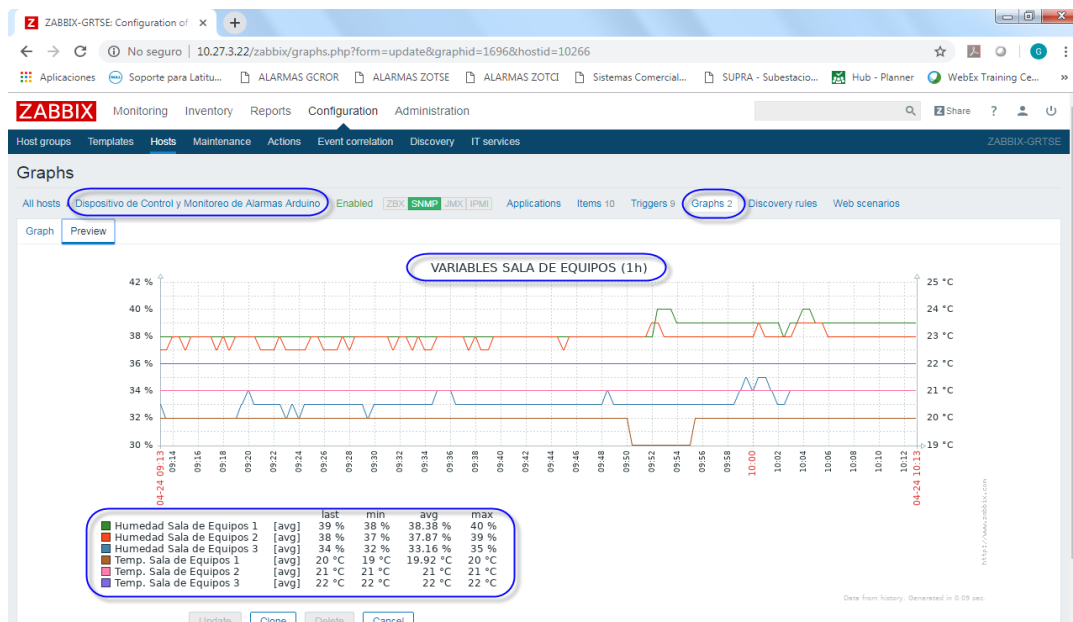


Ilustración 63. Grafica de sala de equipos

Fuente: Elaboración propia, 2019

### 5.3.1 Notificación Telegram

The screenshot shows the ZABBIX configuration interface for triggers. A table lists 9 triggers with columns for Severity, Name, Expression, and Status. Red arrows point to values in the Expression column, labeled 'Límite máximo' and 'Límite mínimo'.

Severity	Name	Expression	Status
Intermedio	ARDUINO: ALARMA DE HUMEDAD SALA DE BATERIAS	{Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_1.last()}<25 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_1.last()}>60	Enabled
Intermedio	ARDUINO: ALARMA DE HUMEDAD SALA DE EQUIPOS 1	{Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_2.last()}<25 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_2.last()}>60	Enabled
Intermedio	ARDUINO: ALARMA DE HUMEDAD SALA DE EQUIPOS 2	{Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_3.last()}<25 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_3.last()}>60	Enabled
Intermedio	ARDUINO: ALARMA DE HUMEDAD SALA DE EQUIPOS 3	{Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_4.last()}<25 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:syshumidity_4.last()}>60	Enabled
Intermedio	ARDUINO: ALARMA DE TEMPERATURA SALA DE BATERIAS	{Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_1.last()}<18 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_1.last()}>30	Enabled
Intermedio	ARDUINO: ALARMA DE TEMPERATURA SALA DE EQUIPOS 1	{Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_2.last()}<18 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_2.last()}>30	Enabled
Intermedio	ARDUINO: ALARMA DE TEMPERATURA SALA DE EQUIPOS 2	{Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_3.last()}<18 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_3.last()}>30	Enabled
Intermedio	ARDUINO: ALARMA DE TEMPERATURA SALA DE EQUIPOS 3	{Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_4.last()}<18 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:systemperature_4.last()}>30	Enabled
Intermedio	ARDUINO: ALARMA NIVEL HIDROGENO	{Dispositivo de Control y Monitoreo de Alarmas Arduino:syshidrogeno.last()}<0 or {Dispositivo de Control y Monitoreo de Alarmas Arduino:syshidrogeno.last()}>1	Enabled

Ilustración 64. Configuración de triggers

Fuente: Elaboración propia, 2019

De esta manera las notificaciones son enviadas cuando un sensor sobrepasa sus límites establecidos, el tipo de comunicación que se escogió para él envío de notificaciones, es la mensajería celular de Telegram, pero con la mejoría que no tendrá ningún tipo de costo gracias a la red de intranet, los mensajes de alertas lo reciben los encargados en turno.

## Pruebas de notificaciones a Telegram



Ilustración 67. Aviso sensor de hidrogeno

Fuente: Elaboración propia, 2019



Ilustración 66. Aviso sensor de temperatura y humedad 2

Fuente: Elaboración propia, 2019



Ilustración 65. Estructura del mensaje en la notificación

Fuente: Elaboración propia, 2019

NOTA: las notificaciones solo son para cierto ingenieros u operadores del COREFO, se limita a solo un número pocas personas, por seguridad.

## 5.4 Etapa 4 Programación de control de actuadores

En el apartado de control, como se mencionó anteriormente en la simulación, se programaron las tareas correspondientes a el sensor de hidrogeno y el sensor de movimiento.

### 5.4.1 Sensor de hidrogeno

Las funciones del sensor de hidrogeno son el monitoreo y control de nivel de hidrogeno dentro de la sala de baterías, los niveles son controlados con un extractor que es operado de dos maneras, manera manual mediante el accionamiento de un interruptor de pared, y la segunda manera es por el accionamiento automático que es realizada por la tecnología Arduino, dependiendo de los niveles que este registrando el sensor, sin importa si la sala se encuentra vacía.

El extractor se encarga de succionar y expulsar el hidrogeno excedente que se encuentra en el aire de la sala de baterías.

**Diagrama de conexión de Sensor de hidrogeno:**

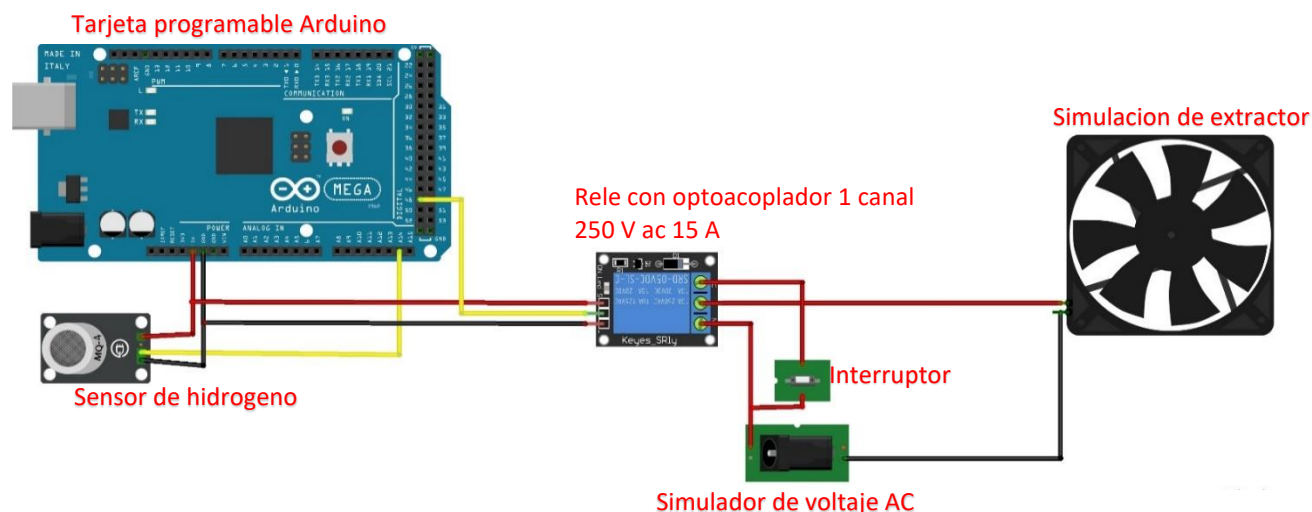


Ilustración 68. Diagrama de conexión de tarea 1

Fuente: Elaboración propia, 2019

Funciones de operación son las mismas que se explican en el apartado de simulaciones, con sus rasgos ya reestablecidos según la norma ISO 10156, que son los siguientes:

- Si el nivel de hidrogeno es mayor a 0.40 %, Arduino activa el extractor y manda la alerta vía SNMP e intranet para proceder a mandar la notificación por Telegram a los encargados de respectiva área.



Ilustración 69. Notificación de alarma de hidrogeno

Fuente: Elaboración propia, 2019

SENSORES CFE COREFO GRTSE x +

No es seguro | 10.27.3.31

Gerencia Regional de Transmision Sureste  
Subgerencia de comunicaciones

**Monitoreo y control de alarmas en Sala de Equipos Hotel Tuxtla**

Descripción (Temperatura)	Actual (°C)	Lim. Inferior (°C)	Lim. Superior (°C)
Sala de Baterías	28	<18	>30
Centro Sala de Equipos	20	<18	>30
Lateral Sala de Equipos	21	<18	>30
Acceso Sala de Equipos	22	<18	>30

Descripción (Humedad)	Actual (%)	Lim. Inferior (%)	Lim.Superior (%)
Sala de Baterías	25	<25	>60
Centro Sala de Equipos	39	<25	>60
Lateral Sala de Equipos	41	<25	>60
Acceso Sala de Equipos	34	<25	>60

Hidrogeno y Movimiento	Actual	Lim. Inferior	Lim. Superior
Nivel de hidrogeno en Sala de Baterías	0.62	<0.0	>1.0
Sensor de movimiento	Desactivado		

Última actualización: Fecha: 26 / 0 4 / 2019 Hora: 8 : 49 :45

Departamento de Conexión de Servicios GRTSE e Instituto Tecnológico de Tuxtla Gutierrez

Escribe aquí para buscar

08:50 a. m. 26/04/2019

Ilustración 70. Alerta de nivel alto de hidrogeno vía web

Fuente: Elaboración propia, 2019

## 5.4.2 Sensor de movimiento

Las funciones que se establecieron al sensor de movimiento son el monitoreo y control de una bocina de para alerta a los operadores, como funcion especial debido a que en la sala de equipos laboran seguidamente entre semana, se reprogramo la funcionalidad del sensor de movimiento a que solamente se mantenga activado los fines de semana.

Rangos de operación del sensor de movimiento:

- Activación: viernes a las 4:00 pm.
- Desactivación: lunes a las 07:30 am.
- Activar la bocina cada vez que se detecte movimiento.
- Días de operación: viernes a domingo

La alarma será para alertar de manera auditiva a los operadores y la notificación será de manera visual por la página web y la notificación por Telegram.



Ilustración 72. Notificación Telegram de movimiento en el COREFO

Fuente: Elaboración propia, 2019

**Gerencia Regional de Transmision Sureste**  
**Subgerencia de comunicaciones**  
**Monitoreo y control de alarmas en Sala de Equipos Hotel Tuxtla**

Descripción (Temperatura)	Actual (°C)	Lim. Inferior (°C)	Lim. Superior (°C)
Sala de Baterías	27	<18	>30
Centro Sala de Equipos	20	<18	>30
Lateral Sala de Equipos	21	<18	>30
Acceso Sala de Equipos	22	<18	>30

Descripción (Humedad)	Actual (%)	Lim. Inferior (%)	Lim.Superior (%)
Sala de Baterías	26	<25	>60
Centro Sala de Equipos	40	<25	>60
Lateral Sala de Equipos	39	<25	>60
Acceso Sala de Equipos	34	<25	>60

Hidrogeno y Movimiento	Actual	Lim. Inferior	Lim. Superior
Nivel de hidrogeno en Sala de Baterías	0.17%	<0.0	>1.0
Sensor de movimiento	Activado		

Última actualización: Fecha: 24 / 04 / 2019 Hora: 10 : 3 : 29  
 Departamento de Conexión de Servicios GRTSE e Instituto Tecnológico de Tuxtla Gutierrez

Ilustración 71. Notificación vía web de movimiento en el COREFO

Fuente: Elaboración propia, 2019



## 5.5 Etapa 5 Pruebas y resultados obtenidos

### Monitoreo y control de alarmas del centro de operación de fibra óptica de la gerencia regional de CFE transmisión sureste (completo)



**Monitoreo y control de alarmas en Sala de Equipos Hotel Tuxtla**

Descripción (Temperatura)	Actual (°C)	Lim. Inferior (°C)	Lim. Superior (°C)
Sala de Baterías	21	<18	>30
Centro Sala de Equipos	21	<18	>30
Lateral Sala de Equipos	21	<18	>30
Acceso Sala de Equipos	22	<18	>30

Descripción (Humedad)	Actual (%)	Lim. Inferior (%)	Lim. Superior (%)
Sala de Baterías	98	<20	>60
Centro Sala de Equipos	98	<20	>60
Lateral Sala de Equipos	98	<20	>60
Acceso Sala de Equipos	98	<20	>60

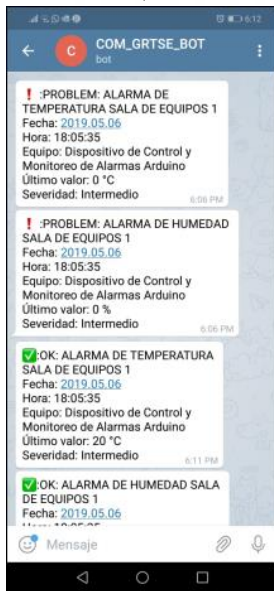
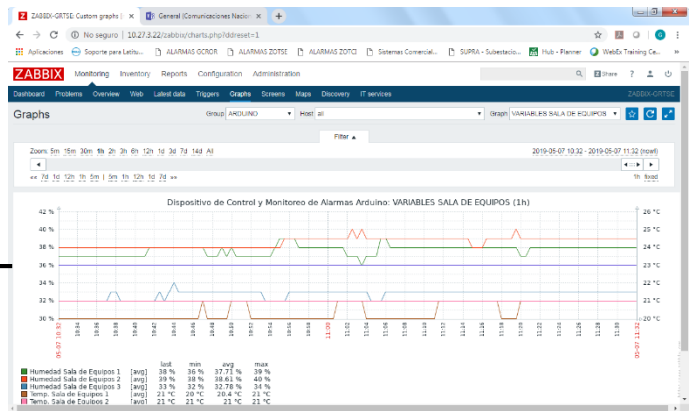
  

Hidrogeno y Movimiento	Actual	Lim. Inferior	Lim. Superior
Nivel de hidrogeno en Sala de Baterías	<0.0	<0.0	>1.0
Sensor de movimiento	0	0	0

El tiempo actualizado fecha: 7/5/2019 Hora: 11:38:05  
Departamento de Control de Servicios GRTSE - Instituto Tecnológico de Tuxtla Oaxteco

**Intranet 10.27.3.31**

Envió de notificación



Notificación en Telegram



Operadores de oficina CFE



## Conclusión

El sistema de alarmas con tecnología y microprocesador Arduino diseñado y puesto en operación en la sala de equipos del Hotel Tuxtla y COREFO Sureste es un dispositivo de bajo costo, fácil armado, programación e implementación, mismo que ha sido de gran utilidad para el monitoreo remoto de niveles de temperatura, humedad, presencia e hidrógeno. Este último sensor es de gran importancia, ya que en caso de niveles superiores a los establecidos (Norma-ISO 10156), se activa en automático el extractor de aire de la sala de baterías, mejorando así la confiabilidad de los equipos y más aún, la seguridad del personal que labora en sitio en caso de alguna contingencia.

La activación en el Arduino del protocolo SNMP permite el registro e historial del comportamiento de los sensores a través de la plataforma ZABBIX (Software utilizado por la CFE GRTSE para el monitoreo de alarmas por Ethernet). Esta plataforma notifica, vía mensajería de texto Telegram, a los responsables de la operación y mantenimiento de la sala de equipos del COREFO Sureste cuando existen niveles inferiores o superiores a los establecidos previamente en la configuración y tomando como base las normas establecidas. Y gracias a la activación del protocolo, se logró eliminar el gasto mensual que implicaba la renta de la línea telefónica que era necesaria para tener el funcionamiento del sistema operando completamente.

El monitoreo vía Intranet (página 10.27.3.31), permite la visualización de las alarmas en tiempo real y los niveles de los sensores, para que de esta manera los operadores e ingenieros a cargos de diversas áreas puedan consultar la información las veces y el tiempo que deseen hacerlo desde su oficina o escritorio.

Adicionalmente se logra apreciar un mayor desarrollo del sistema actual con sus limitantes y alcances.

### Alcances

- En la fase de diseño y pruebas de la página web, se logró demostrar que se pueden implementar la programación de botones digitales de mando para comunicarse con la tarjeta Arduino y realizar diversas acciones de manera manual.
- La visualización de la página tiene un alcance de vista desde Mexicali hasta Chiapas, que es permitida mientras se cumplan ciertas restricciones de seguridad de la red de Comisión Federal de Electricidad (CFE)
- Con la conexión a internet que nos otorga la tarjeta Shield Ethernet, se puede realizar las notificaciones con servidores específicas, como, por ejemplo; correo electrónico, Telegram, todo dependiendo los permisos que se otorguen de la configuración de red del área de trabajo.
- Con la tarjeta Shield Ethernet, se puede trabajar con protocolos de red que facilitan las ordenes de trabajo de una empresa
- La plataforma ZABBIX nos permite tener una mejor organización de información de sistemas, separando los datos por área de trabajo con su respectiva cuenta de usuario.

## limitaciones

- Debido a la poca memoria dedica para procesamiento con la que cuenta el Arduino modelo mega, se tuvo que disminuir el diseño de la página web, a una versión más ligera para poder trabajar de manera estable en conjunto con el protocolo de red SNMP.
- La seguridad en la red de intranet con la que cuenta Comisión Federal de Electricidad.
- El límite financiero otorgado al presupuesto para la realización del proyecto.

## Área de oportunidades a desarrollar

- Monitoreo de corriente del extractor.
- Implementación de más sensores para mejoramiento del monitoreo.
- Implementación de sistema de seguridad en la puerta de acceso

Tabla 5 Análisis costo vs beneficio

No.	ETAPA	COSTO CON RFSIM (ANTES)	COSTO CON ETHERNET (AHORA)	BENEFICIO
1	Microprocesador Arduino	\$ 350.00	\$ 350.00	<ul style="list-style-type: none"> <li>• Detectar niveles anormales de temperatura, humedad y gases en la sala de equipos.</li> <li>• Cuando un sensor detecte un valor superior al normal (temperatura, humedad, hidrogeno, movimiento), se le informara en tiempo real al personal operativo responsable para dar solución al evento, esto en aprovechamiento de la infraestructura con la que ya cuenta CFE.</li> <li>• Evitar daños mayúsculos en la sala de equipos del Hotel Tuxtla o al inmueble del COREFO Sureste, si la alarma no se atiende a tiempo</li> </ul>
2	Temperatura y humedad	\$200.00	\$200.00	
3	Movimiento	\$30.00	\$30.00	
4	Hidrógeno (sala de baterías)	\$60.00	\$60.00	
5	Módulo SIM	\$770.00	---	
6	Módulo Ethernet	---	\$170.00	
7	Varios (RTC, cables, caja, etc.)	\$200.00	\$300.00	
8	Módulos de relevadores (control del extractor)	---	\$30.00	
9	Costo mensual en telefonía celular	\$200.00	---	
	Totales	\$ 1,610.00 Mas + \$200.00 gastos mensuales	\$ 1,140.00	

## Bibliografía

- Arduino. (31 de Mayo de 2019). *Arduino*. Obtenido de <http://arduino.cl/que-es-arduino/>
- Arduino. (31 de Mayo de 2019). *Arduino*. Obtenido de <http://arduino.cl/arduino-mega-2560/>
- Arduino. (31 de Mayo de 2019). *Luis Llamas*. Obtenido de <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>
- Arduino. (31 de Mayo de 2019). *Pinterest*. Obtenido de <https://www.pinterest.com.mx/pin/61431982396527053/>
- CFE. (30 de Mayo de 2019). *CFE*. Obtenido de CFE Corporativo: <https://www.cfe.mx/acercacfe/Estructura%20CFE/Pages/corporativo.aspx>
- GRTS, C. T. (2019). *Especificaciones/ normas de salas de equipos TIA/EIA 942*.
- NALBER. (31 de Mayo de 2019). *tvnalber*. Obtenido de <https://www.tvnalber.com/dht22>
- <https://polaridad.es/conexion-ethernet-tcp-arduino/>
- <https://www.luisllamas.es/arduino-ethernet-shield-w5100/>
- <https://forum.arduino.cc/index.php?topic=138024.0>
- <https://www.instructables.com/id/Ethernet-Switching-with-Arduino/>
- <https://www.arduino.cc/en/Reference/Ethernet>
- [https://naylorlampmechatronics.com/blog/27\\_Tutorial-Ethernet-Shield-y-Arduino.html](https://naylorlampmechatronics.com/blog/27_Tutorial-Ethernet-Shield-y-Arduino.html)
- [https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5\\_lista\\_elementos](https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos)
- <https://www.instructables.com/id/How-to-Add-Arduino-Gas-Sensor-Library-in-to-Proteu/>
- <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>
- <https://hetpro-store.com/TUTORIALES/arduino-millis/>
- [https://www.quackit.com/html/html\\_editors/ck\\_editor\\_full.cfm](https://www.quackit.com/html/html_editors/ck_editor_full.cfm)
- [http://mgphp.blogspot.com/2008/05/php-php-es-un-lenguaje-de-programaci\\_427.html](http://mgphp.blogspot.com/2008/05/php-php-es-un-lenguaje-de-programaci_427.html)
- <https://www.prometec.net/ethernetled/#>
- <http://www.educachip.com/mover-un-servomotor-por-internet-con-arduino-ethernet-shield/>
- <http://html.conclase.net/w3c/html401-es/present/graphics.html>
- [https://create.arduino.cc/projecthub/Arduino\\_Genuino/telegram-bot-library-ced4d4](https://create.arduino.cc/projecthub/Arduino_Genuino/telegram-bot-library-ced4d4)
- <https://leantec.es/como-conectar-arduino-a-twitter/>
- <https://geekytheory.com/como-usar-twitter-con-arduino>
- <https://code.google.com/archive/p/agentuino/wikis/AgentuinoLibrary.wiki>
- <https://forum.arduino.cc/index.php?topic=304540.0>
- <http://arduinoprojexp.blogspot.com/2014/11/agentuino-zabbix-parte-2-implementacao.html>

[https://github.com/jecrespo/Ejemplos-Arduino/blob/master/04\\_Multiserver/04\\_Multiserver.ino](https://github.com/jecrespo/Ejemplos-Arduino/blob/master/04_Multiserver/04_Multiserver.ino)

<https://forum.arduino.cc/index.php?topic=38103.0>

<https://www.microcontroller-project.com/snmp-over-arduino-ethernet-shield.html>

<https://forum.arduino.cc/index.php?topic=50848.45>


<http://cric.grenoble.cnrs.fr/Administrateurs/Outils/MIBS/?module=RFC1213-MIB&fournisseur=ietf>

<http://www.rinconingenieril.es/rtc-control-a-un-rele/?post=true>

<https://forum.arduino.cc/index.php?topic=374139.0>

## Anexos

Tabla 6 Programa de trabajo (Cronograma)

				<b>COMISIÓN FEDERAL DE ELECTRICIDAD</b> GERENCIA REGIONAL DE TRANSMISION SURESTE SUBGERENCIA DE COMUNICACIONES				
				Monitoreo y control de alarmas del centro de operación de fibra óptica de la Gerencia de CFE Regional de Transmisión Sureste (GRTSE)				
No.	ACTIVIDAD	FECHA INICIAL	FECHA DE ENTREGA	Febrero	Marzo	Abril	Mayo	Junio
1	Realizar programa de trabajo.	05 de febrero de 2019	05 de febrero de 2019					
2	Memoria técnica	05 de febrero de 2019	3 de junio de 2019					
3	Observación y estudio de la estructura del proyecto actual	6 de febrero de 2019	6 de febrero de 2019					
4	Reunión para discutir sobre el proyecto	8 de febrero de 2019	8 de febrero 2019					
5	Investigar circuitos y componentes electrónicos necesarios para la integración del sistema.	11 de febrero de 2019						
5.1	Hojas de datos de componentes. (Datasheets)	12 de febrero de 2019	14 de febrero de 2019					
5.2	Códigos de programación de componentes.	14 de febrero de 2019	18 de febrero de 2019					
5.3	Diagramas de conexiones de componentes	14 de febrero de 2019	18 de febrero de 2019					
5.4	Estudio de los actuadores a utilizar	13 de febrero de 2019	15 de febrero de 2019					
6	Diagramas de planta	16 de febrero de 2019						
6.1	Diagramas de planta a mano.							
6.2	Diagramas de planta en power point.							
7	Diseño simulado del proyecto	20 de febrero de 2019						
7.1	Monitoreo de sistema de temperatura.	21 de febrero de 2019	26 de febrero de 2019					
7.2	Monitoreo de sistema de sensores de humedad.	21 de febrero de 2019	26 de febrero de 2019					
7.3	Sensores de oxígeno e hidrogeno (activación de Extractor por tiempo)	27 de febrero de 2019	5 de marzo de 2019					
7.4	Sensores de presencia (encendido de buzzer)	6 de marzo de 2019	12 de marzo de 2019					
8	Diseño de configuración de la	13 de marzo de 2019						

	pantalla LCD. (cambio de mensaje impreso en la pantalla)							
9	Diseño del sistema de conectividad	13 de marzo de 2019						
9.1	Diseño del sistema de ethernet	13 de marzo de 2019	22 de marzo de 2019					
10	Presentación simulada del sistema nuevo	29 de marzo de 2019						
11	Adquisición de materiales y componentes.	1 de abril de 2019						
11.1	Adquisición de cotizaciones y listas de precios	1 de abril de 2019	3 de abril de 2019					
11.2	Componentes locales (MV, L&L, la moderna, steren, etc.)	8 de abril de 2019						
12	Armado integral del sistema	15 de abril de 2019						
12.1	Realización de inventario de material	16 de abril de 2019	16 de abril de 2019					
12.2	Reordenar etapa de detección de temperatura. (cambiar de lugar para mayor espacio)	17 de abril de 2019	19 de abril de 2019					
12.3	Reordenar etapa de detección de humedad. (cambiar de lugar para mayor espacio)	17 de abril de 2019	19 de abril de 2019					
12.4	Armar y programar etapa de detección de oxígeno.	22 de abril de 2019	26 de abril de 2019					
12.5	Armar y programar etapa de detección de presencia.	26 de abril de 2019	2 de mayo de 2019					
12.6	Armar y programar pantalla LCD	2 de mayo de 2019						
13	Armar y programar conectividad	2 de mayo de 2019						
13.1	Pruebas de conexión ethernet	2 de mayo de 2019	10 de mayo de 2019					
14	Primeras pruebas del sistema	13 de mayo de 2019	15 de mayo de 2019					
14.1	Detección de errores encontrados	15 de mayo de 2019	17 de mayo de 2019					
14.2	Solución de errores	20 de mayo de 2019	24 de mayo de 2019					
15	Instalación del sistema de alarmas en sala de equipos del COREFO GRTSE.	28 de mayo de 2019						
15.1	Pruebas finales.	29 de mayo de 2019	30 de mayo de 2019					
16	Presentación del sistema al público	3 de junio de 2019						
Elaboro: Alexander Jiménez Moreno Residente ITTG				Reviso: Ing. Guillermo Iván Huerta Franco Jefe Depto. Conexión de Servicios COM GRTSE				

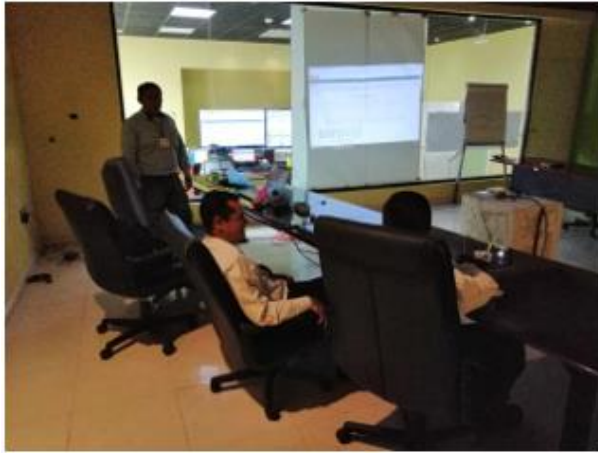


Ilustración 73. Presentación del proyecto a personal operativo de la GRTSE

Fuente: Elaboración propia, 2019



Ilustración 74. Demostración para programar Arduino

Fuente: Elaboración propia, 2019



## URL de datasheet de componentes:

DHT22

<https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>

sensor PIR:

<https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>

MQ-8

<https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Biometric/MQ-8.pdf>

Arduino mega:

<http://www.mantech.co.za/datasheets/products/A000047.pdf>

Shield ethernet W1500:

[https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100\\_Datasheet\\_v1\\_1\\_6.pdf](https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf)

## Programación en software Arduino del sistema de monitoreo y control de alarmas del centro de operación de fibra

### Librerías utilizadas:

```
#include <Streaming.h> // libreria para protocolo SNMP
#include <MemoryFree.h> // libreria para protocolo SNMP
#include <Agentuino.h> // libreria para protocolo SNMP
#include <SPI.h> //libreria para poder usar la tarjeta
shield ethernet
#include <Ethernet.h> //libreria para poder usar la tarjeta
shield ethernet
#include <Wire.h> // libreria para poder utilizar el modulo
I2C
#include "RTClib.h" //libreria para poder usar el modulo de
tiempo de reloj real
#include <LCD.h> //libreria para poder usar el lcd
#include <LiquidCrystal_I2C.h> //libreria para poder usar al adaptador I2C
#include <DHT.h> //libreria para poder utilizar los sensores DHT22
#include <DHT_U.h> //libreria para utilizar los sensores DHT22
#define DHT1PIN30 // declaracion de pines utilizados por los sensores DHT22
#define DHT2PIN31
#define DHT3PIN32
#define DHT4PIN33
#define DHT1TYPE DHT22 // definir que tipo de sensor DHT se esta utilizando
#define DHT2TYPE DHT22
#define DHT3TYPE DHT22
#define DHT4TYPE DHT22
```



### Declaracion de variables:

```
LiquidCrystal_I2C lcd (0x27, 2, 1, 0, 4, 5, 6, 7); //pines del LCD que va a utilizar el adaptador I2C
```

```
int led_rojo=26;
int ledverde=24;
```

```

int ventiarduino=2;

int valor_pir;
int pin_de_pir=34; //pin de entrada donde se recibe la señal del sensor pir
int pin_de_led=36; //salida que se activa cuando el sensor pir detecta movimiento

// variables utilizadas para el control del tiempo
int intervalo=1000;
int tiempo;
int tiempo_anterior=0;
int tiempo_actual;

// variables que se utilizan para el mando del sensor MQ-8
float gasvalor;
int pin_de_salida_gas=48; //pin de salida que se controla con los valores del sensor de hidrogeno
int ventilador=25; //este variable es la que se usa para declarar el pin de salida de donde se controla
el led amarillo
float prender_venti;

int buzzer=35; // variable del pin utilizado por el buzzer

// variables utilizadas para el mando de los sensores DHT22
int DHT1PIN=30; //pin de entrada que recibe la señal del sensor dht_1
int DHT2PIN=31; //pin de entrada que recibe la señal del sensor dht_2
int DHT3PIN=32; //pin de entrada que recibe la señal del sensor dht_3
int DHT4PIN=33; //pin de entrada que recibe la señal del sensor dht_4

int TEMP1;
int TEMP2;
int TEMP3;
int TEMP4;

int HUM1;
int HUM2;
int HUM3;
int HUM4;

DHT dht1 (DHT1PIN,DHT1TYPE); //comando para declarar el sensor dht22 como un objeto dht para
que arduino pueda reconocerlo
DHT dht2 (DHT2PIN,DHT2TYPE);
DHT dht3 (DHT3PIN,DHT3TYPE);
DHT dht4 (DHT4PIN,DHT4TYPE);
RTC_DS1307 RTC; //comando para declarar el módulo de reloj como un objeto rtc para que lo
reconozca arduino

//prender pir en fines de semana
int r_diaSemana;
int hora;
int minuto;
int segundo;
int prender_pir=49;

```

### Declaración de objetos y puerto del protocolo SNMP

```

const static char sysDescr[] PROGMEM = "1.3.6.1.2.1.1.1.0"; // read-only (DisplayString)
const static char sysUpTime[] PROGMEM = "1.3.6.1.2.1.1.3.0"; // read-only (TimeTicks)
const static char sysContact[] PROGMEM = "1.3.6.1.2.1.1.4.0"; // read-write (DisplayString)
const static char sysName[] PROGMEM = "1.3.6.1.2.1.1.5.0"; // read-write (DisplayString)
const static char sysLocation[] PROGMEM = "1.3.6.1.2.1.1.6.0"; // read-write (DisplayString)

```

```

const static char sysServices[] PROGMEM = "1.3.6.1.2.1.1.7.0"; // read-only (Integer)

// oid's de sensor DHT22 1
const static char systemperature_1[] PROGMEM = "1.3.6.1.4.1.36582.5.1.0"; //oid de temperatura 1
const static char syshumidity_1[] PROGMEM = "1.3.6.1.4.1.36582.5.1.1"; //oid de humedad 1

// oid's de sensor DHT22 2
const static char systemperature_2[] PROGMEM = "1.3.6.1.4.1.36582.5.1.2"; // oid de temperatura 2
const static char syshumidity_2[] PROGMEM = "1.3.6.1.4.1.36582.5.1.3"; // oid de humedad 2

// oid's de sensor DHT22 3
const static char systemperature_3[] PROGMEM = "1.3.6.1.4.1.36582.5.1.4"; // oid de temperatura 3
const static char syshumidity_3[] PROGMEM = "1.3.6.1.4.1.36582.5.1.5"; // oid de humedad 3

// oid's de sensor DHT22 4
const static char systemperature_4[] PROGMEM = "1.3.6.1.4.1.36582.5.1.6"; // oid de temperatura 4
const static char syshumidity_4[] PROGMEM = "1.3.6.1.4.1.36582.5.1.7"; // oid de humedad 4

//oid de sensor de hidrogeno
const static char syshidrogeno[] PROGMEM = "1.3.6.1.4.1.36582.5.1.8"; // oid de hidrogeno
//oid de sensor de movimiento
const static char sysmovimiento[] PROGMEM = "1.3.6.1.4.1.36582.5.1.9"; // oid de movimiento

api_status = Agentuino.begin("public","private",161);
//
if ( api_status == SNMP_API_STAT_SUCCESS ) {
//
Agentuino.onPduReceive(pduReceived);
//
delay(10);
//
return;
}
//
delay(10);
}

```

### Void's de mandos:



```

void loop() {
monitoreo_total();
fin_de_semana();
ethernet();
//listen/handle for incoming SNMP requests
Agentuino.listen();
temperature_1 = dht1.readTemperature();
humidity_1 = dht1.readHumidity();
temperature_2 = dht2.readTemperature();
humidity_2 = dht2.readHumidity();
temperature_3 = dht3.readTemperature();

```

```

humidity_3 = dht3.readHumidity();
temperature_4 = dht4.readTemperature();
humidity_4 = dht4.readHumidity();
gas_hidrogeno =analogRead(A14);
pir_movimiento=digitalRead(pin_de_pir);
}
//el void de monitoreo total se uso para poder llamar a los demas void

```

```

void monitoreo_total(){
monitoreo();
temperatura_de_caja();
pir();
prender_ventilador();
}

```

## Parte del void para utilizar shiel ethernet y la creación y diseño de página web

```
void ethernet(){
```

```
    DateTime now=RTC.now(); //obtener la fecha y hora del RTC
```

```

EthernetClient cliente = server.available(); //comprobamos si hay peticiones
if (cliente) { //en caso afirmativo
boolean currentLineIsBlank = true;
while (cliente.connected()) {
if (cliente.available()) { //hay algo pendiente de leer
char c = cliente.read(); //leemos los caracteres de uno a uno
if (c == '\n' && currentLineIsBlank) {
cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println("Connection: close");
cliente.println("Refresh: 60"); // tiempo que se le da a la pagina para que se actualize de manera
automatica
cliente.println();
cliente.println("<!DOCTYPE HTML>"); // tipo de documento que se envia a la pagina web
cliente.println("<html>");
cliente.println("<head>");
cliente.println("<title>SENSORES CFE COREFO GRTSE</title>");
cliente.println("</head>");
cliente.println("<body>");
cliente.println("<img align='left' src='http://www.sise-servicios.com/wp-content/uploads/2017/06/CFE-01-820x421.png'alt='logo de CFE'title='logo hecho por CFE'whidth='100'height='98'/>");
cliente.println("<img align='right'
src='http://tectuxtla.net/pluginfile.php/1/core_admin/logo/0x150/1549680653/00%20logo_ittg_trasparente.png'alt='logo de ITTG'title='logo hecho por el ITTG'whidth='135'height='140'/>");

cliente.println("<p style='text-align: center;'><span style='font-size:22px;'><span
style='color:#008000;'>Gerencia Regional de Transmisi&oacute;n Sureste</span></span></p>");

cliente.println("<p style='text-align: center;'><span style='font-size:18px;'><span
style='color:#008000;'><strong>Subgerencia de comunicaciones</strong></span></span></p>");
cliente.println("<p style='text-align:center;'>");

```

```

cliente.println("<p style='text-align: center;'><span style='font-size:26px'><span
style='color:#008000;'><strong>Monitoreo y control de alarmas en Sala de Equipos Hotel
Tuxtla</strong></span></span></p>");
cliente.println("<br />");

```

```

cliente.println("<table align='center' border='1' cellpadding='1' cellspacing='1' style='width:600px;'>");
cliente.println("<tbody>");
cliente.println("<tr>");
cliente.println("<td style='text-align: center;'><strong>Descripci&oacute;n
(Temperatura)</strong></td>");
cliente.println("<td style='text-align: center;'><strong>Actual (&deg;C)</strong></td>");
cliente.println("<td style='text-align: center;'><strong>Lim. Inferior (&deg;C)</strong></td>");
cliente.println("<td style='text-align: center;'><strong>Lim. Superior (&deg;C)</strong></td>");
cliente.println("</tr>");
cliente.println("<tr>");
cliente.println("<td>Sala de Bater&iacute;as</td>");
if (TEMP1==0){
cliente.println("<td style='text-align: center;' bgcolor='red'>Desconectado</td>");
}
else if (TEMP1<18){
cliente.println("<td style='text-align: center;' bgcolor='red'>");
cliente.println(TEMP1);
cliente.println("</td>");
}
else if (TEMP1>30){
cliente.println("<td style='text-align: center;' bgcolor='red'>");
cliente.println(TEMP1);
cliente.println("</td>");
}
else{
cliente.println("<td style='text-align: center;' bgcolor='gren'>");
cliente.println(TEMP1);
cliente.println("</td>");
}
cliente.println("<td style='text-align: center;'>&lt;18</td>");
cliente.println("<td style='text-align: center;'>&gt;30</td>");
cliente.println("</tr>");
cliente.println("<tr>");
cliente.println("<td>Centro Sala de Equipos</td>");
if (TEMP2==0){
cliente.println("<td style='text-align: center;' bgcolor='red'>Desconectado</td>");
}
else if (TEMP2<18){
cliente.println("<td style='text-align: center;' bgcolor='red'>");
cliente.println(TEMP2);
cliente.println("</td>");
}
else if (TEMP2>30){
cliente.println("<td style='text-align: center;' bgcolor='red'>");
cliente.println(TEMP2);
cliente.println("</td>");
}
else{
cliente.println("<td style='text-align: center;' bgcolor='gren'>");
cliente.println(TEMP2);
cliente.println("</td>");
}
cliente.println("<td style='text-align: center;'>&lt;18</td>");
cliente.println("<td style='text-align: center;'>&gt;30</td>");

```



```
cliente.println("</tr>");
cliente.println("<tr>");
cliente.println("<td>Sala de Bateria</td>");
if (HUM1==0){
cliente.println("<td style='text-align: center;' bgcolor='red'>Desconectado</td>");
}
else if (HUM1<20){
cliente.println("<td style='text-align: center;' bgcolor='red'>");
cliente.println(HUM1);
cliente.println("</td>");
}
else if (HUM1>60){
cliente.println("<td style='text-align: center;' bgcolor='red'>");
cliente.println(HUM1);
cliente.println("</td>");
}
else{
cliente.println("<td style='text-align: center;' bgcolor='green'>");
cliente.println(HUM1);
cliente.println("</td>");
}
```