

ISBN 978-99961-50-75-3 (Impreso)
ISBN 978-99961-50-89-0 (E-Book)

INFORME FINAL DE INVESTIGACIÓN

**DESARROLLO DE UN PROTOTIPO
ELECTRÓNICO E INFORMÁTICO APLICANDO
LA TECNOLOGÍA DE INTERNET DE LAS
COSAS PARA LA MEDICIÓN Y CONTROL DE
MAGNITUDES FÍSICAS.
*En asocio con COCESNA.***

DOCENTE INVESTIGADOR PRINCIPAL:
LIC. MANUEL DE JESÚS GÁMEZ LÓPEZ

DOCENTE CO INVESTIGADOR:
ING. OSCAR ARMANDO SÁNCHEZ SANTOS

CENTRO REGIONAL MEGATEC ZACATECOLUCA

ENERO 2018

ISBN 978-99961-50-75-3 (Impreso)
ISBN 978-99961-50-89-0 (E-Book)

INFORME FINAL DE INVESTIGACIÓN

**DESARROLLO DE UN PROTOTIPO
ELECTRÓNICO E INFORMÁTICO APLICANDO
LA TECNOLOGÍA DE INTERNET DE LAS
COSAS PARA LA MEDICIÓN Y CONTROL DE
MAGNITUDES FÍSICAS.
*En asocio con COCESNA.***

DOCENTE INVESTIGADOR PRINCIPAL:
LIC. MANUEL DE JESÚS GÁMEZ LÓPEZ

DOCENTE CO INVESTIGADOR:
ING. OSCAR ARMANDO SÁNCHEZ SANTOS

CENTRO REGIONAL MEGATEC ZACATECOLUCA

ENERO 2018

Rectora

Licda. Elsy Escolar SantoDomingo

Vicerrector Académico

Ing. Carlos Alberto Arriola Martínez

Vicerrectora Técnica Administrativa

Inga. Frineé Violeta Castillo

**Dirección de Investigación
y Proyección Social**

Ing. Mario Wilfredo Montes, Director

Ing. David Emmanuel Ágreda

Sra. Edith Aracely Cardoza

Director Centro Regional Zacatecoluca

Ing. Christian Antonio Guevara Orantes

629.895

G192d Gámez López, Manuel de Jesús, 1984 -

sv

Desarrollo de un prototipo electrónico e informático aplicando la tecnología del internet de las cosas para la medición y control de magnitudes físicas. / Manuel de Jesús Gámez López, Oscar Armando Sánchez Santos -- 1ª ed. -- Santa Tecla, La Libertad, El Salv. : ITCA Editores, 2018.

58 p. : 28 cm.

ISBN 978-99961-50-75-3 (impreso)

ISBN 978-99961-50-89-0 (E-Book)

1. Automatización de oficinas. 2. Internet de las Cosas. 3. Control de temperatura. 4. Programación orientada a objetos (computadores). Mediciones I. Sánchez Santos, Oscar Armando, 1984-. II. Título.

Autores

Lic. Manuel de Jesús Gámez López

Coautor

Ing. Oscar Armando Sánchez Santos

Tiraje: 13 ejemplares

Año 2018

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica y el sector empresarial, como un aporte al desarrollo del país. Este informe de investigación no puede ser reproducido o publicado parcial o totalmente sin previa autorización de la Escuela Especializada en Ingeniería ITCA-FEPADE. Para referirse a este documento se debe citar al autor. El contenido de este informe es responsabilidad exclusiva de los autores.

Escuela Especializada en Ingeniería ITCA-FEPADE

Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio web: www.itca.edu.sv

Tel: (503)2132-7423

Fax: (503)2132-7599

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA.	4
2.1.	DEFINICIÓN DEL PROBLEMA.....	4
2.2.	ESTADO DE LA TÉCNICA - ANTECEDENTES.....	4
2.3.	JUSTIFICACIÓN.....	5
3.	OBJETIVOS	6
3.1.	OBJETIVO GENERAL	6
3.2.	OBJETIVOS ESPECÍFICOS	6
4.	HIPÓTESIS - PREGUNTA PROBLEMA.....	6
5.	MARCO TEÓRICO DE LA INVESTIGACIÓN.....	6
5.1.	INTERNET DE LAS COSAS.	7
5.2.	REDES INFORMÁTICAS INALÁMBRICAS WIFI.....	9
5.3.	SHIELD ETHERNET - HARDWARE ETHERNET EN ARDUINO	12
5.4.	GPS O SISTEMA DE POSICIONAMIENTO GLOBAL.....	20
5.5.	TECNOLOGÍA MÓVIL O GSM.....	21
5.6.	BASE DE DATOS.	25
5.7.	DESARROLLO DE APLICACIÓN MÓVIL.....	27
6.	METODOLOGÍA DE LA INVESTIGACIÓN.....	31
6.1.	COMPONENTES ELECTRÓNICOS DEL PROYECTO.....	33
6.2.	CONEXIONES INTERNAS EN EL CHASIS	34
6.3.	INTERFACE ELECTRÓNICA DE CONTROL DE CARGAS	35
6.4.	DESARRROLLO DE CÓDIGO ARDUINO	35
7.	RESULTADOS	38
7.1.	CÓDIGO FUENTE DEL PROYECTO.....	38
7.2.	BASE DE DATOS	51
7.3.	CARACTERÍSTICAS DEL PROTOTIPO	52
8.	CONCLUSIONES	54
9.	RECOMENDACIONES.....	54
10.	GLOSARIO	55
11.	REFERENCIAS BIBLIOGRÁFICAS	57

1. INTRODUCCIÓN

Para este proyecto se diseñó y desarrolló un dispositivo electrónico con capacidad de conectividad a la red LAN e INTERNET para la medición de magnitudes físicas a través de sensores. Este dispositivo es capaz de interactuar con otros objetos conectados al mismo (actuadores) si la temperatura aumenta demasiado, teniendo la autonomía suficiente para actuar al respecto, como para activar un ventilador, la calefacción, un aire acondicionado o cualquier otro dispositivo que se decida conectar a la tarjeta electrónica controladora de cargas u objetos; además permite la transmisión de la información recopilada del ambiente a un portal web que contiene como principal elemento pictórico un dashboard o gráficas interactivas donde se puede visualizar la información que envía el dispositivo IoT inmediatamente. Para este proyecto se ha creado una aplicación móvil para Android, la cual interactúa en conjunto para la monitorización en tiempo real de magnitudes físicas de temperatura y humedad, entre otras, dentro de una de las salas con equipos de comunicación en COCESNA, implementando de esta forma la tecnología de Internet de las Cosas IoT.

Como un agregado, se creó en el portal web la opción de poder registrar nuevas salas por país a nivel de Centro América, dejando solamente a COCESNA la facultad de replicar el dispositivo IoT para integrarlo a la red corporativa y escalar este sistema IoT.

2. PLANTEAMIENTO DEL PROBLEMA.

2.1. DEFINICIÓN DEL PROBLEMA.

Derivado de la auditoria de recertificación realizada en el año 2016 por el Bureau Veritas (LÍDER MUNDIAL EN SERVICIOS DE VERIFICACIÓN, INSPECCIÓN, CERTIFICACIÓN Y FORMACIÓN), se identificó que actualmente en las salas de equipos de COCESNA no se cuenta con un registro de parámetros ambientales como temperatura y humedad, alertas o notificaciones oportunas a móviles que permita mantener dichas salas dentro de las tolerancias definidas por COCESNA o por el fabricante. Esta situación representa un riesgo para la estabilidad operativa de los equipos, e incidir en la toma de acciones ante posibles casos de sobrecalentamiento de los mismos y afectar la disponibilidad del servicio brindado por estos.

2.2. ESTADO DE LA TÉCNICA - ANTECEDENTES

En la actualidad, ya existen numerosos proyectos donde se fomenta la aplicabilidad de la tecnología de Internet de las Cosas o de los Objetos (IoT o Internet of Things en inglés) en diferentes áreas empresariales. Entre algunos de estos se pueden mencionar sistemas inteligentes de monitoreo ambiental.

Sistema de monitoreo a distancia para evaluar condición de enfermos graves.

Centro dedicado a internet de las cosas. El proyecto SECURIZA tiene como objetivo la definición e implantación de mejoras a nivel de seguridad en los productos y servicios más innovadores de las entidades de la AEI CIBERSEGURIDAD. Proyecto OPTIMIZE es desarrollar sistemas de rehabilitación integral de edificios que, obteniendo una disminución de al menos el 60% del consumo de energía primaria y mejorando al menos dos letras en la certificación energética, busquen la mayor amortización de la inversión en rehabilitación. Transformación. [Edificios seguros e inteligentes con las tecnologías de Intel® OIT](#). PROMOVER pretende desarrollar un producto para la PROYECTOS AEI 2014 monitorización de la calidad del agua de vertido, a partir de un equipo autónomo, costo-eficiente y de fácil mantenimiento, que permita realizar la monitorización on-line de aquellos parámetros que tengan interés en el conocimiento de la calidad del agua de vertidos, y permita transmitir y procesar la información y clasificar el estado de las aguas.

La mayoría de los dispositivos inteligentes de Internet de las cosas no se encuentran en nuestros hogares o teléfonos, sino en fábricas, empresas y en la asistencia sanitaria. Lo anterior debido a que los objetos inteligentes proporcionan a estas grandes industrias los datos esenciales que necesitan para realizar un seguimiento del inventario, gestionar máquinas, aumentar la eficiencia, ahorrar costes e incluso salvar vidas.

2.3. JUSTIFICACIÓN

En la industria salvadoreña, en su mayoría se carecen de mecanismos tecnológicos para obtener información ágil y oportuna en sus procesos y la toma de decisiones; además de elevados costos en controles y monitoreo, que reduce la productividad de las mismas; por lo que manejar y desarrollar nuevas tecnologías posibilita opciones para la industria de nuestro país.

La tendencia a los bajos costos de la tecnología y la accesibilidad de producir interfaces entre hardware y software, permite crear negocios, en los que se produzcan productos y/o servicios que involucren el Internet de las Cosas (IoT). Nuestra Industria Salvadoreña debe implementar este tipo de tecnología en el manejo de su información, si quiere ser competitivo en el mundo globalizado; Actualmente se carece de aplicación de este tipo de tecnología y si las hay, son con exceso de costo para la adquisición en las empresas, y es la academia la que debe de apoyar a través de las investigaciones en el área, a ser accesible para su utilización.

Cabe destacar que este tipo de investigaciones representará a futuro beneficios de propiedad y de aportaciones a diferentes empresas o negocios que demanden este tipo de sistemas de IoT, para optimizar la forma de manejo y accesibilidad a su información para la toma de decisiones.

La Red de Redes, Internet, es la herramienta de información, formación y comunicación más potente que existe en la actualidad. Internet permite superar la distancia física como factor limitante; el desarrollo de este proyecto de investigación, permite aprovechar el potencial de internet, teniendo especial impacto en la asistencia a las empresas, la docencia y la Investigación, pilares de la actividad científica, que como institución estamos comprometidos a realizar.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar un Prototipo Electrónico e Informático con tecnología de Internet de las Cosas para la medición de magnitudes física en tiempo real a través de internet.

3.2. OBJETIVOS ESPECÍFICOS

1. Diseñar el dispositivo IOT o prototipo electrónico para la monitorización de las variables físicas preseleccionadas.
2. Construir el dispositivo electrónico IOT.
3. Diseñar el Portal Web informático para IOT y verificación de la información y el Dashboard para su implementación en el portal web.
4. Implementar el prototipo IOT en la sala de equipos seleccionada en COCESNA.

4. HIPÓTESIS - PREGUNTA PROBLEMA

¿Cómo registrar y monitorear en tiempo real parámetros ambientales como temperatura y humedad en la sala de equipos de comunicaciones en COCESNA?

5. MARCO TEÓRICO DE LA INVESTIGACIÓN.

El siglo XXI ha sido marcado por grandes avances tecnológicos desde sus primeros años: la creciente miniaturización de las computadoras personales, el nacimiento y auge de los teléfonos inteligentes, las tabletas y el cómputo en la nube son sólo algunos de los ejemplos más importantes que podríamos destacar; que poco a poco han cambiado el comportamiento de las personas en la llamada sociedad de la información. (García, 2015).

Es cotidiano ver a las personas en la calle, en el trabajo o en la casa con sus teléfonos en la mano, ya sea “chateando”, revisando sus redes sociales, o consumiendo algún tipo de recurso en Internet. Un estudio sobre los hábitos de los usuarios de Internet en México 2015, realizado por la Asociación Mexicana de Internet (AMIPCI), revela que de los 53.9 millones de usuarios mexicanos, 5 de cada 10 acceden a Internet desde su teléfono inteligente y pasan en promedio 6 horas y 11 minutos al día conectados y la tendencia es que este número aumente gradualmente en los años siguientes (Asociación Mexicana de Internet, 2015).

Bien, pero ¿Qué relevancia tiene esta información? Esta pregunta tiene una respuesta muy simple: 53.9 millones de personas que están conectadas en promedio 6 horas y 11 minutos al día a Internet, mediante una variedad de dispositivos, genera una cantidad monstruosa de datos al día, por lo que actualmente hay ramas de las tecnologías de la información que se encargan de encarar la ardua tarea

de procesar esos datos y cuyos frutos han sido el desarrollo de sistemas que funcionan bajo enfoques diversos como las implementaciones de almacenes de datos (Data Warehouse) o de datos masivos (Big Data).

En toda esta innovación tecnológica hay un elemento en común que ha servido como apoyo para las tecnologías modernas: el uso de dispositivos especializados que proveen datos útiles del entorno. Por ejemplo, al examinar un teléfono inteligente se identifican dispositivos como los aerómetros, que permiten medir el movimiento de teléfono, o los sensores de proximidad en la toma de fotografías, e incluso los sensores biométricos que ofrecen una capa extra de seguridad de acceso o de autorización a las tradicionales contraseñas.

5.1. INTERNET DE LAS COSAS.

En cuanto a los dispositivos, éstos toman una importancia mayor gracias a su constante miniaturización, su producción en masa que abarata su costo y permite que más y más personas puedan hacerse de ellos, y comenzar a experimentar con su uso. Esto marca el inicio de un concepto llamado M2M, Machine-to-Machine, en el cual dos dispositivos se interconectan entre sí para realizar una acción conjunta: las primeras lámparas que se encendían o apagaban en automático es un ejemplo de las primeras aproximaciones del modelo M2M, en el que un sensor de movimiento se conecta a una red de lámparas; este principio también aplica para los grifos del lavamanos.

Sin embargo, esta interacción se limitaba a la zona donde los dispositivos estaban ubicados, por lo que a pesar del gran potencial que el M2M representaba, sólo se pudo aprovechar por sectores específicos como la construcción de edificios inteligentes (Torres ejecutivas, centros comerciales, instituciones de gobierno y educación, etc.). Esto cambió totalmente con la llegada del Internet al campo de los dispositivos, que produce un cambio de paradigma al que se le ha denominado simplemente como Internet de las cosas.

Internet de las Cosas (IoT, por sus iniciales en inglés, Internet of Things), representa quizás uno de los nichos tecnológicos más importantes de los últimos tiempos. Se entiende por IoT un conjunto de dispositivos o “cosas” interrelacionadas entre si y, a su vez, conectadas a Internet, a través del cual envían o reciben datos. Cabe destacar que estas “cosas” no se refieren a una computadora personal, laptop, teléfono inteligente, Tablet o algún dispositivo similar, por el contrario se trata de objetos diseñados con un propósito en particular, como pulseras que miden el pulso cardiaco del cuerpo humano, calzado deportivo que cuenta el número de pisadas y distancia recorrida por un corredor, etc.. Como lo menciona Adrián McEwen en su libro “Desingning the Internet of things”, “La idea (del IoT) sugiere que en vez de tener un número pequeño de dispositivos con un gran potencial de cómputo en la vida (como laptops, tablets, teléfonos o reproductores de música) se podría tener un gran número de dispositivos de menor potencial (como sombrillas, brazaletes, espejos, refrigeradores, camisas, tarjetas, zapatos, entre muchos otros)”.

Tres elementos que conforman cualquier diseño de solución basado en IoT, estos son:

- ❖ Los objetos físicos y sus características que interesan medir.

- ❖ Los dispositivos que permitirán tener una influencia sobre dichos objetos a través de diferentes aproximaciones: censar (haciendo referencia a la toma de mediciones), controlar (tener un tipo de influencia directa o indirecta en el comportamiento de los objetos) y actuar (permite otorgar cierta autonomía al objeto para actuar según estímulos y eventos externos sobre el).
- ❖ Internet, la principal vía de envío y recepción de datos entre objetos, sin embargo, esto no ha evitado que recientemente se implementen mecanismos alternativos de interconexión.



Elementos que conforman el IoT.

Estos objetos generan ambientes sobre los cuales tienen cierta influencia y realizan un número de actividades. En el hogar, por ejemplo, un sensor puede transmitir datos sobre la temperatura de la sala, si la temperatura aumenta en demasiado tiene la autonomía suficiente para actuar al respecto, como activar un ventilador o la calefacción. Sin embargo, al agregar Internet a estos mismos elementos es posible generar ideas bastante interesantes. Por ejemplo, se podría consultar desde un teléfono o computadora la información derivada de los datos del sensor, e incluso, si se considera conveniente encender la calefacción, si se desea una habitación cálida tras un día frío de invierno, o bien una fresca tras un caluroso día de primavera, todo esto desde cualquier lugar y con efecto inmediato.

Internet de las cosas (en inglés, Internet of things, abreviado OIT) es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet. Alternativamente, Internet de las cosas es la conexión de internet con "cosas u objetos" que personas. También se suele conocer como internet de todas las cosas o internet en las cosas. Si objetos de la vida cotidiana tuvieran incorporados etiquetas de radio, podrían ser identificados y gestionados por otros equipos, de la misma manera que si lo fuesen por seres humanos.

Redes inalámbricas configurables entre los objetos que nos rodean. Cuando todos los objetos que nos rodean estén equipados con dispositivos de identificación podrán ser identificados y gestionados por sistemas informáticos.

Los objetos como tales no se relacionarán, pero pueden ser controlados por algún tipo de agente a través de la red. Por ejemplo, si los libros, termostatos, refrigeradores, la paquetería, lámparas, botiquines, partes automotrices, entre otros estuvieran conectados a internet y equipados con dispositivos de identificación, no existirían, en teoría, artículos fuera de stock o medicinas caducadas; sabríamos exactamente la ubicación, como se consumen en el mundo; el extravió sería cosa del pasado y sabríamos que está encendido o apagado en todo momento.

5.2. REDES INFORMÁTICAS INALÁMBRICAS WIFI.

Una red inalámbrica informática, no es más que un conjunto de ordenadores o computadoras, o de cualquier otro dispositivo informático, comunicados entre sí mediante soluciones que no requieran el uso de cables de interconexión. Para disponer de una red inalámbrica sólo hace falta instalar una tarjeta de red inalámbrica. Este proceso resulta mucho más rápido y flexible que instalar una red cableada. Una vez instalada una red inalámbrica, su utilización es prácticamente idéntica a la de una red cableada. Los ordenadores o computadoras que forman parte de la red pueden comunicarse entre sí y compartir toda clase de recursos. Se pueden compartir archivos, directorios, impresoras, unidades de disco, o incluso, el acceso a otras redes, como puede ser el Internet.

Para el usuario final en general, no hay diferencia entre estar conectado a una red cableada o a una red inalámbrica. De la misma forma, al igual que ocurre con las redes cableadas, una red inalámbrica puede estar formada por tan sólo dos computadoras o por miles de ellas.

MECANISMO DE CONEXIÓN DE DISPOSITIVOS ELECTRÓNICOS DE FORMA INALÁMBRICA.



En algunos países hispanohablantes (/ 'wifi/) es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi, tales como: una computadora personal, una consola de videojuegos, un smartphone o un reproductor de audio digital, pueden conectarse a Internet a través de un punto de acceso de red inalámbrica. Dicho punto de acceso (o hotspot) tiene un alcance de unos 20 metros en interiores y al aire libre una distancia mayor. Pueden cubrir grandes áreas la superposición de múltiples puntos de acceso.

Wi-Fi es una marca de la Wi-Fi Alliance (anteriormente la WECA: Wireless Ethernet Compatibility Alliance), la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

Estándares que certifica Wi-Fi

Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutaban de una aceptación internacional debido a que la banda de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente. En la actualidad ya se maneja también el estándar IEEE 802.11a, conocido como WIFI 5, que opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpios. La banda de 5 GHz ha sido recientemente habilitada y, además, no existen otras tecnologías (Bluetooth, microondas, ZigBee, WUSB) que la estén utilizando, por lo tanto existen muy pocas interferencias. Su alcance es algo menor que el de los estándares que trabajan a 2.4 GHz (aproximadamente un 10%), debido a que la frecuencia es mayor (**a mayor frecuencia, menor alcance**). Existe un primer borrador del estándar IEEE 802.11n que trabaja a 2.4 GHz y a una velocidad de 108 Mbit/s. Sin embargo, el estándar 802.11g es capaz de alcanzar ya transferencias a 108 Mbit/s, gracias a diversas técnicas de aceleramiento. Actualmente existen ciertos dispositivos que permiten

utilizar esta tecnología, denominados Pre-N.

Existen otras tecnologías inalámbricas como Bluetooth que también funcionan a una frecuencia de 2.4 GHz, por lo que puede presentar interferencias con la tecnología Wi-Fi. Debido a esto, en la versión 1.2 del estándar Bluetooth por ejemplo se actualizó su especificación para que no existieran interferencias con la utilización simultánea de ambas tecnologías, además se necesita tener 40 000 k de velocidad.

Seguridad y fiabilidad

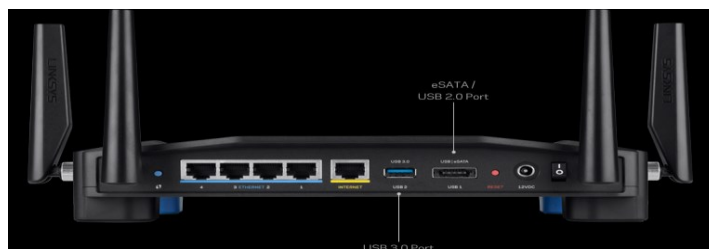
Uno de los problemas a los cuales se enfrenta actualmente la tecnología Wi-Fi es la progresiva saturación del espectro radioeléctrico, debido a la masificación de usuarios, esto afecta especialmente en las conexiones de larga distancia (mayor de 100 metros). En realidad Wi-Fi está diseñado para conectar ordenadores a la red a distancias reducidas, cualquier uso de mayor alcance está expuesto a un excesivo riesgo de interferencias.

Un muy elevado porcentaje de redes son instalados sin tener en consideración la seguridad convirtiendo así sus redes en redes abiertas (o completamente vulnerables ante el intento de acceder a ellas por terceras personas), sin proteger la información que por ellas circulan. De hecho, la configuración por defecto de muchos dispositivos Wi-Fi es muy insegura (Router, por ejemplo) dado que a partir del identificador del dispositivo se puede conocer la clave de éste; y por tanto acceder y controlar el dispositivo se puede conseguir en sólo unos segundos.

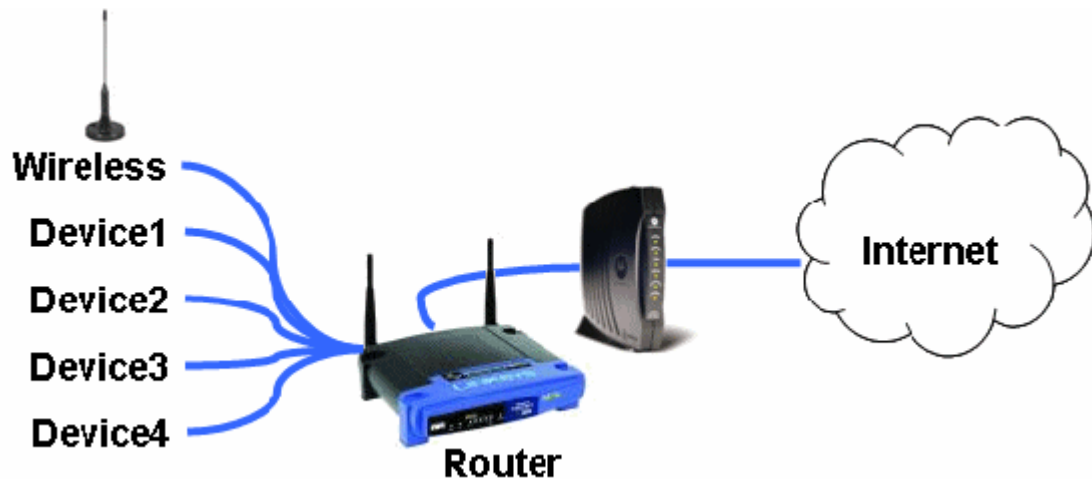
El acceso no autorizado a un dispositivo Wi-Fi es muy peligroso para el propietario por varios motivos. El más obvio es que pueden utilizar la conexión. Pero además, accediendo al Wi-Fi se puede monitorizar y registrar toda la información que se transmite a través de él (incluyendo información personal, contraseñas).

Dispositivos

Existen varios dispositivos Wi-Fi, los cuales se pueden dividir en dos grupos: Dispositivos de Distribución o Red, entre los que destacan los Router, puntos de acceso y Repetidores; y Dispositivos Terminales que en general son las tarjetas receptoras para conectar a la computadora personal, ya sean internas (tarjetas PCI) o bien USB.



Router wifi



Dispositivos de Distribución o Red.

Los puntos de acceso son dispositivos que generan un "set de servicio", que podría definirse como una "Red Wi-Fi" a la que se pueden conectar otros dispositivos. Los puntos de acceso permiten, en resumen, conectar dispositivos en forma inalámbrica a una red existente. Pueden agregarse más puntos de acceso a una red para generar redes de cobertura más amplia, o conectar antenas más grandes que amplifiquen la señal.

Los repetidores inalámbricos son equipos que se utilizan para extender la cobertura de una red inalámbrica, éstos se conectan a una red existente que tiene señal más débil y crean una señal limpia a la que se pueden conectar los equipos dentro de su alcance. Algunos de ellos funcionan también como punto de acceso.

Los Router inalámbricos son dispositivos compuestos, especialmente diseñados para redes pequeñas (hogar o pequeña oficina). Estos dispositivos incluyen, un Router (encargado de interconectar redes, por ejemplo, nuestra red del hogar con internet), un punto de acceso (explicado más arriba) y generalmente un switch que permite conectar algunos equipos vía cable (Ethernet y USB). Su tarea es tomar la conexión a internet, y brindar a través de ella acceso a todos los equipos que conectemos, sea por cable o en forma inalámbrica.

El wifi puede ser desactivado por un terminal del dispositivo.

Las tarjetas PCI para Wi-Fi se agregan (o vienen de fábrica) a los ordenadores de sobremesa. Hoy en día están perdiendo terreno debido a las tarjetas USB. Dentro de este grupo también pueden agregarse las tarjetas MiniPCI que vienen integradas en casi cualquier computador portátil disponible hoy en el mercado.

Las tarjetas PCMCIA son un modelo que se utilizó mucho en los primeros ordenadores portátiles, aunque están cayendo en desuso, debido a la integración de tarjeta inalámbricas internas en estos ordenadores. La mayor parte de estas tarjetas solo son capaces de llegar hasta la tecnología B de Wi-Fi, no permitiendo por tanto disfrutar de una velocidad de transmisión demasiado elevada.

Las tarjetas USB para Wi-Fi son el tipo de tarjeta más común que existe en las tiendas y más sencillo de conectar a un PC, ya sea de sobremesa o portátil, haciendo uso de todas las ventajas que tiene la tecnología USB. Hoy en día puede encontrarse incluso tarjetas USB con el estándar 802.11N (Wireless-N) que es el último estándar liberado para redes inalámbricas.

También existen impresoras, cámaras Web y otros periféricos que funcionan con la tecnología Wi-Fi, permitiendo un ahorro de mucho cableado en las instalaciones de redes y especialmente, gran movilidad.

Ventajas y desventajas

Las redes Wi-Fi poseen una serie de ventajas, entre las cuales podemos destacar:

Al ser redes inalámbricas, la comodidad que ofrecen es muy superior a las redes cableadas porque cualquiera que tenga acceso a la red puede conectarse desde distintos puntos dentro de un rango suficientemente amplio de espacio.

Una vez configuradas, las redes Wi-Fi permiten el acceso de múltiples ordenadores sin ningún problema ni gasto en infraestructura, ni gran cantidad de cables.

La Wi-Fi Alliance asegura que la compatibilidad entre dispositivos con la marca Wi-Fi es total, con lo que en cualquier parte del mundo podremos utilizar la tecnología Wi-Fi con una compatibilidad total.

Pero como red inalámbrica, la tecnología Wi-Fi presenta los problemas intrínsecos de cualquier tecnología inalámbrica. Algunos de ellos son:

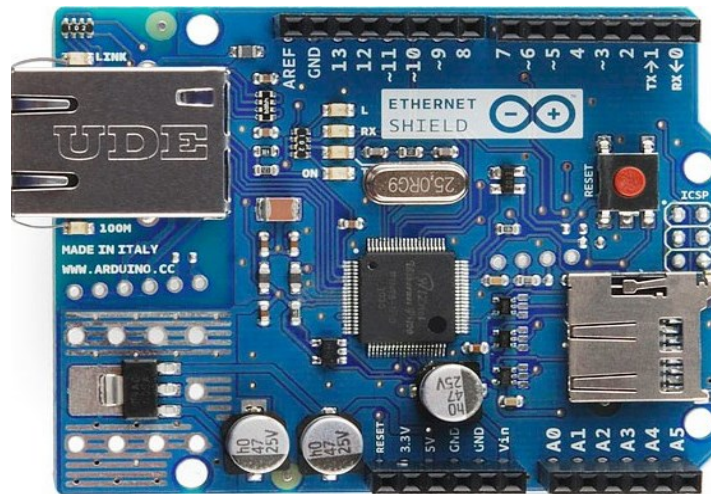
La desventaja fundamental de estas redes existe en el campo de la seguridad. Existen algunos programas capaces de capturar paquetes, trabajando con su tarjeta Wi-Fi en modo promiscuo, de forma que puedan calcular la contraseña de la red y de esta forma acceder a ella. Las claves de tipo WEP son relativamente fáciles de conseguir con este sistema.

La potencia de la conexión del Wi-Fi se verá afectada por los agente físicos que se encuentran a nuestro alrededor, tales como: arboles, paredes, arroyos, una montaña, etc. Dichos factores afectan la potencia de compartimiento de la conexión Wi-Fi con otros dispositivos.

5.3. SHIELD ETHERNET - HARDWARE ETHERNET EN ARDUINO

El Arduino ethernet shield nos da la capacidad de conectar un Arduino a una red ethernet. Es la parte física que implementa la pila de protocolos TCP/IP.

Está basada en el chip ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de soportar TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para leer y escribir los flujos de datos que pasan por el puerto ethernet. Permite escribir sketches que se conecten a internet usando esta shield.



Tarjeta Shield o Escudo Ethernet – Básica

El shield provee un conector ethernet estándar RJ45. La ethernet shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.

Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la ethernet shield. Estos pines no pueden ser usados para e/s genéricas.

El botón de reset en la shield resetea ambos, el W5100 y la placa Arduino.

La shield contiene varios LEDs para información:

- ❖ **ON:** 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
- ❖ **100M:** indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- ❖ **RX:** parpadea cuando el shield recibe datos
- ❖ **TX:** parpadea cuando el shield envía datos

El jumper soldado marcado como “INT” puede ser conectado para permitir a la placa Arduino recibir notificaciones de eventos por interrupción desde el W5100, pero esto no está soportado por la librería Ethernet. El jumper conecta el pin INT del W5100 al pin digital 2 de Arduino.

El slot SD en la shield usa la librería <http://arduino.cc/en/Reference/SD> para manejarlo. El propio chip W5100 incluye el manejo de tarjetas SD.

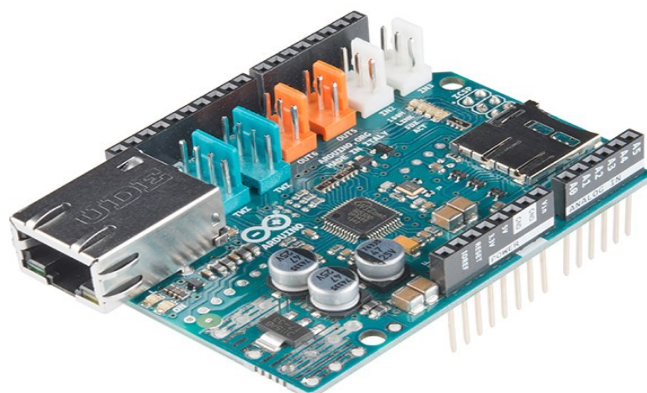
Para usar la Ethernet Shield solamente es necesario montarla sobre la placa Arduino. Para cargar los sketches a la placa con el shield, conectarla al ordenador mediante el cable USB como se hace normalmente. Luego conectar el puerto Ethernet a un ordenador, a un switch o a un router utilizando un cable ethernet standard (**CAT5 o CAT6** con conectores RJ45). La conexión al ordenador puede requerir el uso de un cable cruzado (aunque muchos ordenadores actuales, pueden hacer el cruce de forma interna).

CARACTERÍSTICAS DE LA PLACA.

- ❖ Opera a 5V suministrados desde la placa de Arduino
- ❖ El controlador ethernet es el W5100 con 16K de buffer interno. No consume memoria.
- ❖ El shield se comunica con el microcontrolador por el bus SPI, por lo tanto para usarlo siempre debemos incluir la librería SPI.h: <http://arduino.cc/en/Reference/SPI>.
- ❖ Soporta hasta 4 conexiones simultáneas
- ❖ Usar la librería Ethernet para manejar el shield: <http://arduino.cc/en/Reference/Ethernet>
- ❖ El shield dispone de un lector de tarjetas micro-SD que puede ser usado para guardar ficheros y servirlos sobre la red. Para ello es necesaria la librería SD: <http://arduino.cc/en/Reference/SD>
- ❖ Al trabajar con la SD, el pin 4 es usado como SS.

Arduino UNO se comunica con W5100 y la tarjeta SD usando el bus SPI a través del conector ICSP. Por este motivo los pines 10, 11, 12 y 13 en el UNO y los 50, 51, 52 y 53 en el Mega no podrán usarse. En ambas placas los pines 10 y 4 se usan para seleccionar el W5100 y la tarjeta SD. El Ethernet y el SD no pueden trabajar simultáneamente y se debe tener cuidado al usar ambos de forma conjunta.

Arduino Ethernet Shield 2



Arduino Ethernet Shield es una placa que aparece en la web de arduino.cc como retirado, pero sigue estando disponible como clones o versiones derivadas.

Por otra parte arduino.org sacó el Arduino Ethernet Shield 2 con el nuevo Wiznet 5500.

<http://www.arduino.org/products/shields/arduino-ethernet-shield-2>

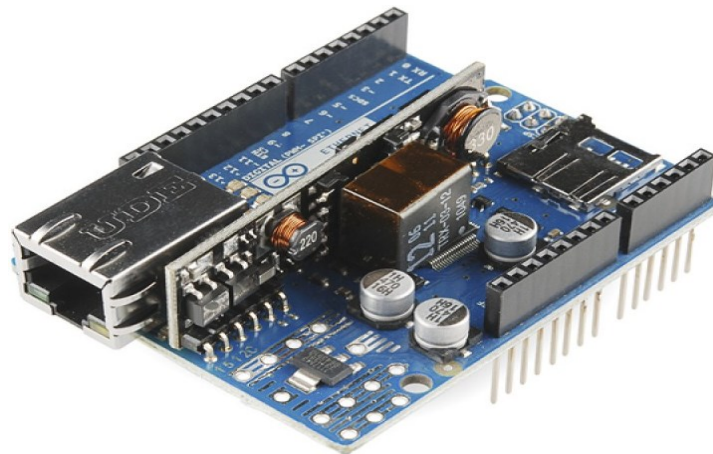
Este Shield usa la librería ethernet2: <http://www.arduino.org/learning/reference/Ethernet-two-Library>

Data sheet de W5500:

https://www.sos.sk/productdata/15/26/12/152612/W5500_datasheet_v1.0.2_1.pdf

Mejoras de W5500: <https://feilipu.me/2014/11/16/wiznet-w5500-ioshield-a/>

Otras variantes Arduino Ethernet SHIELD Rev3 WITH PoE (from arduino.org)



Power over Ethernet

El Ethernet shield es compatible con PoE gracias a un módulo adicional que extrae la energía eléctrica del cable ethernet, anteriormente inyectada desde el switch.

Las características del módulo PoE:

- ❖ IEEE802.3af compliant
- ❖ Low output ripple and noise (100mVpp)
- ❖ Input voltage range 36V to 57V
- ❖ Overload and short-circuit protection
- ❖ 9V Output
- ❖ High efficiency DC/DC converter: typ 75% @ 50% load
- ❖ 1500V isolation (input to output).

Data sheet: <http://arduino.cc/en/uploads/Main/PoE-datasheet.pdf>

Más información sobre el **Power Over Ethernet (POE)**:

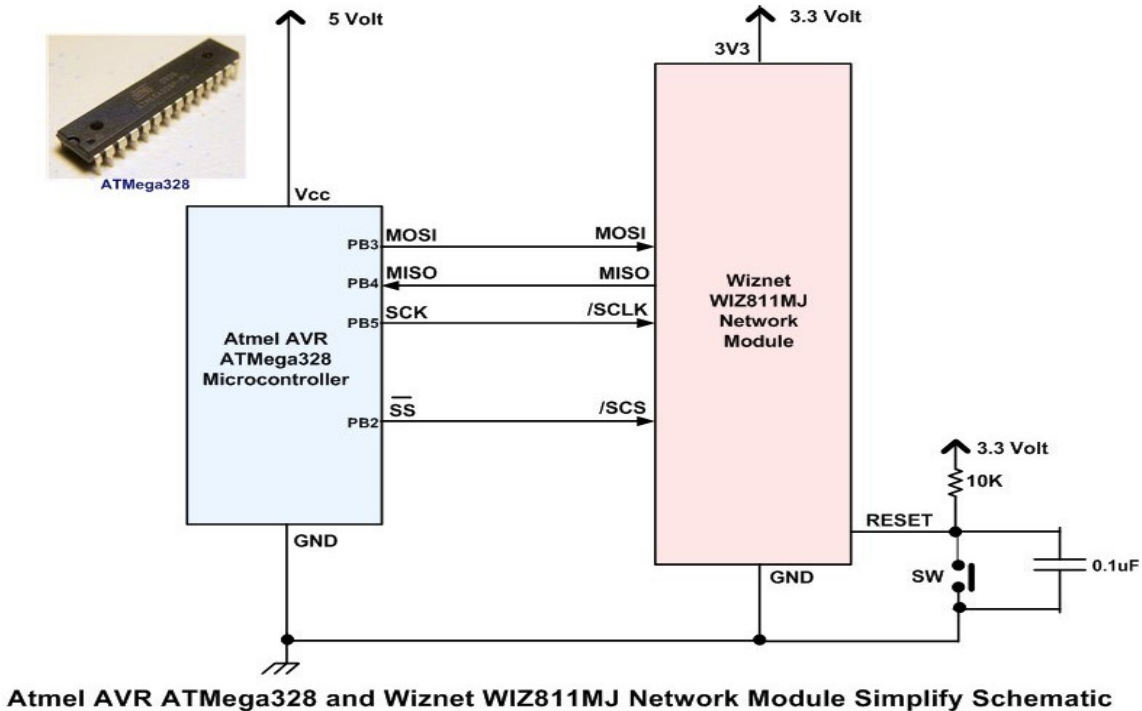
- ❖ http://es.wikipedia.org/wiki/Power_over_Ethernet
- ❖ http://en.wikipedia.org/wiki/Power_over_Ethernet

WIZnet W5100

El integrado W5100 se conecta al arduino mediante SPI.

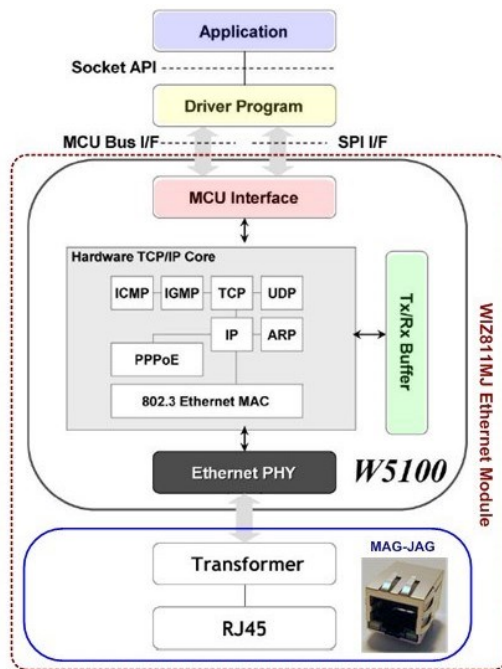
El chip Wiznet W5100 viene con tres métodos para controlar sus registros internos; los dos primeros son utilizar el bus paralelo directo o indirecto, el último es usar una conocida transferencia de datos en serie conocida como SPI (interfaz de periféricos seriales), para este prototipo se utilizó la interfaz SPI para controlar el Wiznet.

La conexión SPI básica entre el módulo de red WIZNET WIZ811MJ y el microcontrolador Atmel AVR ATmega328 se muestra en la siguiente imagen.



El Wiznet W5100 Hardwired TCP / IP Protocol Chip

Básicamente, Wiznet W5100 implementa un completo estándar IEEE 802.3 (capa de enlace físico y de datos de Ethernet) y una potente pila de TCP / IP dentro del chip; esto hace que el chip Wiznet W5100 sea la opción adecuada para integrar el sistema integrado en Internet. La programación del chip Wiznet W5100 también es más sencilla, ya que solo se necesita escribir y leer desde y hacia los registros internos del W5100 para utilizar las características incorporadas del protocolo TCP / IP.



Features

- Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- 10BaseT/100BaseTX Ethernet PHY embedded
- Support Auto Negotiation (Full-duplex and half duplex)
- Support Auto MDI/MDIX
- Support ADSL connection (with support PPPoE Protocol with PAP/CHAP Authentication mode)
- Supports 4 independent sockets simultaneously
- Not support IP Fragmentation
- Internal 16Kbytes Memory for Tx/Rx Buffers
- 0.18 μ m CMOS technology
- 3.3V operation with 5V I/O signal tolerance
- Small 80 Pin LQFP Package
- Lead-Free Package
- Support Serial Peripheral Interface(SPI MODE 0, 3)
- Multi-function LED outputs (TX, RX, Full/Half duplex, Collision, Link, Speed)

The Wiznet W5100 Hardwire TCP/IP Stack Chip and WIZ811MJ Ethernet Module

El Wiznet W5100 actuará como un dispositivo esclavo SPI controlado por microcontrolador ATmega328 como el SPI Maestro. Necesita el protocolo SPI al menos cuatro señales, MOSI (Master Out Serial In), MISO (Master In Serial Out), SCK (señal de reloj proporcionada por el maestro) y CS (the SPI slave chip select). El chip W5100 también proporciona el pin de interrupción.

¿Soporta Arduino IPv6?

El shield oficial de arduino basado en el WizNet 5100 implementa la pila de protocolos IPv4, por lo que está shield no puede ser utilizada para implementar la pila IPv6.

Para implementar la pila IPv6, es necesario usar un shield basado en hardware que permita la gestión de las funciones de las capas de IP y ethernet. Shields basados en el chip MicroChip ENC28J60 son adecuados para la implementación de IPv6. Esto requiere la implementación de una gestión de los estados TCP, resultando un código de arduino más complejo.

Datasheet Chip ethernet IPv6:

- ❖ <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en022889>
- ❖ <http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>

Más información sobre IPv6:

- ❖ <https://sites.google.com/site/ghoelzl/ipv6>

IPv6WebServer:

- ❖ https://sites.google.com/site/ghoelzl/ipv6ethershield/ipv6_http_server

Y aun más...

❖ <http://www.tweaking4all.com/hardware/arduino/arduino-enc28j60-ethernet/>

Y otro ejemplo:

❖ http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Ethernet-Module-ENC28J60-Arduino.pdf

Librerías:

❖ https://github.com/ntruchsess/arduino_uip

❖ <https://github.com/jcw/ethercard>

❖ <https://github.com/ghoelzl/IPv6EtherShield>

❖ <https://sites.google.com/site/ghoelzl/ipv6ethershield>

OTRAS PLACAS DE RED (WIFI) QUE SE PUEDEN UTILIZAR PARA LA ELABORACIÓN DE ESTE PROYECTO.

NodeMCU: La placa de desarrollo NodeMCU está basada en el popular chip que revolucionó el Wifi en sistemas embebidos: Estamos hablando del ESP8266. Con este sencillo módulo podrás realizar el prototipo de cualquier sistema para el OIT (Internet of things) en cuestión de horas. El concepto es muy similar al de arduino, es decir, un microcontrolador conectado a través de un puente USB-Serial que interactúa con un software en la PC.

El IC - ESP8266 en el NodeMCU es más que un simple circuito integrado para WiFi. Se trata de un SoC (System on Chip) que integra en una sola pieza de silicio un procesador de aplicaciones con la electrónica necesaria para la comunicación por (Wifi). Esta placa permite aprovechar el procesador que está dentro del ESP8266 y realizar software que corre en él, no solamente usarlo como un puente entre un microcontrolador y la red wifi.

Esta placa viene cargada con el firmware NodeMCU, sin embargo, también puede usarse como una excelente plataforma para desarrollar, evaluar y experimentar otros firmware para el ESP8266. Internet de las cosas. Los ordenadores actuales y, por tanto, internet son prácticamente dependientes de los seres humanos para recabar información. Una mayoría de los casi 50 petabytes (un petabyte son 1024 terabytes) de datos disponibles en internet fueron inicialmente creados por humanos, a base de teclear, presionar un botón, tomar una imagen digital o escanear un código de barras

Los ordenadores saben más sobre ideas que sobre cosas. Si tuviéramos ordenadores que supieran todo lo que tuvieran que saber sobre las "cosas", mediante el uso de datos que ellos mismos pudieran recoger sin nuestra ayuda, nosotros podríamos monitorizar, contar y localizar todo a nuestro alrededor, de esta manera se reducirían increíblemente gastos, pérdidas y costes. Sabríamos cuando reemplazar, reparar o recuperar lo que fuera, así como conocer si su funcionamiento estuviera siendo correcto. El internet de las cosas tiene el potencial para cambiar el mundo tal y como hizo la revolución digital hace unas décadas.

Vale mencionar que existe diversidad de tarjetas electrónicas (cerebros) que se puede utilizar para la realización de proyectos IoT.

Arduino. Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar.

Arduino puede medir entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos.

El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Pressing).

Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador. Las placas se pueden ensamblar a mano o encargarse pre-ensambladas; el software se puede descargar gratuitamente.

Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

Microcontrolador: Un microcontrolador (abreviado UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

Tipos de Sensores

Sensor: es un objeto capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

Las variables de instrumentación pueden ser, por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Sensores activos: Este tipo de sensores inyectan luz, microondas o sonido en el medio ambiente y detectan si existe algún cambio en él.

Sensores pasivos. Muchas alarmas y sensores utilizados usan la detección de ondas infrarrojas. Estos sensores son conocidos como [PIR \(pasivos infrarrojos\)](#). Para que uno de estos sensores detecte a los seres humanos se debe de ajustar la sensibilidad del sensor para que detecte la temperatura del cuerpo humano.

Actuador: Un actuador es un dispositivo capaz de transformar energía hidráulica neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control, como por ejemplo una válvula. Son los elementos que influyen directamente en la señal de salida del automatismo, modificando su magnitud según las instrucciones que reciben de la unidad de control.

5.4. GPS O SISTEMA DE POSICIONAMIENTO GLOBAL

El **Sistema de Posicionamiento Global**, más conocido por sus siglas en inglés, **GPS** (siglas de **Global Positioning System**), es un sistema que permite determinar en toda la Tierra la posición de un objeto (una persona, un vehículo) con una precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los Estados Unidos. Para determinar las posiciones en el globo, el sistema GPS se sirve de 24 satélites y utiliza la trilateración.

Trilateración: La trilateración es un método [matemático](#) para determinar las posiciones relativas de objetos usando la [geometría](#) de [triángulos](#) de forma análoga a la [triangulación](#). A diferencia de ésta, que usa medidas de [ángulo](#) (junto con al menos una [distancia](#) conocida para calcular la localización del sujeto), la trilateración usa las localizaciones conocidas de dos o más puntos de referencia, y la distancia medida entre el sujeto y cada punto de referencia. Para determinar de forma única y precisa la localización relativa de un punto en un plano [bidimensional](#) usando sólo trilateración, se necesitan generalmente al menos 3 puntos de referencia

El **GPS** funciona mediante una red de 24 satélites en órbita sobre el planeta Tierra, a 20 200 km de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante el método de trilateración inversa, el cual se basa en determinar la distancia de cada satélite al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene las posiciones absolutas o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que lleva a bordo cada uno de los satélites.

La latitud es la distancia que hay desde un punto de la superficie terrestre al ecuador, contada por los grados de su meridiano. Esta se mide en grados, entre 0° y 90° y puede ser representada a través de dos maneras: indicando a qué hemisferio pertenece la coordenada, o en su defecto añadiendo valores, positivos cuando se trata del norte y negativos cuando se trata del sur.

La longitud es un concepto métrico definible para entidades geométricas sobre la que se ha definido una distancia. Más concretamente dado un segmento, curva o línea fina, se puede definir su longitud a partir de la noción de distancia. Sin embargo, no debe confundirse longitud con distancia, ya que para una curva general (no para un segmento recto) la distancia entre dos puntos cualesquiera de la misma es siempre inferior a la longitud de la curva comprendida entre esos dos puntos. Igualmente, la noción matemática de longitud se puede identificar con la una magnitud física que determinada por la distancia física.

5.5. TECNOLOGÍA MÓVIL O GSM.

Tecnología GPRS/SMS.

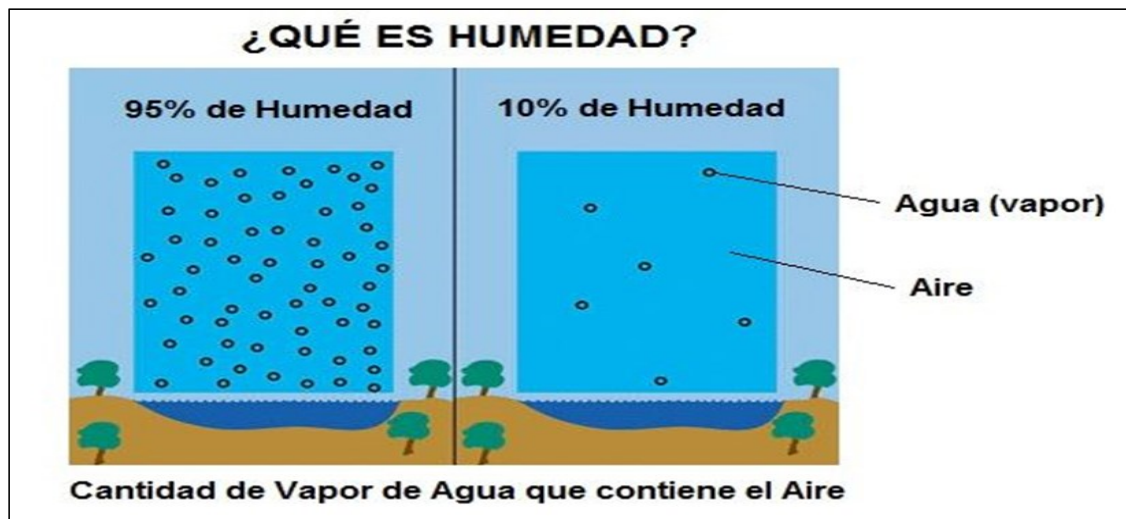
Servicio General de Paquetes Vía Radio

La tecnología GPRS se puede utilizar para servicios como el acceso mediante el Protocolo de Aplicaciones Inalámbrico (WAP), el SMS y MMS, acceso a Internet y correo electrónico.

La **temperatura** es una magnitud referida a las nociones comunes de calor medible mediante un termómetro.

Se denomina **humedad** al agua que impregna un cuerpo o al vapor presente en la atmósfera

La humedad. La mayor parte del aire que te rodea tiene agua en forma de un gas llamado vapor de agua. La humedad es la cantidad de vapor de agua que contiene ese aire, por eso también se llama "**Humedad del Aire**". No podemos ver la humedad porque el vapor de agua es transparente, pero si la podemos sentir.



Como ya sabemos, dos terceras partes de la superficie de la tierra están cubiertas por agua: océanos, lagos y ríos, de las cuales se desprende el vapor de agua.

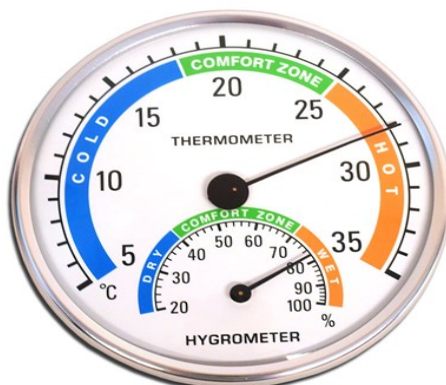
Esto quiere decir que el mismo aire puede estar saturado de vapor de agua a una temperatura, pero si aumentamos su temperatura, podremos introducir en él más vapor.

A más temperatura, tenemos menos humedad relativa pero la cantidad de agua es la misma. Eso es porque al aumentar la temperatura podemos meter más agua en ese aire. La cantidad que a 100° ya no podía entrar más agua en ese aire (al 100%) si le subimos a 20°, como ahora entraría más agua, esa misma cantidad de agua ya no sería el 100% de la cantidad total que podría tener ese aire.

Tres tipos de humedad: humedad relativa, humedad absoluta y humedad específica. En cada caso el aparato que la mide es diferente.

Humedad Relativa: En la atmósfera siempre hay una cantidad de agua en forma de vapor, aunque tú no la veas. La cantidad de agua que contiene el aire se llama humedad y se suele medir como humedad relativa.

Si el valor de la humedad es del 50%, quiere decir que en ese momento el aire contiene un 50% de vapor de agua respecto al máximo que podría contener, que sería el 100%. La humedad relativa se mide con el higrómetro que expresa la humedad relativa del aire en porcentaje.



Humedad Específica: La humedad específica, se refiere a la cantidad de humedad en peso, que se requiere para saturar un kilogramo de aire seco, es decir, los gramos de vapor de agua contenidos en 1 kg de aire seco. Se expresa en g/kg. La humedad específica es muy similar a la humedad absoluta, excepto que esta última, está basada en gramos por metro cúbico, y la humedad específica, está basada en gramos de humedad por kilogramo de aire seco.

Humedad Absoluta: La humedad absoluta se refiere al peso del vapor de agua por unidad de volumen. Se llama humedad absoluta al peso en gramos del vapor de agua contenido en 1 metro cúbico de aire. Se expresa en gm^3 (con valores medios de 10 a 12 g/m^3 , pero puede llegar hasta 40 g/m^3). La humedad absoluta y específica se suelen utilizar, además de en [meteorología](#), para medir por ejemplo la humedad de la madera, en los materiales de construcción, en los cereales y los granos de café o también del papel, el material textil y el algodón.

Estas humedades se miden con unos instrumentos llamados "**medidores de Humedad**".



Al calentarse el aire, pesa menos y asciende, pero también a medida que asciende se va enfriando. Este enfriamiento al subir provoca que el vapor de agua se condense y forme minúsculas gotas de agua que originan las nubes.

Si el vapor de agua se condensa al amanecer, cuando bajan las temperaturas, se deposita sobre los objetos en forma de rocío. En ocasiones, el vapor de agua se condensa en una nube baja de pequeñísimas gotas de agua y forma la niebla. Si la temperatura es menor de 0°C, se hiela y forma la escarcha.

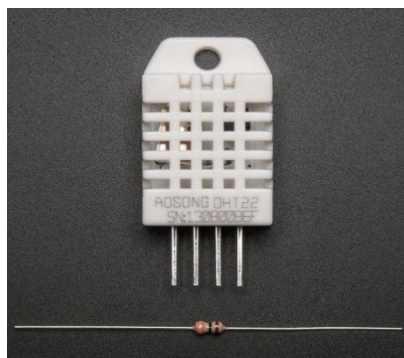
Detector de movimiento Un detector de movimiento, o sensor de presencia, es un dispositivo electrónico equipado de [sensores](#) que responden a un [movimiento](#) físico. Se encuentran generalmente en sistemas de seguridad o en [circuitos cerrados](#).

Sensor infrarrojo pasivo o PIR

El sistema puede estar compuesto, simplemente, por una [cámara](#) de vigilancia conectada a un [ordenador](#) que se encarga de generar una señal de alarma o poner el sistema en estado de alerta cuando algo se mueve delante de la cámara. Aunque, para mejorar el sistema se suele utilizar más de una Arquitectura utilizada arduino uno/ mega

SENSOR DE TEMPERATURA Y HUMEDAD. La captación de los valores de temperatura y humedad en ICRA se ha realizado con un único accesorio, el sensor DHT22. Los sensores DHT11 o DHT22 son unos pequeños dispositivos que permiten medir la temperatura y la humedad. A diferencia de otros sensores, éstos han de conectarse a pines digitales, ya que la señal de salida es digital al integrar un pequeño microcontrolador para hacer el tratamiento de la señal.

Los sensores de gama DHT se componen de un sensor capacitivo para medir la humedad y de un termistor, ambos sensores ya calibrados por lo que no es necesario añadir ningún circuito de tratamiento de señal. Esto es, sin duda, una ventaja ya que simplifican las conexiones a realizar en la placa. Además, como los DHT han sido calibrados en laboratorios, presentan una gran fiabilidad.



Ambos sensores funcionan con ciclos de operación de duración determinada (1s en el caso del DHT11 y 2s en el caso del DHT22). En este tiempo, el microcontrolador externo (Arduino), y el microcontrolador que lleva integrado el sensor, se hablan entre sí de la siguiente manera:

- ❖ El microcontrolador (Arduino) inicia la comunicación.
- ❖ El sensor responde estableciendo un nivel bajo de 80us y un nivel alto de 80us.
- ❖ El sensor envía 5 bytes. Se produce el handshaking.

La siguiente tabla muestra las principales características de los dos modelos de sensores.

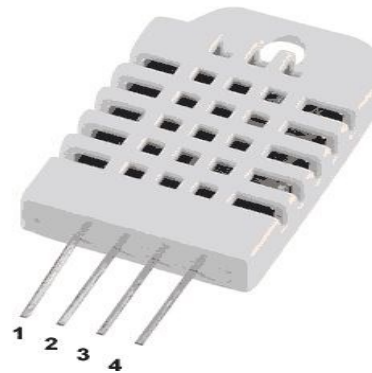
Parámetro	DHT11	DHT22
Alimentación	$3Vdc \leq Vcc \leq 5Vdc$	$3.3Vdc \leq Vcc \leq 6Vdc$
Señal de Salida	Digital	Digital
Rango de medida de Temperatura	De 0 a 50 °C	De -40°C a 80 °C
Precisión Temperatura	± 2 °C	$<\pm 0.5$ °C
Resolución Temperatura	0.1°C	0.1°C
Rango de medida Humedad	De 20% a 90% RH	De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

Tabla: Características sensores DHT

En lo que se refiere al pinout o terminales, los pines del DHT22 siguen el mismo orden. De izquierda a derecha se tiene:

1. VCC: Alimentación del sensor
2. Señal: Información de temperatura y humedad
3. NC: No Conexión 4. GND: Masa del circuito

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



5.6. BASE DE DATOS.

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. Las bases de datos tradicionales se organizan por campos, registros y archivos.

Tipos y función de los gestores de bases de datos.

Los Sistemas Gestores de Bases de Datos (SGBD, por sus siglas en inglés), también conocidos como sistemas manejadores de bases de datos o DBMS (DataBase Management System), son un conjunto de **programas** que manejan todo **acceso a la base de datos**, con el objetivo de servir de **interfaz** entre ésta, el usuario y las aplicaciones utilizadas.

Gracias a este sistema de [software específico](#) el usuario puede **gestionar la base de datos** (almacenar, modificar y acceder a la información contenida en ésta) mediante el uso de distintas herramientas para su análisis, con las que puede realizar consultas y generar informes.

Además de gestionar los datos y mantener su consistencia, su utilización supone numerosas ventajas a la hora de construir y definir la base de datos a diferentes niveles de abstracción para distintas aplicaciones, pues facilita los procesos y también su mantenimiento.

Las funciones de un Sistema Gestor de Bases de Datos (SGBD)

La ejecución de las operaciones sobre la base de datos para luego proporcionarlos al usuario en función de su requerimiento se realiza de un modo eficiente y seguro. Sus características de un [SGDB](#) posibilitan el cumplimiento de una serie de funciones, que pueden agruparse de la siguiente manera:

1. Definición de los datos: El SGBD ha de poder definir todos los objetos de la base de datos partiendo de definiciones en versión fuente para convertirlas en la versión objeto.

2. Manipulación de los datos: El SGBD responde a las solicitudes del usuario para realizar operaciones de supresión, actualización, extracción, entre otras gestiones. El manejo de los datos ha de realizarse de forma rápida, según las peticiones realizadas por los usuarios, y permitir la modificación del esquema de la base de datos gracias a su independencia.

3. Seguridad e integridad de los datos: Además de registrar el uso de las bases de datos, ante cualquier petición, también aplicará las medidas de seguridad e integridad de los datos (adopta medidas garantizar su validez) previamente definidas. Un SGBD debe garantizar su seguridad frente a ataques o simplemente impedir su acceso a usuarios no autorizados por cualquier razón.

4. Recuperación y restauración de los datos: La recuperación y restauración de los datos ante un posible fallo es otra de las principales funciones de un SGBD. Su aplicación se realizará a través de un Plan de recuperación y restauración de los datos que sirva de respaldo.

Tipos de gestores de bases de datos

La [tipología de los SGBD](#) es muy variada, en función del criterio que utilicemos para su clasificación. Agruparlos atendiendo al modelo de datos, número de usuarios o de sitios suele ser lo más habitual, si bien la tipología puede obedecer a otras muchas pautas, según convenga desde un determinado enfoque práctico:

Si atendemos al [modelo de datos](#), los gestores de bases de datos pueden ser:

- Relacionales
- En Red
- Jerárquicos
- Orientados a objetos

MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL, y todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones.

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compete con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos.

5.7. DESARROLLO DE APLICACIÓN MÓVIL

El diseño y desarrollo de la aplicación de monitorización remota se ha implementado bajo el sistema operativo Android.

Android es un sistema operativo móvil que se basa en una versión modificada de Linux. En 2005, como parte de su estrategia para entrar en el mundo móvil, Google compró Android y se hizo cargo de su trabajo de desarrollo (así como de su equipo).

Google quería que Android fuera abierto y libre; de ahí que la mayoría del código Android se pusiera disponible bajo licencia Apache de código abierto, lo que significa que cualquiera que quiera utilizar Android puede hacerlo al descargar su código fuente completo.

La principal ventaja de adoptar Android es que ofrece un enfoque unificado para el desarrollo de aplicaciones. Los desarrolladores solamente necesitan desarrollar para Android, y sus aplicaciones deberían poderse ejecutar en numerosos dispositivos, siempre y cuando los dispositivos utilicen este sistema operativo.

El sistema operativo se divide en cinco secciones, de cuatro niveles principales.

Kernel de Linux: Contiene los controladores de dispositivo de bajo nivel para los diversos componentes de hardware.

Librerías: Contienen todo el código que proporciona las principales características de un sistema operativo Android (SQLite para soporte de base de datos, Web Kit proporciona funcionalidades para navegación Web).

Tiempo de ejecución Android: En el mismo nivel que las librerías, permite a los desarrolladores escribir aplicaciones Android al utilizar el lenguaje de **programación Java**. También incluye la máquina virtual Dalvik, que permite que toda aplicación Android se ejecute en su propio proceso.

Arquitectura de software de aplicación: Presenta las diferentes posibilidades de uso del sistema operativo Android para los desarrolladores de aplicaciones, de modo que pueden hacer uso de ellas en sus aplicaciones.

Aplicaciones: En su capa superior se encuentran las aplicaciones que se distribuyen con el dispositivo Android, al igual que aplicaciones que se descargan e instalan desde Google Play.

Para el desarrollo de aplicaciones es fundamental emplear el Android SDK, el cual incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales.

Una vez se procede a desarrollar una aplicación otro aspecto a considerar es la versión de la API a emplear. La plataforma Android ofrece una API (Interfaz de programación de aplicaciones) que las aplicaciones pueden utilizar para interactuar con la base del sistema Android. Esta API se compone de diversos elementos, tales como un conjunto de paquetes, clases, atributos XML, Intents, permisos que las aplicaciones pueden solicitar.

En cada versión sucesiva de la plataforma Android se incluyen cambios y actualizaciones de la API. Estos cambios están diseñados de manera que cada nueva API sigue siendo compatible con las versiones anteriores. Es decir, la mayoría de los cambios que se realizan en la API son aditivos, introduciendo nuevas funcionalidades. Al actualizar, ciertas partes de la API quedan remplazadas por las nuevas, quedan obsoletas, pero, aun así, no se borran, por lo que las aplicaciones existentes pueden seguir.

Versiones de sistemas operativo android.

Versión	Nombre en código	Fecha de distribución	API level	Cuota(3 de noviembre, 2014)
6.0	Marshmallow	TBA	23	TBA
5.1	Lollipop	06 de abril de 2015	22	0,7%
5.0	Lollipop	3 de noviembre de 2014	21	9,0%
4.4	Kit Kat	31 de octubre de 2013	19	39,8%
4.3	Jelly Bean	24 de julio de 2013	18	5,5%
4.4	Jelly Bean	13 de noviembre de 2012	17	18,1%
4.1	Jelly Bean	9 de julio de 2012	16	15,6% 4
4.0	IceCreamSandwich	16 de diciembre de 2011	14	5,3% 2
2.3	Gingerbread	9 de febrero de 2011	10	5,7%
2.2	Froyo	20 de mayo de 2010	8	0,3%

Tabla. Versiones de la API de Android

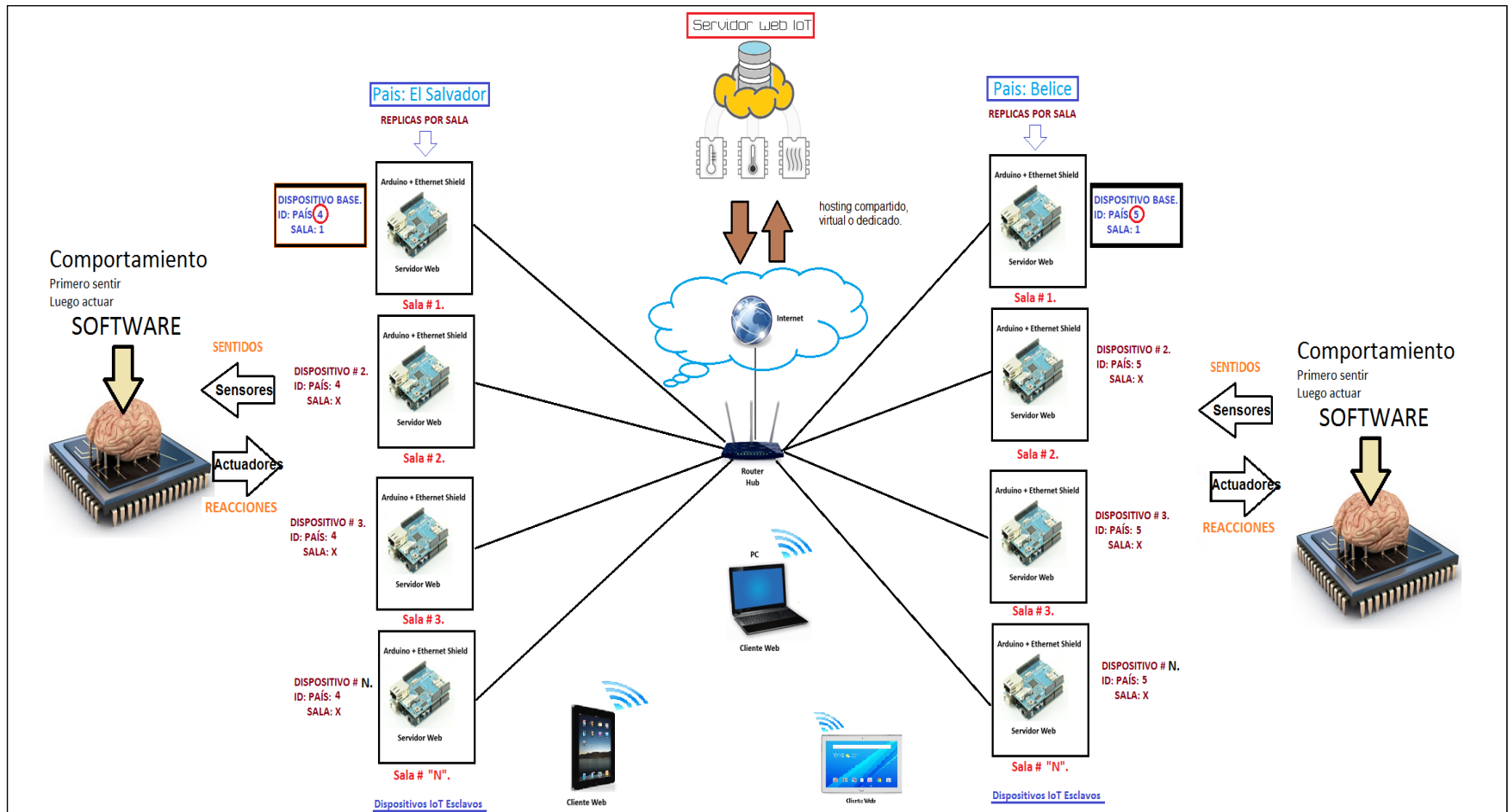
Existen una serie de elementos clave que resultan imprescindibles de conocer y utilizar para desarrollar aplicaciones en Android.

View: Las vistas son los elementos que componen la interfaz de usuario de una aplicación. Son, por ejemplo, un botón, una entrada de texto, Todas las vistas van a ser objetos descendientes de la clase View y por tanto, pueden ser definidos utilizando código Java. Sin embargo, lo habitual va a ser definir las vistas utilizando un fichero XML y dejar que el sistema cree los objetos automáticamente a partir de este fichero. Esta forma de trabajar es muy similar a la definición de una página web utilizando código HTML.

Layout: Un Layout es un conjunto de vistas agrupadas de una determinada forma. Se dispone de diferentes tipos de Layouts para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los Layouts también son objetos descendientes de la clase View, y al igual que las vistas, los Layouts pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML.

Activity: Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación. En Android cada uno de estos elementos, o pantallas, se conoce como actividad y su función principal es la creación del interfaz de usuario. Una aplicación suele necesitar varias actividades para crear el interfaz de usuario, siendo independientes entre sí, aunque todas trabajarán para un objetivo común. Toda actividad ha de pertenecer a una clase descendiente de Activity.

ESQUEMÁTICO DE CONEXIÓN DE DISPOSITIVOS IOT A NIVEL DE AMERICA CENTRAL



6. METODOLOGÍA DE LA INVESTIGACIÓN

Para este prototipo se desarrolló un dispositivo IoT, el cual contiene una serie de sensores para leer los datos de magnitudes físicas preseleccionadas tales como la Temperatura, Humedad, Movimiento, entre otros.

Se pretendió en un inicio desarrollar todo el sistema para la lectura de parámetros provenientes de los sensores antes mencionados según se acordó con la empresa COCESNA, pero a la vez se buscó dejarlo flexible a cambios de programación para añadir o sustituir un sensor por otro y así escalarlo a cualquier nivel o requerimiento según lo demande una determina aplicación.

Se realizó un sistema IoT para la monitorización de los datos proporcionados por los sensores antes mencionados a través de internet en un portal web que contendrá un dashboard para la visualización de la información gráficamente en tiempo real.

Para el desarrollo del proyecto se aplicó la Gestión de proyectos AGILE

Es una metodología de gestión de proyectos ampliamente usada en el sector IT y proyectos de organización empresariales, que tiene como principal virtud la flexibilidad y capacidad de modificar el producto a lo largo del proyecto, ya que estos se van usando al mismo tiempo que se desarrollan.

Esta metodología se basa en dividir el proyecto en fases (sprints), el resultado de las cuales es un producto con una serie de funcionalidades que ya permiten que este sea usado. Estas fases se terminan hasta haber conseguido el total de las funcionalidades definidas para el producto. De forma esquemática estas fases se componen de lo siguiente:

Inicio: Se escoge del total de objetivos del producto aquellos que serán implementados en el sprint, debiendo ser capaces de generar un producto funcional.

En base a estos objetivos se define la duración del sprint (entre una semana y un mes), y las tareas que lo componen.

Desarrollo del sprint: El equipo del proyecto planifica y ejecuta las tareas, las cuales se van supervisando en reuniones diarias donde se miran las tareas ejecutadas, en curso, y pendientes, así como posibles impedimentos y restricciones.

Cierre: Al final del sprint se revisa que se hayan completado las tareas y objetivos definidos al inicio mediante la presentación de un producto funcional. El ensayo de este producto por parte del cliente puede hacer variar los objetivos del proyecto o su prioridad.

La posibilidad de tener un producto funcional y utilizable al final de cada sprint permite ir ajustando los objetivos del proyecto, y por tanto asegurar mejor que el producto final cumplirá con las expectativas del usuario. A parte, la comercialización de estos Sprint permite avanzar los ingresos generados por el proyecto, mejorando su rentabilidad.

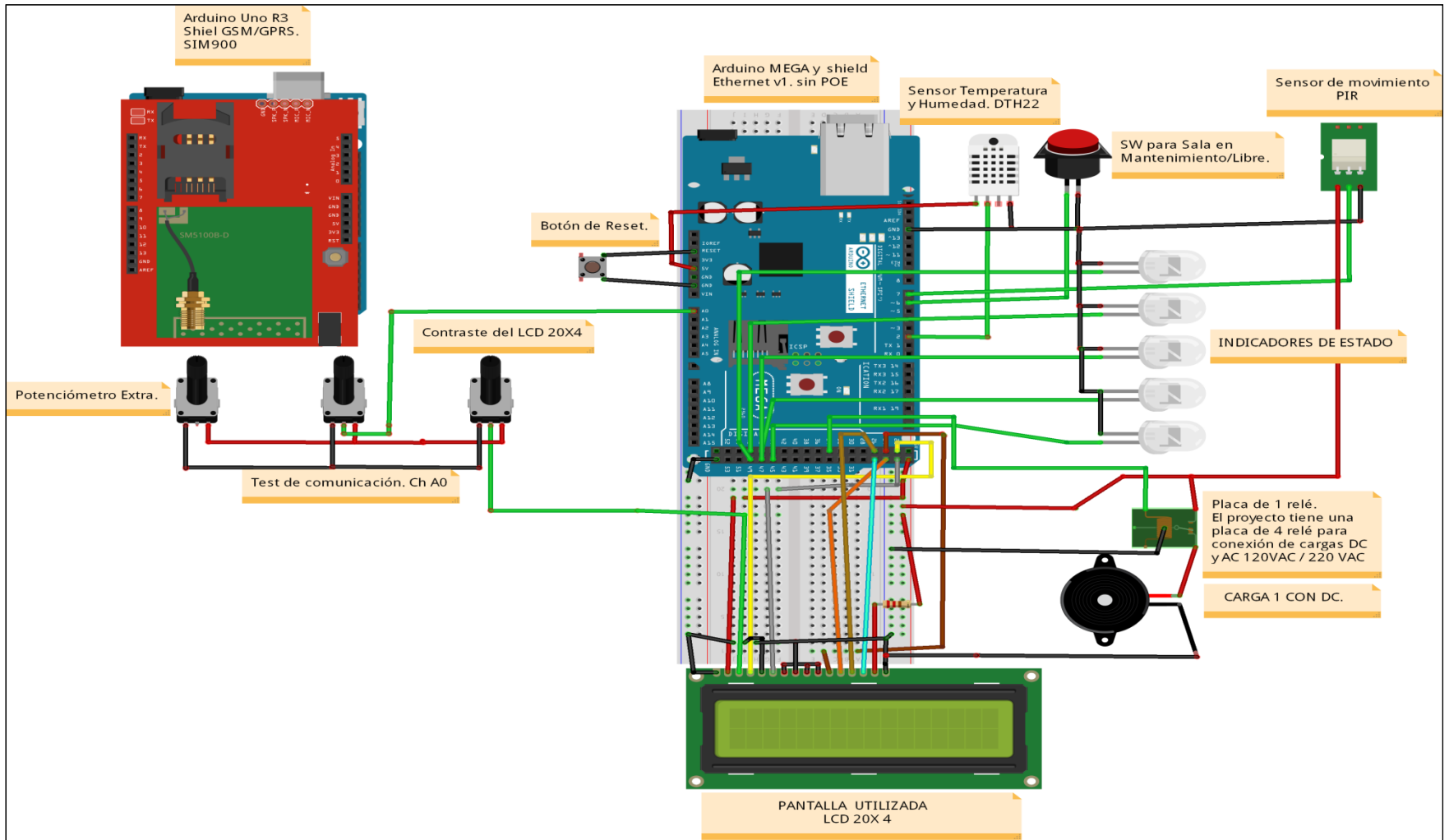
Matriz operacional de la metodología

OBJETIVOS ESPECÍFICOS	ACTIVIDADES A EJECUTAR	RESULTADOS ESPERADOS	MATERIALES
Realizar laplanificación de la investigación.	Investigación bibliográfica. Redacción del Anteproyecto. Presentación de anteproyecto a sede central. Presentación de anteproyecto a COCESNA.	Anteproyecto aprobado por sede central. Anteproyecto aprobado por COCESNA	Computadora.
Diseñar y Crear el sistema informático web aplicando un dashboard para visualizar los datos proporcionados por los captadores.	Diseñar y crear la aplicación web para dispositivo IoT. Desarrollo del dashboard para la aplicación web. Diseñar el dispositivo IoT del sistema. Compra de materiales del proyecto (COCESNA).	Sistema Informático Web. Dashboard Base de datos.	Computadora. Listado de Materiales del proyecto.
Construir el dispositivo IoT.	Construir el dispositivo IoT.	Dispositivo IoT.	Componentes electrónicos, sensores, actuadores, placa de cobre, acido percloruro, plumón permanente, franela, tirro, spray para aislamiento y evitar oxidación de pista, arduino, interface de red, cable header M-M, H-M, H-H, estaño, pasta, cautin, soldapull, PCB, otro básicos.
Instalar y configurar el prototipo electrónico e informático de monitoreo en COCESNA.	Probar la comunicación entre el dispositivo IoT y la aplicación Web. Ajustes finales del dispositivo IoT. Instalar y Configurar el sistema IoT en COCESNA.	Instalación del Proyecto en COCESNA. Entrega final.	Kit de herramientas. Dispositivo IoT.
Documentar la investigación desarrollada, para su divulgación a los sectores interesados.	13. Elaborar el informe final. 14. Presentación de resultados en sede Central.	Reporte final. Articulo técnico	Papel, impresora.

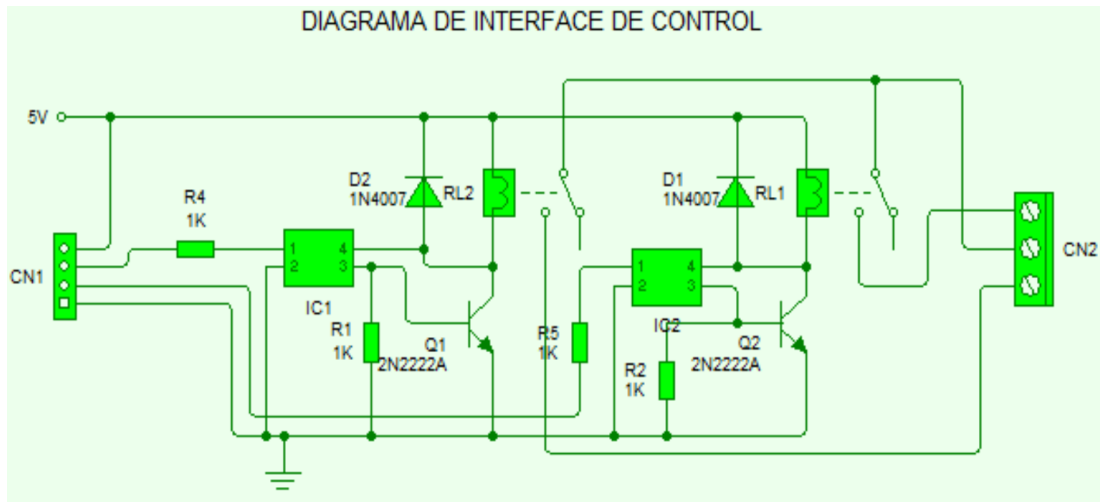
6.1. COMPONENTES ELECTRÓNICOS DEL PROYECTO.

Cantidad	Descripción
1	Fuente de Poder 5VDC A.
2	Arduino MEGA
2	Placa Arduino Mega – Estuche Transparente
1	LCD 20X4
2	Ethernet Shield
2	Rele de 5 V dc
2	Potenciómetro preset de 10K Ω
2	Potenciómetro de 2 K Ω
2	Transistores 2N2222A
2	Bornera Color verde de dos terminales
1	Bornera Color verde de tres terminales
6	Resistencia de 1K Ω
2	Opto acoplador 4N25
2	Diodo rectificador 1N4007
1	Paquete de papel fotográfico de 25 hojas
1	Tirro Normal
3	Plumones permanentes punta fina
1	Placa de cobre
1	Cautín 40 Watts
1 Libra	Estaño
50 gramos	Pasta para soldar
1	Cuchilla
2	Sensor de Temperatura y Humedad DHT11
3	Cable header M-M,H-M,H-H (Uno de cada uno)
5	Pines Macho 1 40 40pin 2.54mm 19mm de largo
5	Pin hembra de una sola fila 2.54mm 40 pin
1	Módulo GSM / GPRS A6 – SIM 900.
3	LED color
3	LED color verde
3	LED color amarillo
1	Interruptor con neón o luz piloto color verde 110 VAC para On / Off del dispositivo OIT
1	Interruptor con neón o luz piloto color rojo 12 DC para programar el mantenimiento de la sala de equipos
9	Resistencia 330 ohm ¼ de watts
1	Extensión 110 VAC de 2 metros para energizar el circuito desde la red eléctrica
2	Cables con Jack Hembra y su Jack macho.
1	Conector RJ45 Hembra
2	Conector RJ45 Macho
1	Yarda de cable UTP Cat 5e
1	Mts o yarda de termoencogible
1	Conversor FTDI a USB (Voltaje de trabajo 3.3 Vdc y 5.0Vdc)

6.2. CONEXIONES INTERNAS EN EL CHASIS



6.3. INTERFACE ELECTRÓNICA DE CONTROL DE CARGAS



Para el diseño de la interfaz de control, se utiliza un transistor y un relé, debido a que los pines (en la mayoría de los casos) de salida del microcontrolador (Arduino), solo son capaz de entregar entre 10 y 30 mA, las exigencia de corrientes superiores pueden dañar el pin programado como salida, tomando en cuenta dicha situación, se utilizan transistores para manejar corrientes mayores, y se utiliza un relé para manejar corrientes mucho mayores a las que se puede manejar un transistor BJT, además de proveer de aislamiento para manejar corriente alterna. En el diagrama se muestra un opto-acoplador conectado a un transistor NPN haciendo lo que se conoce como transistor darlington que permite que la bobina del relé se energice, se utiliza una optocoupla para aislar las corrientes del transistor y de la salida del microcontrolador, para que este maneje un simple LED que alimentara la base del transistor atreves de la optocoupla.

De manera simplificada funciona como un interruptor controlado por un circuito electrónico. Y ese circuito electrónico está compuesto por resistencias limitadoras de corriente y una optocoupla, un transistor y un diodo rectificador que protege al transistor. El transistor se utiliza como switch electrónico en el que controla la salida del relé y el electroimán interno se acciona, permitiendo abrir o cerrar los contactos siendo capaz de controlar un circuito de salida de mayor potencia que el de entrada, en un amplio sentido, como un amplificador eléctrico.

6.4. DESARROLLO DE CÓDIGO ARDUINO

Un programa diseñado para ejecutarse sobre una placa Arduino (un "sketch") siempre se compone, al menos, de tres secciones:

- 1) La sección de declaraciones de variables globales: ubicada directamente al principio del sketch.
- 2) La sección llamada "void setup()": delimitada por llaves de apertura y cierre.
- 3) La sección llamada "void loop()": delimitada por llaves de apertura y cierre.

La primera sección del sketch (que no tiene ningún tipo de símbolo delimitador de inicio o de final) está reservada para escribir, tal como su nombre indica, las diferentes declaraciones de variables que se necesiten.

En el interior de las otras dos secciones (es decir, dentro de sus llaves) deben de escribirse las instrucciones que se deseen ejecutar en la placa, teniendo en cuenta lo siguiente.

Las instrucciones escritas dentro de la sección “void setup()” se ejecutan una única vez, en el momento de encender (o resetear) la placa Arduino.

Las instrucciones escritas dentro de la sección “void loop ()” se ejecutan justo después de las de la sección “void setup ()” infinitas veces hasta que la placa se apague (o se resetee), es decir, el contenido de “void loop ()” se ejecuta desde la primera instrucción hasta la última, para seguidamente volver a ejecutarse desde la primera instrucción hasta la última, para seguidamente ejecutarse desde la primera instrucción hasta la última, y así una y otra vez.

Por tanto, las instrucciones escritas en la sección “void setup ()” normalmente sirven para realizar ciertas pre-configuraciones iniciales y las instrucciones del interior de “void loop ()” son, de hecho, el programa en sí que está funcionando continuamente.

Además de estas funciones básicas, pueden implementarse nuevas funciones, de forma que se puede encapsular tareas específicas que se necesiten utilizar de forma periódica en la función loop (). Estas funciones se pueden crear directamente en el fichero principal, o pueden utilizarse librerías ya implementadas, las cuales ofrecen funciones ya programadas destinadas a tareas concretas (lectura de sensores, conversión de variables...).

Por ejemplo, si se quiere realizar control de un servomotor no hay que programar una función que lleve a cabo las tareas de mando, ya que se dispone en Internet de librerías que incorporan varias funciones de control de servomotores.

Un nivel más allá de la encapsulación es utilizar una Programación Orientada a Objetos.

El principal objetivo de la POO es lograr un código más reutilizable, por lo que se mejora notablemente la eficiencia a la hora de programar.

Un objeto es una unidad que engloba en sí mismo datos y procedimientos necesarios para el tratamiento de esos datos. Cada objeto contiene datos y funciones, y un programa se construye como un conjunto de objetos, o incluso como un único objeto. Los objetos se agrupaban en clases, de la misma forma que en la realidad, existiendo muchas mesas distintas (la de mi cuarto, la del comedor, la del laboratorio, la del compañero o la mía) agrupamos todos esos objetos reales en un concepto más abstracto denominado mesa. De tal forma que, se puede decir, que un objeto específico es una realización o instancia de una determinada clase.

El desarrollo y programación del código se ha llevado a cabo mediante el software Arduino IDE 1.8.1.

El lenguaje C es uno de los lenguajes de programación estructurada más utilizados en nuestros días.

El principal componente estructural de C es la función. En C, las funciones son bloques en los cuales ocurren las actividades de un programa.

La programación orientada a objetos (POO) se fue convirtiendo en el estilo de programación dominante a mediados de los años ochenta, en gran parte debido a la influencia de C++, una extensión del lenguaje de programación C. Para tal caso se ha realizado el sketch respectivo de este proyecto implementando dicho paradigma de programación.

LIBRERIAS

#define

VARIABLES GLOBALES

Definición de CLASES

Creación de objetos

void setup ()

void loop ()

A continuación, se estudian los diferentes bloques que constituyen el programa de la placa Arduino.

Para este proyecto han sido necesarios incluir las siguientes librerías externas:

- ✓ #include <SPI.h>
- ✓ #include <Ethernet.h>
- ✓ //Libreria para Sensores DHT
- ✓ #include <DHT.h>
- ✓ #include <LiquidCrystal.h>
- ✓ #include <SoftwareSerial.h>;
- La librería DHT contiene las funciones necesarias para la adquisición de valores del sensor DHT.
- La librería SPI y Ethernet contiene las funciones necesarias para la comunicación del Arduino Mega con el Shield Ethernet.
- La librería LiquidCrystal contiene las funciones necesarias para la comunicación de Arduino Mega con la LCD 20X4.
- La librería SoftwareSerial se utilizó para la comunicación entre el Arduino mega, el chip FTDI y el Shield SIM900 GSM/GPRS.

7. RESULTADOS

7.1. CÓDIGO FUENTE DEL PROYECTO

```
sketch_Agos_04_OK14_INTERNET1 Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda

sketch_Agos_04_OK14_INTERNET1.$
#include <SPI.h>
#include <Ethernet.h>
//Libreria para Sensores DHT
//#include "DHT.h"
#include <DHT.h>
#include <LiquidCrystal.h>
//LiquidCrystal lcd(2, 3, 4, 5, 6, 7); //Arduino = lcd: - (2=RS, 3=E, 4=D4, 5=D5, 6=D6, 7=D7)
LiquidCrystal lcd(22, 23, 24, 25, 26, 27); //Arduino = lcd: - (22=RS, 23=E, 24=D4, 25=D5, 26=D6, 27=D7)

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0D, 0x4E, 0xD7
  // 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
  //0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02
};

#define DHTPIN 2 // Pin del Arduino al cual esta conectado el pin 2 del sensor
//Descomentar segun el tipo de sensor DHT usado
//#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)

//Inicializa el sensor DHTxx
DHT dht(DHTPIN, DHTTYPE);

//Configuracion de entrada analogica de testing
int fotocelda = 0;
int sensor_A0;
int contador = 0;
int buttonState = 0;
int buttonState1=0;
const int pir = 7;
const int terminal_reset = 8;

//const int buttonPin = 53; //pin para sala en mantenimiento o no.
const int buttonPin = 6; //pin para sala en mantenimiento o no.
const int analogInPin = A0;
int sensorValue = 0;
int outputValue = 0;
#define led_test 46
#define led_power 47
#define led_a 48
#define led_b 49
#define led_c 45 //define led_c 50 //CAMBIE ESTO POR EL CONFLICTO CON LOS PINES DE PROGRAMACION DEL S
//los pines 50, 51, 52 y 53

// Inicializa la instancia client
EthernetClient client;
char server[] = "mjgl.com.sv"; //habilitar esta linea para enviar los datos a INTERNET.
// Ultima conexion al servidor medida en milisegundos
unsigned long ultimaConexion = 0;
```



```

// Intervalo en milisegundos entre conexiones
//const unsigned long intervaloConexion = 25000;
unsigned long intervaloConexion = 25000;

unsigned long tiempo;
//int segundo=0;int minuto=0; int hora=0;
unsigned long hora=0;
unsigned long minuto=0;
unsigned long segundo=0;
int h; int m; int s;
// Leds conectados a Salidas Digitales
int led_1;
int led_2;
int led_3;
int sensor_mantenimiento=0;
int descripcion = 0;

```

```

void setup() {
  pinMode(terminal_reset, OUTPUT);
  digitalWrite(terminal_reset, HIGH);

  //Configura salidas digitales para la placa de 4 relay
  pinMode(34, OUTPUT);
  pinMode(35, OUTPUT);
  pinMode(36, OUTPUT);
  pinMode(37, OUTPUT);

  digitalWrite(34, HIGH);
  digitalWrite(35, HIGH);
  digitalWrite(36, HIGH);
  digitalWrite(37, HIGH);
}

```

```

//Espera 1 segundo para que se inicie la tarjeta Ethernet
delay(1000);

//captura datos obtenidos por el DHCP.
Serial.begin(9600);
Serial1.begin(9600);

//Lo converti en una funcion
while (!Serial) { ; } // So0 para Leonardo
Serial.println("Buscando DHCP...");
Serial1.println("Buscando DHCP...");

if (Ethernet.begin(mac) == 0)
{
  Serial.println("Failed to configure Ethernet using DHCP");
  Serial1.println("Failed to configure Ethernet using DHCP");
  for (;;) ; //No tiene sentido seguir
}

datos_server_dhcp();
//Inicializa Sensor DHTxx
Serial.print("Iniciando Sensor DHT");
Serial1.print("Iniciando Sensor DHT");
Serial.println(DHTTYPE);
Serial1.println(DHTTYPE);
Serial.println();
Serial1.println();
dht.begin();
delay(1000);
lcd.begin(20, 4);
lcd.print("COCESNA -ITCA FEPADE");
lcd.setCursor(0, 1);
lcd.print("SISTEMA DE MONITOREO");
lcd.setCursor(0, 2);
lcd.print("Loading...");

```

```

lcd.setCursor(0, 3);
lcd.print("Please wait...");
pinMode(buttonPin, INPUT);           //Sensor para alertar que la sala esta en mantenimiento. pin 6
pinMode(pir, INPUT);                 //Sensor para detección de movimiento en la sala. pin 7.
pinMode(led_power, OUTPUT);
pinMode(led_test, OUTPUT);          //pinMode(analogOutPin, OUTPUT);
pinMode(led_a, OUTPUT);
pinMode(led_b, OUTPUT);
pinMode(led_c, OUTPUT);             //INDICADOR DE ESTADO DE SWITCH PARA
                                     //PONER LA SALA EN MANTENIMIENTO.

digitalWrite(led_power, HIGH);
for (int i=0; i<=5; i++){
delay(200);
digitalWrite(led_a, HIGH);digitalWrite(led_b, LOW);digitalWrite(led_c, LOW);
delay(200);
digitalWrite(led_a, LOW);digitalWrite(led_b, HIGH);digitalWrite(led_c, LOW);
delay(200);

```

```

digitalWrite(led_a, LOW);digitalWrite(led_b, LOW);digitalWrite(led_c, HIGH);
delay(200);
digitalWrite(led_a, LOW);digitalWrite(led_b, LOW);digitalWrite(led_c, HIGH);
delay(200);
digitalWrite(led_a, LOW);digitalWrite(led_b, HIGH);digitalWrite(led_c, LOW);
delay(200);
digitalWrite(led_a, HIGH);digitalWrite(led_b, LOW);digitalWrite(led_c, LOW);
delay(200);
}
digitalWrite(led_a, HIGH);digitalWrite(led_b, HIGH);digitalWrite(led_c, LOW);
//Espera dos segundos para dar tiempo al sensor mientras obtiene muestra
delay(2000);
}

```

```

void datos_server_dhcp() {
Serial.println();
Serial.println("CONFIGURACION DE LA RED DEL DISPOSITIVO IOT");
Serial.println();
Serial.println("CONFIGURACION DE LA RED DEL DISPOSITIVO IOT");
Serial.print("Direccion IP : ");
Serial.print("Direccion IP : ");
for (byte B = 0; B < 4; B++) {
Serial.print(Ethernet.localIP()[B], DEC);
Serial.print(Ethernet.localIP()[B], DEC);
Serial.print(".");
Serial.print(".");
}
Serial.println();
Serial.println();

```

```

Serial.print("Router por defecto : ");
Serial.print("Router por defecto : ");
for (byte B = 0; B < 4; B++) {
Serial.print(Ethernet.gatewayIP()[B], DEC);
Serial.print(Ethernet.gatewayIP()[B], DEC);
Serial.print(".");
Serial.print(".");
}
Serial.println();
Serial.println();

Serial.print("Subred: ");
Serial.print("Subred: ");
for (byte B = 0; B < 4; B++) {
Serial.print(Ethernet.subnetMask()[B], DEC);
Serial.print(Ethernet.subnetMask()[B], DEC);
Serial.print(".");

```

```

        Serial1.print(".");
    }
    Serial.println();
    Serial1.println();
    Serial.print("DNS: ");
    Serial1.print("DNS: ");
    for (byte B = 0; B < 4; B++) {
        Serial.print(Ethernet.dnsServerIP()[B], DEC);
        Serial1.print(Ethernet.dnsServerIP()[B], DEC);
        Serial.print(".");
        Serial1.print(".");
    }
    Serial.println();
    Serial.println();
    Serial1.println();
    Serial1.println();
}

void loop() {
    Serial1.print("CONTADOR : ");
    Serial1.print(contador);
    Serial1.println();

    conexion_servidor();

    //Obtengo el dato de la señal analogica que da el potenciómetro
    sensorValue = analogRead(analogInPin);
    sensor_A0 = map(sensorValue, 0, 1023, 0, 255);
    analogWrite(led_test, sensor_A0);

    sensor_mantenimiento = digitalRead(buttonPin);
    if (sensor_mantenimiento == HIGH) {
        digitalWrite(led_c, HIGH);
    } else {
        digitalWrite(led_c, LOW);
    }

    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);

    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
        Serial1.println("Failed to read from DHT sensor!");
        return;
    }

    // Compute heat index in Fahrenheit (the default)
    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahreheit = false)
    float hic = dht.computeHeatIndex(t, h, false);

    if(contador == 0){
        datos_server_dhcp();

        Serial1.println("////////////////////////////////////////");
        Serial1.print("INFORMACION DE ESTADO DE LOS SENSORES.");
        Serial1.println();
        Serial1.print("Temperatura: ");
        Serial1.print(t);
        Serial1.print(" oC ~ ");
        Serial1.print(f);
        Serial1.print(" oF");
    }
}

```

```

Serial.println();
Serial.print("Humedad: ");
Serial.print(h);
Serial.print(" % RH");

Serial.println();
Serial.print("Sensor de testing analogico = ");
Serial.print(sensor_A0);

Serial.println();
Serial.print("Sensor de mantenimiento = ");
Serial.print(sensor_mantenimiento);

buttonState1 = digitalRead(pir);
Serial.println();
Serial.print("Sensor de movimiento = ");
Serial.print(buttonState1);

Serial.println();
Serial.println("////////////////////////////////////////");
Serial.println();
contador = 1;
}
if (contador == 3){
    digitalWrite(terminal_reset, LOW);
    delay(200);
    //conexion_servidor();
}
delay(500);
lcd.setCursor(0, 2);
lcd.print("Tem:");
lcd.print(t);
lcd.print("oC ");
lcd.print(f);
lcd.print("oF");

lcd.setCursor(0, 3);
lcd.print("Humedad:");
lcd.print(h);
lcd.print("% ");
lcd.print(" RH.");
delay(100);
}
// Fin del loop principal
void conexion_servidor(){
//Si hay conexion con el servidor
if (client.connected()) {
    if (client.available()) {
        hora=0;minuto=0;segundo=0;
        h=0;m=0;s=0; //intervaloConexion=0;tiempo=0;
        // Obtiene la respuesta del servidor y busca los tags xml <led_1> <led_2> y <led_3>
        if(client.find("<led_1>")) {

            led_1 = client.parseInt();
        }
        if(client.find("<led_2>")) {
            led_2 = client.parseInt();
        }
        if(client.find("<led_3>")) {
            led_3 = client.parseInt();
        }

        if(client.find("<segundo>")) {
            segundo = client.parseInt();
        }

        if(client.find("<minuto>")) {
            minuto = client.parseInt();
        }
    }
}

```

```

if(client.find("<hora>")) {
    hora = client.parseInt();
}

Serial.print("...");
Serial1.print("...");
Serial.print("Conectados con : ");
Serial1.print("Conectados con : ");
Serial.print(server);
Serial1.print(server);
Serial.println();
Serial1.println();

//Seteo tiempo programado desde el portal web.
tiempo = (hora*3600000) + (minuto*60000) + (segundo*1000);
//Serial.print("tiempo: ");
//Serial.println(tiempo);

//Seteo el tiempo calculado.
intervaloConexion=tiempo;
Serial.println("");
Serial1.println("");
Serial.print("Tiempo de lecturas parametrizado desde portal web: ");
Serial1.print("Tiempo de lecturas parametrizado desde portal web: ");
//Serial.print(intervaloConexion); Serial.println(" ms");
Serial.println();
Serial1.println();

Serial.print(intervaloConexion);
Serial1.print(intervaloConexion);
Serial.print(" ms");
Serial1.print(" ms");
Serial.print(" ~ Horas=");
Serial1.print(" ~ Horas=");
Serial.print(hora);

Serial1.print(hora);
Serial.print(" : Minutos=");
Serial1.print(" : Minutos=");
Serial.print(minuto);
Serial1.print(minuto);
Serial.print(" : Segundos=");
Serial1.print(" : Segundos=");
Serial.print(segundo);
Serial1.print(segundo);
Serial.println();
Serial1.println();

//CONVERSIONES PARA MOSTRAR EN EL MONITOR SERIAL LOS TIEMPOS PARAMETRIZADOS DEL PORTAL WEB.
if(hora>0) h=(hora*3600000)/3600000;
if(minuto>0) m=((minuto*60000)/60000);
if(segundo>0) s=((segundo*1000)/1000);

// Imprime valores enviados por servidor
Serial.println();
Serial1.println();
Serial.println("Respuesta enviada por el servidor");
Serial1.println("Respuesta enviada por el servidor");
Serial.print("Carga 1: ");
Serial1.print("Carga 1: ");
Serial.println(led_1);
Serial1.println(led_1);
Serial.print("Carga 2: ");
Serial1.print("Carga 2: ");
Serial.println(led_2);
Serial1.println(led_2);
Serial.print("Carga 3: ");
Serial1.print("Carga 3: ");
Serial.println(led_3);
Serial1.println(led_3);

```

```

//Cierra conexion
Serial.println();
Serial1.println();
Serial.println("Cerrando conexion...");
Serial1.println("Cerrando conexion...");
Serial.println();
Serial1.println();
client.stop();

contador = 0;
// Enciende Leds segun valores obtenidos del portal web.
if (led_1)
    digitalWrite(34, LOW);
else
    digitalWrite(34, HIGH);

if (led_2)

    digitalWrite(35, LOW);
else
    digitalWrite(35, HIGH);

if (led_3)
    digitalWrite(36, LOW);
else
    digitalWrite(36, HIGH);

//if (led_3){
//digitalWrite(34, LOW);
//digitalWrite(35, LOW);
//digitalWrite(36, LOW);
//digitalWrite(37, LOW);
//}else{
//digitalWrite(34, HIGH);
//digitalWrite(35, HIGH);

//digitalWrite(36, HIGH);
//digitalWrite(37, HIGH);
//}

} //else{
//conexion_servidor();
//}

}
else if(millis() - ultimaConexion > intervaloConexion) {
    httpRequest();
    contador=3;
}

//delay(1);

//if (!client.connected())
//{
//    /*Serial.println();
//    Serial.println("disconnecting.");
//    client.stop();
//    // do nothing
//    while (true);
//    */
//}
}

// Funcion que realiza la conexion http al servidor y obtiene la respuesta
void httpRequest() {
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();

```

```

// Read temperature as Celsius (the default)
float t = dht.readTemperature();
// Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  Serial.println("Failed to read from DHT sensor!");
  return;
}

// Control de errores, valida que se obtuvieron valores para los datos medidos
// if (isnan(h) || isnan(t)) {
//   Serial.println("Falla al leer el sensor DHTxx!");
//   Serial.println("Falla al leer el sensor DHTxx!");
//   return;
// }

// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);

if (t>0 && t<=17){
  descripcion = 1;
}else if (t>17 && t<=40){
  descripcion = 2;
}else if (t>40 && t<=60){
  descripcion = 3;
}else if (t>60){
  descripcion = 4;
}

// Obtiene datos de la fotocelda
//sensorValue = analogRead(analogInPin);
//sensor_A0 = map(sensorValue, 0, 1023, 0, 255);

//analogWrite(led_test, sensor_A0);
//sensor_A0 = analogRead(0);

//sensor_mantenimiento = digitalRead(buttonPin);
// buttonState1 = digitalRead(pir);
Serial.println("*****");
Serial.println("*****");
Serial.print("INFORMACION DE ESTADO DE LOS SENSORES.");
Serial.print("INFORMACION DE ESTADO DE LOS SENSORES.");
Serial.println();
Serial.println();
Serial.print("Temperatura: ");

Serial.print("Temperatura: ");
Serial.print(t);
Serial.print(t);
Serial.print(" oC ~ ");
Serial.print(" oC ~ ");
Serial.print(f);
Serial.print(f);
Serial.print(" oF");
Serial.print(" oF");

Serial.println();
Serial.println();
Serial.print("Humedad: ");
Serial.print("Humedad: ");
Serial.print(h);
Serial.print(h);
Serial.print(" % RH");

```

```

Serial.print(" % RH");

Serial.println();
Serial.println();
Serial.print("Sensor de testing analogico = ");
Serial.print("Sensor de testing analogico = ");
Serial.print(sensor_A0);
Serial.print(sensor_A0);

Serial.println();
Serial.println();
Serial.print("Sensor de mantenimiento = ");
Serial.print("Sensor de mantenimiento = ");
Serial.print(sensor_mantenimiento);
Serial.print(sensor_mantenimiento);

Serial.println();

Serial.println();
Serial.print("Sensor de movimiento = ");
Serial.print("Sensor de movimiento = ");
Serial.print(buttonState1);
Serial.print(buttonState1);
Serial.println();
Serial.println();
Serial.println("*****");
Serial.println("*****");

Serial.println();
Serial.println();

// Se conecta al servidor en el puerto 80 (web)
if (client.connect(server, 80)) {
// Envia el dato al puerto serial
//Serial.println();

client.print(descripcion);
client.print("&temperatura=");
client.print(t);
client.print("&humedad=");
client.print(h);
client.print("&A0=");
client.print(sensor_A0);
client.print("&manto=");
client.print(sensor_mantenimiento);

client.print("&pir=");
client.print(buttonState1);
client.print("&ip=");
client.print(Ethernet.localIP());
client.print("&pais=");
client.print("4");
client.print("&sala=");
client.print("1");

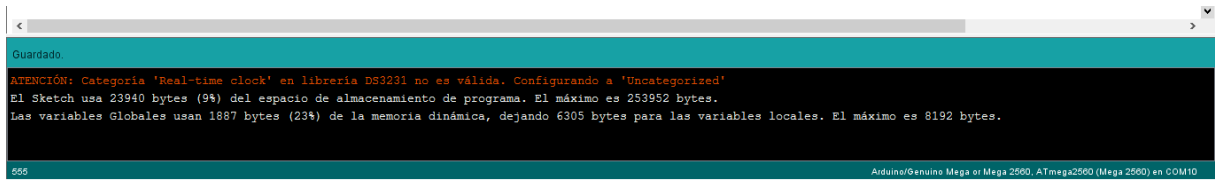
client.println(" HTTP/1.1");
client.print("Host: ");
client.println(server);
client.println("User-Agent: Arduino-Ethernet");
client.println("Connection: close");
client.println();

client.println ("*****");

client.println ("Copyright COCESNA - A traves de MEGATEC-ZACATECOLUCA");
client.println ("*****");

client.println();
// Actualiza el tiempo en milisegundos de la ultima conexion
ultimaConexion = millis();
}
}

```

CÓDIGO BÁSICO PARA VERIFICACIÓN DE LA CONFIGURACIÓN DEL SHIELD ETHERNET ARDUINO POR DHCP DESDE EL SERVIDOR.

ROUTER O S.O CON SERVICIO DE DHCP.

```
sketch_jan29a Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda
sketch_jan29a $
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0D, 0x4E, 0xD7
  // 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
  //0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02
};
// Inicializa la instancia client
EthernetClient client;
void setup() {
  // put your setup code here, to run once:
  //captura datos obtenidos por el DHCP.
  Serial.begin(9600);
  Serial1.begin(9600);

  //Lo converti en una funcion
  while (!Serial) { ; } // Soo para Leonardo

  Serial.println("Buscando DHCP...");
  Serial1.println("Buscando DHCP...");

  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("Failed to configure Ethernet using DHCP");
    Serial1.println("Failed to configure Ethernet using DHCP");
    for (;;) ; //No tiene sentido seguir
  }

  Serial.println();
  Serial.println("CONFIGURACION DE LA RED DEL DISPOSITIVO IOT");
  Serial1.println();
  Serial1.println("CONFIGURACION DE LA RED DEL DISPOSITIVO IOT");
  Serial.print("Direccion IP : ");
  Serial1.print("Direccion IP : ");
  for (byte B = 0; B < 4; B++) {
    Serial.print(Ethernet.localIP()[B], DEC);
```

```

    Serial1.print(Ethernet.localIP()[B], DEC);
    Serial.print(".");
    Serial1.print(".");
}
Serial.println();
Serial1.println();

Serial.print("Router por defecto : ");
Serial1.print("Router por defecto : ");
for (byte B = 0; B < 4; B++) {
    Serial.print(Ethernet.gatewayIP()[B], DEC);
    Serial1.print(Ethernet.gatewayIP()[B], DEC);
    Serial.print(".");
    Serial1.print(".");
}
Serial.println();
Serial1.println();

Serial.print("Subred: ");
Serial1.print("Subred: ");
for (byte B = 0; B < 4; B++) {
    Serial.print(Ethernet.subnetMask()[B], DEC);
    Serial1.print(Ethernet.subnetMask()[B], DEC);
    Serial.print(".");
    Serial1.print(".");
}
Serial.println();
Serial1.println();
    Serial.print("DNS: ");
    Serial1.print("DNS: ");
for (byte B = 0; B < 4; B++) {
    Serial.print(Ethernet.dnsServerIP()[B], DEC);
    Serial1.print(Ethernet.dnsServerIP()[B], DEC);
    Serial.print(".");
    Serial1.print(".");
}
Serial.println();
Serial1.println();
}

void loop() {
    // put your main code here, to run repeatedly:
}

```

CODIGO DEL MÓDULO GSM / GPRS SIM900.

```
ya_FUNCIONO Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda
ya_FUNCIONO
int sensor=0;
int salir=0;
int indicator=5;
#include <SoftwareSerial.h>
//SoftwareSerial SIM900(2, 3); // Configura el puerto serial para el SIM900
SoftwareSerial SIM900(7, 8); // RX, TX Configura el puerto serial para el SIM900...FUNCIONA PERFECTAMENTE CON ESTO
//SoftwareSerial SIM900(22, 23); // Configura el puerto serial para el SIM900
char incoming_char=0; //Variable que guarda los caracteres que envia el SIM900
int pir = 2;

void setup() {
  pinMode(indicator, OUTPUT); //CARGA A CONTROLAR
  pinMode(9, OUTPUT);
  pinMode(pir, INPUT);

  SIM900power(); //Encender el chip

  //inicializaSIM900();

  SIM900.begin(19200); //Configura velocidad serial para el SIM900
  //SIM900power(); //Encender el chip
  delay(25000); //Retardo para que encuentra a una RED
  Serial.begin(19200); //Configura velocidad serial para el Arduino
  Serial.println("OK"); //Mensaje OK en el arduino, para saber que todo va bien.

  for(int i=0;i<=4;i++){
    digitalWrite(indicator,HIGH);
    delay(200);
    digitalWrite(indicator,LOW);
    delay(200);
  }
}

void loop() {
  int sensor = digitalRead(pir);
  if (sensor == 1){
    digitalWrite(indicator, HIGH);
    // Serial.println("Sensor de movimiento a detectado intruso");
    // Serial.print("Estado del sensor: ");
    // Serial.println(sensor);

    mensaje();

    // Serial.println("Sr. Manuel Gamez, Lamentamos informarle que hay INTRUSOS.");
    // Serial.println("La temperatura ambiente dentro de su casa es: ");
    // Serial.println(analogRead(0)); // message to send
    // Serial.println("SMS ENVIADO.");
    // Serial.println("");

    //llamar();

  }else{
    //digitalWrite(5, LOW);

    //Serial.println(sensor);
  }
  //delay(500);

  digitalWrite(indicator, LOW);
}

void inicializaSIM900()
{
```

```

// digitalWrite(9, HIGH); // Descomentar para activar la alimentación de la tarjeta por S
// delay(1000);
// digitalWrite(9, LOW);

delay (5000);
SIM900.begin(19200); //Configura velocidad del puerto serie para el SIM900
Serial.begin(19200); //Configura velocidad del puerto serie del Arduino
Serial.println("OK");
delay (1000);
SIM900.println("AT + CPIN = \"1234\""); //Comando AT para introducir el PIN de la tarjeta
delay(25000); //Tiempo para que encuentre una RED
Serial.println("PIN OK");
SIM900.print("AT+CLIP=1\r"); // Activa la identificación de llamada
delay(100);
SIM900.print("AT+CMGF=1\r"); //Configura el modo texto para enviar o recibir mensajes

delay(1000);
SIM900.print("AT+CNMI=2,2,0,0,0\r"); // Saca el contenido del SMS por el puerto serie del GPRS
delay(1000);
}
void llamar()
{
//SIM900.println("ATD 10725275;"); //Numero local
//SIM900.println("ATD 61107065;"); //Celular MIO
SIM900.println("AT+CMGS=\"71271994\""); //BRENDA.
delay(100);
SIM900.println();
delay(15000); // wait for 30 seconds...
SIM900.println("ATH"); // Cuelta el telefono
delay(1000);
}

void SIM900power(){
digitalWrite(9, HIGH); // Descomentar para activar la alimentación de la tarjeta por Software
delay(1000);
digitalWrite(9, LOW);
delay (5000);
}

void mensaje(){
//Funcion para mandar mensaje de texto
SIM900.print("AT+CMGF=1\r"); // AT command to send SMS message
delay(100);
//SIM900.println("AT + CMGS = \"61107065\""); // recipient's mobile number, in international format

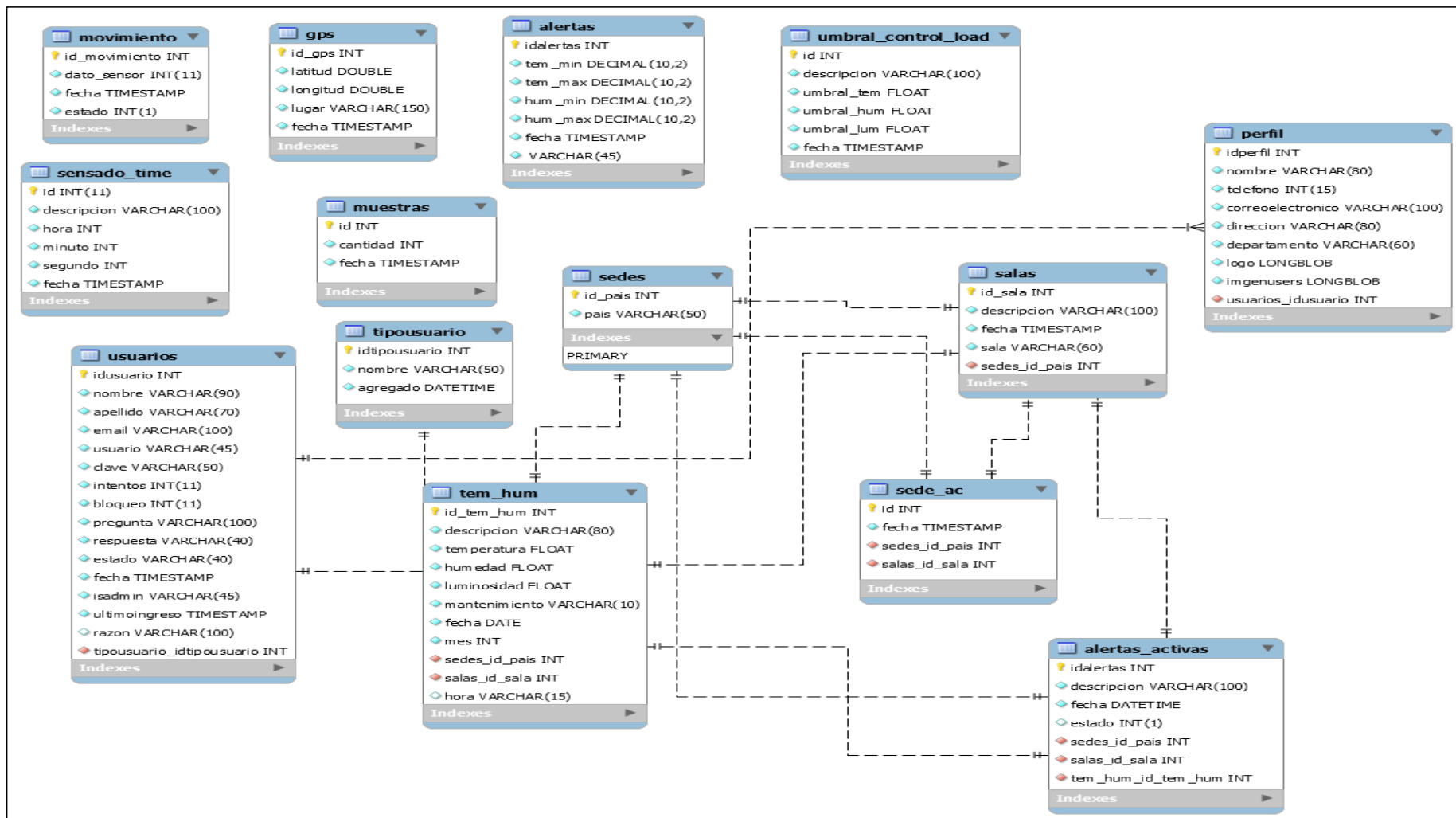
//SIM900.println("AT + CMGS = \"78484255\""); // recipient's mobile number, in international format
SIM900.println("AT + CMGS = \"78700872\""); // recipient's mobile number, in international format num. Ing. mixco

//SIM900.println("AT + CMGS = \"71271994\""); // recipient's mobile number, in international format
delay(100);
SIM900.println("Sr. Manuel Gamez, SALA EN AMBIENTE PELIGROSO.");
SIM900.println("La temperatura ambiente dentro de su casa es: ");
SIM900.println(analogRead(0)); // message to send

delay(100);
SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26 //Comando de finalizacion
delay(100);
SIM900.println();
delay(5000); // Tiempo para que se envíe el mensaje
}

```

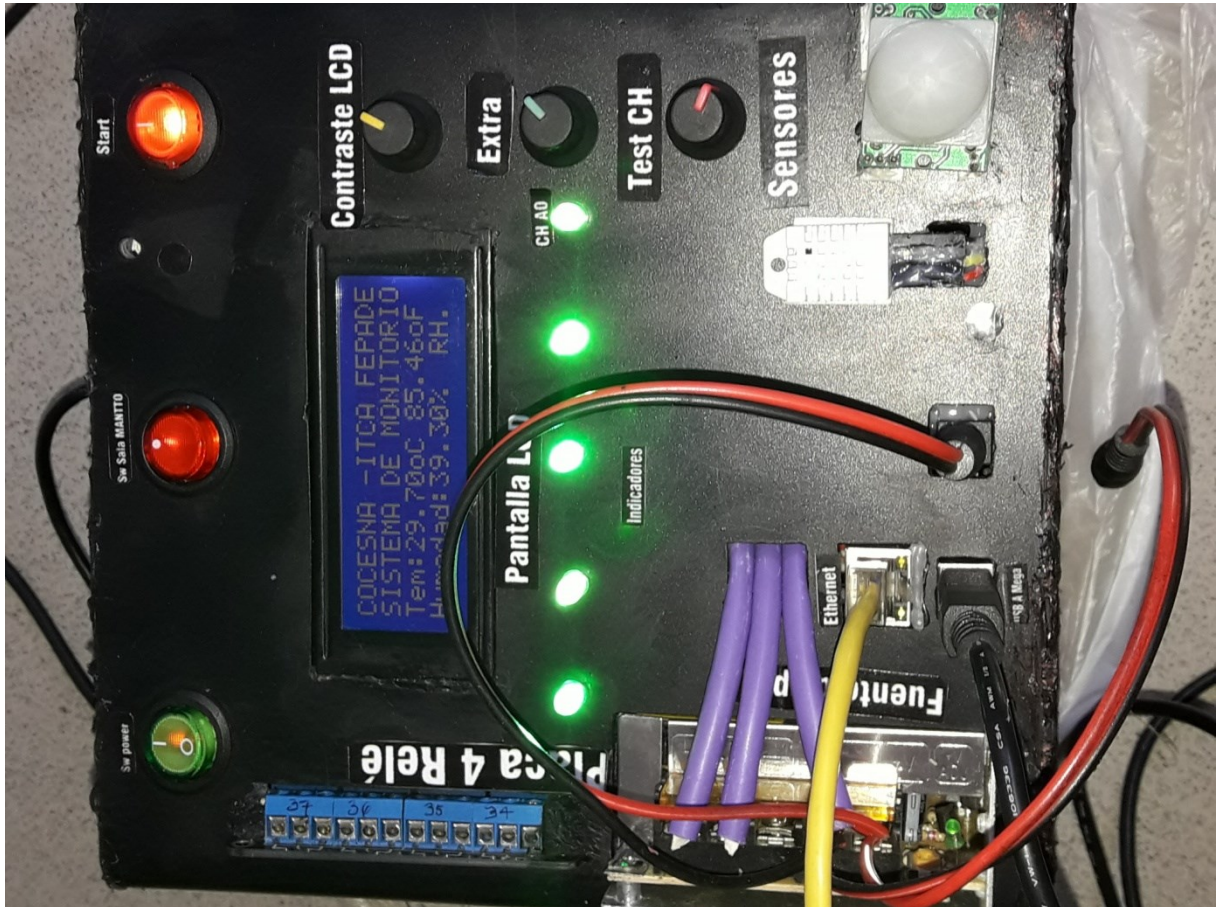
7.2. BASE DE DATOS



7.3. CARACTERÍSTICAS DEL PROTOTIPO

- ❖ Prototipo electrónico IoT con su respectivo chasis (Hardware).
- ❖ Portal Web para el monitoreo en tiempo real (Dashboard) on-line.
- ❖ Control de ON-OFF de cargas de voltaje DC y AC (120VAC-220VAC) según criterios de parametrización desde el portal web.
- ❖ Visualización de notificaciones en pantalla según Parametrización.
- ❖ Envío de mensajes a correo según cumplimiento de criterios previamente establecidos desde la opción configuración incluida en el portal web.
- ❖ Software necesario para monitoreo y control on-line en tiempo real para microcontrolador Arduino.
- ❖ App para celular con sistema operativo android 4.0 en adelante.
- ❖ Monitoreo remoto a través de app.
- ❖ Envío de mensajes a correo, whatsapp, fb, entre otros del último registro en BD.
- ❖ Envío de SMS a móvil desde la App.
- ❖ Navegación por sitio web oficial de COCESNA desde la app.
- ❖ Manual de usuario.





8. CONCLUSIONES

Se presenta una propuesta basada en el modelo Internet de las cosas para la creación de soluciones, el cual consiste en la interconexión de objetos con acceso a Internet con el fin de obtener datos relevantes de un ambiente.

Se demuestra que es posible la creación de este tipo de proyectos utilizando dispositivos como sensores y actuadores con el fin de generar soluciones acordes a modelos de negocio atractivos; que integren hardware y software de bajo costo, e incluya la capacidad de comunicarse con otros aplicativos, para generar mayor valor a los procesos de negocio.

9. RECOMENDACIONES.

Se hace énfasis en seguir las siguientes recomendaciones para obtener el máximo aprovechamiento del presente proyecto.

- ❖ Disponer de una conectividad segura y abierta para la red del dispositivo IoT.
- ❖ Configurar adecuadamente el router que se disponga.
- ❖ Configurar y habilitar el servicio DHCP de su router.
- ❖ Conseguir que la conexión IP entre el dispositivo IoT (Arduino Ethernet Shield) y el Router (área local) no cambie. **Dirección MAC: 0x90, 0xA2, 0xDA, 0x0D, 0x4E, 0xD7** del shield Ethernet para crearle una reserva por DHCP.
- ❖ Abrir los puertos del Router para permitir que entre y salga información.
- ❖ Conseguir que la conexión entre el Router e Internet no cambie (para que se pueda acceder siempre desde la misma dirección).
- ❖ Abrir únicamente el puerto que va a utilizar tu shield de Arduino. Para ello se debe buscar en los menús del Router algo de nombre **NAT, Virtual Servers o Port Forwarding**. El nombre varía en función del Router.

Port Forwarding

Entries in this table allow you to automatically redirect common network services to a specific machine behind the NAT firewall. These settings are only necessary if you wish to host some sort of server like a web server or mail server on the private local network behind your Gateway's NAT firewall.

Port Forwarding: Disable Enable

Protocol: Both Enable

Remote IP Address: Public Port: 80 - 80

Local IP Address: 192.168.1.177 Local Port: 80 - 80

Interface: ppp0

Current Port Forwarding Table:

Select	Local IP Address	Protocol	Local Port	Comment	Enable	Remote Host	Public Port	Interface
<input type="checkbox"/>	192.168.1.177	TCP+UDP	80		Enable		80	ppp0

- ❖ Instalar el portal web en un servidor on-line o hosting compartido, virtual o dedicado.

Nota: Se puede ir buscando desde la web de administración del router palabras claves como: modificar los puertos o guiarse por la palabra "Port". Si no se termina de encontrar la manera de llegar a ese menú, realizar búsqueda en Google (poniendo el modelo del Router).

10. GLOSARIO

Internet of Things IOT: Redes inalámbricas y auto-configurables entre los objetos que nos rodean. Cuando todos los objetos que nos rodean estén equipados con dispositivos de identificación podrán ser identificados y gestionados por sistemas informáticos. Los objetos como tales no se relacionarán, pero pueden ser controlados por algún tipo de agente a través de la red.

Internet: Red global compuesta por redes de ordenadores interconectadas mediante el protocolo TCP/IP, para facilitar la comunicación y el intercambio de datos. Abarca todo el mundo y permiten el acceso a millones de usuarios a información prácticamente sobre cualquier tema.

Wireless network. Red inalámbrica: Conjunto de dispositivos que transmiten y reciben datos vía ondas electromagnéticas. Generalmente se emplea entre equipos situados en un radio de acción reducido. Los sistemas de comunicaciones remotas disponen de características de seguridad como el cifrado y el salto entre frecuencias.

AP: (Access Point). Punto de acceso. Es un dispositivo que conecta dispositivos inalámbricos a otra red. Por ejemplo, a una red local inalámbrica, un módem de Internet, etc.

API: (Application Programming Interface). Interfaz de programación de aplicaciones. Interfaz que permite que dos programas informáticos independientes se comuniquen entre sí.

Ambiente: Con respecto a la temperatura, el ambiente se refiere a la temperatura del aire circundante.

Autenticación: Verifica la identidad de un usuario que inicia sesión en la red. Para comprobar la identidad del cliente en la red se utilizan contraseñas, certificados digitales, tarjetas inteligentes y biométricas. Las

contraseñas y los certificados digitales se utilizan también para que el cliente identifique la red.

DNS: (DomainNameSystem). Sistema de nombres de dominio. Sistema que asigna nombres de dominio como www.intel.com a las direcciones IP.

DIY: (Do It Yourself). Hágalo usted mismo. Es hacer una tarea usted mismo en lugar de depender de un profesional.

Firmware: Programa informático dentro de un dispositivo de hardware.

Sensor: es un objeto capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, [pH](#), etc. Una magnitud eléctrica puede ser una [resistencia eléctrica](#) (como en una [RTD](#)), una [capacidad eléctrica](#) (como en un [sensor de humedad](#)), una [tensión eléctrica](#) (como en un [termopar](#)), una [corriente eléctrica](#) (como en un [fototransistor](#)), etc.

Actuador: es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control, como por ejemplo una válvula. Son los elementos que influyen directamente en la señal de salida del automatismo, modificando su magnitud según las instrucciones que reciben de la unidad de control.

Activity: Representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual.

APP: Tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos para dispositivos móviles o tabletas

Bitbucket: Servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git. Bitbucket ofrece planes comerciales y gratuitos.

Concurrencia: La computación concurrente es la simultaneidad en la ejecución de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa.

Handshaking: La traducción de "handshaking" es "apretón de manos" y viene a significar que el procesador y los periféricos intercambian señales de control que les permiten sincronizar sus acciones y "colaborar" conjuntamente en la transferencia de información.

HTTP: Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la **World Wide Web**. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

IGU: Interfaz gráfica de usuario. Está construida mediante una jerarquía de View y objetos ViewGroup.

Interrupción: Señal recibida por el procesador de una computadora, para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

IU: Interfaz de usuario. Todo aquello que el usuario puede ver e interactuar.

JSON: acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Microcontrolador: Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una **computadora:** unidad central de procesamiento, memoria y periféricos de entrada/salida.

MySQL: MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

Opensource: Código abierto es la expresión con la que se conoce al software o hardware distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

Parser: Un analizador sintáctico (o parser) es una de las partes de un compilador que transforma su entrada en un árbol de derivación. El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada.

Processing: Lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.

Puerto Serie: Interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

PWM: Técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

Serializar: Proceso de codificación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible como XML o JSON, entre otros.

Smartphone: El teléfono inteligente o “teléfono” (en inglés: smartphone) es un tipo teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.

11. REFERENCIAS BIBLIOGRÁFICAS

Datasheet Chip ethernet IPv6:

- ❖ <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en022889>
- ❖ <http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf>

Más información sobre IPv6:

- ❖ <https://sites.google.com/site/ghoelzl/ipv6>

IPv6WebServer:

- ❖ https://sites.google.com/site/ghoelzl/ipv6ethershield/ipv6_http_server

Y aún más...

- ❖ <http://www.tweaking4all.com/hardware/arduino/arduino-enc28j60-ethernet/>

Y otro ejemplo:

- ❖ http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Ethernet-Module-ENC28J60-Arduino.pdf

Librerías:

- ❖ https://github.com/ntruchsess/arduino_uip
- ❖ <https://github.com/jcw/ethercard>
- ❖ <https://github.com/ghoelzl/IPv6EtherShield>
- ❖ <https://sites.google.com/site/ghoelzl/ipv6ethershield>

[1] KEN SCHWABER Y JEFF SUTHERLAND,
La Guía Definitiva de Scrum: Las Reglas del Juego,
Scrum.Org and Scrumming,
2014.

[2] ALBERT GARRIGA
Metodología de gestión de proyectos
<http://www.rekursosenprojectmanagement.com/metodologia-de-gestion-de-proyectos/>



VISIÓN

Ser una institución educativa líder en educación tecnológica a nivel nacional y regional, comprometida con la calidad, la empresarialidad y la pertinencia de nuestra oferta educativa.

MISIÓN

Formar profesionales integrales y competentes en áreas tecnológicas que tengan demanda y oportunidad en el mercado local, regional y mundial, tanto como trabajadores y como empresarios.

VALORES

EXCELENCIA: *Nuestro diario quehacer está fundamentado en hacer bien las cosas desde la primera vez.*

INTEGRIDAD: *Actuamos congruentemente con los principios de la verdad en todas las acciones que realizamos.*

ESPIRITUALIDAD: *Desarrollamos todas nuestras actividades en la filosofía de servicio, alegría, compromiso, confianza y respeto mutuo.*

COOPERACIÓN: *Actuamos basados en el buen trabajo en equipo, la buena disposición a ayudar a todas las personas.*

COMUNICACIÓN: *Respetamos las diferentes ideologías y opiniones, manteniendo y propiciando un acercamiento con todo el personal.*

SEDE Y REGIONALES EL SALVADOR



La Escuela Especializada en Ingeniería ITCA-FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro Centros Regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.

1 SEDE CENTRAL SANTA TECLA

Km. 11.5 carretera a Santa Tecla, La libertad.

Tel.: (503) 2132-7400

Fax: (503) 2132-7599

2 CENTRO REGIONAL SANTA ANA

Final 10a. Av. Sur, Finca Procavia.

Tel.: (503) 2440-4348

Tel.: (503) 2440-4348

Tel./Fax: (503) 2440-3183

3 CENTRO REGIONAL LA UNIÓN

Calle Sta. María, Col. Belén, atrás del Instituto Nacional de La Unión

Tel.: (503) 2668-4700

4 CENTRO REGIONAL ZACATECOLUCA

Km. 64.5, desvío Hacienda El Nilo sobre autopista a Zacatecoluca.

Tel.: (503) 2334-0763 y

(503) 2334-0768

5 CENTRO REGIONAL SAN MIGUEL

Km. 140 carretera a Santa Rosa de Lima.

Tel.: (503) 2669-2298

Fax: (503) 2669-0061