

DESARROLLO Y CONSTRUCCIÓN DE UN SISTEMA DE MONITOREO DE ALARMAS A TRAVES DE
GPRS Y ARDUINO PARA EL NODO BUENOS AIRES (BUGA)

Una Tesis Presentada Para Obtener El Título De:

Ingeniero de Telecomunicaciones

Universidad Nacional Abierta y a Distancia

UNAD

Jhon Freddy Reinoso Ojeda.

Abril 2019

Copyright © 2017 por Jhon Freddy Reinoso. Todos los derechos reservados.

Dedicatoria

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A mis padres que aunque no me lo expresan con palabras sé que siempre han querido lo mejor para mí. A Leydi García por ser tan paciente conmigo durante el tiempo que estuve realizando este trabajo...Que

Dios te bendiga!

A mi hermano Nelson que siempre me deseo lo mejor durante la realización de esta tesis....Gracias hermano

Agradecimientos

Quiero agradecer a Dios primeramente el haberme permitido terminar este trabajo así como a mi familia que sé que siempre estuvo a mi lado durante este proceso, también a Leydi García por su comprensión y apoyo en este caminar. No puedo dejar de mencionar a mis asesores de proyecto que me ayudaron en las correcciones de este trabajo; especialmente a la tutora Diana Gisela victoria que siempre estuvo atenta en este proceso de realizar un buen trabajo. ¡Muchas gracias a todos!

Tabla de contenido

Resumen.....	2
Capitulo 1 Antecedentes y Contexto.....	7
1.1 Planteamiento del problema.....	7
1.1.1 Situación Problema	7
1.1.2 Formulación del Problema.....	8
1.2 Justificación	9
1.3 Objetivos.....	12
1.3.1 Objetivo General.....	12
1.3.2 Objetivos Específicos.....	12
Capitulo 2 Revisión de la Literatura	13
2.1 Marco Contextual.....	13
2.2 Marco Teórico.....	15
2.2.1 Monitoreo y Gestión de Red	15
2.2.1.1 Funciones de los protocolos de gestión	17
2.2.1.1 Estándares de Gestión.....	19
2.2.1.2 Herramientas de monitoreo.....	22
2.2.2 Redes Celulares.....	28
2.2.2.1 Arquitectura de una red GSM.....	32
2.2.2.2 General Paquet Radio Service (GPRS).....	37

	v
2.2.3 Arduino	39
2.2.3.1 IDE Arduino.....	43
2.2.4 Internet de las cosas	44
2.2.4.1 La miniaturización del hardware	46
2.2.4.2 Campos de aplicación del internet de las cosas	47
2.2.4.3 Protocolo MQTT (Message Queue Telemetry Transport).....	48
Capitulo 3 Metodología de la Investigación	51
3.1 Tipo de Investigación.....	51
3.2 Ruta Metodológica.....	51
3.2.1 Fase de desarrollo de hardware.....	52
3.2.1.1 Diseño de la interfaz de alarmas	52
3.2.1.2 Uso de tarjeta arduino	57
3.2.1.3 Implementación del módulo GPRS	58
3.2.2 Fase de desarrollo de software.....	65
3.2.2.1 Programación de la tarjeta Arduino	65
Capitulo 4 Resultados y Discusión	68
4.1 Conexión del módulo GSM con Arduino a través del puerto serial.....	68
4.2 Conexión a la red celular GPRS	72
4.3 Configuración NSMP para envío de mensajes	78
4.4 Conexión a un servidor MQTT para el envío de mensajes.....	81
4.5 Construcción de la interfaz de alarmas	85

4.6 Centro de gestión	vi
4.6 Centro de gestión	87
4.7 Simulación de alarmas desde la interfaz	88

Lista de figuras

Figura 1. Elementos de la gestión de red	16
Figura 2. Tareas de los protocolos de gestión.....	18
Figura 3. Pila de protocolos CMIP.....	19
Figura 4. Pila de protocolos SNMP	21
Figura 5. Interface de FMS	23
Figura 7. Interface de Nagios.....	28
Figura 8. Arquitectura de la red GSM.....	30
Figura 9. Evolución de la telefonía móvil.....	31
Figura 10. Esquema de una red GSM	32
Figura 11. Esquema de una célula teórica vs real	36
Figura 12 BTS.....	36
Figura 13. Evolución de GSM (Datos)	38
Figura 14. Esquema de una red GPRS	39
Figura 15. Tarjeta Arduino Uno.....	40

Figura 16. Distribución de pines Arduino.....	41
Figura 17. Funcionamiento de un Microcontrolador	42
Figura 18. IDE Arduino	44
Figura 19. Conceptos de Internet de las cosas	45
Figura 20. Miniaturización del hardware.....	46
Figura 21. Funcionamiento de MQTT.....	50
Figura 22. Estructura de los topic	50
Figura 23. Diagrama de bloques sistema de monitoreo.....	52
Figura 24. Esquema de una transferencia eléctrica.....	53
Figura 25 Transferencia nodo Buenos aires.....	53
Figura 26. Planta eléctrica Nodo B Aires	54
Figura 27. Estructura interna de un relé.....	55
Figura 28. Diagrama de la Interfaz alarmas.....	56
Figura 29. Conexión arduino con módulo GSM.....	58
Figura 30. Módulo GSM.....	60
Figura 31. Adecuación niveles TTL	61
Figura 32. Entradas y salidas Arduino.....	62
Figura 33. Banco de baterías nodo B aires	63
Figura 34. Diagrama circuito completo	63
Figura 35. PCB layout.....	64
Figura 36. PCB en 3D.....	64

Figura 37. Comunicación con arduino	65
Figura 38. Algoritmo Arduino	67
Figura 39. Módulo GPRS	68
Figura 40. Prueba módulo GSM	69
Figura 41. Prueba de conexión al módulo	70
Figura 42. Características puerto serial sim800	71
Figura 43. Adaptador de nivel sim800	72
Figura 44. Antenas para módulo GSM	73
Figura 46. Conexión a GPRS	76
Figura 47. Código para la conexión a GPRS	77
Figura 48. Prueba GPRS	78
Figura 49. Librerías para activar SNMP	79
Figura 50. Prueba de conexión a snmp por J F Reinoso, 2018..... 80	
Figura 51. Envío de trap desde arduino	80
Figura 52. Prueba de puertoSNMP	81
Figura 53. Servidor MQTT	82
Figura 54. Librería pubsubclient.....	83
Figura 55. Configuración MQTT.....	83
Figura 56. Conexión a MQTT desde Arduino.	84
Figura 57. Prueba MQTT	84

Figura 58. PCB tarjeta alarmas	85
Figura 59. PCB tarjeta alarmas	86
Figura 60. Conexión PCB tarjeta alarmas.....	86
Figura 61. MQTTBox	87
Figura 62. Recepción de alarmas	88

Lista de tablas

Tabla 1. Modelos de arduino	42
Tabla 2. Funcionamiento del relé.....	55

Resumen

El presente proyecto consiste en el monitoreo remoto de la red eléctrica del nodo Buenos Aires ubicado en el centro del Departamento del Valle del Cauca, cerca al municipio de Buga. Para esto se plantea la posibilidad de utilizar la tarjeta de desarrollo Arduino, junto a un módulo de comunicación GPRS (*General Packet Radio Service*), los cuales permiten el envío de dos señales de alarmas a través de la Internet a un nodo de monitoreo ubicado en la ciudad de Cali, para verificar el estado de la red eléctrica. (Huidobro Moya, 2006, p.190-191)

Para el envío de señales de alarma se utiliza la tarjeta Arduino y el módulo GPRS, se escoge el protocolo MTQ (*Message Queue Telemetry Transport*), el cual ya ha sido implementado en aplicaciones similares y brinda ventajas adicionales. Este protocolo permite la transmisión de dos señales de alarma a través de la red GPRS, con las que se conocerá el estado de la red eléctrica del nodo buenos aires, tal como se plantea en los objetivos del proyecto.

La fase de desarrollo de hardware consiste en la construcción de las etapas necesarias para el funcionamiento del sistema de alarmas. Estas son la etapa de potencia o fuente de alimentación, la etapa de procesamiento de alarmas; la cual está compuesta por los relés y compuertas tipo AND, dicha etapa envía la señal de alarma a la tarjeta Arduino, y la etapa de comunicación la cual posee un módulo GPRS quien realiza la función de transmitir la información procesada por la tarjeta Arduino a través de la red celular. La fase de desarrollo de

software consiste en el desarrollo del código fuente que requiere la tarjeta Arduino para poder procesar los datos provenientes del módulo de Alarmas y enviarlas al módulo GPRS, para esto se requiere el uso de una librería que permita conectarse a la red celular.

Para lograr los objetivos planteados se utilizó la investigación aplicada, la cual es una metodología que se caracteriza por utilizar las teorías o conocimientos científicos actuales para generar nuevos productos que permitan el desarrollo de la sociedad, tal es el caso del presente proyecto, el cual aporta a la sociedad y a las empresas un nuevo producto que ayuda a mejorar procesos a nivel técnico y logístico, logrando así mejorar la productividad y el servicio al cliente. De lo anterior se puede decir que se cumplieron y se lograron los objetivos planteados en el proyecto, obteniéndose un producto que permite monitorear la red eléctrica del nodo buenos aires utilizando la red GPRS la cual será de utilidad para las empresas y para todo aquel que quiera utilizarlo y/o mejorarlo.

Abstract

This work it is about of the remote monitoring of the electrical network node Buenos Aires, located in the Valle del Cauca Department center, near the Buga municipality. For this I think the possibility of using the Arduino development card together with a GPRS communication module, which allows the sending of two alarm signals through the internet network to a node located in the city of Cali, to checking the status of the electrical network.

Of the sending of alarm signals the Arduino card and the GPRS is used, the MTQ protocol is chosen, which has already been implemented in similar applications and offers additional advantages. This protocol allowed the transmission of the two alarm signals through the GPRS network, with which the status of the electricity network of the Buenos Aires node will be known, as it was stated in the project's objectives.

The hardware development phase consists of the construction of the necessary steps for the operation of the alarm system. These are the power stage or power supply, the alarm processing stage which is composed of relays and AND gates, this stage sends the alarm signal to the Arduino card, and the communication stage which has a module GPRS who performs the function of transmitting the information processed by the Arduino card through the cellular network. The software development phase consists of the development of the source code required by the Arduino card to process the data coming from the Alarms module and send them to the GPRS module, for this the use of a library that allows to connect to the cellular network is

required .To achieve the proposed objectives applied research was used, which is a methodology that is characterized by using theories or scientific knowledge to generate new products that allow the development of society, such is the case of the present project, which contributes to society and companies a new product that helps improve processes at a technical and logistic level, thus achieving improving productivity and customer service.

From the above it can be said that the objectives set out in the project were met and achieved, obtaining a product that allows monitoring the electrical network of the Buenos Aires node using the GPRS network, which will be useful for companies and for anyone who wants to use it and / or improve it

Introducción

El presente proyecto nace de la necesidad que tiene la empresa Radionet de realizar el monitoreo remoto de la red eléctrica del nodo Buenos Aires, el cual se encuentra ubicado a más de 3 horas de la ciudad de Cali y presenta fallas constantes, ocasionando dificultades en la atención del evento, primero porque no se cuenta con un dispositivo que informe de manera inmediata a la oficina central ubicada en la ciudad de Cali, sobre la falla del sistema y segundo porque el desplazamiento demanda mucho tiempo y por ende conocer la causa del evento.

Teniendo en cuenta lo anterior se planteó como objetivo la construcción de un sistema de alarmas que permita mediante la red GPRS, conocer si el nodo ha tenido eventos en la parte eléctrica, para lo cual se procesaran dos variables: falla de AC y falla de planta.

Para el desarrollo del proyecto se trabajaron dos fases: La fase de desarrollo de hardware que consiste en la construcción de las etapas (potencia, procesamiento y comunicación) necesarias para el funcionamiento del sistema de alarmas y la fase de desarrollo de software que consiste en el desarrollo del código fuente que requiere la tarjeta Arduino para poder procesar los datos provenientes del módulo de Alarmas y enviarlas al módulo GPRS.

Capítulo 1 Antecedentes y Contexto

1.1 Planteamiento del problema

1.1.1 Situación Problema

El nodo Buenos Aires se encuentra ubicado en el centro del Departamento del Valle del Cauca, en el municipio de Buga, y es uno de los concentradores o repetidores de red que la empresa Radionet utiliza para prestar los servicios de telecomunicación a los clientes de municipios del centro y norte del Valle del Cauca: Buga, Tuluá, San Pedro, entre otros. El concentrador funciona desde hace más de 15 años, y durante este tiempo se ha visto afectado en muchas ocasiones por fallas eléctricas, como la ausencia de tensión en una de las tres fases del circuito principal o la caída total del circuito que alimenta el nodo.

La atención de este evento resulta difícil y demorado por la ubicación de la repetidora, al estar en un cerro, lo que ocasiona que el tiempo de desplazamiento sea alrededor de 3 horas desde la ciudad de Cali, sumado al mal estado de la carretera y temas de orden público, ya que el área es catalogada como zona roja. Esta situación de seguridad implica más tiempo en el desplazamiento de lo normal al tener que buscar vías alternas que incrementan el desplazamiento hasta en 2 horas. Dadas las circunstancias, el repetidor requiere un monitoreo regular y constante, de por lo menos 4 veces en el mes, con el fin de estar verificando que el nodo esté operando normalmente y tomar las medidas para encontrar una falla en la planta o en la red

eléctrica, lo cual acarrea costos de \$300.000 por viaje para desplazamiento desde la ciudad de Cali.

1.1.2 Formulación del Problema

Por lo anterior se plantea la siguiente pregunta:

¿Cómo monitorear de forma remota el estado de la red eléctrica del nodo buenos aires, mediante la tarjeta de desarrollo Arduino y la red GPRS?

1.2 Justificación

Los sistemas de monitoreo son muy utilizados hoy en día en diferentes áreas de la industria, como son las líneas de producción, telecomunicaciones, electricidad, etc. Estos sistemas se construyen para evaluar diferentes situaciones o variables (dependiendo de la exigencia del cliente o de las condiciones del lugar), que permiten tomar decisiones en el corto o mediano plazo, mejorando la productividad de la empresa que requiere del monitoreo. Para esto, existen diferentes métodos o técnicas, siendo algunas de las más utilizadas los sistemas Scada, estas utilizan PLC para recolectar las variables y llevarlas a un servidor que administre la red.

Otra forma de hacer el monitoreo es a través de líneas telefónicas o ADSL esta técnica transporta los datos enviados por una tarjeta electrónica que a su vez captura las variables o señales de los sensores.

En el sector de las telecomunicaciones es muy utilizada la tecnología Ethernet para el monitoreo de alarmas en equipos de comunicación como servidores, Switches y Routers, estos utilizan varios protocolos para leer las variables.

Por lo anterior y debido a que las condiciones para realizar el monitoreo son cada vez más diversas, se realizará el desarrollo de este proyecto usando la red celular como un medio que permita realizar la vigilancia de dos variables para la implementación de un sistema de alarma, con lo cual se podrá conocer si el nodo buenos aires (Buga) cuenta con suministro eléctrico. Se

escoge este método de transmisión debido a que los equipos que operan en el nodo Buenos aires no cuentan con puerto de monitoreo en Ethernet y tampoco existen Switches para realizar una implementación cableada. Otra razón es que los equipos de energía están en cuartos separados por una distancia muy larga y no se cuenta con canalizaciones para conectarlos, por estas razones vemos conveniente utilizar la tecnología celular ya que además minimiza las posibles fallas de cableado.

Para llevar a cabo esto se trabajará en el uso y funcionamiento de la red celular, especialmente en el campo de la transmisión de paquetes de datos vía GPRS, ya que este es uno de los sistemas con mayor disponibilidad en el país y el más asequible como modulo para proyectos de investigación ya que es muy fácil conseguirlo en comercios de electrónica.

Como componente fundamental para este proyecto usaremos la tecnología Arduino, esta tecnología es hoy en día una de las impulsoras del desarrollo tecnológico debido a su facilidad de manejo, así como a su comunidad de colaboradores que día tras día sacan y actualizan nuevas librerías para poder conectar a Arduino con el mundo exterior.

Para este proyecto Arduino cuenta con unas librerías que permiten trabajar con GPRS y el protocolo de comunicación MQTT lo cual no es fácil de hacer en otras plataformas como Microchip o Rasberry ya que las mismas exigen diseñar las librerías necesarias para el proyecto o en su defecto modificar librerías similares, lo cual requiere de buenos conocimientos en programación.

También se trabaja en el uso de Arduino y su comunicación con el módulo GPRS, debido a su facilidad de manejo ya que posee las librerías necesarias para trabajar con los módulos existentes y además de utilizar lenguaje de alto nivel lo que permite una fácil programación e integración con otras plataformas de software.

Por lo anterior este proyecto contribuirá a mejorar o desarrollar nuevas líneas de investigación en el campo de la telemática, la domótica y el IoT (internet de las cosas) el cual cada día incorpora nuevos dispositivos con capacidad de monitoreo y de configuración debido al crecimiento de la internet.

Como resultado podremos obtener una nueva forma de hacer el monitoreo de variables eléctricas a través de la red celular utilizando la tecnología Arduino, y que además permitirá mejorar las condiciones de operación y servicio prestado por este nodo en los municipios del norte del valle. Estos son: Buga, San pedro, Tuluá, Zarzal.

1.3 Objetivos

1.3.1 Objetivo General

Construir un sistema de monitoreo de alarmas mediante el uso de la tarjeta Arduino y la red GPRS, que permita supervisar el estado de la red eléctrica del nodo Buenos aires ubicado en el cerro buenos aires (Buga).

1.3.2 Objetivos Específicos

- Diseñar la interfaz de alarmas para conectar al módulo Arduino.
- Programar y configurar el módulo Arduino para que procese dos alarmas (falla AC, falla planta)
- Evaluar el funcionamiento y comunicación de las alarmas entre el módulo Arduino y la herramienta de gestión

Capítulo 2 Revisión de la Literatura

2.1 Marco Contextual

El monitoreo o gestión de alarmas ha sido un área exclusiva de las empresas en especial de las industrias, y por lo tanto sus aplicaciones suelen ser muy específicas, llegando al punto de solo disponer de poca tecnología para implementar estos sistemas, es por ello que actualmente y gracias a internet este paradigma ha empezado a cambiar ya que empiezan a surgir nuevas alternativas para realizar el monitoreo de alarmas ya no solo en la industria sino en el hogar(Llaneza, G, 2018) , algunos ejemplos son los siguientes:

Sistema de monitoreo para pacientes de alto riesgo (Olguer Sebastián, et al., 2015), este sistema tiene como función mejorar el servicio que prestan los hospitales a los pacientes más críticos en la ciudad de Bogotá, para lo cual se desarrolló un sistema que toma la variable ECG (electrocardiograma) del paciente y las envía vía GPRS a un computador con interfaz gráfica LabVIEW, la cual se va almacenando en una base de datos para su posterior análisis.

Monitoreo de humedad en suelo a través de sensores inalámbricos (Flórez, medina, et al, 2015) esto se realiza con el fin de hacer un uso eficiente del agua ya que este es un recurso escaso y por lo tanto se requiere de un sistema que permita cumplir con este objetivo. Para ello se desarrolló un sistema que integra sensores de húmeda los cuales envían datos a través de GPRS.

Monitoreo, control y diagnóstico en bancos de capacitores (Peña, W, et al., 2018) este sistema se desarrolló para hacer un control más preciso de los bancos de condensadores y así poder controlar el factor de potencia en las empresas ya que este repercute en la facturación de energía incrementando los gastos operacionales.

Aunque en las bases de datos utilizadas para este proyecto no se encontró información sobre aplicaciones o desarrollos similares a este proyecto, en el comercio si se encuentran desarrollos similares en especial los utilizados para la seguridad privada o seguridad perimetral. Estos desarrollos han sido implementados por las propias empresas de seguridad aquí en Colombia y en otros países, pero los mismos solo se dedican a monitorear puertas, ventanas y cámaras de seguridad, estos no contemplan el monitoreo de equipos de energía ni de redes ya que no es su especialidad.

En el caso de las empresas de telecomunicación los sistemas de monitoreo son muy comunes, pero los mismo tienen como objetivo la vigilancia de la red, en aspectos propios de esta área como son las caídas del sistema o la medición de tráfico. En ocasiones se utilizan estos mismos sistemas para monitorear equipos de energía que utilicen el protocolo SNMP (*Simple Network Management Protocol*), pero esto limita la posibilidad de utilizar otras tecnologías.

2.2 Marco Teórico

2.2.1 Monitoreo y Gestión de Red

El monitoreo es un término muy empleado en el área de las telecomunicaciones y la telemática, el cual se refiere a la posibilidad que tienen muchos equipos de comunicación como Routers y Switches, de enviar información sobre su estado de funcionamiento general o particular, así como de los servicios que presta a un cliente. Esto quiere decir, que es posible visualizar desde un lugar remoto, el estado de un Router o Switch que esté conectado a una red privada o pública, lo que permitirá hacer un diagnóstico aproximado de las fallas que se puedan presentar en el equipo monitoreado.

Para entender mejor lo que es el monitoreo vamos a definir algunos conceptos clave que nos permitan entender mejor como funciona un sistema de monitoreo o gestión de red.

- **Entidad gestora.** Es el equipo (pueden ser varios equipos) donde se encuentra el software (servicio administrador de red) que se utiliza para controlar la gestión de la red. Será el punto desde donde trabaja el administrador de la red. Ese software realiza las siguientes tareas:
 - Recoger información de los dispositivos gestionados.
 - Procesar y analizar la información obtenida.
 - Presentar dicha información para su interpretación y para la toma de decisiones.

- Dispositivos gestionados.** Contienen los objetos gestionados. Por ejemplo, puede tratarse de un equipo (dispositivo gestionado) que contiene diversos elementos que se quieren gestionar, como la CPU (*Central Unit Processor*), la tarjeta de red, la memoria, etc. La entidad gestora obtiene y, a lo mejor, modifica la información de esos objetos. Para hacer esto, en el dispositivo gestionado se encuentra el agente de gestión, el software que permite la gestión remota en el dispositivo gestionado. Un dispositivo gestionado puede ser cualquier equipo que se pueda conectar a la red y comunicarse, cualquier ordenador, impresora, servidor, etc.
- Protocolos de gestión.** Son los que establecen las reglas de comunicación entre la entidad gestora y los agentes de gestión. Mediante estas reglas se definen los tipos de mensajes, las secuencias de comandos que se pueden emplear y la seguridad a utilizar en la comunicación. (Calvo García, 2014, p.32-95)

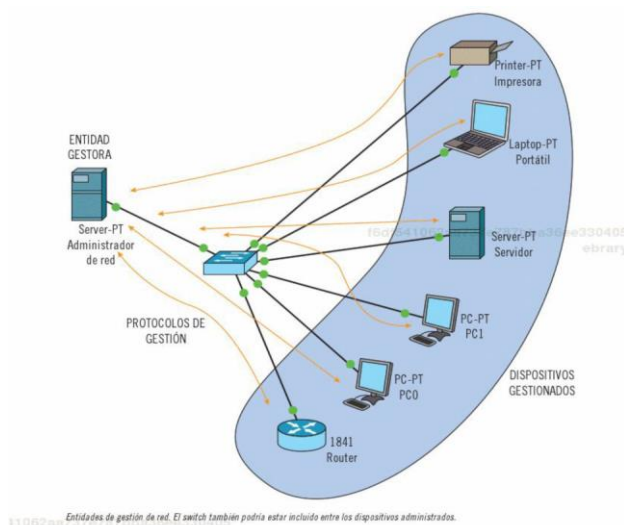


Figura 1. Elementos de la gestión de red. Copyright 2014 por Gestion de redes telematicas.

Se puede concluir a partir de las definiciones anteriores, que un sistema de gestión requiere de varias entidades para su operación. Estas entidades han de poder comunicarse eficientemente para no confundir al administrador de la red, por lo que además del software de gestión o monitoreo se deberá utilizar el protocolo adecuado según las necesidades de la empresa o proyecto.

A continuación, profundizaremos más sobre los protocolos y sus funciones en una red de gestión.

2.2.1.1 Funciones de los protocolos de gestión

- **Administrar el rendimiento:** en la administración de la red, vigilar el rendimiento de la misma es una de sus funciones principales. Cualquier usuario demanda ante todo un buen rendimiento, y eso es lo que le exigirá al responsable de la administración
- **Monitorizar:** es indispensable para realizar lo que se le exige. Realmente, es de donde parte casi todo el conocimiento de la red. Podrá averiguar:
 - Nivel de utilización de cada enlace.
 - Tipos de tráfico que circulan por la red (voz IP, datos normales, telefonía, video, etc.).
 - Si se pierden paquetes de información.
 - Utilización de los recursos. Cómo y cuándo se utilizan los diferentes recursos.
- **Analizar:** del análisis se podrán sacar conclusiones.
 - Si hay abusos de recursos.
 - Si hay tráfico inusual.

- Cuáles son los servicios más demandados.
 - Valores para establecer QoS (calidad de servicio).
 - Control de tráfico para balancear carga por enlaces menos demandados/saturados.
- **Administrar fallas:** arreglar problemas. Pruebas de diagnóstico y localización de fallas.
 - **Administrar Informes:** para toma de decisiones.
 - **Administrar seguridad:** para que los servicios funcionen como se espera.
 - **Administrar alarmas:** establecer alarmas de seguridad y contra fallas. (Calvo García, 2014, p.121)

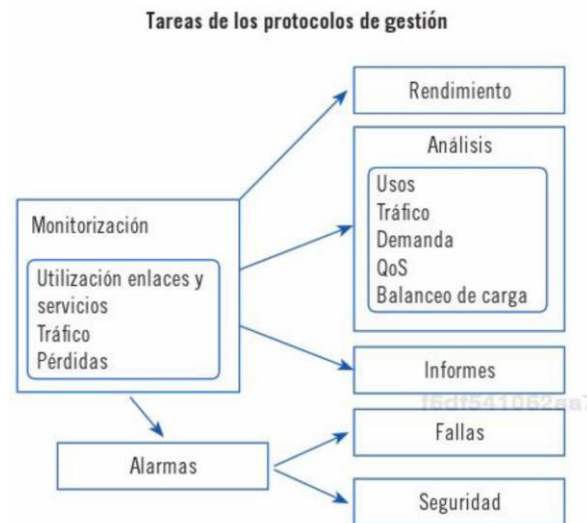


Figura 2. Tareas de los protocolos de gestión. Copyright 2014 por Gestion de redes telematicas.

2.2.1.1 Estándares de Gestión

Para realizar las tareas de gestión vistas anteriormente se han desarrollado varios estándares. Entre los más destacados están:

Common Management Information Protocol (CMIP)

“Es un protocolo de gestión desarrollado por ISO que se basa en los servicios aportados por CMISE (*Common Management Información Service Element*)”. El protocolo CMIP también se denomina CMOT (*CMip over TCP/IP*). (Calvo García, 2014, p.124).

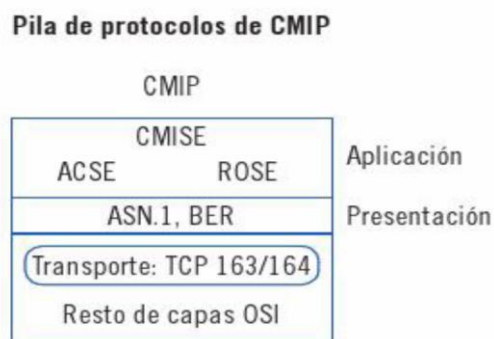


Figura 3. Pila de protocolos CMIP. Copyright 2014 por Gestion de redes telematicas.

Las características de CMIP son:

- Trabaja con la estructura NMS
- Utiliza TCP (protocolo orientado a conexión lo que lo hace más complejo)
- Trabaja en la capa alta del modelo OSI

- Estructura distribuida
- Puerto de escucha: 164 agente y 163 administrador
- Permite múltiples peticiones de un solo mensaje
- Utiliza MIB((*Management Information Base*))
- Se considera costosa su implementación

Simple Network Management Protocol (SNMP)

Es el protocolo de gestión más implantado, quizá por ser nativo de TCP/IP, el protocolo de internet. “SNMP (*Simple Network Management Protocol*) sigue la filosofía de cierta simplicidad en los encabezados de datos, pero ello le pone ciertas trabas, de manera que a veces se complica su configuración dependiendo del nivel de control que se desea sobre la red. Se define en las RFC de la 3410 a la 3418 en las versiones más recientes” (Calvo García, 2014, p.126).

Las características de SNMP son las siguientes:

- Trabaja con la estructura NMS
- No es orientado a conexión
- Arquitectura cliente-servidor
- Trabaja en la capa más alta del modelo OSI (aplicación)
- Utiliza los puertos 161 y 162 UDP
- No realiza solicitudes múltiples
- Utiliza MIB((*Management Information Base*))

- Su implementación es económica, pero su ampliación costosa.

Pila de protocolos de SNMP

SNMP	Aplicación
UDP, puerto 161/162	Transporte
Resto de capas TCP/IP	

Figura 4. Pila de protocolos SNMP. Copyright 2014 por Gestion de redes telematicas.

Estos protocolos son los más utilizados en la actualidad especialmente en el campo de la informática ya que funcionan en la capa de aplicación del modelo OSI, lo que permite una implementación más rápida, aunque no menos costosa, dependiendo de las necesidades que requiera el administrador de red.

Otro protocolo que no comentamos es el RMON, el cual está basado en SNMP el cual suple algunas carencias que tiene este pero que no es muy utilizado por los costos de implementación, ya que se requiere mínimo una sonda por cada subred.

2.2.1.2 Herramientas de monitoreo

En el mercado existen muchas herramientas de monitoreo, las cuales trabajan con uno o varios de los protocolos anteriormente mencionados. Además, poseen características particulares que los hacen mejor en algunas situaciones.

A continuación, se expondrán algunas de las herramientas de código abierto más usadas actualmente y sus características.

Pandora FMS

Pandora FMS es un software de monitorización para gestión de infraestructura TI. Esto incluye equipamiento de red, servidores Windows y Unix, infraestructura virtualizada y todo tipo de aplicaciones. Pandora FMS tiene multitud de funcionalidades, lo cual lo convierte en un software de nueva generación que cubre todos los aspectos de monitorización. Entre los que se encuentran:

- Monitorización de redes (SNMP, WMI,ICMP)
- Control remoto de equipos de red
- Consola personalizable
- Gestión de errores y eventos
- Tecnología de autodescubrimiento de redes
- Monitorización de traps

- Geolocalizacion
- Escalabilidad (maneja hasta 10.000 servidores)
- Monitoreo netflow
- Agentes multiplataforma (WINDOWS,LINUX,AIX,HP, SOLARIS,BSD)
- Control de acceso basado en roles
- Agentes para android y sistemas empotrados

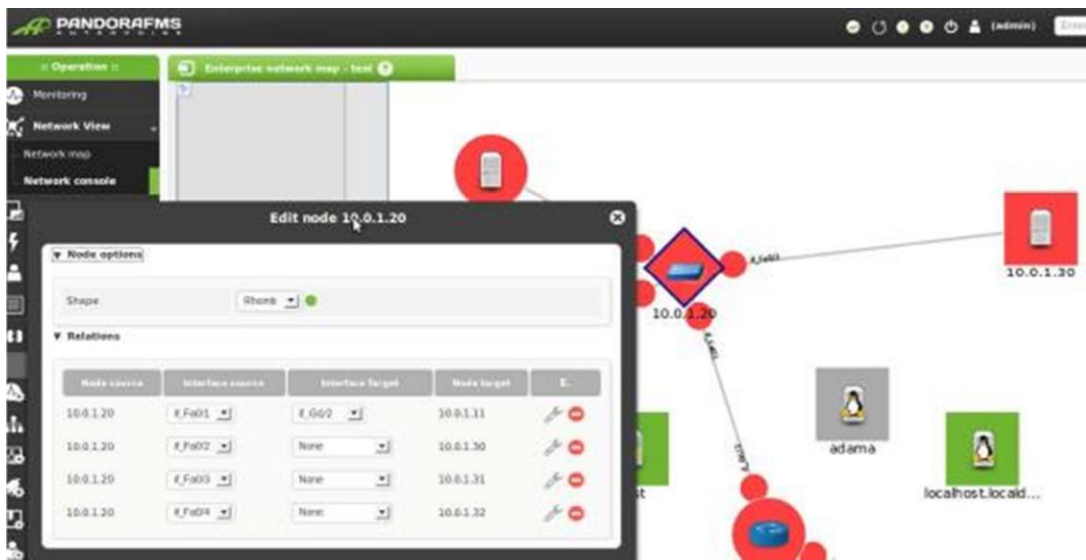


Figura 5. Interface de FMS. Copyright 2018 por Pandora FMS Network Management Review.

Zabbix

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

- Chequeos simples que pueden verificar la disponibilidad y el nivel de respuesta de servicios estándar como SMTP o HTTP sin necesidad de instalar ningún software sobre el host monitorizado.
- Un agente Zabbix puede también ser instalado sobre máquinas UNIX y Windows para monitorizar estadísticas como carga de CPU, utilización de red, espacio en disco, etc.
- Como alternativa a instalar el agente sobre los host, Zabbix incluye soporte para monitorizar vía protocolos SNMP, TCP y ICMP, como también sobre IPMI, JMX, SSH, telnet y usando parámetros de configuración personalizados. Zabbix soporta una variedad de mecanismos de notificación en tiempo real, incluyendo XMPP.

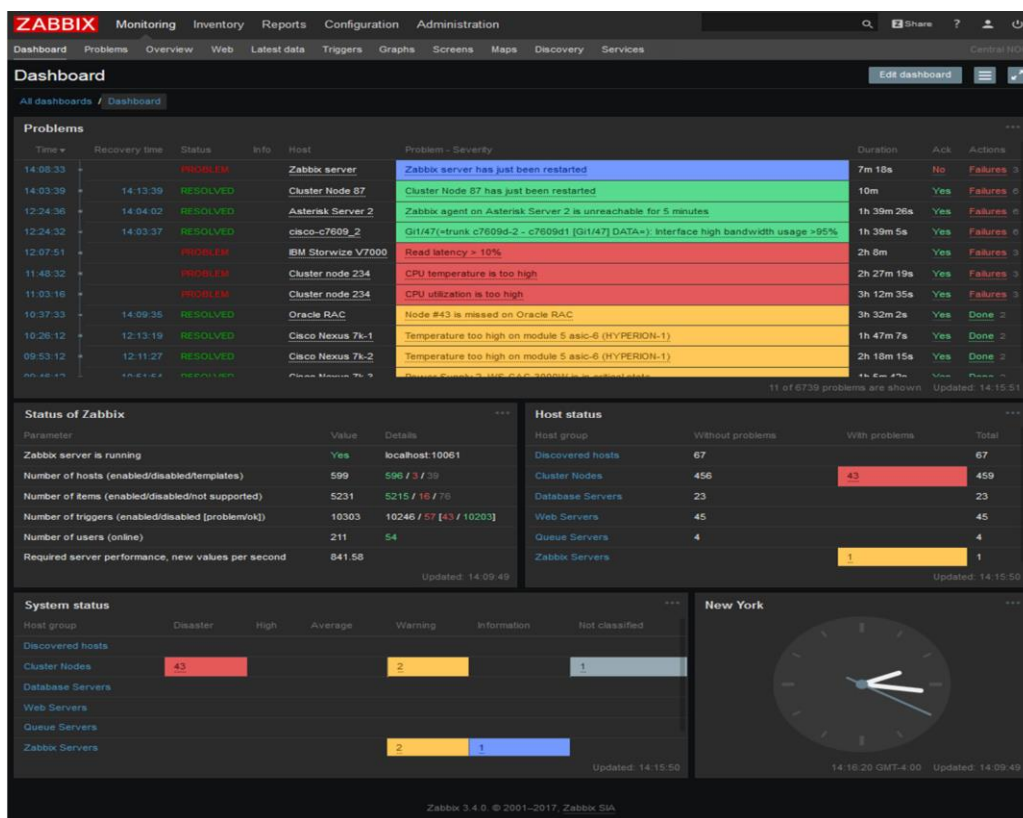


Figura 6. Interface de zabbix. Copyright 2018 por Zabbix.

Nagios

Nagios es un sistema de monitorización de redes ampliamente utilizado, de código abierto, que vigila los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado. Entre sus características principales figuran la monitorización de servicios de red (SMTP, POP3, HTTP, SNMP...), la monitorización de los recursos de sistemas hardware (carga del procesador, uso de los discos, memoria, estado

de los puertos...), independencia de sistemas operativos, posibilidad de monitorización remota mediante túneles SSL cifrados o SSH, y la posibilidad de programar plugins específicos para nuevos sistemas.

Se trata de un software que proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los responsables correspondientes mediante (entre otros medios) correo electrónico y mensajes SMS, cuando estos parámetros exceden de los márgenes definidos por el administrador de red.

Llamado originalmente Netsaint, nombre que se debió cambiar por coincidencia con otra marca comercial, fue creado y es actualmente mantenido por Ethan Galstad, junto con un grupo de desarrolladores de software que mantienen también varios complementos.

Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix y ofrece los siguientes servicios:

- Monitorización de servicios de red (SMTP, POP3, HTTP, NNTP, ICMP, SNMP).
- Monitorización de los recursos de equipos hardware (carga del procesador, uso de los discos, logs del sistema) en varios sistemas operativos, incluso Microsoft Windows con los plugins NRPE_NT o NSClient++.
- Monitorización remota, a través de túneles SSL cifrados o SSH.

- Diseño simple de plugins, que permiten a los usuarios desarrollar sus propios chequeos de servicios dependiendo de sus necesidades, usando sus herramientas preferidas (Bash, C++, Perl, Ruby, Python, PHP, C#...).
- Chequeo de servicios paralizados.
- Posibilidad de definir la jerarquía de la red, permitiendo distinguir entre host caídos y host inaccesibles.
- Notificaciones a los contactos cuando ocurren problemas en servicios o hosts, así como cuando son resueltos (a través del correo electrónico, buscapersonas, Jabber, SMS, o cualquier método definido por el usuario junto con su correspondiente complemento).
- Posibilidad de definir manejadores de eventos que ejecuten al ocurrir un evento de un servicio o host para resoluciones de problemas proactivas.
- Rotación automática del archivo de registro.
- Soporte para implementar hosts de monitores redundantes.
- Visualización del estado de la red en tiempo real a través de interfaz web, con la posibilidad de generar informes y gráficas de comportamiento de los sistemas monitorizados, y visualización del listado de notificaciones enviadas, historial de problemas, archivos de registros.

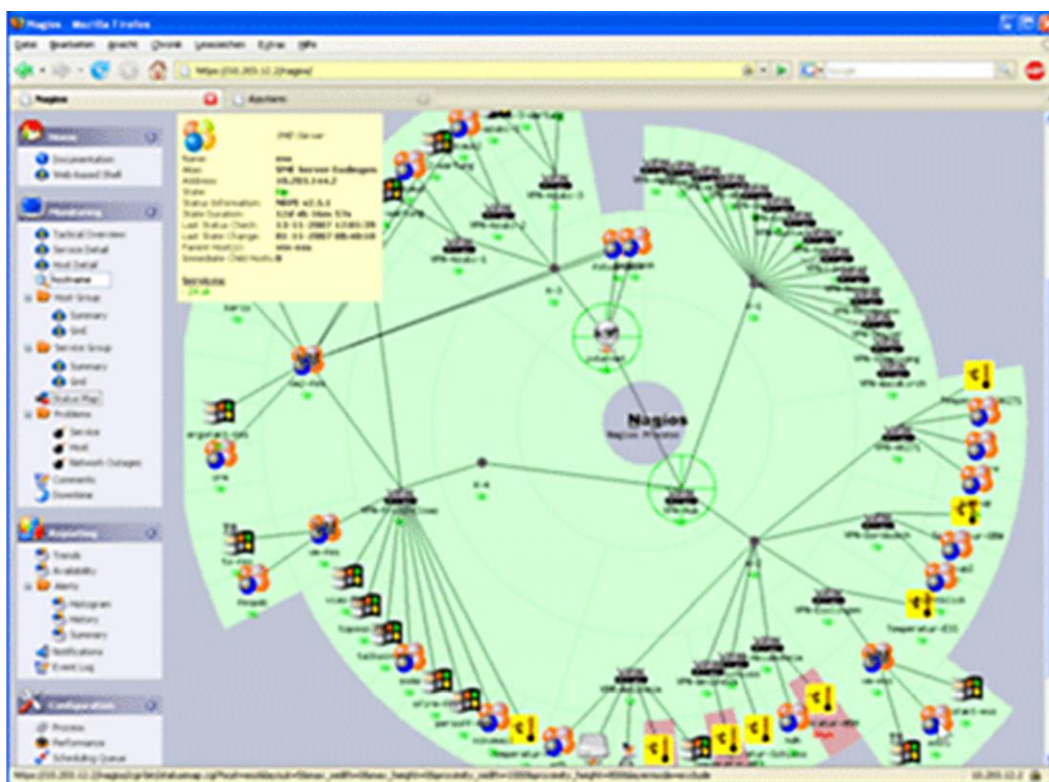


Figura 7. Interface de Nagios. Copyright 2018 por Nagios

2.2.2 Redes Celulares

Las redes celulares son sistemas de telecomunicaciones inalámbricos que se caracterizan por su movilidad, esto quiere decir que se puede establecer una comunicación desde cualquiera de los puntos de la red y esta continuara así se desplace hacia otro punto de la misma red.

Sus orígenes se dieron en estados unidos gracias a AT&T quienes desarrollaron el primer sistema de telefonía móvil conocido como AMPS (*sistema de telefonía móvil avanzado*) y que luego empezó a funcionar en la ciudad de chicago.

Con el tiempo estas redes han ido evolucionando debido a sus limitaciones en ancho de banda, y de prestación de servicios. Por lo que, en sus inicios, se desarrollaron varios estándares que aportaran soluciones en cada una de las regiones o países donde se desarrollaron. Esta evolución se ha dividido en generaciones, las cuales se han enumerado así:

- **Generación (1G):** La primera generación de redes celulares se caracterizó por ser totalmente analógica, lo cual represento problemas para su ampliación y cobertura y solo podía prestar servicios de voz. Los estándares más usados fueron **NMT** (telefonía móvil nórdica), **TACS** (sistema de comunicaciones de acceso total) y **AMPS**. (Figuroa, D. L. C. M. (2008) p.14)
- **Generación (2G):** La segunda generación de redes celulares se caracteriza por utilizar técnicas digitales para la transmisión de datos. Esto permitió aumentar la capacidad de usuarios por celda, así como la mejora en la calidad de los servicios. Los estándares desarrollados para esta generación son: **GSM** (sistema global de comunicaciones móviles), **D-AMPS** o **TDMA** (acceso múltiple por división de tiempo), **CDMA** (acceso múltiple por división de código) y **PDC** (comunicaciones digitales del pacifico). (Figuroa, D. L. C. M. (2008 p.15)

De estos estándares el de mayor acogida fue **GSM**, el cual al igual que los otros estándares ofrece las siguientes ventajas:

- Mayor calidad de voz
- Menores costos de operación

- Mayor nivel de seguridad
- Roaming internacional
- Soporte para terminales de menor potencia
- Variedad de servicios

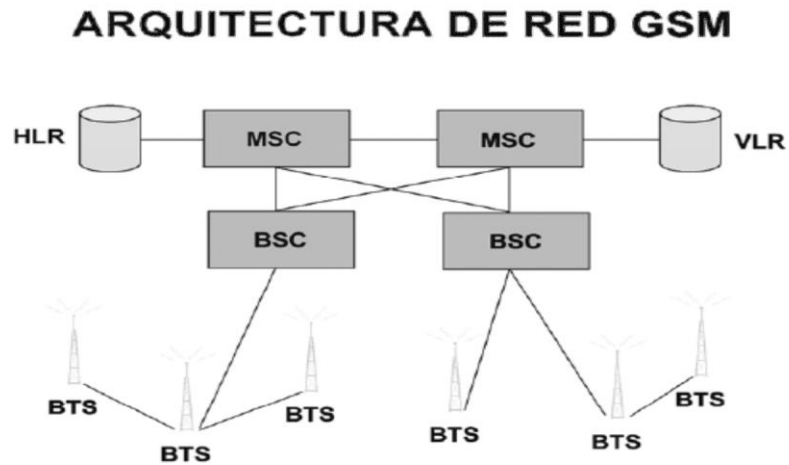


Figura 8. Arquitectura de la red GSM. Copyright 2006 por Redes y servicios de telecomunicaciones.

- **Generación (3G):** La tercera generación de redes celulares ayuda a solucionar los dos problemas principales que tenía la red celular 2G. Primero el agotamiento de espectro radioeléctrico, lo cual no permitía atender la demanda de usuarios demandantes. Segundo los servicios, esta nueva red permitió ampliar la cantidad y calidad de los servicios ofrecidos por los operadores. Además de telefonía, se ofrece el servicio de mensajería SMS y el servicio de datos bajo el protocolo IP, lo cual permitió la integración de servicios; logrando así que los usuarios disfrutaran de los servicios de audio e imágenes.

Para lograr esta convergencia de servicios 3G mejoro la velocidad de transferencia en los canales de datos al pasar de 144Kbps a 384Kbps típicamente con picos de 2Mbps. (

Figueroa, D. L. C. M. (2008 p.19)

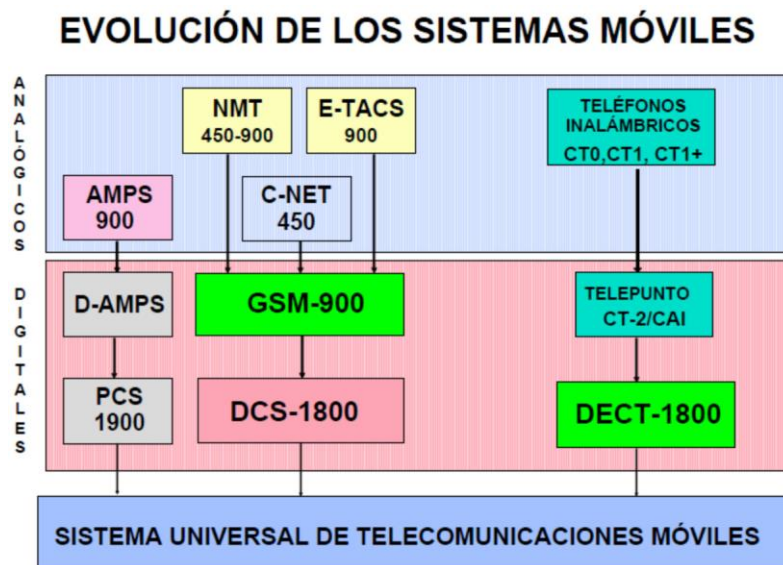


Figura 9. Evolución de la telefonía móvil. Copyright 2006 por Redes y servicios de telecomunicaciones.

- **Generacion (4G):** Actualmente las redes celulares trabajan con la tecnología de cuarta generación (**4G**), la cual permite transferencias de datos aún mayores a 3G, obteniéndose velocidades de hasta 10Mbps. Esto permite a los usuarios disfrutar de servicios multimedia como los videos y los juegos en línea. Esta generación de tecnología se caracteriza por mejorar la velocidad de transferencia de datos debido a la proliferación de nuevos servicios basados en internet. (Figueroa, D. L. C. M. (2008 p.26)

2.2.2.1 Arquitectura de una red GSM

La estructura de la red del sistema GSM es una estructura de tipo jerárquico, en la que los diferentes elementos están conectados mediante interfaces definidos en el estándar. En la Figura 11 se muestran los diferentes elementos y sus interfaces. A continuación, se enumeran los elementos de la estructura y se muestra el significado de las siglas:

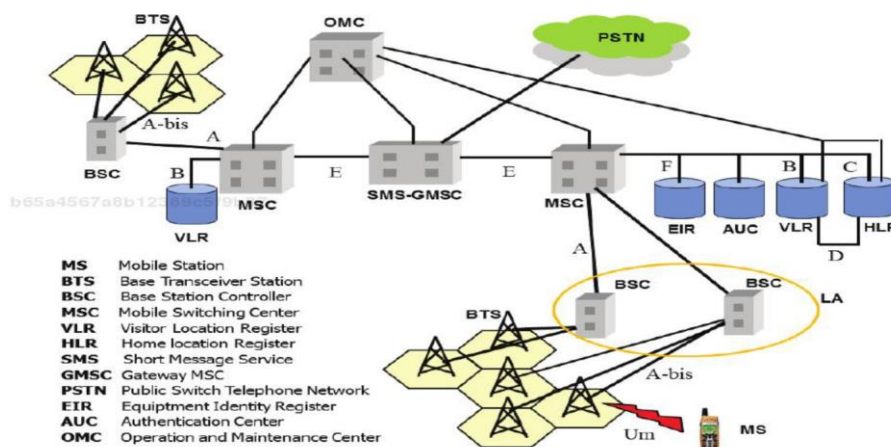


Figura 2.22. Estructura de la red GSM [GSM Switching, Services and Protocols]. En algunos textos el AuC se coloca conectado al HLR, sin conexión directa con el MSC [GSM, cdmaOne and 3G Systems, Comunicaciones Móviles 02].

Figura 10. Esquema de una red GSM. Copyright 2014 por Sistemas de comunicaciones móviles: segunda, tercera y cuarta generación.

- **Estación móvil (MS, “Mobile Station Equipment”)**: Es el dispositivo móvil con el que el usuario se conecta a la red a través del interfaz Um. Este interfaz describe el enlace radio entre la MS y el siguiente elemento de la red, el BTS. (Huidobro Moya, J. M. (2006) p.196)

- **Estación Base (BTS, “Base Transceiver Station”)**: Son los equipos de radiofrecuencia, antenas y los equipos electrónicos que dan servicio a una célula o a varias células (célula sectorizada). La palabra “transceiver” hace referencia a la capacidad de la BTS de transmitir y de recibir. Se conecta con el interfaz Um con la MS y mediante el interfaz A-bis con el BSC. (Huidobro Moya, J. M. (2006) p.196)
- **Controlador de estaciones base (BSC, “Base Station Controller”)**: Como su propio nombre indica controla varias estaciones base. Se conecta a sus BTS mediante el interfaz A-bis y su MSC a través del interfaz A. (Huidobro Moya, J. M. (2006) p.196)
- **Área de localización (LA, “Location Area”)**: No es propiamente un elemento sino una agrupación de BSC, pero lo mencionamos debido a su importancia en la búsqueda de MS como veremos más adelante. (Huidobro Moya, J. M. (2006) p.196)
- **Centro de conmutación de servicios móviles (MSC, “Mobile Service Switching Center”)**: Es el encargado de realizar casi todas las funciones de conmutación. Es un elemento con múltiples conexiones. Se conecta con sus BSC a través del interfaz A. Además, se conecta con el SMS-GMSC a través del interfaz E, al HLR (si posee uno) mediante el interfaz C y a su VLR mediante el interfaz B. (Huidobro Moya, J. M. (2006) p.196)
- **Registro de localización local (HLR, “Home Location Register”)**: Es una base de datos que contiene información de todos los usuarios abonados de la red móvil. Se conecta mediante el interfaz D al VLR y mediante el interfaz C al SMS-GMSC.

Puede existir uno o varios HLR en una red GSM. (Huidobro Moya, J. M. (2006) p.196)

- **Registro de localización de visitantes (VLR, “Visited o Visitor Location Register”)**: Es una base de datos que contiene información sobre los abonados que se encuentran actualmente en alguna de las LA que forman parte del MSC, al que está asociado el VLR. Se conecta con otros VLR mediante el interfaz G y con su MSC mediante el interfaz B. (Huidobro Moya, J. M. (2006) p.196)
- **Puerta de Enlace o Pasarela-Centro de conmutación de servicios móviles (“SMS-GMSC” Gateway Mobile Service Switching Center)**: es un MSC especial que se encarga de la comunicación con la red de telefonía fija (PSTN “Public Switching Telephone Network”). También se encarga de la gestión de los SMS. Se conecta a los MSC a través del interfaz E. (Huidobro Moya, J. M. (2006) p.196)
- **Centro de operación y mantenimiento (OMC, “Operation and Maintenance Center”)**: Se encarga de las operaciones de mantenimiento y realiza tareas de gestión. Posee conexiones con todos los elementos de la red jerárquica. (Huidobro Moya, J. M. (2006) p.196)
- **Registro de identificación de registros (EIR, “Equipment Identity Register”)**: Es una base de datos que guarda información sobre los equipos móviles de los abonados a la red. (Huidobro Moya, J. M. (2006) p.196)

- **Centro de Autenticación (AUC, “Authentication Center”)**: Es una base de datos que también posee equipos para realizar cálculos de autenticación de los usuarios abonados a la red. (Huidobro Moya, J. M. (2006) p.196)

Asimismo, se explican con detalle las funciones y propiedades de cada uno de los elementos de la red. Como se puede apreciar en la figura 11 una red celular como GSM es compleja de entender a primera vista, ya que tiene muchos elementos los cuales cumplen diferentes funciones para poder establecer una comunicación óptima entre el terminal móvil o (MS) y la red celular. Las funciones más importantes para establecer la comunicación con una red celular son:

- **Identificación de suscriptor**: Esto se hace a través de la SIMCARD que se asigna al usuario la cual se instala luego al teléfono móvil o MS.
- **Identificación del MS**: Esto consiste en identificar cada uno de los equipos que se conectaran a la red celular, para lo cual cada equipo debe poseer un número que lo identifique. Conocido como IMEI.
- **Área de localización**: Esta consiste en informar a la red donde se encuentra ubicado un usuario a través de su SIMCARD.

La **BTS** es el elemento que nos permite conectarnos a la red celular. Esta consta de una estructura llamada torre de telecomunicación, la cual contiene las antenas y equipos de

transmisión que conectan a los usuarios de la red celular, los cuales son agrupados en áreas llamadas células, que a su vez se agrupan para cubrir áreas más grandes dentro de un territorio, ya sea este rural o urbano.

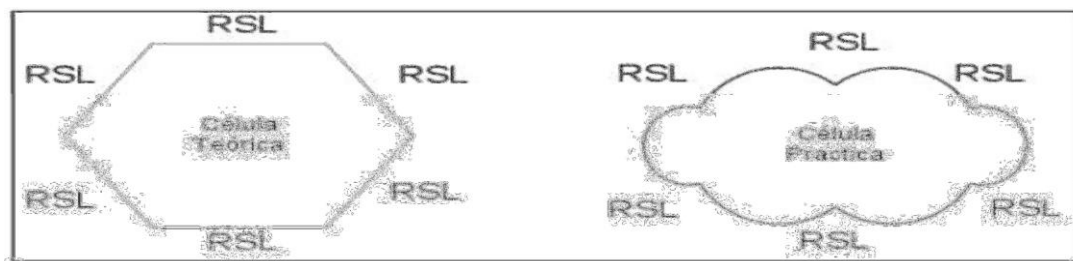


Figura 11. Esquema de una célula teórica vs real. Copyright por Introducción a los sistemas de telefonía celular.

Como se puede ver en la figura 13 la **BTS** es el elemento más visible para los usuarios, ya que este está ubicado relativamente cerca al usuario, y está diseñada para cubrir áreas pequeñas o medianas dependiendo de la cantidad de usuarios que tenga el territorio donde se presta el servicio, por esta razón en ambientes urbanos es común verlas en avenidas o edificios pequeños para cubrir el área requerida.



Figura 12 BTS. Copyright 2018 por Estudios sanitarios sobre radio frecuencias.

Como se mencionó anteriormente, la BTS cumple varias funciones, pero la más importante es la comunicación con el terminal móvil o MS, para lo cual, se establecieron las bandas de RF en que puede operar una red celular. Estas bandas en Colombia son de 850 MHz y 1900 MHz para los servicios de 3G y 1700 y 2600 MHz para 4G. Por lo tanto, solo los terminales que trabajen en estas bandas, podrán conectarse a la red celular y establecer una comunicación con otro usuario, ya sea que este en la misma área o fuera de ella.

Debido a que en los sistemas celulares se debe hacer reusó de frecuencias, por la limitación del espectro electromagnético, cada radio basé (BTS), debe hacer un control de potencia de su señal radiada para evitar el solapamiento de frecuencias entre BTS. Esto se hace, con el fin de, garantizar una buena calidad de la señal para el usuario y a su vez permite reducir el consumo energético de los terminales móviles, los cuales cuentan con baterías pequeñas, permitiendo que los terminales puedan operar por largo tiempo.

2.2.2.2 General Paquet Radio Service (GPRS)

Actualmente las redes celulares prestan muchos servicios, esto se debe a que las exigencias del mercado han obligado a que las redes celulares ofrezcan no solo los servicios tradicionales de voz, sino nuevos servicios de valor agregado como internet y voz sobre IP. Estos nuevos servicios tienen actualmente mayor demanda que los servicios de voz, por esta razón, los fabricantes de tecnología celular, han tenido que buscar maneras de solucionar el principal problema para prestar estos servicios, como lo es la tasa de transmisión de datos y ha desarrollado diferentes soluciones para ello. Una de las primeras fue el sistema GPRS (*servicio*

general de paquetes por radio), el cual permitió incrementar la velocidad de 9.6 Kbps que ofrecían los sistemas 2G a 115 Kbps, en lo que se conoce como la evolución de GSM.

EVOLUCIÓN DE GSM (DATOS)

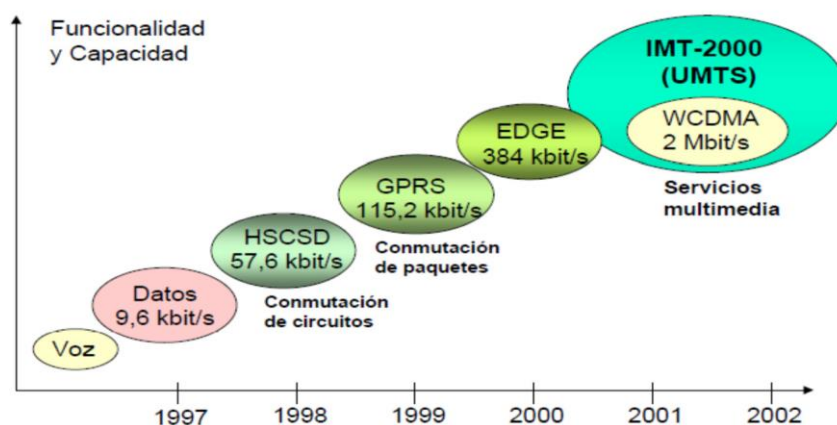


Figura 13. Evolución de GSM (Datos). Copyright 2018 por Redes y servicios de telecomunicaciones.

Las siglas GPRS corresponden a General Packet Radio Service. Se basa en, la conmutación de paquetes realizando la transmisión sobre la red GSM. Esto permite que, los recursos de red se compartan entre los usuarios del servicio permitiendo de esta manera llegar a más usuarios.

Para tener acceso a este servicio, solo es necesario estar afiliado a un operador de telefonía celular, ya que la tecnología GPRS aprovecha la infraestructura de las redes GSM. Razón por la cual, la implementación ha sido fácil de realizar, al reutilizar la mayoría de

elementos como los sistemas radiantes (radios) y solo agregar dos nuevos elementos como el SGSN (Serving GPRS Support Node, en español Nodo de Soporte GPRS) y GGSN (Gateway GSN, en español Interfaz hacia las redes de paquetes de datos externos), los cuales se pueden observar en la figura 16, donde se puede apreciar la arquitectura de la red GPRS.

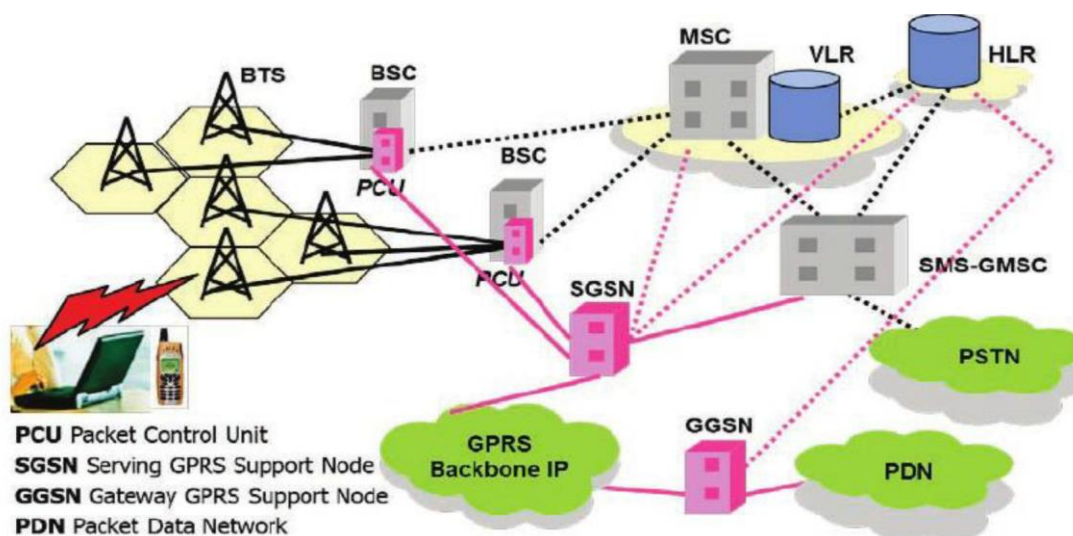


Figura 14. Esquema de una red GPRS. Copyright 2017 en sistemas de comunicaciones móviles segunda tercera y cuarta generación.

La característica que más destaca a esta tecnología, es que trabaja bajo el protocolo IP, lo que permite aún más su integración y operación con muchos servicios o aplicaciones que utilizan este protocolo como medio de transporte. Por lo tanto, se deben cumplir con unos requisitos para poder conectarse a esta red, esto se conoce como el contexto previo a la conexión. Este define la asignación de dirección IP y el tipo de cifrado que tendrá la comunicación.

2.2.3 Arduino

Arduino es una plataforma de hardware y software que permite desarrollar proyectos a la medida del usuario, gracias a su versatilidad y fácil configuración. El mismo, integra un micro-controlador tipo AVR fabricado por ATMEL y montado sobre una tarjeta o PCB, que está diseñada para usar cada una de las entradas y salidas de este microcontrolador, así como una interface de comunicación serial, que se utiliza para comunicarse con el microcontrolador utilizando la interface de desarrollo Arduino o IDE. (Bayle, J. (2013) p.xx)

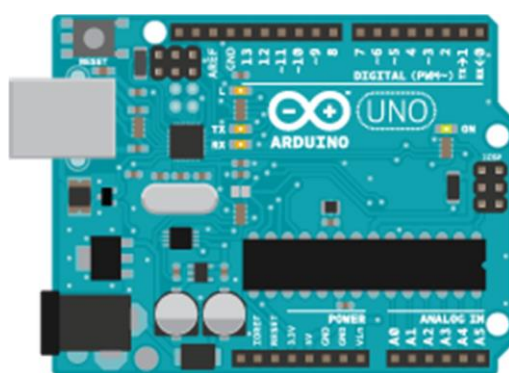


Figura 15. Tarjeta Arduino Uno. Copyright 2017 por <https://www.arduino.cc/>

Una de las grandes ventajas de esta plataforma es su facilidad de programación, ya que está basado en lenguaje C++, el cual, es un lenguaje muy conocido y de fácil comprensión. Por lo tanto, realizar una aplicación sencilla no tomara mucho tiempo, debido a que su lenguaje es de fácil comprensión, al contar con una amplia comunidad, que da soporte a la plataforma. También es posible utilizarse en múltiples plataformas como (Windows, Mac y Linux).

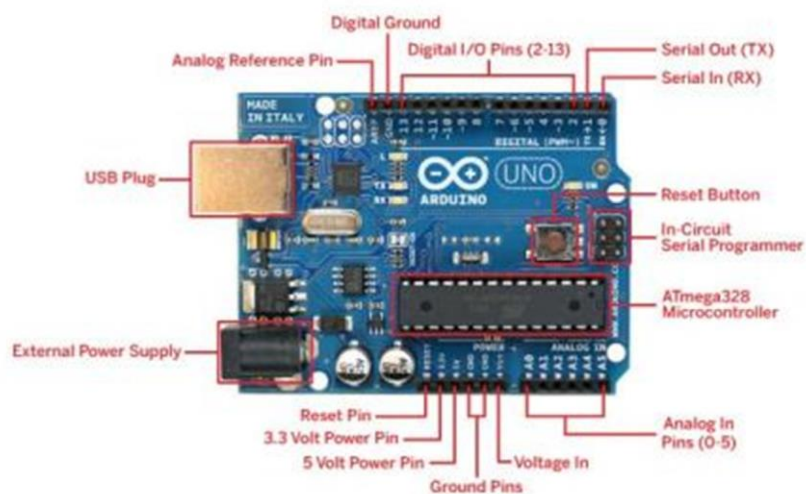


Figura 16. Distribución de pines Arduino. Copyright 2017por <https://eeeproject.com/arduino-uno-board/>

Como se puede ver en la figura 18, la placa Arduino tiene los elementos básicos para el desarrollo de proyectos, al contar con pines de entrada y salida tanto análogos como digitales y pines para señales PWM y puertos de comunicación serial, SPI (*Serial Peripheral Interface*) e I2C (*Inter-Integrated Circuit*).

Su núcleo principal es el microcontrolador tipo AVR, el cual posee todo lo necesario para realizar las mismas funciones que un computador, debido a que tiene internamente una CPU, memoria y buses de datos para interconectarse con el exterior.

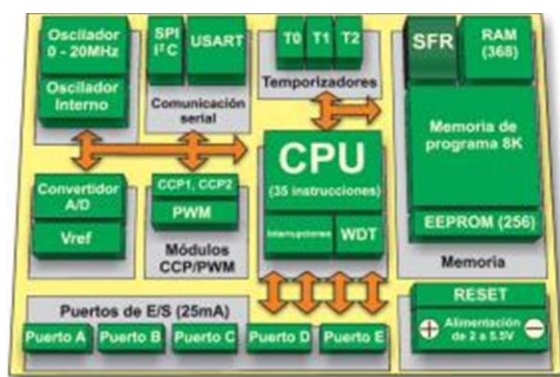


Figura 17. Funcionamiento de un Microcontrolador. Copyright 2018 por microcontroladores.

Además de la placa arduino uno, existen otras placas que permiten realizar casi cualquier proyecto académico o personal, ya que cada una de estas placas, posee unas características especiales para una u otra aplicación. Las más conocidas son:

Tabla 1. Modelos de arduino

Nombre	Microcontrolador	Mem Ram	Mem Flash	Clock	Input/output
Arduino Uno	ATmega 328	2Kb	32Kb	16Mhz	20
Arduino Mega	ATmega 2560	8Kb	256Kb	16Mhz	70
Arduino Leonardo	ATmega 32u4	2.5Kb	32Kb	16Mhz	32
Arduino Yun	ATmega 32u4	2.5Kb	32Kb	16Mhz	20
Arduino 101	Intel Curie	24Kb	196Kb	32Mhz	20

Fuente: Elaboración propia.

2.2.3.1 IDE Arduino

El IDE es el entorno de desarrollo integrado de Arduino, este es un paquete de software que contiene todas las herramientas necesarias para programar cualquiera de las placas Arduino vistas en la tabla anterior.

Para utilizar esta herramienta, solo es necesario instalarla en cualquiera de las plataformas soportadas (Windows, Linux o Mac). La interface es fácil de utilizar, gracias a que solo se requiere conocer la estructura de programación de Arduino, la cual es muy similar a la utilizada en C++, haciendo más fácil de entender su uso.

Esta herramienta cuenta con todo lo necesario para poner a funcionar la tarjeta Arduino, al tener librerías estándar, que permiten interactuar con elementos básicos como leds, motores, sensores, etc. Aunque cuenta con librerías básicas, se pueden agregar o crear librerías para poder interactuar con otros elementos como sensores o dispositivos electrónicos.

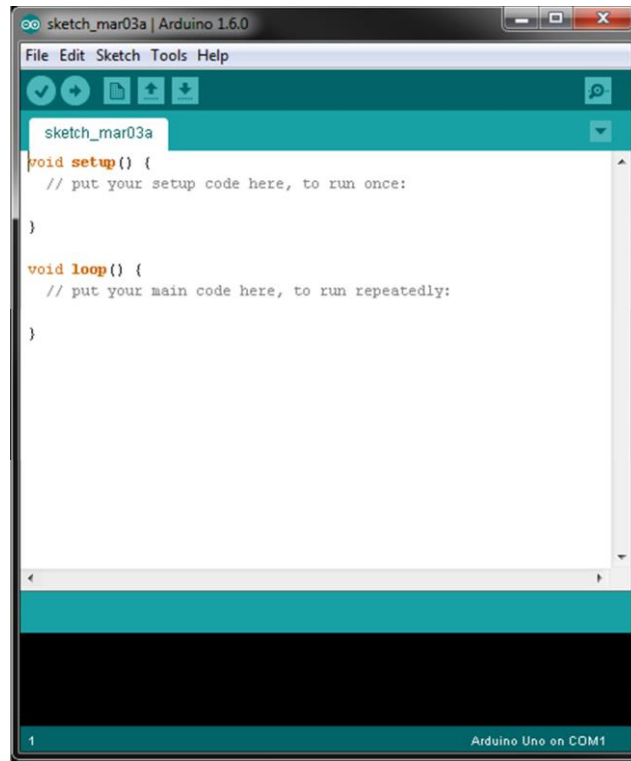
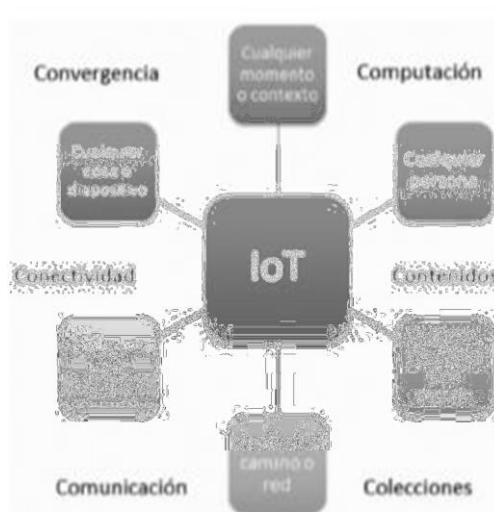


Figura 18. IDE Arduino. Copyright 2018 por Aprendiendo arduino.

2.2.4 Internet de las cosas

El internet de las cosas es una idea que se comenzó a desarrollar desde los años noventa, a partir del ingreso del concepto de computación ubicua que introdujo Mark Weiser, director científico del Xerox Palo Alto Research Center (PARC); Este concepto viene dado por la integración de los elementos cotidianos. mediante la tecnología y el manejo de una gran red de información por medio de una simple caja (PC).

El internet de las cosas les permite a las personas y cosas que puedan ser conectados en cualquier momento, en cualquier lugar, con cualquier cosa o persona, idealmente usando cualquier Path/red y cualquier servicio. Esto implica abordar elementos como la convergencia, contenidos, colecciones (repositorios), comunicación, informática, y de conectividad en el contexto donde no hay interconexión perfecta entre las personas y cosas o entre cosas y cosas, de modo que los elementos estén presentes en el direccionamiento. En la figura 20 se muestra la relación de todos estos aspectos. (Cama Pinto, De la Hoz Franco, 2012)



Fuente: adaptada de Saint-Exupery (2009).

Figura 19. Conceptos de Internet de las cosas. Copyright 2012 por Tecnologías involucradas en la internet del futuro.

2.2.4.1 La miniaturización del hardware

Una pieza clave para lograr llegar al internet de las cosas son los sensores; con estos se logra la recopilación de información sobre el entorno en que se encuentran las cosas; gracias a los avances en nanotecnología, se ha logrado que el tamaño de los microprocesadores sea cada vez menor, sin pérdidas de velocidad de procesamiento.

La idea de la miniaturización, es que cada vez elementos más comunes puedan interactuar con la red de internet, sin observar cambios considerables en los equipos, gracias a los sensores conectados, desde donde se obtendrá información en tiempo real y se podrá acceder a ella desde lugares remotos. Con la información recolectada por los sensores, se podrán tomar decisiones remotas sobre las acciones por tomar o qué acciones se realizaron de manera automática. En la figura 21 se muestra la reducción de tamaño que se ha tenido a lo largo del tiempo y el costo que conlleva esta reducción. (Cama Pinto, De la Hoz Franco, 2012)

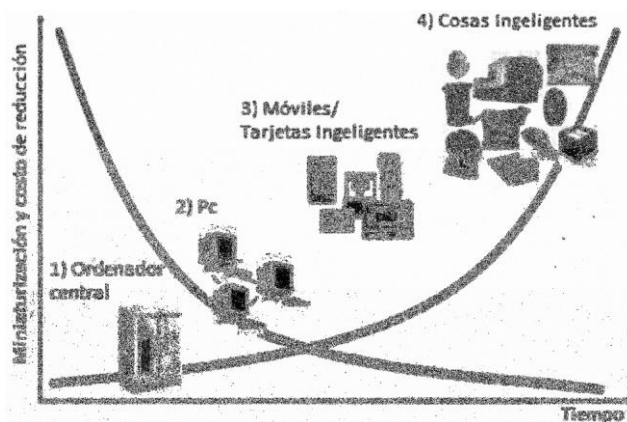


Figura 20. Miniaturización del hardware. Copyright 2012 por Tecnologías involucradas en la internet del futuro.

2.2.4.2 Campos de aplicación del internet de las cosas

El internet de las cosas tiene aplicación en muchos ámbitos de la vida y de los procesos industriales. A continuación, se muestran algunos de los sectores y aplicaciones que tiene el IoT:

- **Aeroespacial y de Aviación:** El monitoreo inalámbrico en vuelo de la aeronave mediante el uso de dispositivos inteligentes con sensores disponibles dentro de la cabina o en el exterior y conectados al sistema de monitoreo de la aeronave.
- **Seguridad.** El desafío de mantener la confianza tanto de los pasajeros en vuelos comerciales; no solo seguirá siendo extremadamente seguro, a pesar del mayor tráfico, sino se reducirá la incidencia de accidentes y se mejorará la eficiencia de los sistemas. En este contexto, sistemas inalámbricos identificables se desarrollarán mediante:
- **Automotor:** Aplicaciones en la industria automotriz incluyen el uso de cosas inteligentes para controlar e informar de todo, desde la presión en los neumáticos a la proximidad de otros vehículos. La tecnología RFID sirve para racionalizar la producción de vehículos, mejorar la logística, aumentar el control de calidad y mejorar el servicio al cliente. Los dispositivos conectados a partes contienen información relacionada con el nombre del fabricante y cuándo y dónde se fabrican los productos, su número de serie, tipo de producto, código, y en algunas aplicaciones la localización precisa de la instalación en ese momento. RFID proporciona datos en tiempo real en el proceso de fabricación, operaciones de mantenimiento y ofrece una nueva forma de gestionar memorias con mayor eficacia.

Las aplicaciones anteriormente descritas, son solo una pequeña parte del amplio campo de aplicación de la internet de las cosas, que se pueden aplicar también en los siguientes campos: telecomunicaciones, edificios inteligentes, medicina (salud), vida independiente (seguimiento), farmacéutica, logística, ventas, fabricación, gestión del ciclo de vida del producto, procesamiento de las industrias (petróleo y gas), seguridad, protección y privacidad, vigilancia del ambiente, las personas y los bienes de transporte, trazabilidad alimentaria, agricultura y cría, medios de comunicación, entretenimiento y venta de entradas, seguros, reciclaje, etc. (Cama Pinto, De la Hoz Franco, 2012)

2.2.4.3 Protocolo MQTT (Message Queue Telemetry Transport)

Es un protocolo de código abierto creado por IBM y Eurotech, que luego fue donado al proyecto Eclipse «Paho» M2M. Es un protocolo usado para la comunicación **machine-to-machine** (M2M) en el "**Internet of Things**". Este protocolo está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos empotrados con pocos recursos (CPU, RAM). Este protocolo se apoya en la capa TCP/IP, pero se ejecuta en la capa de aplicación es por esto que funciona bien para el IoT, además de ser un protocolo liviano y fácil de manejar.

La trama del protocolo MQTT consta de un Encabezado fijo (2 bytes), Encabezado variable (logitud variable) y un Payload del mensaje (hasta 256MB) y puede ser de tipo Fijo(tipo paquete, longitud del payload, QoS) o Variable (depende del tipo de paquete, ID, topic, etc)

Arquitectura de MQTT

La arquitectura de MQTT sigue una topología de estrella, con un nodo central que hace de servidor o "broker" con una capacidad de hasta 10000 clientes. El broker es el encargado de gestionar la red y de transmitir los mensajes, para mantener activo el canal, los clientes mandan periódicamente un paquete (**PINGREQ**) y esperan la respuesta del broker (**PINGRESP**).

El protocolo tiene varias características distintivas:

- Es un protocolo de publicación/suscripción.
- Además de proporcionar distribución 'de uno a muchos', la publicación/suscripción desacopla las aplicaciones. Ambas funciones resultan útiles en aplicaciones que tengan muchos clientes.
- No depende en modo alguno del contenido del mensaje.
- Se ejecuta sobre TCP/IP, que proporciona conectividad de red básica.
- Tiene tres calidades de servicio para la entrega de mensajes
- Dispone de una función Última voluntad y testamento que notifica a los suscriptores si se produce una desconexión de un cliente de un servidor MQTT.

Funcionamiento de MQTT

La comunicación se basa en unos "**topics**"(temas), que el cliente que publica el mensaje crea y los nodos que deseen recibirlo deben suscribirse a él. La comunicación puede ser de uno a uno, o de uno a muchos.

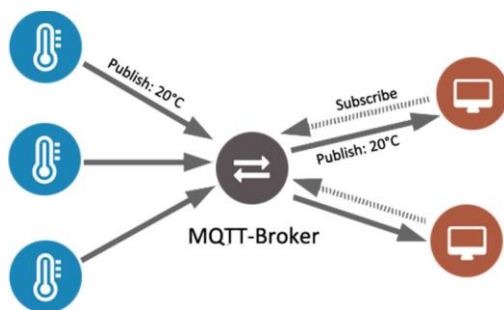


Figura 21. Funcionamiento de MQTT. Copyright 2017 por Introducción a IoT usando protocolo mqtt.

Un "topic" se representa mediante una cadena y tiene una estructura jerárquica. Cada jerarquía se separa con '/'. Por ejemplo: **"edificio1/planta5/sala1/raspberry2/temperatura"** o **"edificio3/planta0/sala3/arduino4/ruido"**. De esta forma se pueden crear jerarquías de clientes que publican y reciben datos, como podemos ver en esta forma un nodo puede suscribirse a un "topic" concreto ("edificio1/planta2/sala0/arduino0/temperatura") o a varios ("edificio1/planta2/#")



Figura 22. Estructura de los topic. Copyright 2017 por Introducción a IoT usando protocolo mqtt.

Capítulo 3 Metodología de la Investigación

3.1 Tipo de Investigación

Para desarrollar este proyecto fue necesario realizar una investigación de tipo aplicada, esta consiste en la búsqueda y utilización de conocimientos específicos ya existentes para solucionar problemas de la cotidianidad y que tienen impacto a nivel social ya que permiten el desarrollo de las comunidades. De esta manera se logra enriquecer el desarrollo cultural y científico de una región o país.

3.2 Ruta Metodológica

Para desarrollar el proyecto es necesario dividir la implementación en dos fases principales: La fase de desarrollo de hardware y la fase de desarrollo de software.

La fase de desarrollo de hardware consiste en la construcción de las etapas necesarias para el funcionamiento del sistema de alarmas. La primera etapa es la de **potencia o fuente de alimentación**, seguida de la **etapa de procesamiento de alarmas**, la cual está compuesta por los relés y las compuertas tipo AND, y es la encargada de enviar la señal de alarma a la tarjeta Arduino, finalizando con **la etapa de comunicación**, quien cuenta con un módulo GPRS

encargado de realizar la función de transmitir la información procesada por la tarjeta Arduino a través de la red celular.

La fase de desarrollo de software consiste en el desarrollo del código fuente que requiere la tarjeta Arduino, para procesar los datos provenientes del módulo de Alarmas y enviarlas al módulo GPRS, para esto se requiere el uso de una librería que permita conectarse a la red celular.

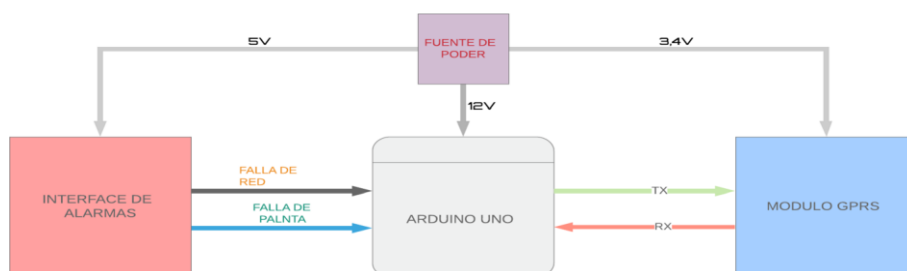


Figura 23. Diagrama de bloques sistema de monitoreo. Copyright 2019 por J F reinoso.

3.2.1 Fase de desarrollo de hardware

3.2.1.1 Diseño de la interfaz de alarmas

Para diseñar la interfaz de alarmas, es necesario tener en cuenta el concepto de transferencia eléctrica, pues las variables a monitorear dependen del estado de este elemento.

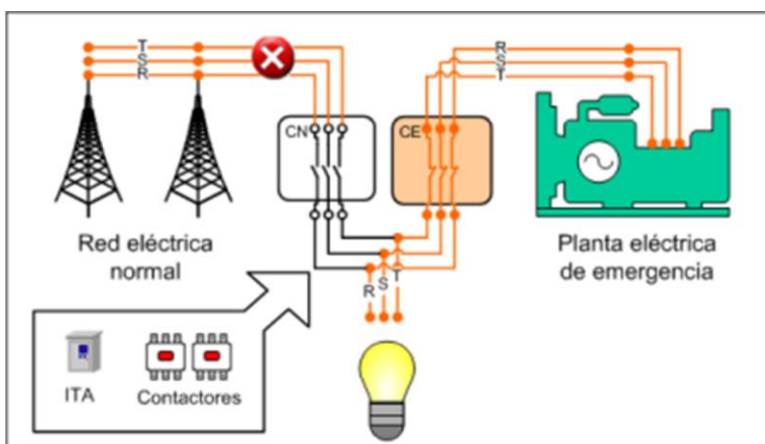


Figura 1. Esquema de una transferencia eléctrica. Copyright 2018 por Transferencias Automáticas con Contactores.

La transferencia del nodo Buenos Aires se caracteriza por tener una acometida eléctrica AC, que entrega el operador EPSA y se respalda con una acometida AC que entrega la planta eléctrica que tiene el nodo.



Figura 2 Transferencia nodo Buenos aires. Copyright 2017por J F reinos.

Como toda transferencia eléctrica, ésta determina por cual circuito trabajar dependiendo de si hay tensión de red normal, como lo muestran las figuras 25 y 26



Figura 3. Planta eléctrica Nodo B Aires. Copyright 2017 Por J F reinosa.

Debido a que las variables a monitorear se deben extraer de la transferencia eléctrica, se hace necesario aislar de alguna manera estas señales, pues manejan alta tensión comparado con la que requiere el módulo Arduino. Este aislamiento evita un daño a la tarjeta Arduino.

Para realizar este procedimiento, se utilizan relés con bobina de 120V, los cuales aíslan las tensiones altas que entrega la transferencia eléctrica y permiten conocer el estado de la transferencia, ya que poseen dos tipos de conmutadores o switches: uno normalmente abierto (NA) y otro normalmente cerrado (NC).

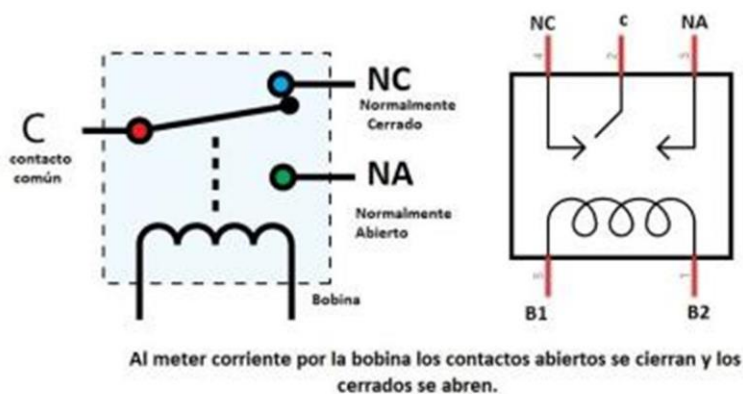


Figura 4. Estructura interna de un relé. Copyright 2018 por Areatecnologia.

Para entender su funcionamiento podemos utilizar una tabla de verdad como la siguiente:

Tabla 2. Funcionamiento del relé

Entrada (Bobina)	Salida NA	Salida NC
0	0	1
1	1	0

Fuente: Elaboración propia.

Teniendo en cuenta lo anterior, se diseñó el circuito de interfaz, el cual contara con tres relés para censar la acometida normal o de red y otros tres para censar el estado de la acometida de planta. Esto permitirá saber realmente, por cual sistema está operando el Nodo o si uno de los dos sistemas presenta falla. Aunque, se podría solo censar una de las acometidas para establecer el estado de la transferencia, se optó por censar las dos acometidas, para de esta manera saber si realmente la planta opera cuando falla la red normal.

El diseño se realizó con la herramienta Proteus, herramienta utilizada para el diseño y simulación de modelos de circuitos electrónicos, el cual, es uno del software más populares y sencillos de utilizar. Este diseño posee, como se dijo anteriormente, tres relés para censar la red normal y otros tres para censar la carga de la planta, quienes enviarán dos posibles estados a las compuertas que tiene el circuito, lo que permitirá determinar el estado de la transferencia.

Los datos obtenidos por las compuertas se enviarán al módulo Arduino quien las procesara y enviara al módulo GPRS (sim8001)

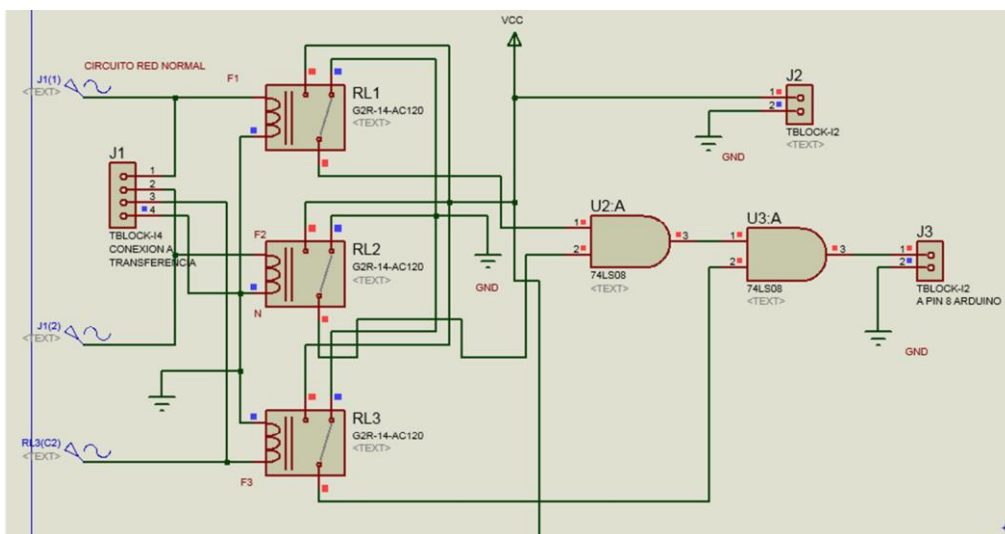


Figura 5. Diagrama de la Interfaz alarmas. Copyright 2017 por J F reinosa.

Tabla 3. Tabla de verdad compuerta

RELE 1	RELE2	RELE3	SALIDA
1	1	1	1
1	1	0	0
1	0	1	0
0	1	1	0
0	0	1	0

Fuente: Elaboración propia.

3.2.1.2 Uso de tarjeta arduino

La tarjeta o módulo Arduino es un elemento de hardware, que nos permite desarrollar con mayor rapidez, todo tipo de proyectos, al necesitar la interacción con el exterior, para obtener desarrollos más rápidos y exitosos. Esto es debido a que, solo debemos conectar al módulo Arduino los elementos necesarios para poner en funcionamiento una idea o proyecto. Para este proyecto, usaremos la tarjeta Arduino uno, la cual cumple lo requerido para el proyecto, especialmente en lo referente a memoria.

Estos elementos son por lo general, sensores o actuadores, los cuales se conectan a las entradas y salidas que posee el módulo Arduino. Para este proyecto, el módulo Arduino debe

interactuar con un módulo GPRS, quien será el encargado de realizar la comunicación con el software de monitoreo, utilizando la red celular. La comunicación de este módulo con la tarjeta Arduino se hace utilizando el puerto serial de la tarjeta, ya sea por los pines 1 y 2 o por otros pines dependiendo de la librería utilizada.

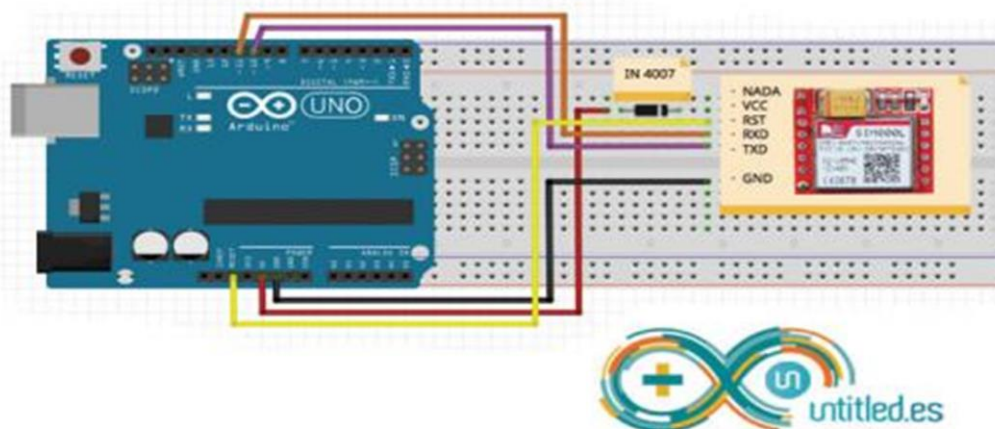


Figura 6. Conexión arduino con módulo GSM. Copyright 2018 por Arduino.

3.2.1.3 Implementación del módulo GPRS

Este módulo es un dispositivo electrónico que integra en una tarjeta el chip sim800 fabricado por Simcom, así como un slot para una simcard de cualquier operador de red celular, esto permite al usuario realizar múltiples tareas, desde llamadas telefónicas hasta la conexión a internet, utilizando el canal de datos de la red celular que para este módulo es GPRS.

Adicionalmente, el modulo posee terminales para poder conectarse a otra tarjeta o protoboard, así como un conector UFL para la conexión de antena. Las características que nos interesan son las siguientes:

- **Características Generales**

- Quad-band 850/900/1800/1900MHz
- GPRS multi-slot class 12/10
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
 - Class 4 (2 W @ 850/900 MHz)
 - Class 1 (1 W @ 1800/1900MHz)
- FM: 76~109MHz worldwide bands with 50KHz tuning step
- Dimensions: 15.8*17.8*2.4 mm
- Weight: 1.35g
- Control via AT commands (3GPP TS 27.007, 27.005 and SIMCOM enhanced AT Commands)
- Supply voltage range 3.4 ~ 4.4V
- Low power consumption
- Operation temperature:-40°C~85°C

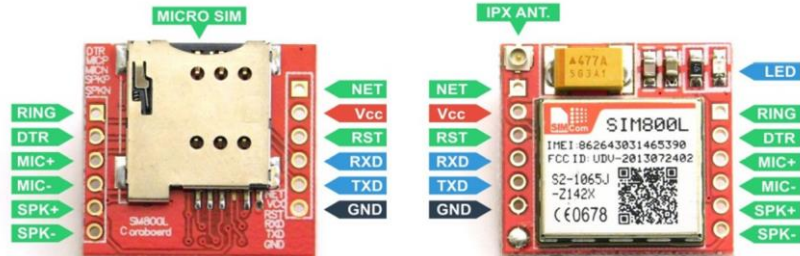
- **Especificaciones para Datos GPRS**

- GPRS class 12: max. 85.6 kbps (downlink/uplink)
- PBCCH support

- Coding schemes CS 1, 2, 3, 4
- PPP-stack
- CSD up to 14.4 kbps
- USSD
- Non transparent mode
- Specifications for SMS via GSM/GPRS
- Point to point MO and MT
- SMS cell broadcast
- Text and PDU mode

(sim800h_hardware_design_v1.00 (2018), Recuperado de

<http://www.simcom.com/product/showproduct.php?id=271&lang=en>)



NETTIGO

Figura 7. Módulo GSM. Copyright 2018 por Sim800L.

Como otros módulos utilizados por arduino, el método de comunicación es por comandos AT, estos son palabras en formato string, que le dicen al módulo que función realizar; para cada módulo existe una serie de comandos dados por el fabricante en sus hojas de datos o datashet.

En el caso del SIM800 el puerto serial no es TTL, por lo que se debe adecuar para poder comunicarse de manera eficaz con la tarjeta Arduino cuando el modulo no trae incluido el conversor de nivel, esto se hace con transistores bipolares según recomienda el propio fabricante.

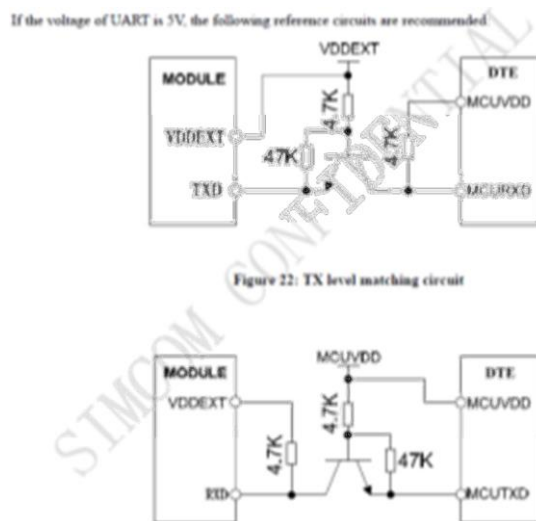


Figura 8. Adecuación niveles TTL. Copyright 2018 por Sim800.

Otra configuración que se debe hacer en la tarjeta Arduino es la de las entradas para alarmas proveniente de la salida de las compuertas de interface de alarmas. Para esto se seleccionaron dos entradas digitales de las 10 disponibles en la tarjeta Arduino uno.

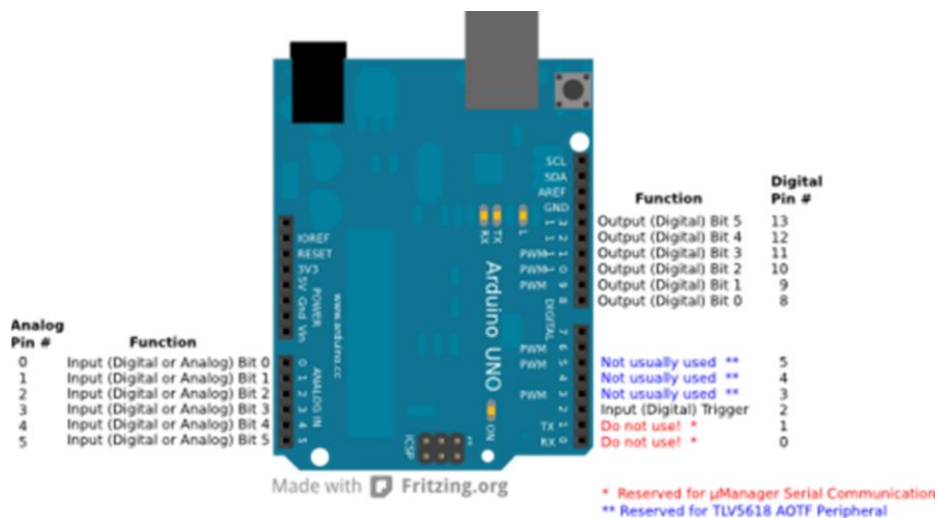


Figura 9. Entradas y salidas Arduino. Copyright 2018 por Arduino.

Para la alimentación del módulo GSM se requieren entre 3.4 y 4.4V por lo tanto se debe utilizar un regulador ajustable para poder activar el modulo correctamente, adicionalmente la tarjeta Arduino uno, también requiere de un voltaje de 5V estable por lo que debe agregar un regulador de voltaje fijo a 5V, estos reguladores se conectaran a un adaptador de 12 V y como respaldo se debe conectar a una batería.



Figura 10. Banco de baterías nodo B aires. Copyright 2017 por J F reinosa.

Finalmente se podrán integrar en una sola tarjeta todas las etapas nombradas anteriormente, para esto se utiliza la herramienta Proteus, que permite simular y diseñar la PCB requerida para el proyecto.

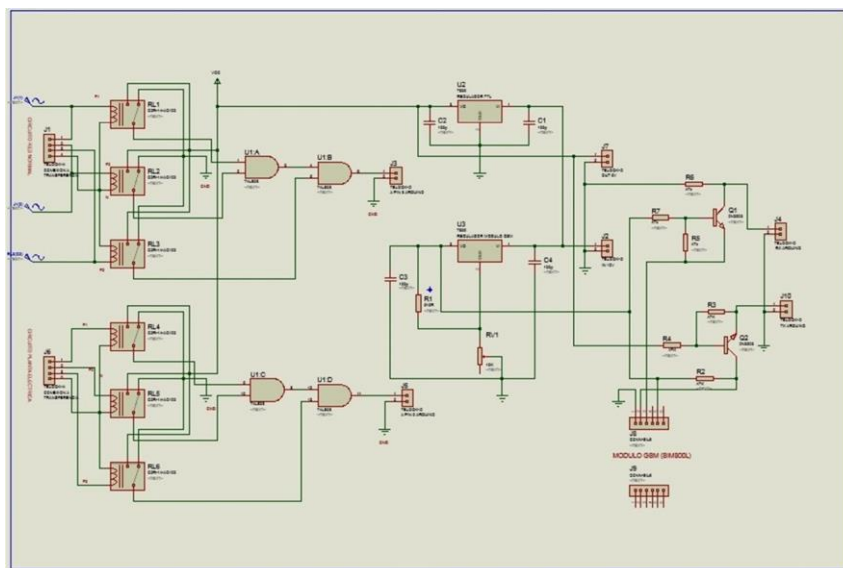


Figura 11. Diagrama circuito completo. Copyright 2018 por J F reinosa.

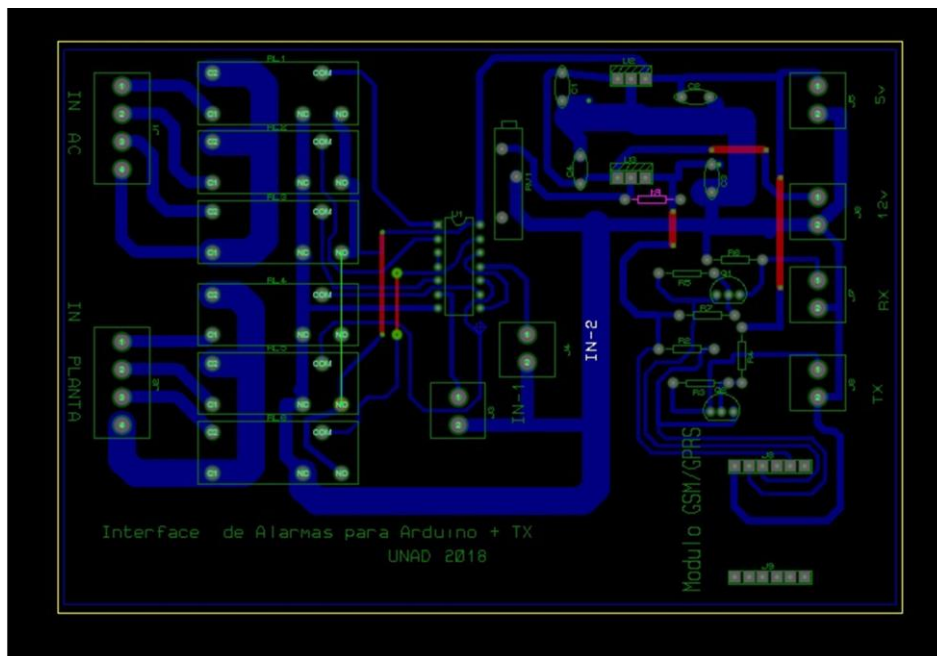


Figura 12. PCB layout. Copyright 2018 por J F reinosa.

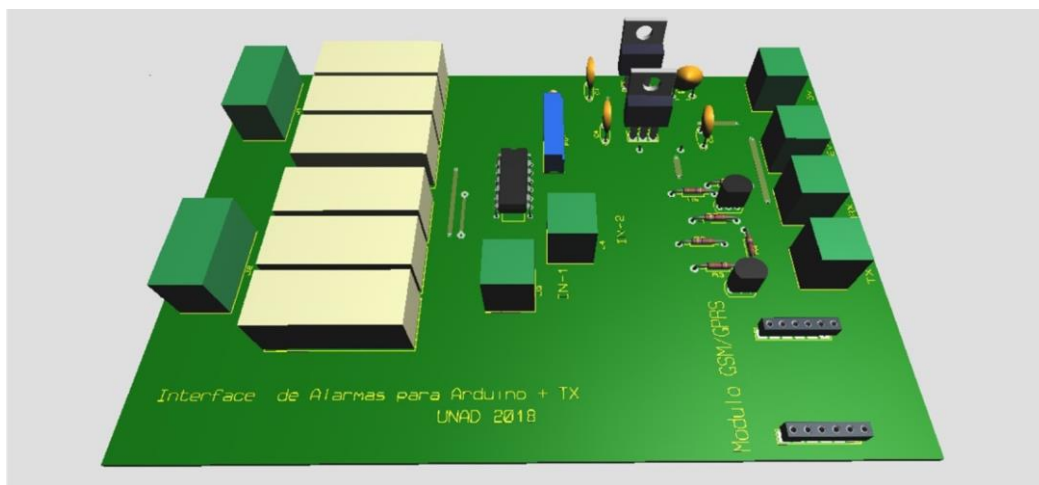


Figura 13. PCB en 3D. Copyright 2018 por J F reinosa.

3.2.2 Fase de desarrollo de software

3.2.2.1 Programación de la tarjeta Arduino

Para la programación de la tarjeta o módulo Arduino se utilizó el IDE Arduino el cual es una herramienta propia de esta tecnología y con la cual se desarrolló el código para el proyecto. Esta herramienta solo requiere instalarse en una computadora con sistema operativo Windows o Linux y posteriormente conectar la tarjeta Arduino por medio de un cable USB para poder crear el código y compilarlo.

El lenguaje de programación es muy fácil de entender ya que es un lenguaje orientado a objetos, muchas de sus librerías vienen escritas en C++ y pueden ser mejoradas o ampliadas si se tiene conocimiento del lenguaje. En la práctica las funciones creadas en las librerías son muy intuitivas.

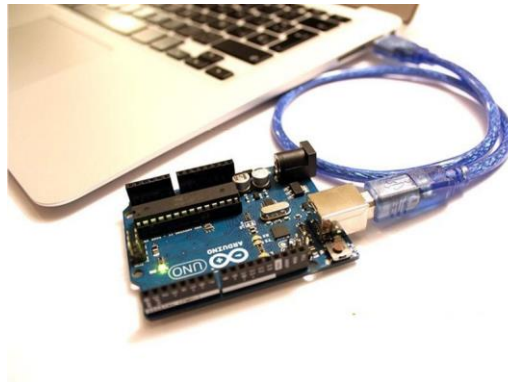


Figura 14. Comunicación con Arduino. Copyright 2019 por Alimentar el arduino.

El desarrollo del código debe partir de la necesidad primaria que es la conexión con una red celular GSM, a través de la cual se enviarán los datos necesarios para que en la central de monitoreo puedan verse las alarmas generadas en la transferencia eléctrica y procesadas por el módulo Arduino. Para entender mejor esto se elaboró un algoritmo que nos permitiera realizar la programación correspondiente.

Inicialmente se requirió de la librería **Tiny** la cual es la que permite comunicar el Arduino con la red GSM, y la cual solo requiere ser llamada desde el programa principal, esta librería tiene las funciones necesarias para conectar el módulo GPRS a una red celular que permita tener acceso a internet ya que será a través de este medio que se establecerá la comunicación con la central de monitoreo. Después de hacer esto se continuará con el proceso de envío de los mensajes según la alarma presentada en la interface de alarmas, para esto se utilizará otra librería que permita enviar la información del mensaje al sitio remoto de manera satisfactoria.

La librería escogida es **pubsubclient** la cual usa el protocolo MQTT para el envío de mensajes a un servidor de reenvío o publicación el cual retransmite la información enviada a todos los clientes que se suscriban al tema o topic configurado aquí.

Con estas dos librerías se logrará la comunicación hacia el centro de gestión pudiendo visualizarse la información enviada a través de una interfaz web que estará previamente conectada al servidor de publicaciones.

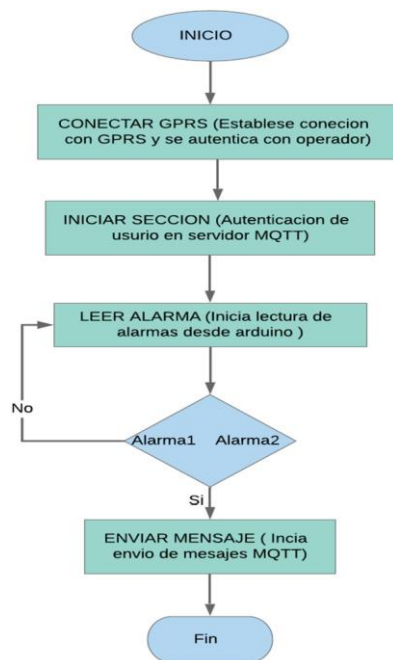


Figura 15. Algoritmo Arduino. Copyright 2017 por J F reinoso.

Para ponerlo en funcionamiento y logran la comunicación vía serial desde Arduino se requirió de una librería que fuera compatible con el módulo A6 o que pudiera enviar comandos AT manualmente (consola) al módulo, ya que esta es la forma de comunicación que utilizan muchos módulos GSM. Inicialmente se consiguieron hacer pruebas a través de la consola de Arduino tal como se ve en la figura 42 aunque antes se presentaron algunos problemas de sincronismo ya que estos módulos vienen configurados para trabajar en diferentes velocidades de baudios teniendo que probar con varias opciones estableciéndose la de 9600 como la opción correcta para realizar las pruebas pero al revisar el alcance del proyecto se vio la necesidad de buscar otra librería que pudiera manejar cadenas de *string* ya que se requerirá para enviar los mensajes, a través de la red celular

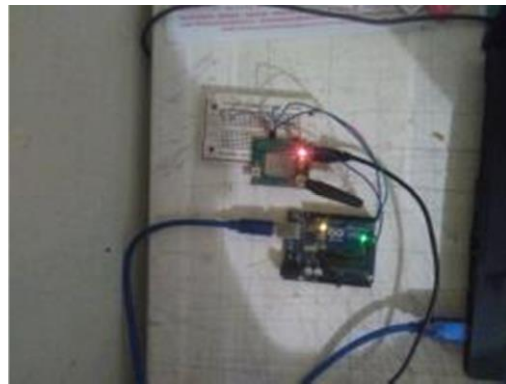
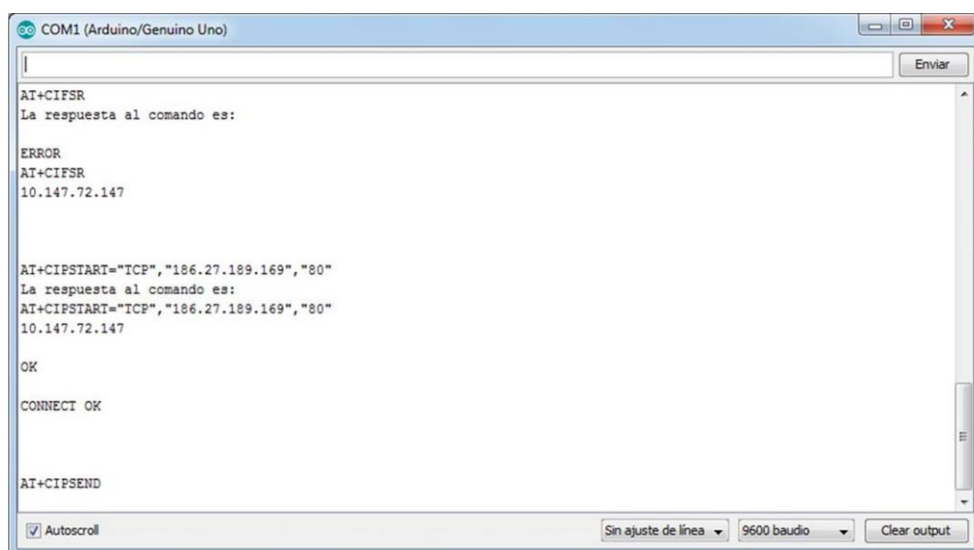


Figura 17. Prueba módulo GSM. Copyright 2017 por J F reinoso.



```
COM1 (Arduino/Genuino Uno)
|
| Enviar
|
AT+CIFSR
La respuesta al comando es:
ERROR
AT+CIFSR
10.147.72.147

AT+CIPSTART="TCP","186.27.189.169","80"
La respuesta al comando es:
AT+CIPSTART="TCP","186.27.189.169","80"
10.147.72.147

OK

CONNECT OK

AT+CIPSEND

 Autoscroll
Sin ajuste de línea 9600 baudio Clear output
```

Figura 18. Prueba de conexión al módulo. Copyright 2017 por J F reinosa.

Para la implementación se hace uso de una librería llamada TinyGSM que trabaja con varios módulos GSM entre ellos el **A6** la cual ya trae integradas varias funciones para trabajar con los módulos GSM entre ellas la conexión automática a la red celular y el envío de datos a través de TCP/IP.

Durante el proceso de pruebas para el envío de mensajes al sitio remoto se presenta la desactivación del módulo A6 por parte del operador de red Claro lo cual me obliga a cambiar de módulo ya que el anterior no estaba homologado en Colombia. Consulto en la página de la CRC los módulos Homologados y encuentro uno de los ofertados en las tiendas de electrónica el SIM800L el cual cumple con las características para el proyecto, por lo tanto decido continuar el trabajo con este módulo que además también es compatible con la librería **Tyni** utilizada con el módulo A6.

Para trabajar con el módulo SIM800 se debieron agregar dos elementos, uno es un regulador ajustable para poder alimentarlo debidamente ya que el mismo trabaja a no más de 4.4V y el otro un adaptador de nivel para el puerto serial, ya que este módulo trabaja con señales de máximo 2.8V lo cual dificulta la comunicación con el módulo Arduino debido a que este trabaja con señales TTL es decir de 5V.

Symbol	Min	Max	Unit
V_{IL}	0	0.3	V
V_{IH}	2.5	2.8	V
V_{OL}	0	0.1	V
V_{OH}	2.7	2.8	V

Figura 42. Características puerto serial sim800. Copyright 2018 por sim800.

Como se puede apreciar en la figura 44. Los niveles de voltaje que maneja el módulo sim800 con diferentes al estándar TTL lo cual hace necesario realizar una adaptación de estos voltajes o señales para poder tener una buena comunicación entre los dos elementos (Arduino y módulo GSM).

Después de lograr la comunicación entre el Arduino y el módulo GSM se continuo con la prueba de librerías que fueran compatibles con el sim800 y que permitieran trabajar con *string* ya que fue fundamental para el desarrollo del proyecto, como se digo anteriormente la librería Tyni es perfectamente compatible con este módulo, aunque hay muchas más dependiendo de la aplicación que le quiera dar al módulo.

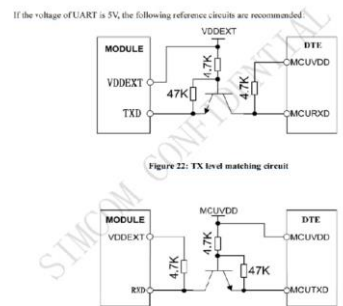


Figura 43. Adaptador de nivel sim800. Copyright 2018 por sim800.

4.2 Conexión a la red celular GPRS

Para la conexión a la red celular se requieren varias cosas, una es la debida instalación del módulo GSM/GPRS es decir que este alimentado con el voltaje recomendado por el fabricante, así como la instalación de una buena antena para poder captar la mejor señal posible. Lo siguiente es haber establecido una buena comunicación con el modulo a través del puerto serial, este último corresponde a los pines tx y rx del módulo el cual debe estar configurado con un *baud-rate* igual al usado por el elemento de control que en este caso es el Arduino uno.

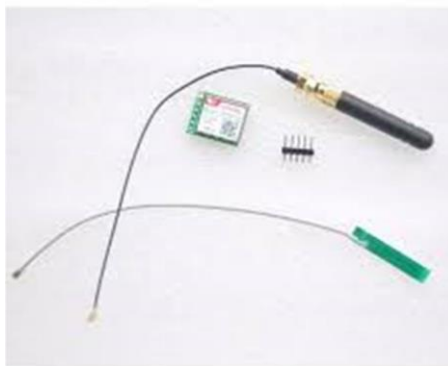


Figura 44. Antenas para módulo GSM. Copyright 2018 por módulo gsm gprs sim800 con antena.

Establecida la comunicación con el modulo se carga una librería en el IDE Arduino para poder llamarla desde el código fuente realizado para la conexión a la red, este código tiene tres elementos básicos para poder establecer la conexión. El primer elemento es la configuración del *baud-rate* del puerto serial para poder enviar los comandos AT necesarios para la conexión estos comandos ya vienen definidos en la librería utilizada, el segundo elemento es la verificación de disponibilidad de red, esto se realiza con la función “*waitForNetwork()*”, la cual verifica que el modulo este habilitado para trabajar en la red celular elegida.

Por último el modulo solicita la conexión a la red, para lo cual se deben ejecutar una serie de comandos AT que vienen establecidos en la función “*gprsConnect()*”. Desde el punto de vista de red lo que realiza esta función es asignar al cliente (módulo GPRS) los recursos necesarios para conectarse a internet pero para obtenerlos se debe activar el contexto PDP apropiado según

lo establece el estándar 3GPP quien define como se debe realizar la comunicación a nivel físico y lógico en los equipos de tecnología celular, a continuación se muestra el procedimiento de activación definido por 3GPP

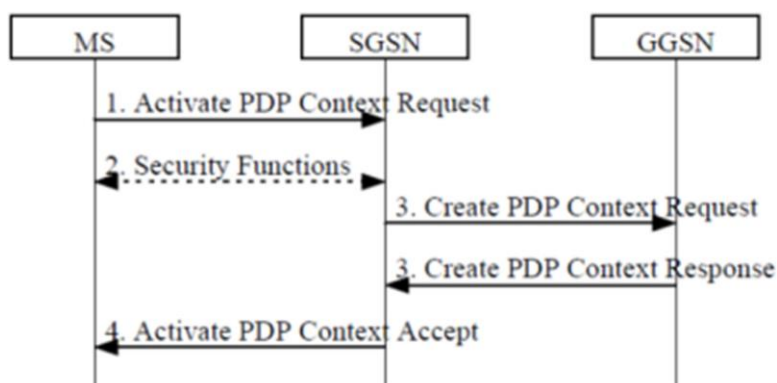


Figura 45. Procedimiento de activación PDP. Copyright 2018 por Technical Specification Group Services and System Aspects.

Para entenderlo mejor miremos el siguiente ejemplo.

Formato:

`AT+CGDCONT= <cid> [,<PDP_type> [,<APN> [,<PDP_addr> [,<d_comp> [,<h_comp>]]]]]`

Uso:

AT+CGDCONT=1,"IP","internet.comcel.com.co" (internet comcel es el apn del operador claro utilizado para este proyecto")

Como se pudo apreciar en el ejemplo, se debe definir el tipo de PDP a utilizar que para este caso es IP con esto le indicamos a la red que vamos a utilizar el protocolo ip para el transporte de la información, a continuación se define el APN que como se puede ver en el ejemplo es el correspondiente a claro Colombia. Adicionalmente el estándar 3GPP define poder asignar la IP desde el módulo para obtener una IP fija pero esto es algo que solo se utiliza a nivel corporativo la mayoría de usuarios de estas redes se les asigna la dirección automáticamente después de definir el contexto PDP y autenticarse en la red.

Como se puede observar para la conexión del módulo GPRS se deben tener en cuenta varios aspectos técnicos pero con la utilización de una librería apropiada se puede simplificar el proceso ya que de lo contrario se debería hacer uso de una extensa lista de comandos para la conexión a la red. El conocer esto además nos sirve para poder hacer las modificaciones a que haya lugar si se requiere trabajar en otros ambientes como es la conexión punto a punto o por frame relay protocolos que también se pueden usar en este tipo módulos.

En la siguiente figura se muestra como se realiza la conexión entre los diferentes elementos de una red GPRS según el estándar 3GPP.

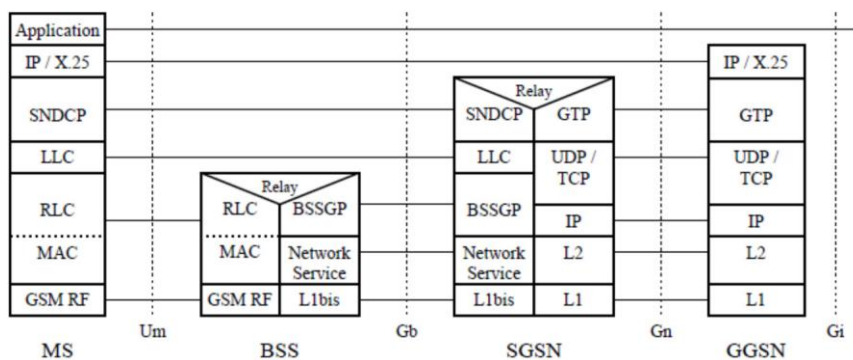


Figura 46. Conexión a GPRS. Copyright 2018 Especificación técnica GSM 03.60.

A continuación se muestra como se realizó la conexión del módulo GPRS(sim800) a la red de claro Colombia utilizando la librería tyni, esta librería tiene implementadas las funciones para realizar una conexión a internet de manera normal, para ello solo es necesario indicar el APN de la red con su respectivo usuario y contraseña o password. Esta librería también cuenta con implementaciones para ver el nivel de señal (rs1) el operador y la Ip asignada al módulo.

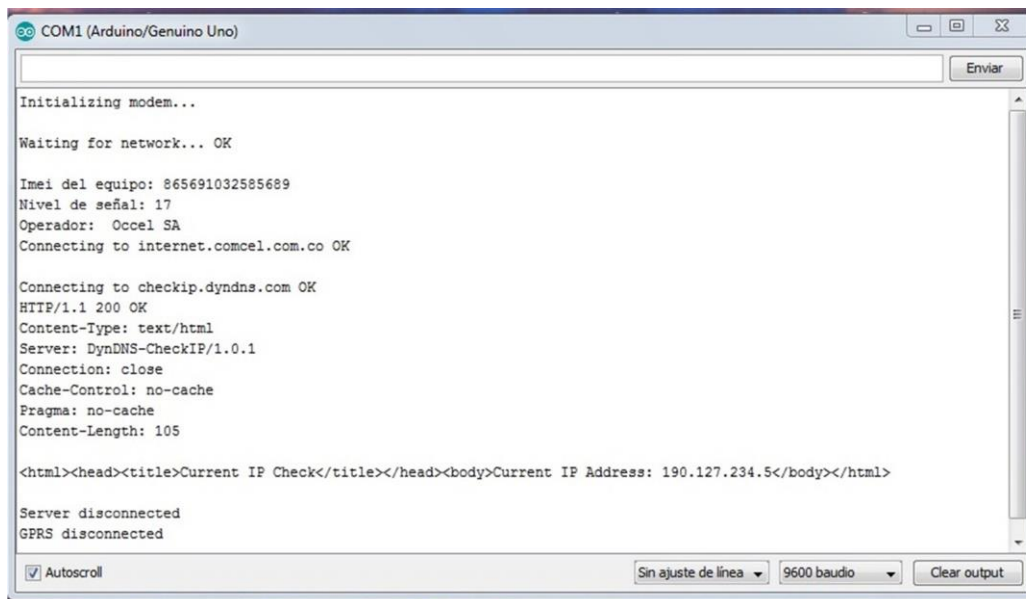


```
PruebaHiveMQ Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda

PruebaHiveMQ $
63 // Set GSM module baud rate
64 SerialAT.begin(1200);
65 delay(3000);
66
67 // Restart takes quite some time
68 // To skip it, call init() instead of restart()
69 Serial.println("Initializing modem...");
70
71 String modemInfo = modem.getModemInfo();
72 Serial.print("Modem: ");
73 Serial.println(modemInfo);
74
75 Serial.print("Waiting for network...");
76 if (!modem.waitForNetwork()) {
77     Serial.println(" fail");
78     while (true);
79 }
80 Serial.println(" OK");
81
82 Serial.print("Connecting to ");
83 Serial.print(apn);
84 if (!modem.gprsConnect(apn, user, pass)) {
85     Serial.println(" fail");
86     while (true);
87 }
88 Serial.println(" OK");
89
90 // MQTT Broker setup
```

Figura 47. Código para la conexión a GPRS. Copyright 2017 por J F reinosa.

Debido a que se requería conocer la IP pública y para probar la conexión a internet se debió buscar la manera de certificar esto, ya que la librería `tyini` no tiene implementada una función para realizar estas funciones, se debió implementar en el código una función que llamara a la “`checkip.dyndns.com`” la cual es una página diseñada para este tipo de pruebas lográndose probar tanto la conexión así como conocer la IP pública que nos servirá para pruebas posteriores.

The image shows a screenshot of the Arduino IDE's serial monitor window. The window title is "COM1 (Arduino/Genuino Uno)". The output text is as follows:

```
Initializing modem...  
Waiting for network... OK  
Imei del equipo: 865691032585689  
Nivel de señal: 17  
Operador: Ocel SA  
Connecting to internet.comcel.com.co OK  
  
Connecting to checkip.dyndns.com OK  
HTTP/1.1 200 OK  
Content-Type: text/html  
Server: DynDNS-CheckIP/1.0.1  
Connection: close  
Cache-Control: no-cache  
Pragma: no-cache  
Content-Length: 105  
  
<html><head><title>Current IP Check</title></head><body>Current IP Address: 190.127.234.5</body></html>  
  
Server disconnected  
GPRS disconnected
```

At the bottom of the window, there are controls for "Autoscroll" (checked), "Sin ajuste de línea" (line wrap), "9600 baudio" (baud rate), and "Clear output".

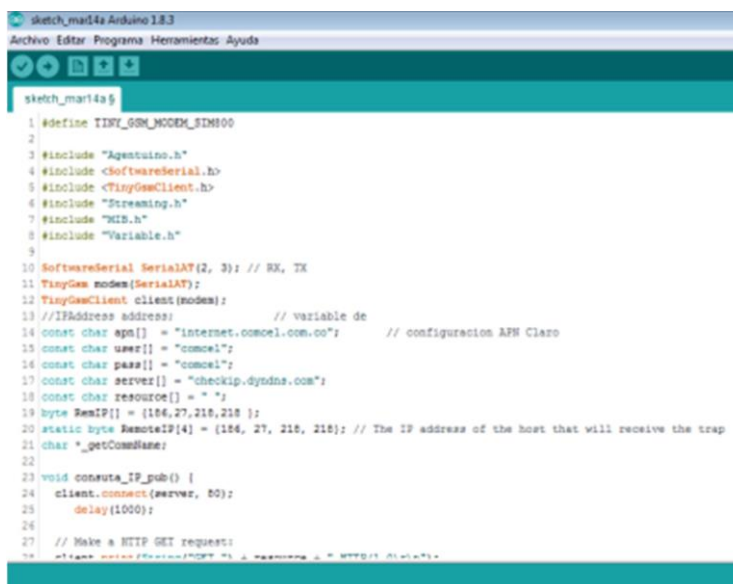
Figura 48. Prueba GPRS. Copyright 2017 por J F reinosa.

4.3 Configuración NSMP para envío de mensajes

Con el modulo conectado al GPRS se continua el proceso para enviar los mensajes de alarmas a la estación remota o centro de monitoreo, para esto se escogió el protocolo NMP el cual es muy usado por los administradores de red para el monitoreo de eventos o alarmas en servidores y demás elementos de hardware además de que es un protocolo fácil de gestionar desde diferentes plataformas. Para ello se encontró la librería **agentuino** la cual ha sido desarrollada para trabajar con la tarjeta Arduino lo cual la hacía idónea para el proyecto, aunque se presentaron varios inconvenientes.

1. Los ejemplos de uso solo han sido con módulos **WIFI** o Ethernet, por lo que se debieron realizar cambios a esta librería para poder integrarla con **tiny** ya que se presentaban errores de compilación del código.
2. Cambio del protocolo de transporte en la librería **tiny** ya que viene por defecto para trabajar con TCP y snmp trabaja con **UDP**.
3. Las redes públicas no permiten el uso de UDP, se realizaron pruebas de puestos y los mismos están cerrados tanto en los operadores móviles como en los fijos.

Una de las ventajas que tiene este protocolo es el de poder integrarse a redes SNMP activas ya que el protocolo es escalable pudiendo anexar dispositivos nuevos a medida que se amplía la red, pero como se verá en las pruebas realizadas no fue posible implementarlo con redes celulares ya que es un protocolo que trabaja con puertos UDP lo que no es común en las redes de publicas por motivos de seguridad.



```

1 #define TINY_GSM_MODEM_SIM900
2
3 #include "Agentuno.h"
4 #include <SoftwareSerial.h>
5 #include <TinyGSMClient.h>
6 #include "Streaming.h"
7 #include "MD5.h"
8 #include "Variable.h"
9
10 SoftwareSerial SerialAT(2, 3); // RX, TX
11 TinyGSM modem(SerialAT);
12 TinyGSMClient client(modem);
13 //IPAddress address; // variable de
14 const char apn[] = "internet.comcel.com.co"; // configuracion APN Claro
15 const char user[] = "comcel";
16 const char pass[] = "comcel";
17 const char server[] = "checkip.dyndns.com";
18 const char resource[] = "/";
19 byte RemIP[] = {186, 27, 218, 218};
20 static byte RemoteIP[4] = {186, 27, 218, 218}; // The IP address of the host that will receive the trap
21 char *_getCmdName;
22
23 void consulta_IP_pub() {
24     client.connect(server, 80);
25     delay(1000);
26
27     // Make a HTTP GET request
28     client.print(HTTP_GET_REQUEST);
  
```

Figura 49. Librerías para activar SNMP. Copyright 2018 por J F reinosa.

```

208 void loop() {
209     consulta_IP_pub();
210     remote host and the locUpTime
211
212
213     Serial.println("Send TRAP");
214     int TRAP("Arduino SNMP trap", RemoteIP, locUpTime);
215     Agentuino.Trap("Arduino SNMP trap", RemoteIP, locUpTime, "1.3.6.1.4.1.28032.1.1.1");
216     delay(3000);
217     locUpTime = locUpTime + 100;
218
219     if (millis() - prevMillis > 1000) {
220         prevMillis += 1000;
221         locUpTime += 100;
222     }
223
224 }

```

Figura 50. Prueba de conexión a snmp. Copyright 2018 por J F Reinosa.

Como se puede ver en la figura a y b no se logró obtener conexión con el módulo GPRS a través de SNPM ya que se encuentra cerrado el puerto 162 UDP en el router que recibe los mensajes enviados por el modulo, se consultó con el operador de internet y me informa que no se pueden abrir esos puertos por seguridad. Se verifica con los operadores móviles si pueden abrir los puertos udp para poder hacer las pruebas pertinentes y contestan de igual forma. Por lo que se debió buscar otra forma de enviar los mensajes.

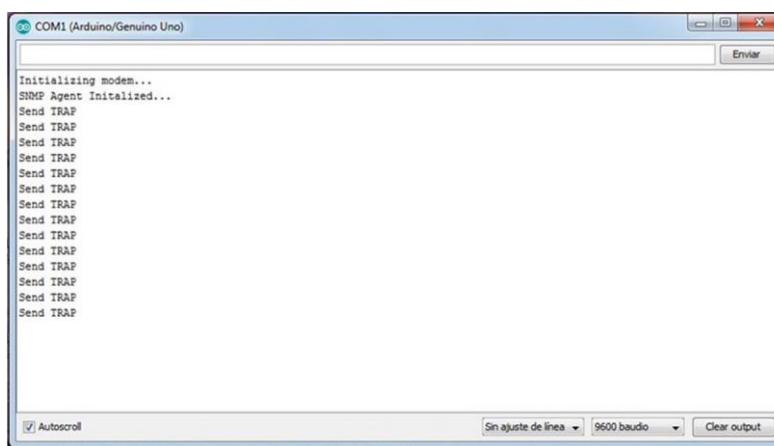


Figura 51. Envío de trap desde Arduino. Copyright 2018 por J F reinosa.

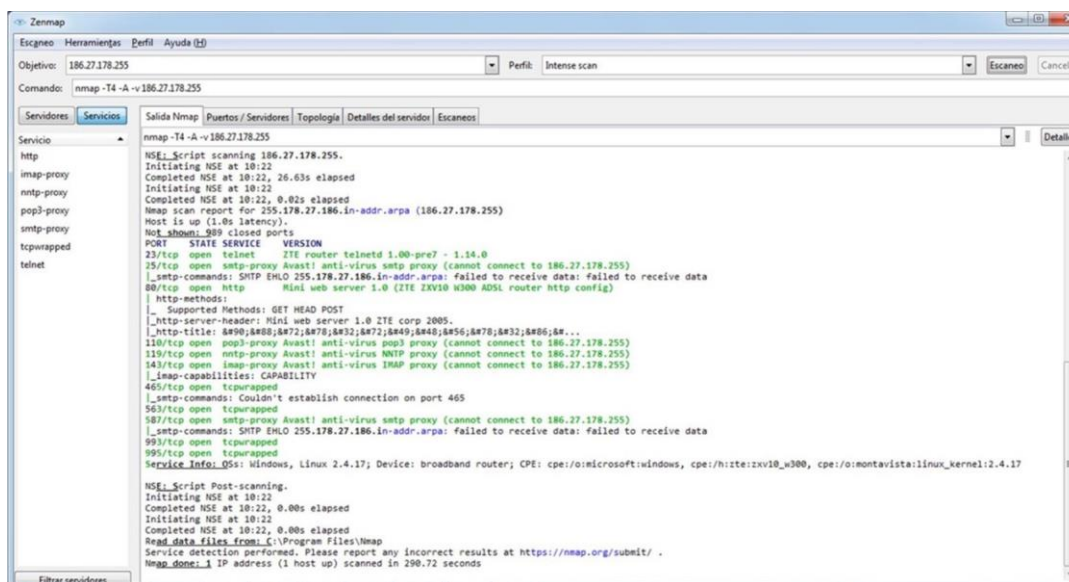


Figura 52. Prueba de puertoSNMP. Copyright 2018 por J F reinosa.

4.4 Conexión a un servidor MQTT para el envío de mensajes

Debido al problema presentado con el protocolo snmp se buscó una alternativa para enviar el mensaje que informara de la alarma transmitida por la interface de alarmas, se encontraron varias opciones para hacerlo entre ellas están:

1. Usar una VPN para redireccionar el puerto UDP
2. Crear un servidor para almacenar los datos enviados por Arduino
3. Reenviar el puerto desde el router ISP
4. utilizar el protocolo MQTT

La opción 4 fue la elegida ya que es una combinación de las demás opciones, este protocolo permite el reenvío de mensajes recibidos previamente desde un sensor o aplicación

hacia otra aplicación. Adicionalmente es un protocolo simple que no utiliza muchos recursos de software ni de hardware por lo tanto es la mejor opción para superar los inconvenientes tenidos con snmp.

Otra de las ventajas es que no se requiere tener una ip fija para el módulo GPRS ya que el protocolo se implementa sobre un servidor público o también se puede implementar uno propio lo que supera otro problema que se tenía con el protocolo snmp. Para implementarlo solo se requiere crear una cuenta en uno de los servidores públicos que hay disponibles actualmente o instalar un servidor propio, en este caso también hay opciones tanto en Linux como en Windows.

Para este proyecto se utilizó un servidor público el cual está siempre disponible y al cual se accede solo con ingresar a <http://www.hivemq.com/demos/websocket-client/> aunque hay otros disponibles este es de los más fáciles de usar ya que posee su propia interface.

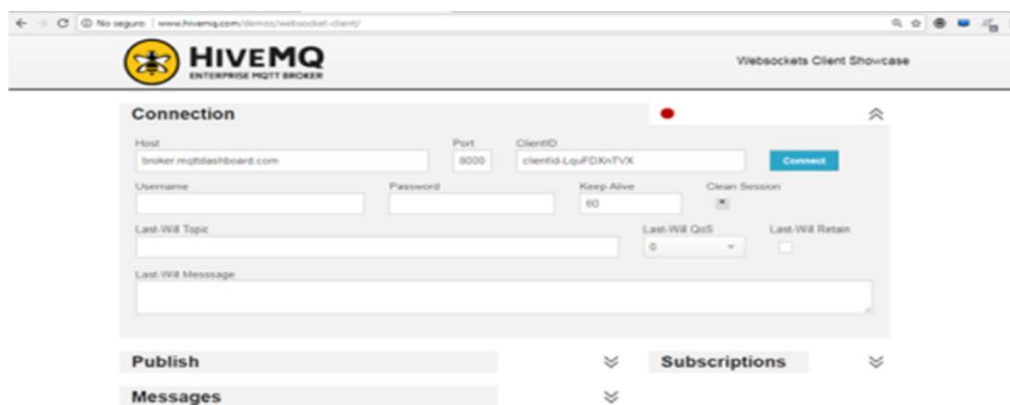
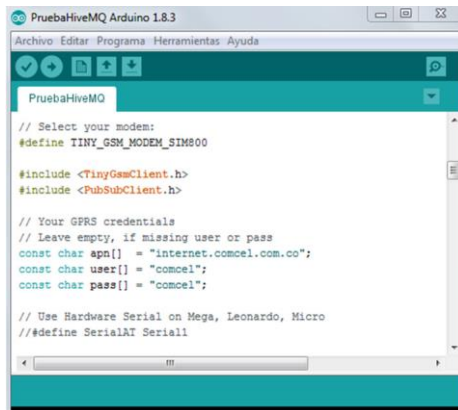


Figura 53. Servidor MQTT. Copyright 2018 por J F reinosa.

Para integrar este protocolo al proyecto solo se requiere agregar al IDE arduino la librería Pubsubclient, esta librería está desarrollada para trabajar con cualquier servidor que soporte el protocolo MQTT. Esta librería realiza dos funciones básicas enviar mensajes o recibirlos.



```

PruebaHiveMQ Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda
PruebaHiveMQ
// Select your modem:
#define TINY_GSM_MODEM_SIM800

#include <TinyGsmClient.h>
#include <PubSubClient.h>

// Your GPRS credentials
// Leave empty, if missing user or pass
const char apn[] = "internet.comcel.com.co";
const char user[] = "comcel";
const char pass[] = "comcel";

// Use Hardware Serial on Mega, Leonardo, Micro
// #define SerialAT Serial1

```

Figura 54. Librería pubsubclient. Copyright 2018 por J F reinos.



```

PruebaHiveMQ Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda
PruebaHiveMQ 5
--
// MQTT Broker setup
90 mqtt.setServer(broker,1883);
91 mqtt.setCallback(mqttCallback);
92 }
93 }
94
95 boolean mqttConnect() {
96   Serial.print("Connecting to ");
97   Serial.print(broker);
98   if (!mqtt.connect("454fd10-1815-4f9a-ba5b-b91fecf30502")) {
99     Serial.println(" fail");
100    return false;
101   }
102   Serial.println(" OK");
103   mqtt.publish("Arduino", "UserClientTest");
104   mqtt.subscribe("Arduino");
105   return mqtt.connected();
106 }
107
108 void loop() {
109
110   if (mqtt.connected()) {
111     mqtt.loop();
112   } else {
113     // Reconnect every 10 seconds
114     unsigned long t = millis();
115     if (t - lastReconnectAttempt > 10000L) {
116       lastReconnectAttempt = t;

```

Figura 55. Configuración MQTT. Copyright 2018 por J F reinos.

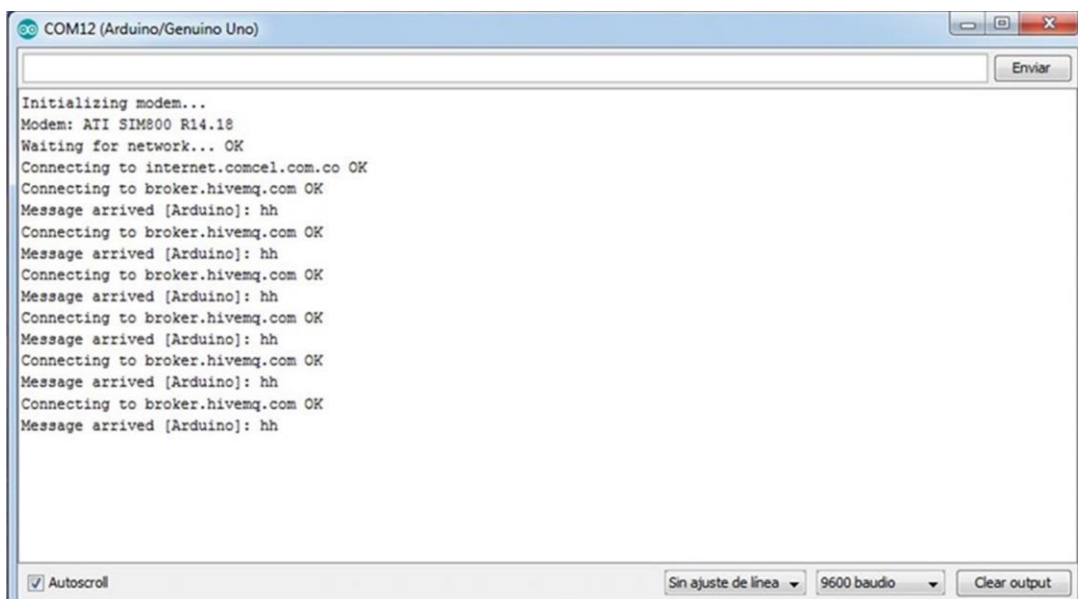


Figura 56. Conexión a MQTT desde Arduino. Copyright 2018 por J F reinoso.

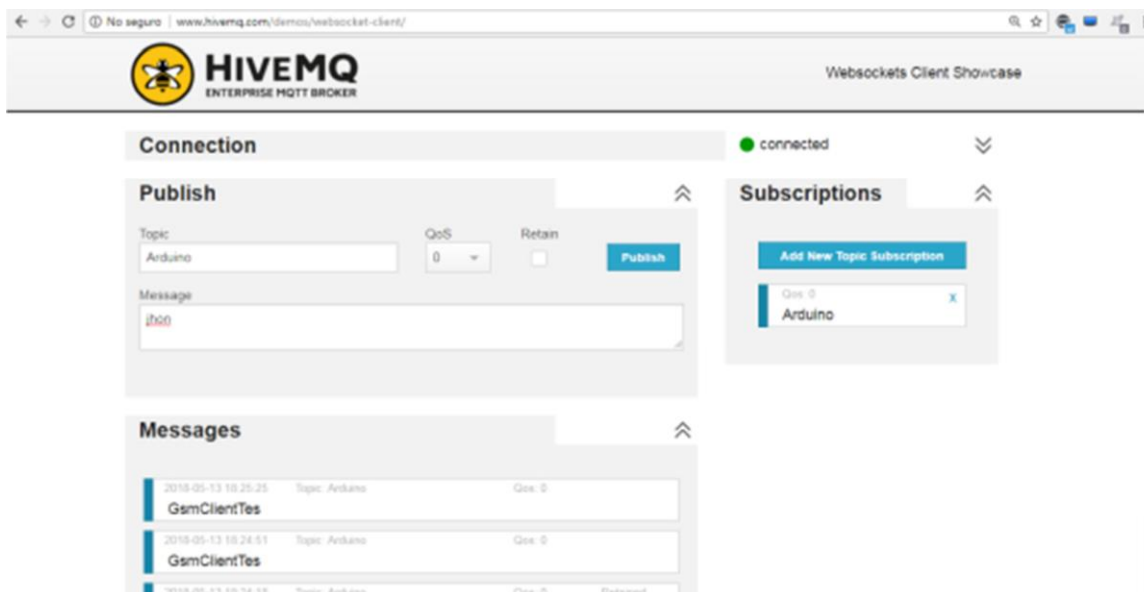


Figura 57. Prueba MQTT. Copyright 2018 por J F reinoso.

4.5 Construcción de la interfaz de alarmas

La interfaz de alarmas se construyó utilizando el software Proteus, la cual es una herramienta para desarrollo electrónico muy versátil con la cual se pudo realizar tanto la simulación de la interface así como el diseño de la tarjeta de circuito o PCB.

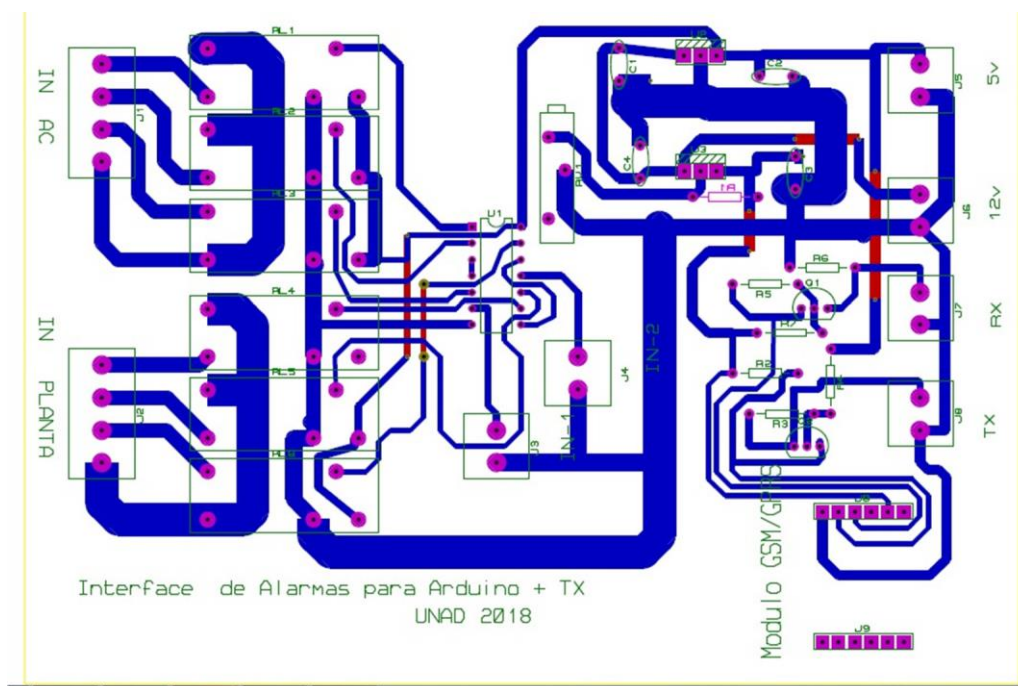


Figura 58. PCB tarjeta alarmas. Copyright 2018 por J F reinosa.

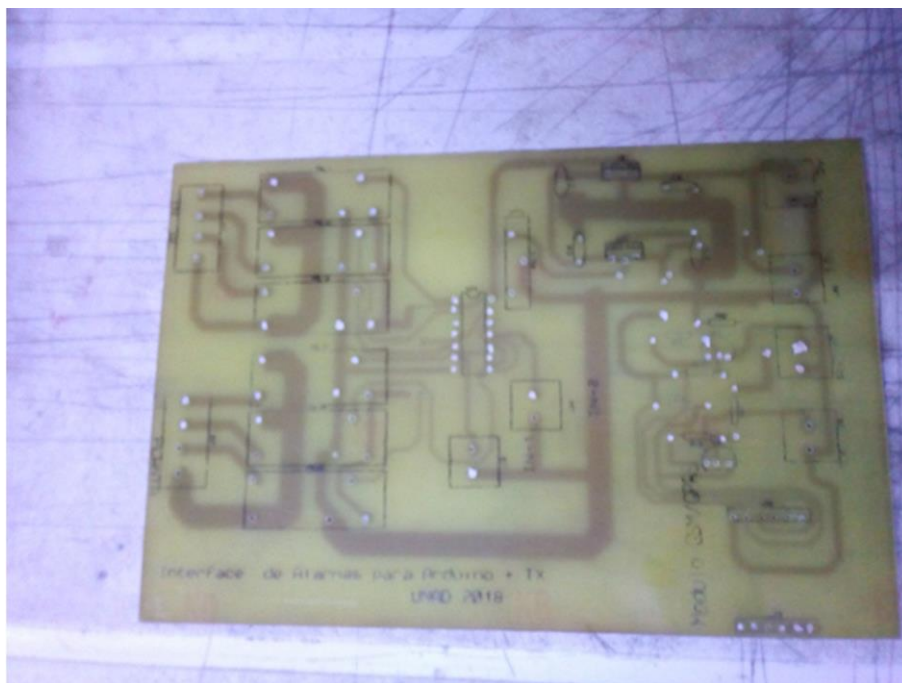


Figura 59. PCB tarjeta alarmas. Copyright 2018 por J F reinoso.

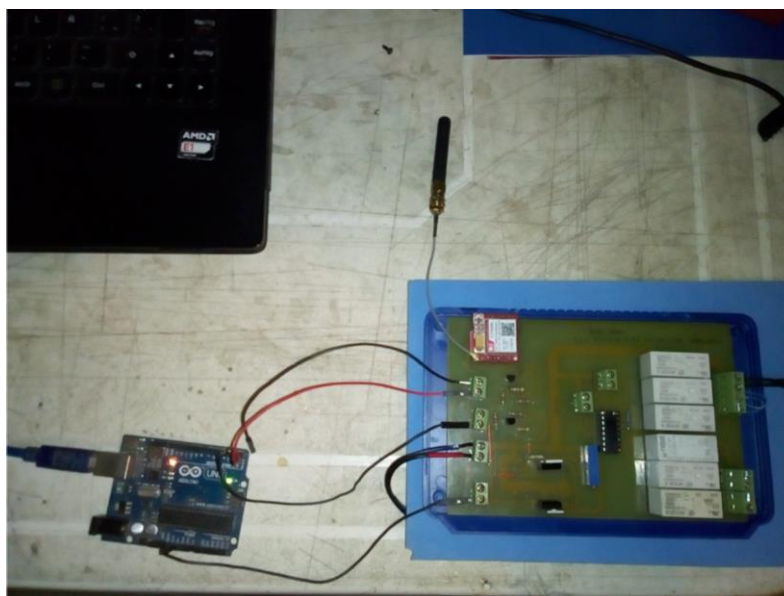


Figura 60. Conexión PCB tarjeta alarmas. Copyright 2018 por J F reinoso.

4.6 Centro de gestión

Para la gestión o monitoreo de las alarmas generadas por el sistema se pueden utilizar varias aplicaciones entre ellas está el popular Nagios muy utilizado por los administradores de redes, pero también existen otras opciones más fáciles de implementar y de configurar estos son los clientes MQTT que han sido desarrollados para trabajar como complementos de los navegadores de internet como crome o Mozilla. Para este proyecto se escogió a MQTTBox, el cual trabaja como complemento del navegador Google Chrome. Esta aplicación nos permite recibir los mensajes enviados por el sistema de monitoreo a través de la ventana de *topic* o suscripción.

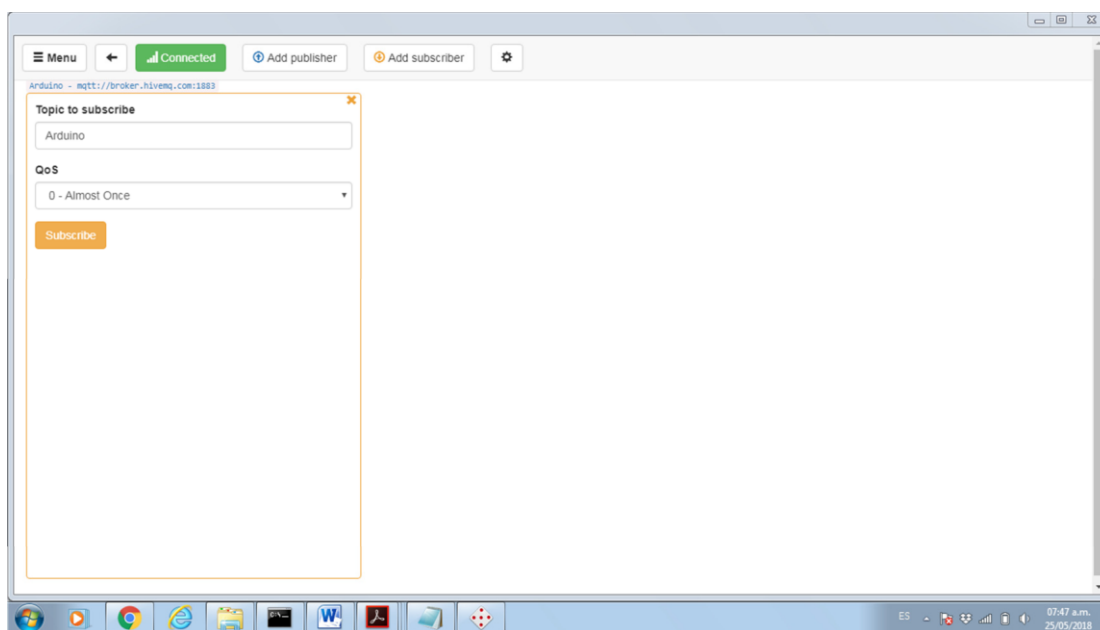


Figura 61. MQTTBox. Copyright 2018 por J F reinosa.

4.7 Simulación de alarmas desde la interfaz

Para la simulación de alarmas se utilizó MQTTbox el cual permite gestionar las alarmas que se vayan presentando, para esto en el aplicativo se deben configurar los parámetros utilizados por el sistema de monitoreo, estos son el servidor utilizado y el nombre del tema. Con estos parámetros ya podemos empezar a ver los eventos presentados en el sistema.

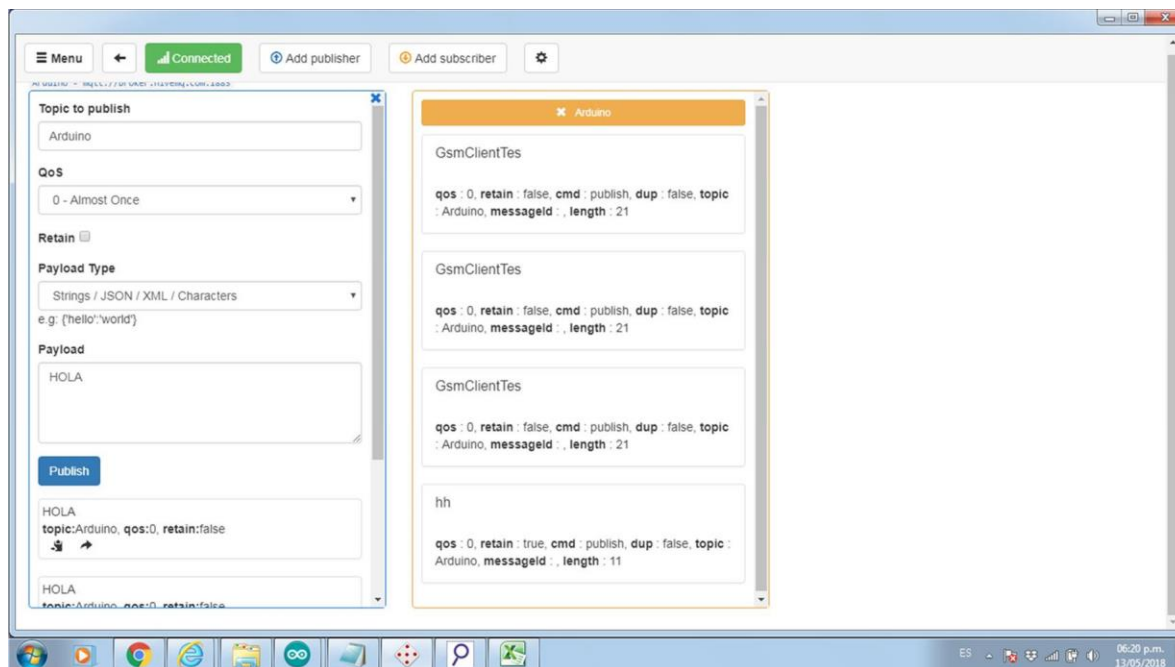


Figura 62. Recepción de alarmas. Copyright 2018 por J F reinoso.

Conclusiones

Como conclusión podemos decir que este proyecto cumple con el objetivo planteado desde el inicio, ya que permite la gestión de alarmas que se presenten en el nodo de manera oportuna gracias a la tecnología Arduino y a la posibilidad de usar la red celular para enviar dicha información.

Es de destacar que el protocolo MQTT fue de gran ayuda para la implementación del proyecto, ya que este protocolo posee características excepcionales con respecto a SNMP. Una de ellas es el tamaño de trama, la cual es muy pequeña lo que permite implementarse en redes con poco ancho de banda como lo son las redes celulares.

Con respecto a la tecnología Arduino podemos decir que fue fundamental en el desarrollo del proyecto ya que permitió hacer una implementación rápida del sistema, ya que el mismo es compatible con muchas librerías de comunicación por lo cual se pueden utilizar módulos GSM de muchos fabricantes. Adicionalmente al poseer entradas y salidas digitales permitió implementar de manera rápida la interface de alarmas.

Otro aspecto a destacar es la posibilidad de añadir nuevas funciones o variables al sistema ya que, aunque en los objetivos solo se plantean dos variables, en un futuro se pueden adicionar nuevas variables a monitorear. Además, estas nuevas variables pueden ser de diferente naturaleza ya que la tecnología Arduino permite incorporar varios tipos de sensores como lo son de temperatura, húmeda, presión, etc.

Respecto al software de gestión se puede decir que se deben hacer mejoras ya que el utilizado en este proyecto es una versión de uso libre, y que funciona en complemento con el explorador chrome, por lo tanto sería interesante agregarle otras funciones para mejorar el aspecto grafico así como una base de datos para almacenar estadísticas del sistema.

Lo anterior no solo permite utilizar este sistema para nodos de telecomunicación, como es en este caso, sino poder ampliar su uso a nuevos ambientes como la domótica.

Recomendaciones

Algunas recomendaciones para la mejora del sistema de monitoreo son:

1. Añadir una pantalla para visualizar el estado de alarmas y la operación del sistema.
2. Agregar una función para desactivar el sistema por mantenimiento del mismo.
3. Agregar una base de datos en Arduino para guardar los eventos localmente y poder visualizarlos en pantalla.

Lista de referencias

Areatecnologia. (2018). Relés. [Figura 26]. Recuperado de

<http://www.areatecnologia.com/electricidad/rele.html>

Aprendiendo Arduino. (2018). IDE Arduino y Configuración. [Figura 18]. Recuperado de

<https://aprendiendoarduino.wordpress.com/category/ide/>

Arduino. (2018). ¿Que es arduino?. [Figura 15]. Recuperado de <https://www.arduino.cc/>

Arduino. (2018). Reset usando sim8001. [Figura 28]. Recuperado de

<https://arduino.stackexchange.com/questions/50557/arduino-uno-reset-using-sim8001>

Bayle, J. (2013). C Programming for Arduino: Learn How to Program and Use Arduino Boards with a Series of Engaging Examples, Illustrating Each Core Concept. Birmingham: Packt Publishing. Retrieved from

<http://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=581716&lang=es&site=eds-live&scope=site>

Blake, R. (2004). GSM. In Sistemas electrónicos de comunicaciones (2nd ed., pp. 832-835).

Mexico City: Cengage Learning. Retrieved from

Calvo, G. Á. L. (2014). Gestión de redes telemáticas (UF1880). Madrid, ES: IC Editorial.

Retrieved from

<https://bibliotecavirtual.unad.edu.co:2538/lib/unadsp/detail.action?docID=4310542&query=gestion+de+redes>

- Cama Pinto, A., De la Hoz Franco, E., & Cama Pinto, D. (2012). Las redes de sensores inalámbricos y el internet de las cosas. INGE CUC, 8(1), 163-172. Recuperado de <http://revistascientificas.cuc.edu.co/index.php/ingecuc/article/view/253/232>
- Eeproject. (2018). Arduino uno board - full description. [Figura 16]. Recuperado de <https://eeproject.com/arduino-uno-board/>
- Figueroa, D. L. C. M. (2008). Introducción a los sistemas de telefonía celular. Buenos Aires, AR: Editorial Hispano Americana HASA. Recuperado de
- Flores-Medina, M., Flores-García, F., Velasco-Martínez, V., González-Cervantes, G., & Jurado-Zamarripa, F. (2015). Monitoreo de humedad en suelo a través de red inalámbrica de sensores/Monitoring soil moisture using a wireless sensor network. Tecnología y Ciencias Del Agua, 6(5), 75-88. Recuperado de <https://bibliotecavirtual.unad.edu.co:2357/docview/1786636543?accountid=48784>
- Fórum arduino. (2018). Sim800l. [Figura 29]. Recuperado de <https://forum.arduino.cc/index.php?topic=473845.15>
- Fpune. (2017). Introducción a IoT usando protocolo mqtt.[Figura 21 y 22]. Recuperado de <http://www.fpune.edu.py/ect2017/doc/Materiales/Introducci%C3%B3n%20a%20IoT%20usando%20protocolo%20MQTT.pptx>

- García, J. P., Molina, G. J. M., & Llácer, L. J. (2014). Sistemas de comunicaciones móviles: segunda, tercera y cuarta generación. Cartagena, ESPAÑA: Universidad Politécnica de Cartagena. Recuperado de <https://bibliotecavirtual.unad.edu.co:2538/lib/unadsp/detail.action?docID=4794898&query=sistemas+de+comunicaciones+moviles>
- García, L., & Ortiz, J. (2012). TECNOLOGÍAS INVOLUCRADAS EN LA INTERNET DEL FUTURO. Revista vínculos, 9(2), 115-127. Recuperado de <https://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4283>
- Geekfactory. (2019). Alimentar el arduino. [Figura 36]. Recuperado de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/>
- Gestión de redes telemáticas. (2014). Entidades que participan en la gestión [Figura 1]. Recuperado de <https://bibliotecavirtual.unad.edu.co:2538/lib/unadsp/detail.action?docID=4310542&query=gestion+de+redes>
- Gestión de redes telemáticas. (2014). Protocolos de gestión [Figura 2]. Recuperado de <https://bibliotecavirtual.unad.edu.co:2538/lib/unadsp/detail.action?docID=4310542&query=gestion+de+redes>
- Gestión de redes telemáticas. (2014). CMI SE/CMIP de OSI [Figura 3]. Recuperado de <https://bibliotecavirtual.unad.edu.co:2538/lib/unadsp/detail.action?docID=4310542&query=gestion+de+redes>

Gestión de redes telemáticas. (2014). SNMP de TCP/IP [Figura 4]. Recuperado de

<https://bibliotecavirtual.unad.edu.co:2538/lib/unadsp/detail.action?docID=4310542&query=gestion+de+redes>

Hivemq. (2018). Conexión. [Figura 57]. Recuperado de <https://www.hivemq.com/try-out/>, 2018,

<http://www.hivemq.com/demos/websocket-client/>

Humidor Moya, J. M. (2006). Los Servicios de Comunicaciones Móviles. In Redes y servicios de telecomunicaciones (pp. 173-221). Madrid: Paraninfo. Recuperado de

Introducción a IoT usando protocolo MQTT. Recuperado de

<http://www.fpune.edu.py/ect2017/doc/Materiales/Introducci%C3%B3n%20a%20IoT%20usando%20protocolo%20MQTT.pptx>

Introducción a los Sistemas de telefonía celular. (2008). Celula practica y analitica. [Figura 11].

Recuperado de

<http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=74&docID=10337342&tm=1504744704874>

Kocjan, W. (2014). Learning Nagios 4 (Vol. 2nd ed). Birmingham, UK: Packt Publishing.

Recuperado de

<http://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=734985&lang=es&site=eds-live&scope=site>

Los Servicios de Comunicaciones Móviles. (2006). Arquitectura de una red GSM. [Figura 8].

Recuperado de

<http://bibliotecavirtual.unad.edu.co:2081/ps/i.do?p=GVRL&sw=w&u=unad&v=2.1&it=r&id=GALE%7CCX3647500008&asid=6e803c95e8f5ee55fc432f39962ad7ab>

Los Servicios de Comunicaciones Móviles. (2006). La telefonía vía radio. [Figura 9].

Recuperado de

<http://bibliotecavirtual.unad.edu.co:2081/ps/i.do?p=GVRL&sw=w&u=unad&v=2.1&it=r&id=GALE%7CCX3647500008&asid=6e803c95e8f5ee55fc432f39962ad7ab>

Los Servicios de Comunicaciones Móviles. (2006). HSCSD, GPRS y EDGE: Datos a alta velocidad. [Figura 13]. Recuperado de

<http://bibliotecavirtual.unad.edu.co:2081/ps/i.do?p=GVRL&sw=w&u=unad&v=2.1&it=r&id=GALE%7CCX3647500008&asid=6e803c95e8f5ee55fc432f39962ad7ab>

Mactronica. (2018). Módulo gsm gprs sim800 con antena. [Figura 44]. Recuperado de

<http://www.mactronica.com.co/modulo-gsm-gprs-sim800l-con-antena-57044197xJM>

micro-manager. (2018). Arduino. [Figura 31]. Recuperado de <https://micro-manager.org/wiki/Arduino>.

Llaneza, G. P. (2018). *Seguridad y responsabilidad en la internet de las cosas (iot)*. Retrieved from <https://ebookcentral-proquest-com.bibliotecavirtual.unad.edu.co>

Olguer Sebastián, M. V., Julián Rolando, C. L., & Elvis Eduardo, G. G. (2015). Sistema de monitoreo para pacientes de alto riesgo integrando módulos GPS, GSM/GPRS y zigbee.

Tecnura, 19, 97-111. doi:

<http://bibliotecavirtual.unad.edu.co:2469/10.14483/udistrital.jour.tecnura.2>

Olups, R. (2010). Zabbix 1.8 Network Monitoring: Monitor Your Network's Hardware, Servers, and Web Performance Effectively and Efficiently. Birmingham, UK: Packt Publishing.

Retrieved from

<http://bibliotecavirtual.unad.edu.co/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=333462&lang=es&site=eds-live&scope=site>

Oswagar. (2018). Microcontroladores. [Figura 17]. Recuperado de

<http://oswagar.galeon.com/enlaces2530563.html>

Peña, W. Y., Rodríguez, L. R., Bient, J. R., & León, P. G. (2018). Monitoreo, control y diagnóstico en bancos de capacitores automáticos en baja tensión. *Ingeniare : Revista Chilena De Ingeniería*, 26(1), 28-42. Retrieved from

<https://bibliotecavirtual.unad.edu.co:2357/docview/2014455301?accountid=48784>

Pandorafms. Recuperado de <https://pandorafms.org/es/>

Pandorafms. Recuperado de <https://pandorafms.org/es/producto/introduccion-a-pandora-fms/>

pcwldd. (2018) .Pandora FMS Network Management Review. [Figura 5]. Recuperado de

<https://www.pcwldd.com/pandora-fms-network-management-review>.

Raymondtunning. (2018). A6 Gsm/Gprs module pinout and data. [Figura 36]. Recuperado de

<https://raymondtunning.wordpress.com/2016/05/31/a6-gsmgprs-module-pinout-and-data/>

Revistas unidistrital. (2012). Tecnologías involucradas en el internet del futuro. [Figura 19].

Recuperado de <https://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4283>

Revistas unidistrital. (2012). Tecnologías involucradas en el internet del futuro. [Figura 20].

Recuperado de <https://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4283>

Simcom. (2018). Sim800. [Figura 30]. Recuperado de

<http://simcomm2m.com/En/module/detail.aspx?id=138>

Simcom. (2018). Sim800. [Figura 41]. Recuperado de

<http://simcomm2m.com/En/module/detail.aspx?id=138>

Simcom. (2018). Sim800. [Figura 42]. Recuperado de

<http://simcomm2m.com/En/module/detail.aspx?id=138>

Sistemas de Comunicaciones Móviles. (2014). Estructura de la red GSM. [Figura 10].

Recuperado de

<http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=31&docID=11335528&tm=1504743923615>

Sistemas de Comunicaciones Móviles. (2014). General packet radio service (GPRS). [Figura 14].

Recuperado de

<http://bibliotecavirtual.unad.edu.co:2077/lib/unadsp/reader.action?ppg=31&docID=11335528&tm=1504743923615>

Steelintheair. (2018). Estudios sanitarios sobre radio frecuencias. [Figura 12]. Recuperado de

<http://www.steelintheair.com/es/estudios-de-salud-de-radio-frecuencias.html#.WvxC1--FPIV>

Software. (2018) .Nagios. [Figura 7]. Recuperado de <http://www.software.com.ar/p/nagios>

Velásquez. (2018). Transferencias Automáticas con Contactores.[Figura 23]. Recuperado de

http://www.velasquez.com.co/paginas/transferencia_automatica_con_contactores.php

3gpp. (2018). SP-99547. [Figura 45]. Recuperado de

http://www.3gpp.org/ftp/tsg_sa/tsg_sa/tsgs_06/docs/pdf/SP-99547.PDF

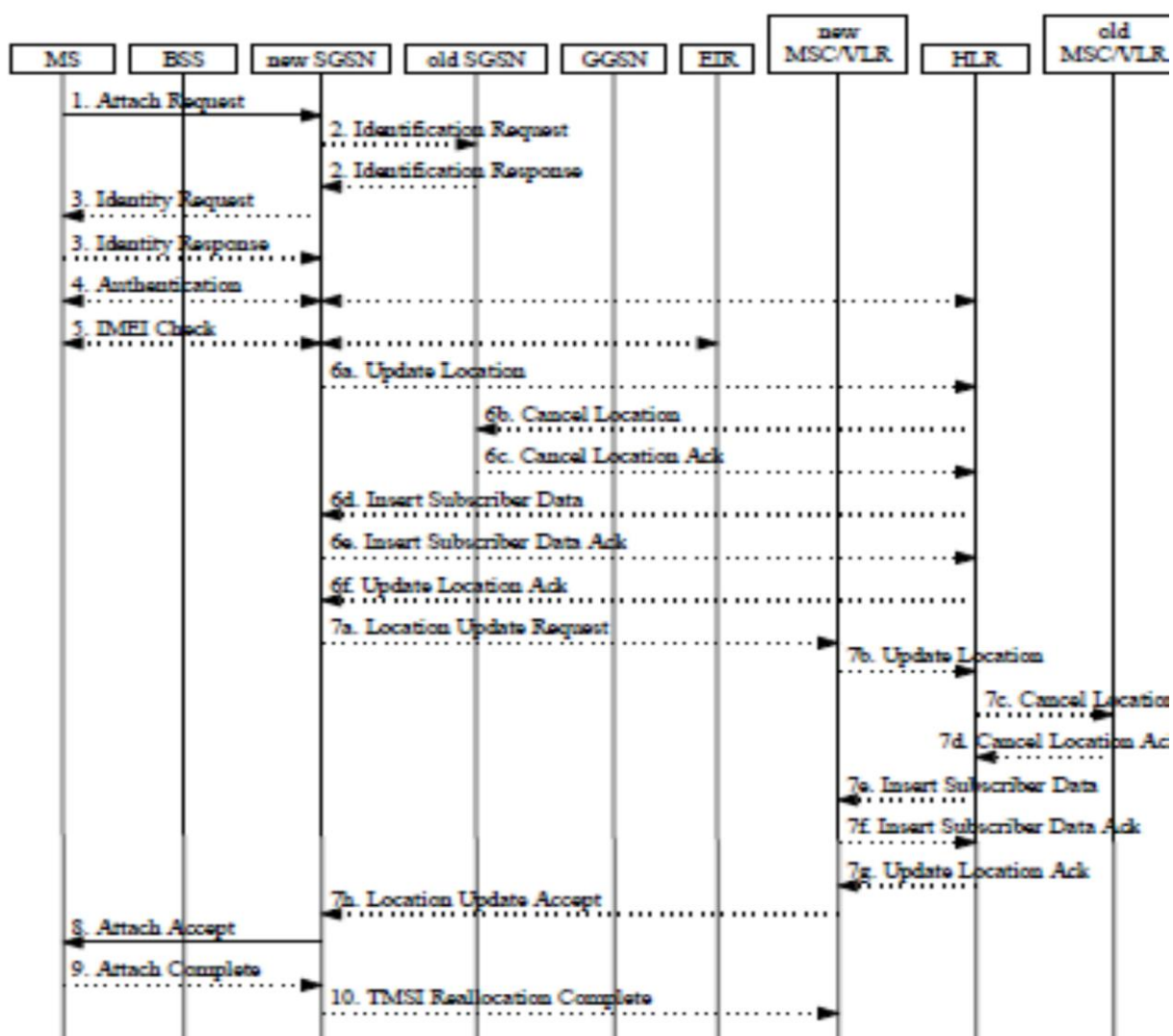
3gpp. (2018). especificación técnica GSM 03.60. [Figura 46]. Recuperado de

http://www.3gpp.org/ftp/tsg_sa/tsg_sa/tsgs_06/docs/pdf/SP-99547.PDF

Anexos

Apéndice A

Mecanismo de adhesión a la red GPRS



Apéndice B

Comandos AT (sim800)

Comandos GPRS soportados

AT+CGATT Attach or detach from GPRS service

AT+CGDCONT Define PDP context

AT+CGQMIN Quality of service profile (minimum acceptable)

AT+CGQREQ Quality of service profile (requested)

AT+CGACT PDP context activate or deactivate

AT+CGDATA Enter data state

AT+CGPADDR Show PDP address

AT+CGCLASS GPRS mobile station class

AT+CGEREP Control unsolicited GPRS event reporting

AT+CGREG Network registration status

AT+CGSMS Select service for MO SMS messages

Apendice C

Librería tyni

Nombre	Fecha de modifica...	Tipo	Tamaño
examples	05/09/2018 07:38 a...	Carpeta de archivos	
extras	05/09/2018 07:38 a...	Carpeta de archivos	
tools	05/09/2018 07:38 a...	Carpeta de archivos	
keywords		Documento de tex...	1 KB
library.json		Archivo JSON	1 KB
library.properties		Archivo PROPERTI...	1 KB
LICENSE		Archivo	8 KB
Makefile		Archivo	1 KB
README.md		Archivo MD	7 KB
TinyGsmClient.h	04/03/2018 03:19 ...	Archivo H	2 KB
TinyGsmClientA6.h	04/03/2018 10:10 ...	Archivo H	17 KB
TinyGsmClientESP8266.h		Archivo H	9 KB
TinyGsmClientM590.h		Archivo H	16 KB
TinyGsmClientSIM800.h	25/03/2018 10:53 a...	Archivo H	22 KB
TinyGsmClientSIM808.h		Archivo H	4 KB
TinyGsmCommon.h		Archivo H	3 KB
TinyGsmFifo.h		Archivo H	3 KB

Apendice D

Librería pubsubclient

Nombre	Fecha de modifica...	Tipo	Tamaño
examples	05/09/2018 07:38 a...	Carpeta de archivos	
src	05/09/2018 07:38 a...	Carpeta de archivos	
tests	05/09/2018 07:38 a...	Carpeta de archivos	
.gitignore	17/04/2018 06:45 a...	Archivo GITIGNORE	1 KB
.travis.yml	17/04/2018 06:45 a...	Archivo YML	1 KB
CHANGES	17/04/2018 06:45 a...	Documento de tex...	3 KB
keywords	17/04/2018 06:45 a...	Documento de tex...	1 KB
library.json	17/04/2018 06:45 a...	Archivo JSON	1 KB
library.properties	17/04/2018 06:45 a...	Archivo PROPERTI...	1 KB
LICENSE	17/04/2018 06:45 a...	Documento de tex...	2 KB
README.md	17/04/2018 06:45 a...	Archivo MD	2 KB