

**SISTEMA DE COMUNICACIÓN DE UN TERMINAL MÓVIL PARA
NARICES ELECTRÓNICAS BASADAS EN IOT**

**JOHN KEVIN CONDE VILLAMIZAR
JOHAN ALFONSO RANGEL RUIZ**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y
TELECOMUNICACIONES
BUCARAMANGA
2015**

**SISTEMA DE COMUNICACIÓN DE UN TERMINAL MÓVIL
PARA NARICES ELECTRÓNICAS BASADAS EN IOT**

**JOHN KEVIN CONDE VILLAMIZAR
JOHAN ALFONSO RANGEL RUIZ**

**Trabajo de Grado presentado como requisito para obtener el título de
Ingeniero Electrónico**

Director:

HOMERO ORTEGA BOADA

Ingeniero Electrónico, Ph.D

Codirector:

DIANA CAROLINA CARRILLO ARENAS

Ingeniera Electrónica, Ms(c)

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y
TELECOMUNICACIONES
BUCARAMANGA
2015**

TABLA DE CONTENIDO

	Pág.
INTRODUCCION.....	11
1. DESCRIPCIÓN DE LAS TECNOLOGÍAS, TEORÍAS Y CONCEPTOS QUE APOYAN EL DESARROLLO.....	13
1.1 PLATAFORMA DE RADIOGIS.....	13
1.2 WEB SERVICE COMO HERRAMIENTA DE COMUNICACIÓN.....	14
2. METODOLOGÍA PARA EL DESARROLLO DEL TERMINAL MÓVIL.....	17
2.1 PRIMER PASO: IDENTIFICACIÓN DE LOS REQUERIMIENTOS PARA EL DESARROLLO....	17
2.2 SEGUNDO PASO: MODELO DE CAPAS PARA LA SOLUCIÓN.....	18
2.3 TERCER PASO: SELECCIÓN DE LAS TECNOLOGÍAS.....	19
2.3.1 Tarjetas Desarrolladoras.....	20
2.3.2 GPS Shield's.....	21
2.3.3 Módem 3G.....	22
3. IMPLEMENTACIÓN DEL TERMINAL MÓVIL.....	24
3.1 OBTENCIÓN DE DATOS DE GPS.....	25
3.2 DATOS SIMULADOS.....	26
3.3 DISEÑO DEL CLIENTE WEB SERVICE DE APOYO A LA COMUNICACIÓN.....	27
3.4 CONEXIÓN POR RED 3G.....	29
4. ESTUDIO DE DESEMPEÑO.....	31
4.1 PRUEBAS DE DESEMPEÑO.....	31

4.2 COSTOS.....	33
5. CONCLUSIONES.....	34
CITAS.....	36
BIBLIOGRAFIA.....	37

LISTA DE FIGURAS

	Pág.
Figura 1. Esquema General de la Plataforma.....	13
Figura 2. Explicación de los Web Services.....	15
Figura 3. Modelo de capas del terminal móvil.....	18
Figura 4. 3G/GPRS Shield.....	22
Figura 5. Modem Wi-Fi 3G.....	23
Figura 6. Esquema general de la solución e integración de las tecnologías.....	24
Figura 7. Tiempos estimados para el envío de datos por medio del terminal móvil de narices electrónicas.....	32

LISTA DE TABLAS

	Pág.
Tabla 1. Tarjetas desarrolladoras más comerciales.....	20
Tabla 2. Shield´s compatibles con las tarjetas de desarrollo.....	21
Tabla 3. Tiempo de Empaquetamiento.....	32
Tabla 4. Costo del TM desde cero.....	33

RESUMEN

TÍTULO: SISTEMA DE COMUNICACIÓN DE UN TERMINAL MÓVIL PARA NARICES ELECTRÓNICAS BASADAS EN IOT*

AUTORES: JOHN KEVIN CONDE VILLAMIZAR
JOHAN ALFONSO RANGEL RUIZ**

PALABRAS CLAVE: 3G, 4G, IOT, TARJETA DE DESARROLLO, TERMINAL MÓVIL, WEB SERVICES.

DESCRIPCIÓN:

Este artículo presenta una solución innovadora al problema de transmisión de los datos de un sistema de narices electrónicas. Esto lleva a la construcción de un terminal móvil, pues se apoya en las redes móviles 3G y puede adaptarse a las 4G para enlazar los sensores de las narices ubicados remotamente a un sistema de almacenamiento y cómputo en la nube. La solución se caracteriza por ser sencilla, económica, portable y escalable. Este artículo presenta el diseño realizado, pero su mayor valor está representado en una metodología original que se describe con suficiente detalle y que aplica para el diseño de cualquier otro terminal móvil que se requieran en sistemas SensIT con apoyo de los avances más modernos en Internet de las Cosas (IoT). Precisamente, esta metodología representa el principal aporte de este artículo y se describe de manera que la solución pueda ser fácilmente replicada en otros campos, brindando incluso enlaces donde se comparte el código desarrollado. La metodología combina un modelo de capas para la comunicación, una aplicación de software, un receptor GPS, un módem 3G y técnicas de Web Services. El trabajo resulta útil para identificar las competencias que deben tener los nuevos ingenieros electrónicos que deben trabajar de manera mancomunada con ingenieros de sistemas para desarrollar soluciones convergentes que hoy reclama la industria y la sociedad.

* Trabajo de grado.

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director. PhD Homero Ortega Boada

ABSTRACT

TITLE: MOBILE TERMINAL COMMUNICATION SYSTEM FOR IOT-BASED ELECTRONIC NOSES*

AUTHORS: JOHN KEVIN CONDE VILLAMIZAR
.. JOHAN ALFONSO RANGEL RUIZ**

KEYWORDS: 3G, 4G, IOT, DEVELOPMENT BOARD, MOBILE TERMINAL, WEB SERVICES.

DESCRIPTION:

This paper presents a simple and innovative solution to the problem of transmitting data retrieved from electronic nose systems. This leads to the construction of a mobile terminal, since it supports itself on the 3G mobile networks and can be adapted to 4G linking nose sensors remotely located with storage and cloud computing systems. The solution is characterized for being simple, inexpensive, portable and scalable. This paper presents the design made, but its greatest value consists in an original methodology described in sufficient detail and which applies to the design of any other mobile terminal required in SensIT supported by the latest developments in the Internet of Things (IoT). In fact, this methodology represents the main contribution of this paper and is described in a way that it can be easily replicated in other fields, even providing links where the developed code was shared. The methodology combines a layered model for communication, a software application, a GPS receiver, a 3G modem and Web Service techniques. This work is useful to identify the skills that new electronic engineers must have, they are required to work along with systems engineers to develop convergent solutions that the industry and the society are claiming.

* Bachelor Thesis

** Faculty of Mechanical and Physical Engineering. School of Electrical Electronics and Telecommunications engineering. Director. PhD Homero Ortega Boada

INTRODUCCIÓN

Atrás quedaron los tiempos en que solo las personas podían conectarse a Internet, hoy lo hacen los equipos de medición, las prendas de vestir, las neveras, y tienden a hacerlo todas las cosas para potenciar las actividades humanas. Es por eso que el término de moda en las TIC es “Internet of Things” (IoT). Entre esos nuevos avances tecnológicos, no dejan de impresionar los lentes de Google y otros dispositivos que potencian los sentidos humanos. El reto que se aborda en este artículo nació precisamente de la necesidad de potenciar uno de estos sentidos con una tecnología llamada narices electrónicas.

Las narices electrónicas se vienen estudiando desde hace varios años, pero nunca se ha explotado las capacidades que pueden tener si se proyectan como una solución IoT, como se lo ha planteado el CentroTIC¹ de la Universidad Industrial de Santander. La ventaja que tiene esta propuesta es la optimización del costo del dispositivo que se ubica cerca al usuario, pues la conexión que mantiene con internet permite realizar el procesamiento de datos de forma remota en la nube, potencializando las necesidades de cómputo. Pero las ventajas no paran allí, pues cientos o miles de estos dispositivos pueden ser atendidos de manera centralizada por una única solución. Otra ventaja está en la flexibilidad que este tipo de implementación tiene para superar progresiva y ágilmente diferentes niveles de madurez, usando métodos ágiles de desarrollo. En términos de las Telecomunicaciones Móviles Internacionales (IMT) definidas por la Unión Internacional de Telecomunicaciones (UIT), todos esos dispositivos de usuarios que hacen uso de las redes móviles se conocen como Terminales Móviles (TM). Precisamente, el desarrollo de un terminal móvil para narices electrónicas representa un reto para el grupo de investigación en Radiocomunicaciones y sistemas de información geográficos (RadioGIS) que forma parte del CentroTIC.

El desarrollo del TM ha sido hecho en dos subproyectos: uno orientado a la captura de las señales que entregan los sensores y otro a la comunicación que naturalmente debe darse entre un TM con las

¹ Centro de Investigación Científica y de Desarrollo en Tecnologías de la Información y las Comunicaciones

redes móviles, el internet y los servicios en la nube. De esta manera, el problema que específicamente aborda el presente artículo consiste en la necesidad de identificar y diseñar una solución sencilla para que esa comunicación se dé usando redes 3G ó 4G, y georreferenciación con el fin de facilitar la masificación de las soluciones IoT que se den más adelante con este aporte. El artículo comienza con un estado del arte, para luego dedicarse a entregar al lector suficiente información para que esta visión pueda ser replicada, pues los resultados obtenidos aplican para cualquier desarrollo que use sensores, razón por la cual este artículo contribuye también a la nueva ciencia, la cual surgió del programa del DARPA-EEUU en el año 2000 conocido como *Sensor Information Technology* (SensIT).

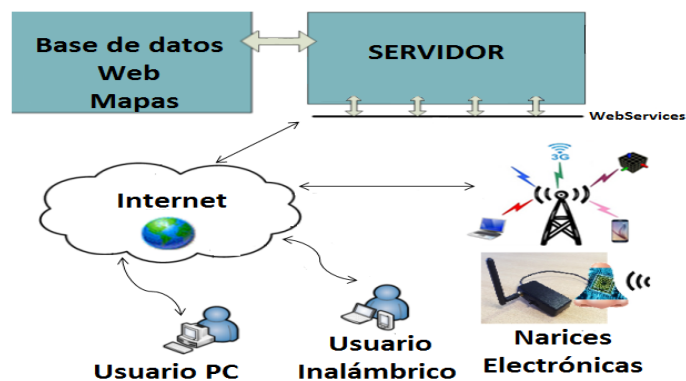
1. DESCRIPCIÓN DE LAS TECNOLOGÍAS, TEORÍAS Y CONCEPTOS QUE APOYAN EL DESARROLLO

Esta sección se aborda los conceptos y definiciones esenciales que todo ingeniero debe conocer para la implementación de terminales móviles. El capítulo comienza introduciendo el concepto de plataforma de servicios, analizando el caso particular de la plataforma del grupo de investigación RadioGIS (la cual ya cuenta con algunos servicios de Telecomunicaciones) para luego pasar a una descripción de los Web Services, dentro de cada uno de estos temas se tratan los diversos subtemas pertinentes a cada uno de ellos. De esta manera se obtiene un capítulo de vital importancia para el lector ya que todas estas tecnologías hacen parte de la solución de TM.

1.1. PLATAFORMA DE RADIOGIS

Una buena parte de las tecnologías, teorías y conceptos que apoyan el desarrollo son parte de una plataforma tecnológica para soportar soluciones IoT, como la que posee el grupo RadioGIS [1] y que se resumen en la Figura 1.

Figura 1. Esquema General de la Plataforma



En la parte inferior del esquema general de la plataforma están las tecnologías que se encuentran más cerca a los usuarios, pueden ser computadores conectados a Internet, pero que para el caso de IoT es básicamente cualquier cosa conectada a Internet. También se puede ver que existen diversos medios de acceso a la Internet como las redes Wi-Fi, las conexiones físicas por Ethernet o el xDSL y se resalta la expansión cada vez más segura de las redes de los operadores móviles, así mismo en la figura anterior el diseño propuesto en este trabajo aparece identificado como narices electrónicas. Como se había explicado en la introducción cualquier dispositivo conectado a las redes de comunicaciones móviles son considerados por la UIT como TM. Lo importante es que cualquiera que sea el medio de acceso, todos los dispositivos y usuarios resultan conectados a la Internet y a través de ella con uno o varios servidores en la nube donde se aloja la capacidad de almacenamiento y de cómputo. Uno de los logros del trabajo consiste en trazar los límites entre el trabajo que debe ser realizado por ingenieros de programación en la nube y el que debe ser realizado por ingenieros de terminales móviles. Un ingeniero de terminales móviles debe tener claro los modelos de capas, pero también ser hábil para usar Web Services (WS) ya que los WS son la herramienta predilecta para que cualquier aplicación remota intercambie información con otra sin importar el lenguaje de programación que se ha usado para desarrollar dichas aplicaciones, o en qué sistema operativo estén corriendo, o dónde están alojadas. Por eso, en la Figura 1 los WS aparecen como los intermediarios entre los recursos que se alojan en la nube y las tecnologías de acceso.

1.2. WEB SERVICE COMO HERRAMIENTA DE COMUNICACIÓN

Los WS representan la forma más sencilla y confiable para intercambiar datos con cualquier cosa que esté conectada a la Internet, permitiendo la comunicación de otros programas sin tener en cuenta el lenguaje en el que estén escritos, ya que manejan una forma estandarizada de comunicación. También pueden permitir una comunicación tan segura como sea necesaria, pues en ellos se pueden programar restricciones de acceso a los programas sobre los que una aplicación no debería tener permiso[2]. El siguiente ejemplo permite lograr suficiente claridad sobre los WS para los propósitos de la propuesta que se realiza en el presente artículo. La manera más fácil de entenderlos es haciendo un símil con un restaurante, tal como se ve en la figura 2.

Figura 2. Explicación de los Web Services.



El cliente encuentra un nuevo restaurante en las páginas amarillas, se dirige al lugar donde está ubicado, allí sobre la mesa encuentra un menú que le permite elegir la comida que prefiere. Entonces emite la orden al mesero, quien ha de llevarla a la cocina y posteriormente trae la comida al cliente. Finalmente, el cliente queda satisfecho. Es importante destacar que cualquier cliente que llegue a este restaurante lo único que debe saber es hablar el idioma español, pues el mesero resolverá las demás dificultades. Para el caso de los WS las páginas amarillas equivalen a directorios UDDI (Universal Description, Discovery and Integration) que están disponibles en la Internet para que los WS sean fáciles de ubicar; el mesero representa el web service que es una aplicación con una dirección URL dispuesta a prestar un servicio y el menú equivale a un WSDL (Web Services Description Language) que es el lenguaje usado para programar el WS; la orden equivale a una solicitud SOAP para la cual se obtiene una respuesta SOAP (Simple Object Access Protocol); finalmente, la cocina equivale a una base de datos o a un sistema de procesamiento en la nube. El lenguaje de comunicación entre el cliente de un WS y el WS en sí es usualmente texto escrito en XML, pero también se está imponiendo el lenguaje JSON, donde se usa otra terminología, por ejemplo en vez de SOAP se habla de REST.

Para el envío y almacenamiento remoto de los datos, es imprescindible contar con un WS que se encargue de recibirlos, validarlos, preprocesarlos y guardarlos en una base de datos ubicada en algún lugar remoto que llamaremos nube. El diseño de un WS usualmente estará a cargo de un ingeniero experto en soluciones en la nube, pero es necesario que el diseñador de un TM conozca de qué se trata el WS, los servicios que ofrece, la manera en que se invoca el servicio, lo cual equivale a diseñar un WS-cliente que se encargue de consumir el WS. Existen tres factores particularmente relevantes que el diseñador del TM debe tener presente: El formato de recepción de los datos (JSON, XML u otro), la dirección URL del WS y el método HTTP a usarse en el envío (GET, POST u otro).

La URL es el identificador del Web Service, una dirección web que apunta a su ubicación en el servidor, que permite invocarlo y establecer la conexión. El formato que se defina para los datos, permite al cliente enviarlos de forma que puedan reconocerse del otro lado de la comunicación, garantizando así que el mensaje transmitido sea interpretado de manera adecuada, conservando la integridad y la naturaleza de los datos. Por último, el método es un acuerdo entre el cliente y el servidor que le indica a este último dónde debe buscar los datos que el usuario le ha transmitido.

El cliente puede escribir en cualquier lenguaje de programación que permita establecer conexiones web; los lenguajes más comunes cuentan por lo general con módulos que se encargan de dicha labor facilitando la tarea al usuario. Para el trabajo con terminales es altamente recomendable utilizar Python, pues está bien soportado y su sintaxis es notablemente sencilla. Entre las alternativas para la conexión está el uso del módulo Requests que permite realizar peticiones HTTP de forma sencilla y transparente para el usuario. Los Web Services habitualmente indican al usuario si la operación concluyó exitosamente y suelen reportar en forma de error cualquier inconsistencia que haya sido detectada; es importante considerar dicha información al momento de diseñar el cliente, a fin de garantizar que los datos sean almacenados en la nube o tomar las medidas necesarias en caso contrario.

2. METODOLOGÍA PARA EL DESARROLLO DEL TERMINAL MÓVIL

El desarrollo realizado ha permitido identificar una metodología original para el desarrollo de un TM aprovechando al máximo las tecnologías actuales para llegar a una solución sencilla pero muy eficiente. Esta se describe a continuación.

2.1. PRIMER PASO: IDENTIFICACIÓN DE LOS REQUERIMIENTOS PARA EL DESARROLLO.

El primer paso consiste en determinar los requerimientos mínimos que deberá cumplir la solución, partiendo del propósito central del proyecto (satisfacer las necesidades de transmisión de datos de los dispositivos de narices electrónicas).

Para este proyecto se formularon los siguientes requerimientos:

- Permitir el envío de los datos mediante una comunicación confiable, lo cual quiere decir que se debe garantizar que los datos enviados lleguen satisfactoriamente a su destino.
- Usar redes móviles, 3G o tecnologías superiores, para enviar los datos a internet. Se precisa que la información que envía el TM esté asociada a la ubicación geográfica de los sensores, es decir, los datos deben georreferenciarse.
- Se busca que el TM sea tan sencillo, económico y portable como sea posible, de este modo, se requiere minimizar el procesamiento de datos necesario dentro del mismo. El procesamiento se trasladará a la nube, lográndose además el mejor aprovechamiento de los recursos con los que se cuenta.

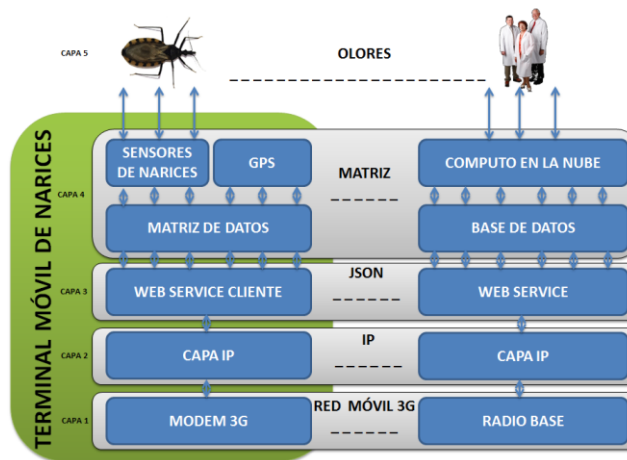
- El TM debe ser modular y escalable, esto quiere decir que podrá actualizarse o complementarse con nuevas tecnologías o métodos para acomodarse a nuevos retos.

Se tiene entonces un conjunto de requerimientos completos para visionar el desarrollo.

2.2. SEGUNDO PASO: MODELO DE CAPAS PARA LA SOLUCIÓN

En esta fase se debe comprender el sistema de comunicación para lo cual ha resultado útil usar un modelo de capas, como se presenta en la figura 3.

Figura 3. Modelo de capas del terminal móvil



Aquí cada capa tiene un protocolo que se encarga de la comunicación entre elementos de cada capa. Las capas del modelo que define cómo se realiza la transmisión de datos desde el TM a desarrollar, se explican a continuación:

Capa 1 - Capa de acceso inalámbrico: monta la información sobre ondas electromagnéticas que la llevan hasta una radiobases usando tecnologías avanzadas de acceso móvil como las llamadas 3G y las 4G.

Capa 2 - Capa IP: Permite que la información llegue a cualquier parte del mundo donde exista una dirección IP.

Capa 3 - Capa Web Services: facilita el acceso a servicios que se encuentran en la web sin importar en qué lenguaje de programación, ni en qué plataformas o sistemas operativos estén alojados, ya que usan un formato de datos estándar, común para el intercambio de información entre ellos. Realmente hay varios formatos como el WSDL (Web Services Description Language) o el JSON.

Capa 4 - Capa de adquisición y entrega de las señales provenientes de la fuente de información: Permite la traducción de señales de un mundo físico a un medio digital. En nuestro proyecto se usaron sensores de la nariz electrónica y GPS que generan la información que debe moverse por todas las capas hasta llegar al usuario final. Como se muestra en la figura, una matriz de datos georreferenciados es esperada por algoritmos de cómputo en la nube.

Capa 5 - Capa de usuarios: Se encarga de poner en contacto, aunque de manera remota, al usuario con los sensores. Por eso, en nuestro caso, los médicos pueden percibir remotamente la información de los olores captados por los sensores de una nariz electrónica desarrollada para detectar triatominos, unos insectos capaces de transmitir el vector del mal de Chagas.






2.3. TERCER PASO: SELECCIÓN DE LAS TECNOLOGÍAS

Una vez se determina el modelo de capas, se debe seleccionar las tecnologías más apropiadas para el desarrollo del TM. Hoy en día hay una alta gama de equipos y tecnologías que ayudan a

solucionar diferentes problemas de comunicación, con una gran variedad de precios y funcionalidades. A continuación se describen los elementos que se necesitan para el desarrollo del TM y las opciones que se encuentran en el mercado.

2.3.1 Tarjetas Desarrolladoras. En la tabla 1, se comparan las tarjetas que abarcan la mayor parte del mercado actual.

Tabla 1. Tarjetas desarrolladoras más comerciales.

Imagen de la Tarjeta Desarrolladora					
Tarjeta	Arduino Uno	Raspberry Pi 2	BeagleBone Black	Intel Galileo	Intel Edison
Precio (USD)	\$25.53	\$41.85	\$56.95	\$42.99	\$95.99
Procesador	ATmega328	BCM2836 ARMv7	AM3359A Cortex-A8	Intel Quark SoCX1000	Dual Intel Quark SoCX1000
Velocidad del procesador	16 MHz	900 MHz	1 GHz	400 MHz	500 MHz
Pines Analógicos	6	-	7	6	6
Pines Digitales	14 (6 PWM)	40 Digitales GPIO	65 GPIO (8PWM)	14 (6 PWM)	20 (4 PWM)
Memoria	SRAM 2 KB EEPROM 1KB	1 GB RAM	512 MB RAM DDR3L, eMMC 2GB	SRAM 512 KB DRAM 256 MB EEPROM 11KB	SRAM 512 KB DRAM 256 MB EEPROM 11 KB
Lenguajes de Programación	Arduino IDE C Variant, Matlab	Cualquiera	Cualquiera	Cualquiera	Cualquiera

De este análisis, se escogió como mejor opción la Intel Edison para este desarrollo, ya que cuenta con un módulo Wi-Fi incorporado, permitiendo una conexión inalámbrica a internet.

2.3.2. GPS Shield's. A continuación se analizan diferentes shields disponibles como soluciones de geoposicionamiento. En la Tabla 2 se presentan tres GPS shield's para cada una de las tarjetas de la Tabla 1.

Tabla 2. Shield's compatibles con las tarjetas de desarrollo.

TARJETA	SHIELDS GPS COMPATIBLES	RUTA DE HALLAZGO	PRECIO (USD)
ARDUINO	Adafruit Ultimate GPS Logger Shield - Includes GPS Module	goo.gl/UFwyT0	\$49.95
	Arduino M2MShield	goo.gl/UFwyT0	\$67.253
	3G/GPRS shield	goo.gl/qxZ0EN	\$265
RASPBERRY	Navigatron v2 -I2C	goo.gl/Cqa06w	\$39.09
	Raspberry PI GPS Add-on V1.0	goo.gl/IMdtXF	\$28.50
	3G/GPRS shield	goo.gl/qxZ0EN	\$265
BEAGLE BOND	Tracking Cape for Beagle Bone	goo.gl/xVG4cO	\$85
	VAYU GPS/TIM	goo.gl/9MDa6x	\$62.00
	GPS/GPRS Cape	goo.gl/WCnUZM	\$115.22
INTEL GALILEO	Módulo GPS para Arduino, Raspberry Pi e Intel Galileo	goo.gl/jQJgSG	\$46.05
	Grove - GPS	goo.gl/O2W0SC	\$29.90
		goo.gl/qxZ0EN	\$265

Tabla 2. Continuación

3G/GPRS shield			
INTEL EDISON	Adafruit Ultimate GPS Logger Shield- Includes GPS Module	goo.gl/SLA9Ub	\$49.95
	Kit Sparkfun Shield GPS	goo.gl/PZTvPa	\$ 59.95
	3G/GPRS shield	goo.gl/qxZ0EN	\$265
	3G/GPRS shield		

Al contar con el shield 3G/GPRS, el cual se muestra en la figura 4, resultó siendo compatible con las necesidades de la tarjeta de desarrollo seleccionada. Así, para el caso de la conexión del dispositivo se usan los puertos RX y TX, que corresponden al puerto de recepción y el de transmisión de los datos respectivamente. También se utiliza un pin de alimentación que suministra el voltaje para el funcionamiento de shield y la tierra (GND) de la tarjeta. Los 20 pines digitales que funcionan como entrada o salida y los 6 pines análogos de la tarjeta Intel Edison permiten conectar los diferentes shields y establece una comunicación serial entre los puertos de entrada y salida de la tarjeta de desarrollo y del shield.

Figura 4. 3G/GPRS Shield



2.3.3. Módem 3G. Se trata de nuevos dispositivos que facilitan la conexión a internet de cualquier cosa. Para este proyecto se eligió el TP LINK Wi-Fi Móvil 3G M5350, el cual se observa en la figura 5, cuyas cualidades son: tamaño diminuto, peso muy bajo, autonomía, se conecta a internet por redes móviles que están en todas partes, comparte internet a otros dispositivos por Wi-Fi.

Figura 5. Modem Wi-Fi 3G

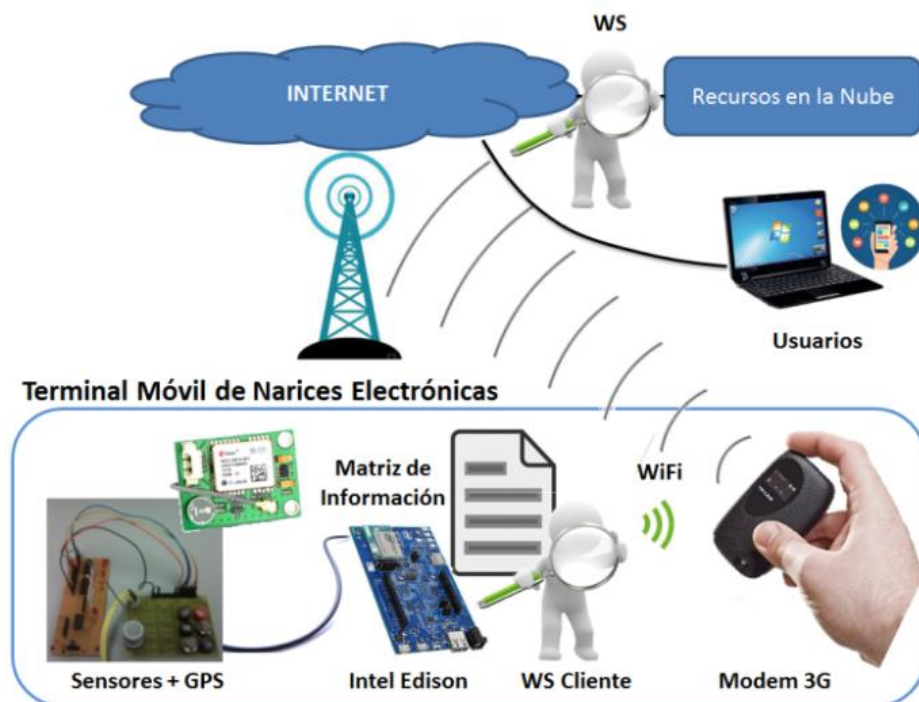


Como puede verse, la metodología presentada es muy sencilla y eficiente para el tipo de soluciones que requieren sensado remoto de información

3. IMPLEMENTACIÓN DEL TERMINAL MÓVIL

Para el desarrollo del prototipo se deben tener en cuenta las tecnologías apropiadas, además del acople de cada una de ellas para poder desarrollar el TM. En la figura 6, se puede observar el esquema de la solución planteada en este artículo, esta solución consta de varios elementos entre los cuales se encuentra, un módulo GPS, la tarjeta desarrolladora, la organización de los datos en una matriz, el WS cliente, un módulo de comunicación inalámbrica, las radiobases, el WS y los usuarios. La solución debe cumplir con los requisitos dados en el capítulo tres.

Figura 6. Esquema general de la solución e integración de las tecnologías



Todos estos elementos integran una solución que aprovecha los recursos ya disponibles, como el Web Service de RadioGIS, las radiobases de los operadores de telecomunicaciones que son el

medio por el cual recibe y envía la información, y los diferentes equipos que se encuentran en el mercado actual como lo son las tarjetas desarrolladora, los módulos GPS y los módems inalámbricos. A continuación se explican algunos de los elementos que hacen parte de la solución:

3.1. OBTENCIÓN DE DATOS DE GPS

Al georreferenciar la información los datos transmitidos toman mayor importancia, porque esto da conocimiento de su lugar de procedencia. Esto se puede lograr a través de diferentes módulos GPS. Algunos de ellos ya vienen compatibles con las tarjetas desarrolladoras, los cuales se conocen en el mercado como “*shield's*”, diseñadas para dar soluciones sencillas a problemas comunes.

Es de suma importancia tener en cuenta los puertos de entrada y salida que tienen los shield's para poder lograr la comunicación con el módulo GPS. En este caso se tiene el shield 3G/GPRS, el cual fue seleccionado por su valor didáctico, permitiendo acceder al módulo GPS por medio del puerto miniUSB o por los pines de expansión que posee, los cuales se comportan como entrada o salida. En ambos casos se establece una comunicación serial. Pero existen diferentes “*shield's*” que se encuentran en el mercado y para lograr la recolección de datos todos ellos deben cumplir los siguientes pasos:

Paso 1: Reconocimiento del hardware adquirido (Puerto de Entrada, Salida y Alimentación).

Paso 2: Modelo de Trabajo para el desarrollo de software (Familia de Microcontrolador, Procesador y SO).

Paso 3: Selección del lenguaje de Programación (C, C++, Python, etc.).

Paso 4: Pruebas Iniciales de Comunicación.

Paso 5: Organización de la Información a través del software.

Todos estos pasos permiten entablar la comunicación entre la tarjeta desarrolladora y el GPS, de esta forma se logra recolectar los datos de georreferenciación.

3.2. DATOS SIMULADOS

Para desarrollar pruebas de transmisión y utilizar el WS cliente se crearon archivos de datos con las siguientes características:

- Cada dato está conformado hasta por 6 caracteres.
- Los datos están organizados en una matriz, la cual cuenta con 16 columnas y n filas.

Las 16 columnas corresponden al número de sensores que conforman la nariz electrónica, este número de sensores puede variar, cada columna se encuentra separada por espacios. Las n filas corresponden a la cantidad de muestras que se están capturando por sensor. Las unidades de estos valores son irrelevantes para la transmisión ya que solo se busca transmitir los archivos que contiene los datos.

Con esta información se crearon archivos similares que contenían las mismas características, también se utilizaron los archivos reales generados por sensores de narices electrónicas para realizar pruebas de transmisión. Es importante aclarar que estos datos se organizaron en forma de matriz siendo el formato que arroja la nariz electrónica real.

3.3. DISEÑO DEL CLIENTE WEB SERVICE DE APOYO A LA COMUNICACIÓN

A continuación se describe el proceso puntual llevado a cabo en este proyecto y que puede servir de guía para futuros proyectos.

Estudio previo WS disponibles para los TM:

El grupo de investigación RadioGIS disponía de un WS a ser usado por los TM cuyas especificaciones es necesario analizar previamente:

- El WS está destinado para recibir valores de mediciones georreferenciadas.
- El WS se podría acceder a través de la URL <http://radiogis.uis.edu.co/sensores/medida>.
- El WS estaba configurado para recibir datos en formato JSON a través de una petición POST.
- El WS da a conocer al usuario el resultado del servicio mediante códigos HTTP de la siguiente manera: El número 200 significa que el servicio se prestó completamente en forma exitosa, mientras el número 400 significa que se presentó alguna falla. En este último caso, se incluye además en el cuerpo del mensaje una breve descripción de la causa del error.

Diseño del WS cliente:

A continuación se crea el consumidor de Web Service que será el encargado de enviar los datos a la nube. Para este diseño se tuvieron en cuenta las siguientes especificaciones:

- Usar Python como lenguaje de programación.
- Cumplir con las especificaciones que provienen del WS.
- Leer los datos a enviar de un archivo que es usado por el sistema de conversión análogo digital para almacenar los datos medidos.
- Georreferenciar los datos del punto anterior con coordenadas leídas de un archivo de texto en el que el GPS almacena la posición del terminal.
- Como una etapa previa el WS cliente se crea, ejecuta y se prueba en la computadora personal con un sistema operativo similar al que se encontrarán en la tarjeta donde el WS cliente se alojará.
- Una vez probado, el WS cliente se pasará definitivamente a la tarjeta, de manera embebida.

Descripción del WS Cliente:

Durante el consumo del Web Service los datos pasan por un proceso de tres etapas: Recopilación, adaptación y envío. En total se elaboraron dos scripts en Python, uno encargado de cubrir las dos primeras etapas y otro responsable por la última.

Recopilación y adaptación: Este script² se ubica en la carpeta que le indique el usuario, identifica los archivos de texto allí presentes y elabora un listado de los mismos. Utilizando como referencia dicho listado, abre uno a uno los archivos y extrae de ellos los datos. A continuación, toma y agrega la fecha del sistema y georreferenciar los datos con coordenadas leídas de otro archivo de texto (almacenadas allí por el GPS), los datos requeridos faltantes son definidos manualmente por el usuario en la configuración del script. Antes de cerrar el documento y proceder a su eliminación, llama al otro script para el envío de los datos al servidor.

Envío: Valiéndose de los módulos Request (para establecer la conexión y enviar las peticiones) y JSON (para formatear los datos), envía los datos al servidor. El script³ está compuesto por dos funciones: Una primera, que es la que hace propiamente el envío, a la cual deben proveer los datos formateados y una segunda que hace las veces de auxiliar ahorrándole al usuario el trabajo de dar formato a los datos.

Para la solución de narices electrónicas se utilizó el lenguaje de programación Python, el cual permite a través de librerías y funciones definir el proceso como un todo y los puertos de acceso a la tarjeta como variables del sistema; el lenguaje se utilizó como herramienta de desarrollo para este proyecto, permitiendo leer, editar o eliminar archivos, extraer datos, y organizar el protocolo de datos en formato JSON, estableciendo la comunicación y consumo del Web Service.

3.4. CONEXIÓN POR RED 3G

La portabilidad de un sistema es una necesidad primaria en los equipos, pues las conexiones virtuales y la transmisión de datos se han hecho un factor vital para el día a día. Para independizar el Terminal y hacer “Móvil” se necesita un canal de comunicación portable, por esto se selecciona una

² Disponible en <https://goo.gl/0DjdP9>

³ Disponible en <https://goo.gl/dQE3EB>

red 3G como una alternativa portátil de comunicación. Actualmente en el mercado existen “shield” que permiten la conectividad a través de estas redes, con el acceso a través de las SIMCARD que ofrecen las empresas de telefonía Móvil. El uso de estos shield puede definirse en 6 pasos, que permiten cumplir a cabalidad con la transferencia de datos a través de una red 3G.

Paso 1: Selección y Revisión de los Puertos de Entrada y Salida (Petición y Obtención de la información).

Paso 2: Selección del Proveedor de Servicios (Internet) y adquisición de una Tarjeta (SIMCARD) con su respectivo PIN, PUK, número de identificación en la red (Número del teléfono celular) y VPN (Virtual Private Network - red privada virtual).

Paso 3: Selección del protocolo de envío de paquetes a través de la red (FTP, SFTP, Socket).

Paso 4: Conocimiento previo de la dirección IP, hostname o dominio de envío de los paquetes.

Paso 5: Definición del tamaño del paquete de datos.

Paso 6: Evaluación de tiempos de envío y respuesta sobre el protocolo de selección.

Para el desarrollo del prototipo fue importante llevar a cabo cada uno de los pasos mencionados anteriormente esto con el fin de lograr la transmisión de los datos, si se llegara a saltar alguno de los primeros cuatro pasos la transmisión fallaría y no se cumpliría con los requerimientos planteados en el capítulo tres.

4. ESTUDIO DE DESEMPEÑO

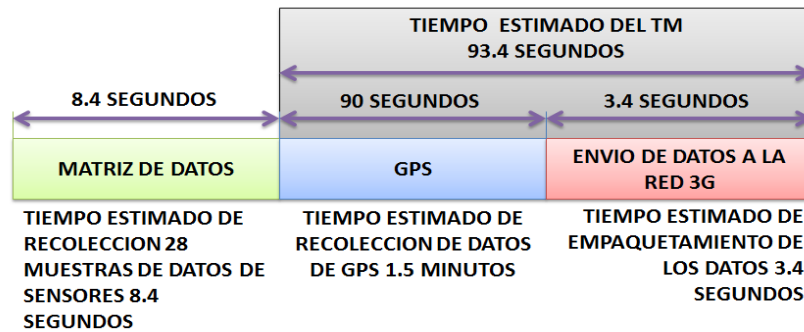
Al tener el prototipo funcionando correctamente se hace unas pruebas de desempeño con el fin de verificar el correcto funcionamiento del TM y del WS cliente desde la tarjeta desarrolladora, primero se calcula el tiempo estimado que se demora el sistema en recolectar y empaquetar los datos para transmitirlos, utilizando una matriz de información de 2.7Kb y la segunda prueba consiste en calcular el tiempo de empaquetamiento de los datos de diferente tamaño. A continuación se describe cada uno de ellos.

4.1. PRUEBAS DE DESEMPEÑO

La nariz electrónica está diseñada con 16 sensores de gases y un tiempo de muestreo de 300 ms, esto quiere decir que durante este intervalo se están generando 16 datos cada uno de un sensor diferente y a su vez se genera un archivo de texto plano el cual almacena cada dato separado por espacio, de esta forma se genera una matriz de 16 columnas por muestra. Se estableció que la cantidad de muestras que tomará la nariz electrónica es de 30, lo que corresponde al número de filas, dando un total de 480 datos generados por los sensores, demorando un tiempo de 8.4 segundos para generar la matriz de datos que serán enviados a la nube. Además a ello la toma de datos de GPS en modo Stand Alone demora 90 segundos en detectar la ubicación y en generar el archivo con las coordenadas del dispositivo.

El tiempo para empaquetar los datos de la matriz y realizar el envío toma 3.4 segundos. Todo lo anterior nos permite determinar tiempo mínimo estimado de 101.4 segundos para el envío de datos de sensores de narices electrónicas tal como se observa en la figura 7.

Figura 7. Tiempos estimados para el envío de datos por medio del terminal móvil de narices electrónicas.



La segunda prueba desarrollada consistió en crear y utilizar diferentes tamaños de paquetes de datos, de esta forma se pudo comprobar que el empaquetamiento y envío de datos tiene diferentes tiempos, dependiendo directamente del tamaño de cada una de ellas, en la tabla 3 se observa el tiempo que demora por tamaño de paquete.

Tabla 3. Tiempo de Empaquetamiento.

Peso de la Matriz	Tiempo de Empaquetamiento y Envío
1 Kb	3.37 segundos
10 Kb	3.51 segundos
100 Kb	4.82 segundos
1000 Kb	17.57 segundos

Con estas dos pruebas realizadas se pudo hallar el tiempo de empaquetamiento y envío de datos al WS, además se observa el comportamiento que presenta al aumentar el tamaño de la matriz de datos enviada.

4.2. COSTOS

El valor del terminal móvil para hacerlo desde cero se ve en la Tabla 4.

Tabla 4. Costo del TM desde cero.

Unidad	Valor(USD)
Intel Edison	\$95.99
3G/GPRS shield	\$265
Wi-Fi Móvil 3G M5350	\$91.46
Total	\$452.45

Para este terminal móvil el grupo de investigación RadioGIS contaba con el 3G/GPRS Shield y la tarjeta desarrolladora Intel Edison, por lo tanto dio un costo ideal de \$91.46 Se recomienda usar un GPS diferente al 3G/GPRS, existen otros mucho más económicos que no brindan tantas funcionalidades como el anterior mencionado, puesto que este además del GPS posee para hacer llamadas, enviar mensajes, y otras muchas funcionalidades de los celulares de alta gama de hoy en día.

5. CONCLUSIONES

Como fruto de este trabajo, se consiguió desarrollar uno de los elementos claves para el diseño de servicios IoT: El terminal móvil. Este aporte, junto con otros que lo complementan a nivel de captura de variables de sensado, procesamiento en la nube y aplicaciones para los usuarios, representa un importante impulso al desarrollo de servicios no solo en Colombia sino a nivel internacional.

La divulgación de este desarrollo promueve la creación de soluciones similares que aprovechan el potencial de las tarjetas de desarrollo por los usuarios. Así mismo, beneficia a la sociedad en general, demandando una mejor cobertura nacional a las empresas prestadoras de telefonía móvil.

Este avance hace posible la caracterización de una zona de forma remota, ya que por este medio se puede obtener la información que se desee. Así por ejemplo, en la industria se podrán monitorear procesos industriales que requieran supervisión continua 24 horas del día, sin necesidad de tener personal humano altamente capacitado.

Se logró aprovechar al máximo los recursos del grupo RadioGIS referente a la asociación con Intel al hacer uso de la tarjeta desarrolladora, siendo una herramienta valiosa para adquirir conocimiento. El desarrollo en esta tarjeta permitió ser los pioneros a nivel del grupo de investigación de la universidad, en comenzar a utilizar Intel Galileo como un procesador en pro del internet de las cosas.

Las cualidades de las tarjetas motiva al desarrollo de proyectos más complejos que aprovechen al máximo las capacidades de la tarjeta, la puesta en marcha de otros prototipos para mejorar las características de las mismas y el desarrollo de mejores versiones que cumplan y faciliten el trabajo de los usuarios. Se abre paso para trabajos futuros donde se pueda llegar a mejorar el proyecto y a solucionar problemas más complejos, que presten un servicio a la comunidad, a la industria, etc.

El terminal móvil desarrollado es un prototipo que puede transferir los datos que se deseen sin importar de dónde provengan, convirtiéndolo en un sistema versátil que facilita la transmisión de datos de cualquier sistema que los adquiera y los necesite para transmitir de forma inmediata. La sencillez y modularidad de la tarjeta desarrolladora Intel Edison son ventajas a la hora de la implementación de dicha terminal, puesto que hacen que sus dimensiones sean aún más pequeñas con respecto a una similar construida sobre la Intel Galileo

Al usar tarjetas de desarrollo como la Intel Edison, los medios de transmisión se vuelven modulares, dándole mayor importancia al sistema embebido de la tarjeta desarrolladora por encima de la transmisión. El desarrollo realizado, aunque se enfocó a narices electrónicas, resulta útil para conectar cualquier tipo de sensor gracias a la visión puesta en el desarrollo.

Es importante hacer pruebas del correcto funcionamiento de cada uno de los elementos que conforman la solución del TM, esto se da desde comprobar los script de Python que se encargan de consumir el WS, hasta los dispositivos electrónicos adquiridos para desarrollar el TM, antes de llevar a cabo todo el proceso sobre la tarjeta desarrolladora, de esta forma se tiene la certeza que todo funcionará correctamente.

Por último, este proyecto permite tomar conciencia de la importancia del ingeniero electrónico en manejar conceptos claves de programación: Web Services, programación de componentes electrónicos con Python, Arduino, entre otro

CITAS

- [1] RODRÍGUEZ, Cesar Camilo. “Servicio prototipo de telecomunicaciones basado en localización para monitorear en línea el espectro radioeléctrico”, Trabajo de grado Maestría en Ingeniería Electrónica. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Maestría en Ingeniería Electrónica, 2013.
- [2] TEDESCHI, Nicolás. Web Services, un ejemplo práctico. Microsoft [en línea] [Redmond, Washington, E.U.]: [Consulta: 27 de julio de 2015] Disponible en: <https://msdn.microsoft.com/es-es/library/bb972248.aspx>

BIBLIOGRAFIA

INTEL® DEVELOPER ZONE. Connecting your Intel Edison board using Wi-Fi*. Intel [online] [Santa Clara, California E.U.]: Intel Developer Zone [cited 30 april 2015]. Available from: <https://software.intel.com/en-us/connecting-your-intel-edison-board-using-wifi>.

INTEL® GALILEO. Ficha Técnica Galileo. Intel [en línea] [Santa Clara, California E.U.]: Intel® Galileo Septiembre 2014 [consultado 10 de enero 2015]. Disponible en: https://engage.intel.com/servlet/JiveServlet/previewBody/50545-102-1-62396/Es_Intel_Galileo_Ficha_Tecnica.pdf

INTEL® QUARK™. Intel® Quark™ SoC X1000 BSP Build and Software User Guide Intel [online] [Santa Clara, California E.U.]: Intel® Quark™ 16 June 2014. [cited 22 february 2015]. Available from: https://downloadmirror.intel.com/23962/eng/Quark_BSP_BuildandSWUserGuide_329687_007.pdf.

----- Intel® Quark™ SoC X1000 Software Developer's Manual for Linux* Intel [online] [Santa Clara, California E.U.]: Intel® Quark™ February 2015 [cited 22 february 2015]. Available from: <http://www.intel.com/content/www/us/en/embedded/products/quark/quark-x1000-linux-sw-developers-manual.html>.

LIBELIUM COMUNICACIONES DISTRIBUIDAS S.L. 3G + GPS shield over Arduino and Raspberry Pi: Difficulty Level: Expert. [online] Cooking-Hacks. [Zaragoza, España] Libelium: Cooking-Hacks. [Cited: 3 Agosto 2015] Available from: <https://www.cooking-hacks.com/documentation/tutorials/3g-gps-shield-arduino-raspberry-pi-tutorial/>.

LÓPEZ, Paula Andrea, RODRIGUEZ, Yojanes Andrés y RUIZ, Luisa Fernanda. Caracterización de un módulo de reconocimiento de patrones para un servicio TIC basado en narices electrónicas. Trabajo de grado Ingeniería Electrónica. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, 2015.

RODRÍGUEZ, Cesar Camilo. Servicio prototipo de telecomunicaciones basado en localización para monitorear en línea el espectro radioeléctrico, Trabajo de grado Maestría en Ingeniería Electrónica. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Maestría en Ingeniería Electrónica, 2013.

TEDESCHI, Nicolás. Web Services, un ejemplo práctico. Microsoft [en línea] [Redmond, Washington, E.U.]: [Consulta: 27 de julio de 2015] Disponible en: <https://msdn.microsoft.com/es-es/library/bb972248.aspx>