

# “Sistema Prototipo de Monitoreo Remoto para Pacientes”

## Trabajo Terminal No. 2016 -B013

Presentan: Alejandra Melina Cuadra Camacho, Gabriela Saldaña Aguilar

Directores: Dr. Rubén Ortega González, Dr. Miguel Santiago Suárez Castañón

Escuela Superior de Cómputo I.P.N. México D.F. Tel. 57-29-6000 ext. 52000 y 52021. e-mail:

[ale\\_cmae@hotmail.com](mailto:ale_cmae@hotmail.com)

**Resumen-** El trabajo terminal presentado consiste en el desarrollo de un sistema de cómputo que permite el monitoreo remoto de los signos vitales ritmo cardíaco y temperatura corporal basado en: tecnologías para el desarrollo de aplicaciones web, el servicio de máquinas virtuales Google Cloud Platform para el alojamiento de la aplicación, la tarjeta NodeMCU para aplicaciones de telemetría, el sensor de ritmo cardíaco PulseSensor y el sensor digital de temperatura ds18b20, se logró el despliegue de la aplicación.

**Palabras clave-** Internet de las cosas, Telemetría, e-Health, Aplicación REST, La Nube, Plataforma como servicio.

## I. INTRODUCCIÓN

Los sistemas Web han mostrado un avance precipitado en los últimos 10 años, este cambio se debe principalmente a los siguientes factores:

1. Evolución del Uso/ Demanda
2. Evolución de las tecnologías de Navegación
3. Evolución de las tecnologías de Servidores
4. Marketing y Ventas
5. Evolución de las Tecnologías de Información

El número de usuarios que generan o descargan contenido en internet ha crecido rápidamente como se puede observar en la Fig. 1, por lo que las tecnologías web tuvieron que evolucionar a una arquitectura más robusta dejando a tras la arquitectura vertical [1] donde se agregaba más capacidad

a un mismo servidor y se generaban cuellos de botella como consecuencia.

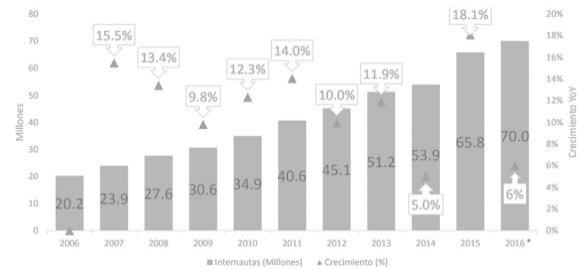


Fig. 1. Incremento de usuarios de internet en los años 1990 a 2007.

Al ver que un escalamiento vertical de recursos no era mantenible a largo plazo, comenzaron a migrar a arquitecturas horizontales [1]. En vez de agregar capacidad a un solo servidor hasta el punto de saturarlo; se optó por la distribución de trabajo en un número mayor de servidores independientes que no necesariamente necesitan tener correlación entre sí. Así es como surgió la necesidad de diseñar programas adaptables a estas nuevas arquitecturas distribuidas, a este conjunto de programas se les llamó aplicaciones REST por sus siglas en inglés Representational State Transfer.

## REST

Las aplicaciones de Transferencia de Estado Representacional son un estilo de arquitectura de software para sistemas hipertexto. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding [3].

Los principios básicos para una arquitectura REST son:

\*Todo lo que se mueve a través de las comunicaciones de internet es un recurso, y un recurso es representado por el tipo de formato que tiene, es decir se describe por su contenido.

\*Cada recurso debe ser accesible e identificable mediante una única URI.

\*El protocolo de transmisión de datos debe de usar verbos o acciones nativas de HTTP.

\*Los recursos pueden tener distintas representaciones o formatos.

\*Las comunicaciones deben ser sin estado, no se debe de mantener ningún tipo de persistencia en las transacciones, el servidor debe enviar el recurso completo y la solicitud debe contener todos los datos necesarios para obtenerlo. Una vez regresado el recurso el servidor olvidara quien era el cliente.

REST se ha hecho popular por el incremento en servicios web. RESTful son los servicios web que sigues los principios REST.

Ventajas:

\*Separación del recurso y la representación:

\*Visibilidad

\*Seguridad

\*Escalabilidad

\*Rendimiento

Las aplicaciones REST tienen este impacto hoy en día gracias a la popularización de la Nube, es decir, se aprovechan del hecho que los recursos computacionales se encuentran en una arquitectura descentralizada, puesto que muchas de las tareas que se solían hacer en un mismo ordenador ahora han sido separadas y alojadas en servidores remotos a los cuales uno se conecta para consumir un servicio [2].

Basándose en el protocolo HTTP, así como en sus operaciones básicas (GET, PUT, POST, DELETE) y en el almacenamiento y computación de datos en la Nube, los

servicios web o SAS por sus siglas en inglés Software As a Service, han tenido un papel fundamental hoy en día en el tipo de aplicaciones que desarrollan empresas que tienen un gran poder de control de masas como lo son Facebook, Amazon, Microsoft etc.

## **Internet de las Cosas**

Las aplicaciones REST abren paso a un mundo más conectado, donde el número de objetos conectados a internet supera al número de personas conectadas a internet, a este punto en el tiempo se le conoce como Internet de las cosas (en inglés Internet of Things).

El concepto de Internet de las cosas fue propuesto por Kevin Ashton en el Auto-ID Center del MIT en 1999 [3]. Esta vasta red de objetos conectados a Internet es un campo de estudio muy completo, sus infinitas aplicaciones hacen que sea presentes en áreas como lo son: edificios inteligentes, salud, automatización, monitoreo energético, entre otros.

Este trabajo se desarrolla dentro del área de medicina, en específico e-Health.

## **II. METODOLOGÍA**

El sistema prototipo de monitoreo remoto para pacientes está constituido por tres módulos principalmente; adquisición de mediciones, envío de mediciones y procesamiento e interpretación de mediciones. El primer módulo consta de 2 pasos que son la lectura y el procesamiento, el segundo módulo consta de la conexión y envío de datos a un servidor remoto y el tercer módulo consta de 3 pasos que son lectura de datos, procesamiento y despliegue.

A continuación, se describe a detalle cada uno de estos módulos.

### A. Adquisición de mediciones

Lectura: Es la parte inicial del lado del cliente, por medio de la tarjeta de adquisición de datos NodeMCU se obtienen las lecturas de las mediciones entregadas por el sensor de temperatura ds18b20 y el de ritmo cardiaco PulseSensor; gracias al ADC que viene integrado a la entrada analógica de la tarjeta es posible leer el sensor de pulso sin necesidad de utilizar un ADC externo.

La NodeMCU nos permite comunicarnos con el sensor digital de temperatura mediante el protocolo de comunicación One-Wire. De esta forma se obtienen los valores digitalizados de ambos sensores.

Procesamiento: Una vez realizada la lectura en los pines correspondientes de la tarjeta, se procede a calcular las pulsaciones por minuto (BPM) usando el algoritmo descrito en su página oficial.

El algoritmo encuentra el tiempo entre pulsaciones (IBI) no consideradas como ruido, las últimas 10 mediciones de IBI son almacenadas y promediadas, este valor promedio se toma como el valor de una pulsación, de tal manera que si uno quiere contar el número de pulsaciones que hubo en un periodo de tiempo igual a un minuto.

La fórmula es como se puede observar en la Fig. 2:

```
word runningTotal = 0;

for(int i=0; i<=8; i++){
    rate[i] = rate[i+1];
    runningTotal += rate[i];
}

rate[9] = IBI;
runningTotal += rate[9];
runningTotal /= 10;
BPM = 60000/runningTotal;
```

Fig 2. Cálculo del BPM.

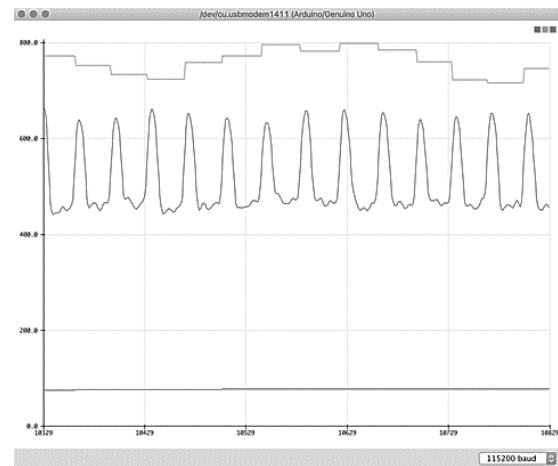


Fig. 3 Gráfica del IBI, BPM y el valor de la señal emitida por el sensor de ritmo cardiaco.

### B. Envío de mediciones

Conectividad: Cuando se hayan obtenido ambas lecturas de las mediciones, se ejecuta un programa encargado de realizar la conexión con el servidor vía protocolo HTTP. En esta etapa uno puede comprobar los distintos modos de funcionamiento de la NodeMCU, usando los comandos AT que trae por defecto podemos obtener la IP que se le asigna mediante DHCP, la dirección física que le corresponde y además se puede

configurar una red específica para poder conectarse a ella.

TABLA I  
 COMANDO AT PARA ESTABLECER CONEXIÓN A  
 UNA RED DE INTERNET

| COMANDO  |
|--|
| "AT+CWMODE=1" se usa como modo estación                                |
| "AT+CWLAP" chequea las redes Wifi disponibles                          |
| AT+CWLAP_CUR="ARRIS-1892","AA3ECEF9977CCAHY" se conecta a una red Wifi |
| AT+CIFSR para obtener la IP local asociada al módulo                   |

Envío: Se formula una petición HTTP que es enviada al servidor web con parámetros que siguen la siguiente estructura (1):

<http://35.193.190.76/measures/measure/18:FE:34:D6:2F:58/34.6/80> (1)

Primero se coloca la dirección de nuestro servidor en la nube, seguido de la dirección MAC y finalmente los valores de las mediciones.

### C. Procesamiento de Mediciones

Obtención: El servidor en la nube es una máquina virtual de la empresa Google la cual tiene por sistema operativo Ubuntu 16.04, la aplicación REST fue desplegada sobre un servidor Apache usando Laravel como framework de desarrollo Web MVC.

Procesamiento: El servidor está siempre escuchando por peticiones HTTP, y estas peticiones están ligadas a alguna de las operaciones básicas de HTTP (GET, PUT,

POST, DELETE) que ejecutan una acción en específico. En este caso todas las mediciones son recibidas mediante peticiones tipo POST, las cuales son tratadas de la siguiente manera: Se recibe una petición POST, se extrae el valor de las variables y se almacena el valor en una bitácora en la base de datos.

Despliegue: Se revisa si estas mediciones están dentro de los rangos normales de funcionamiento y se generan las notificaciones necesarias. Finalmente, los datos son graficados dinámicamente usando el servicio web ThingSpeak [6].

## III. RESULTADOS

Antes de la elección del NodeMCU, se hicieron pruebas con las tarjetas de adquisición Raspberry y Arduino, debido a lo portable y a la gran capacidad con la que cuenta la NodeMCU se determinó que era la más adecuada para la aplicación.

TABLA II  
 COMPARACIÓN DE TARJETAS DE ADQUISICIÓN  
 DE DATOS

|                             | RASPBERRY Y PI3  | ARDUINO UNO                           | NODEMCU                                 |
|-----------------------------|--|---------------------------------------|---|
| Potencia de cálculo         | Cuenta con 4 núcleos de procesamiento.                         | Cuenta con 1 núcleo de procesamiento. | Cuenta con 1 núcleo de procesamiento.   |
| Capacidad de almacenamiento | Cuenta con 1GB.  | Cuenta con 32KB.                      | Cuenta con 4MB.                         |
| Velocidad                   | Corre a 900 MHZ.   | Corre a 16 MHZ.                       | Corre a 160 MHZ.                        |
| Interfaces de comunicación  | SPI, I2C, GPIO, UART, MINI USB, ETHERNET, USB, HDMI, MICRO SD. | GPIO, UART, USB, I2C, SPI, 1-Wire.    | Mini USB, UART, GPIO, SPI, I2C, 1-Wire. |

|                      |  |   |  |
|----------------------|--|---|--|
| ADC                  | 8 canales de lectura a 10 bits.  | 6 canales de lectura a 10 bits.                               | 1 canal de lectura a 10 bits.                            |
| Voltaje de operación | 5v   | 5v  | 3v   |
| Consumo de corriente | 230 mA   | 46 mA   | 80 mA  |
| Adecuado para IOT    | Posee módulo de Wifi, ocupa demasiado espacio y consume mucha corriente. | No tiene modulo Wifi, por lo que hay que agregar uno externo. | Consumo bajo de corriente. Cuenta con un módulo de Wifi. |

Se realizó la monitorización de 8 distintas personas adultas, de edades 22, 30, 40, y 70 años. Después de efectuarse la medición con los sensores y al compararlos contra las mediciones de instrumentos médicos, se determinó que el sistema tiene una eficiencia del 80%. Cabe mencionar que el 30% de las personas a las que se les efectuaron mediciones, no se colocaban los sensores de la forma indicada o tuvieron una conexión a internet lenta. El sistema pudo ser accedido desde diferentes navegadores web por distintos usuarios y no mostró problema alguno. Internamente se generó un tráfico de 100 pacientes y 50 doctores dados de alta en la Base de datos, los cuáles fueron soportados por la aplicación. El mayor índice de error que presentó el sistema fue con los sensores, por una parte, el de pulso cardiaco puesto que es extremadamente sensible, cualquier pequeño movimiento podía ocasionar una medición errónea y por otro parte el sensor de temperatura tiene un tiempo de medición de aproximadamente 5 minutos, bastante lento a comparación del sensor de ritmo cardiaco.

#### IV. CONCLUSIONES

Hoy en día las tecnologías de la web nos permiten estar cada vez más conectados, Internet de las Cosas abre paso a aplicaciones remotas de medicina como es el caso de este Trabajo Terminal, en México aún falta más arquitectura para poder soportar la siguiente generación de redes donde se involucrará IOT, pero es viable realizar pequeñas aplicaciones donde conectemos un número de objetos a internet.

Con este proyecto terminal se dio solución al problema de monitorear pacientes de la tercera edad de forma remota.

El sistema se desarrolló utilizando las tecnologías más nuevas para el desarrollo web, así como, siguiendo las mejores prácticas para una aplicación portable y escalable. Además de que el campo de aplicación de este proyecto es muy amplio, como lo son la telemedicina, el monitoreo de rendimiento para atletas, intervenciones a distancia, etc.

Se desarrolló un sistema de cómputo con la capacidad de monitorear el ritmo cardiaco y la temperatura corporal a una eficiencia del 80%, el cual puede ser aplicado para monitorear personas de cualquier edad y con la posibilidad de agregar más variables para monitorear todos los signos vitales.

#### V. RECONOCIMIENTOS

*Los Autores agradecen a la Escuela Superior de Cómputo del Instituto Politécnico Nacional por el apoyo recibido y las facilidades otorgadas para el desarrollo del presente trabajo terminal.*

#### VI. REFERENCIAS

[1] Richardson, C. (2017). Microservices.  
Obtenido de Microservice Architecture:  
<http://microservices.io/patterns/microservices.html>

[2] Berners-Lee, T. (mayo de 1996). Hypertext  
Transfer Protocol. Obtenido de:  
<http://1.0>: <https://tools.ietf.org/html/rfc1945>

[3] Lopez Research. (Noviembre de 2013). Part  
1 of "The IoT Series". Obtenido de:  
[http://www.cisco.com/c/dam/en\\_us/solutions/trends/iot/introduction\\_to\\_IoT\\_november.pdf](http://www.cisco.com/c/dam/en_us/solutions/trends/iot/introduction_to_IoT_november.pdf)