

# Prototipo de Parqueadero Inteligente Mediante Red de Sensores Inalámbricos

Marcelo D. Márquez

Departamento de  
Eléctrica y Electrónica

Universidad de las Fuerzas Armadas  
ESPE

Quito-Ecuador

Email: mdmarquez@espe.edu.ec

Román A. Lara

Departamento de  
Eléctrica y Electrónica

Universidad de las Fuerzas Armadas  
ESPE

Quito-Ecuador

Email: ralara@espe.edu.ec

Rodolfo X. Gordillo

Departamento de  
Eléctrica y Electrónica

Universidad de las Fuerzas Armadas  
ESPE

Quito-Ecuador

Email: rxgordillo@espe.edu.ec

**Abstract**—El desarrollo e implementación de proyectos tecnológicos en el ámbito de la movilidad vehicular pueden mejorar la calidad de vida de los ciudadanos en las grandes ciudades. Si bien en los últimos años se ha buscado solucionar el problema de la falta de plazas de parqueo la ciudadanía no ha sentido un cambio significativo, el desarrollo de tecnologías que ayuden al usuario a encontrar plazas de parqueo libres pueden ayudar a reducir el tiempo perdido en hallar una plaza de parqueo, así como también reducir la contaminación y mejorar el tráfico. En el presente proyecto se realiza el desarrollo de un prototipo de parqueadero inteligente, el cual se desarrolla en tres plazas de parqueo. La información es obtenida por una red de sensores y enviada a un programa en Java que presenta el uso de las plazas de parqueo propuestas en el prototipo.

**Keywords:** WSN (Wireless Sensor Network), Sensor, ZigBee, IEEE 802.15.4, Xbee, Waspote, Java

## I. INTRODUCCIÓN

Es muy difícil creer hoy en día que hace algunos años atrás no había automóviles en las calles, en los últimos 50 años el paisaje humano se ha visto invadido por vehículos ya que estos se han vuelto en elementos imprescindibles de la vida moderna, un problema grave que conlleva el aumento de vehículos es el tráfico y mucho más la falta de gestión en el tema de parqueaderos.

Es por eso que se requiere de soluciones para el problema de la movilidad de los automóviles, además se debe tomar en cuenta otro problema que es la estancia y permanencia en el espacio público de los mismos. Una solución al problema de las plazas de parqueo en las calles podría ser hacer más parqueaderos o se puede buscar un sistema para gestionar las plazas de modo que hallar un parqueadero no sea una pérdida de tiempo y combustible [1].

Debido a las posibilidades que brinda el vehículo privado tanto lo que se refiere a la versatilidad para realizar desplazamientos y el confort de los usuarios, este se ha convertido en el medio de transporte preferido por muchos usuarios. Analizando el caso de la ciudad de Quito, en la ciudad se estima que el parque automotor alcanza los 400.000 vehículos; esto se debe a problemas muy marcados como las debilidades del transporte público y el bajo costo de los combustibles.

En los últimos años el crecimiento del parque automotor ha sido de forma desproporcionada complicando directamente la escasez de las plazas de parqueo en la ciudad. En la Figura 1 podemos observar una ubicación de la ciudad de Quito con sus puntos de referencia cercanos.

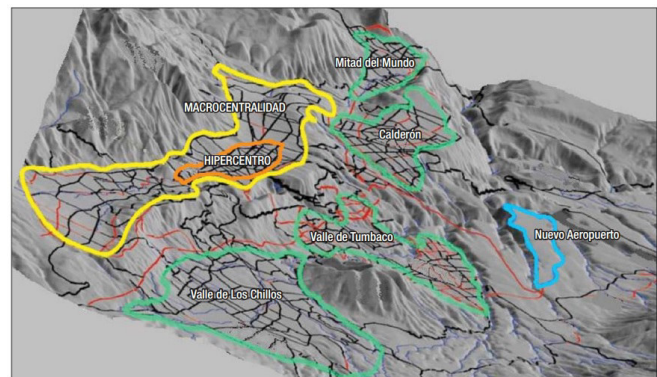


Fig. 1. Centralización de servicios en Quito

El número de viajes en transporte privado crece a razón de 10,8% anual por lo tanto en 16 años se considera que el 60% de los viajes motorizados será de este tipo; los viajes se concentran principalmente en el hipercentro de Quito donde se producen 700.000 viajes diarios; de ellos el 64% son de fuera del hipercentro y el 36% son internos como se muestra en la Figura 2.

El parque automotor en Quito se incrementa anualmente en 30.000 vehículos, de mantenerse así la tendencia para el año 2025 sería del triple de vehículos lo que llevaría a que sean en el futuro 1'290.000 vehículos lo que aumenta el número de viajes como se muestra en la Figura 3, además de una alta deficiencia de plazas de parqueo, lo que conlleva a que parquear un vehículo en la ciudad se vuelva imposible.

En el hipercentro de la ciudad circulan aproximadamente 800.000 vehículos a diario, lo que significa que en horas pico alrededor de 35.000 vehículos demandan de plazas de parqueo, no siempre disponibles en esa proporción sobre todo en las zonas de mayor actividad, lo que conlleva a ineficiencias de

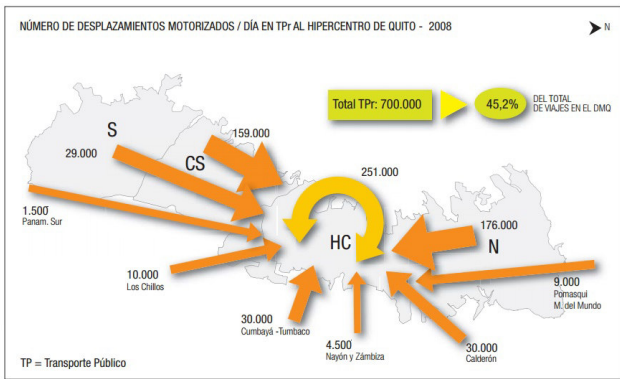


Fig. 2. Número de desplazamientos motorizados

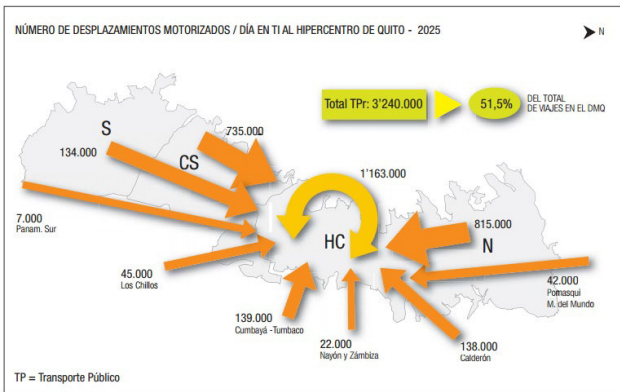


Fig. 3. Tendencia de viajes para el año 2025

la red vial, contaminación y mal uso del espacio público.

La deficiencia de parqueaderos produce conflictos en el tráfico y afectaciones ambientales, puesto que los conductores circulan de manera excesiva alrededor de su destino buscando plazas de estacionamiento lo que multiplica los efectos de ruido y emisión de gases contaminantes [2].

## II. FUNDAMENTO TEÓRICO

### A. Origen de las redes de sensores

Las redes de sensores tienen su origen militar, por este motivo no existe mucha información sobre su origen. En la Guerra Fría se colocaban en el mar boyas a modo de una red, cada boya contaba con un sensor ultrasónico que podía detectar la presencia de submarinos o barcos, este sistema se conocía como SOSUS (Sound Surveillance System) [3] [4].

Las redes de sensores inalámbricos como los conocemos ahora nacieron como una propuesta del Dr. Pister que era profesor de la Universidad de Berkeley, para esa época los objetivos de desarrollo de las redes de sensores eran reducir el alto consumo de energía, el tamaño de los equipos y el alto costo de mantenimiento; lamentablemente para la época muchas cosas no se pudieron cumplir debido a limitaciones propias de la época pero fue el inicio para lo que conocemos ahora [5].

### B. Protocolo de red de tiempo sincronizado

Este protocolo asume que los nodos en una red están sincronizados con una diferencia de 1 ms entre cada uno; suponiendo que un nodo A quiere transmitir datos a un nodo B, el nodo B va a despertar de un modo de espera, en el tiempo que el nodo B despierta y espera los datos ya habrán transcurrido 2 ms, si dentro de este tiempo el nodo A no transmite datos ya sea porque no tenía nada que enviar o no llegaron los datos por problemas en el canal el nodo B vuelve a quedarse en espera, por otro lado si el nodo A si envía información, se envía un ACK de confirmación de recepción y los nodos vuelven a entrar en modo de espera [6].

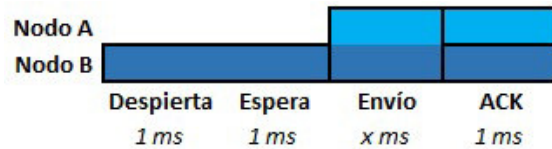


Fig. 4. Envío y recepción de datos

La gran ventaja de este protocolo es que cumple con el objetivo del ahorro energético ya que mientras los nodos están dormidos la energía de consumo es muy baja además este protocolo puede cambiar el canal de operación si es que se tiene interferencias en el mismo [7].

Este protocolo también puede hacer uso del multisalto, que consiste en que un paquete no se envíe directamente a su destino sino que haga uso de otros nodos intermedios para llegar a este, es decir que pueden existir nodos que toman información que no les pertenece pero la reenvían a modo de repetidoras agregando su propia información; el gran problema de este modo es que dependiendo del número de nodos y la cantidad de saltos la información puede ser muy grande y retardada hasta llegar a su destino.

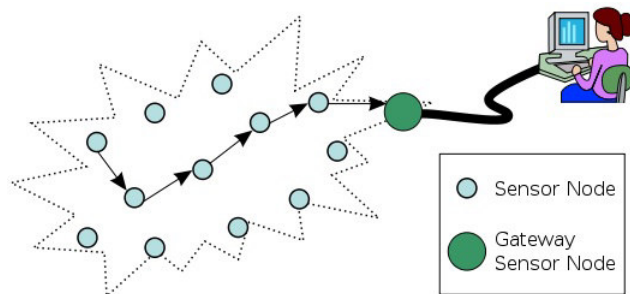


Fig. 5. Multisalto

### C. Estándar IEEE 802.15.4

Este estándar define una capa de comunicación en el modelo OSI, el propósito del estándar es definir los niveles de red básicos para dar servicio a un tipo específico de red inalámbrica de área personal centrada en la comunicación entre dos dispositivos con bajo costo de energía y velocidad. Por lo general estas redes son sin infraestructura.



Fig. 6. Pila de protocolos IEEE 802.15.4

1) *Capa física*: Provee la transmisión de datos sobre el medio físico propiamente dicho. Esta capa controla el transceptor de radio frecuencia y realiza la selección de canales junto con el control de consumo y de la señal.

TABLE I  
BANDAS DE FRECUENCIAS PARA EL ESTÁNDAR

Frecuencia (MHz)	Uso	Rev. 2003	Rev. 2006
868 - 868.8	Europa	1	3
902 - 928	US	16	30
2400 - 2483.5	-	16	16

TABLE II  
TASA DE DATOS

Banda (MHz)	Codificación	Tasa Datos (kbps)
868	DSSS	20
915	DSSS	40
2450	DSSS	250

2) *Control de acceso al medio*: Ofrece servicio de datos, interfaz de control y regula el acceso al canal físico además del envío de la trama beacon; por otro lado controla la validación de las tramas, asociaciones entre los nodos y garantiza la existencia de los slots de tiempo [8].

#### D. ZigBee

Es un estándar propuesto por la ZigBee Alliance que se utiliza en la transmisión de datos digitales, esta basado en el estándar IEEE 802.15.4, los objetivos principales de este es tener redes de baja tasa de envío de datos y maximización de la vida útil de las baterías.

ZigBee utiliza las bandas de frecuencia no licenciadas es decir 868 MHz en Europa, 915 MHz en Norte América y 2,4 GHz en todo el mundo, los fabricantes por lo general producen equipos que trabajan en la banda de 2,4 GHz debido que esta es no licenciada en todo el mundo.

1) *Tipos de dispositivos*: Existen varios tipos de dispositivos dentro de una red de sensores dependiendo de las funciones que cumplen:

- **Coordinador ZigBee (ZC)**: Es el dispositivo principal dentro de la red por lo que siempre debe existir mínimo uno dentro de ella, se encarga de controlar la red y el enrutamiento de los datos de los dispositivos conectados a este.
- **Router ZigBee (ZR)**: Puede interconectar dispositivos separados en la topología de la red.
- **Dispositivo Final (ZED)**: Posee la capacidad de comunicarse con un coordinador o un router pero no puede transmitir información hacia otros dispositivos terminales, por eso este dispositivo puede estar dormido la mayoría del tiempo aumentando la duración de las baterías; este dispositivo es mucho más sencillo que el coordinador y el router por lo que es mucho más barato.

2) *Topologías de red*: Existen varios tipos de topologías de red posibles:

- **Estrella**: El coordinador se sitúa en el centro de la red.
- **Árbol**: El coordinador es la raíz del árbol dentro de la red.
- **Malla**: Donde al menos uno de los nodos tiene más de dos conexiones en la red.

El uso de la red en malla (mesh) es el éxito de ZigBee y este también es uno de los principales motivos por lo que se ha popularizado su uso; ésta topología permite que los nodos puedan enviar información que no les pertenece si uno de los nodos cercanos falla esto se hace enrutando la información por otro camino y de este modo no se pierde.

Para que exista una topología en malla debe haber nodos que actúen como router y el nodo coordinador se encarga del enrutamiento [9].

### III. PROTOTIPO DE PARQUEADERO INTELIGENTE

Un parqueadero inteligente está definido por la capacidad del sistema de estacionamientos para darnos información importante sobre el estado de las plazas de parqueo, en este caso lo que se busca es que el usuario sepa que plazas de parqueo están desocupadas dentro de una zona específica de modo que pueda dirigirse de inmediato a un puesto vacío y no perder tiempo ni consumir combustible de forma innecesaria [10] [11].

#### A. Sensor ultrasónico

Los sensores ultrasónicos son detectores de proximidad, el sensor emite un sonido de una frecuencia determinada y mide el tiempo que demora el sonido en regresar al origen, dependiendo del tiempo de rebote del sonido se puede determinar la distancia a la que se encuentra el objeto, el principio de este sensor es el sonar.

Estos sensores solo trabajan en el aire y pueden detectar muchos tipos de materiales siempre y cuando sean deflectores de sonido. El inconveniente de estos sensores son las zonas ciegas ya que los objetos para ser detectados deben estar siempre dentro del rango del sonido generado por este [12].



1) *Sensor LV-MaxSonar-EZ0*: El sensor utilizado para el prototipo es de la marca MaxBotix modelo MaxSonar – EZ0 que tiene como principales características:

- Alimentación: 2.5V - 5.5V
- Consumo: 2mA
- Operación: 42kHz
- Rango de detección: 0m - 6.45m

Se utilizó este tipo de sensores debido a su bajo consumo de energía, bajo costo, facilidad de uso y precisión al detectar objetos sin dar falsas alarmas. En la Figura 7 se puede observar una fotografía del aspecto físico del sensor [13].



Fig. 7. Aspecto físico

Se utilizó tres pines del sensor, el primer pin es el de GND para la conexión a tierra de la fuente, el segundo pin es el de +5V de alimentación (como se indicó en las características puede variar entre un voltaje mínimo y un máximo) y el tercer pin es el de AN que nos da el valor de voltaje analógico dependiendo de la distancia a la que se encuentra el objeto detectado. En la Figura 8 se puede observar los pines con los que cuenta el sensor.

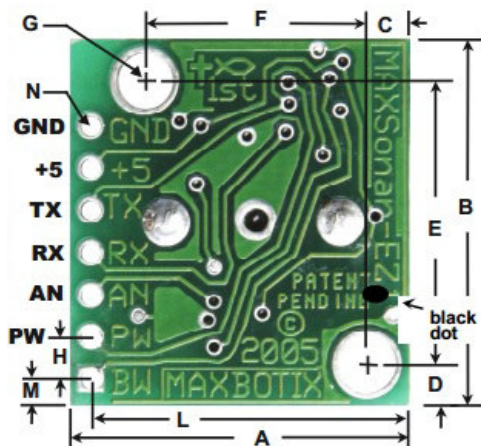


Fig. 8. Distribución de pines

En la Figura 9 se puede observar cuatro gráficas aproximadas de los lóbulos de detección que se generan. Para 0.0063 m se genera el lóbulo (A), para 0.025 m se genera

el lóbulo (B), para 0.082 m se genera el lóbulo (C) y para 0.27 m se genera el lóbulo (D). Sabiendo que el lóbulo (D) es muy teórico ya que al haber distancias tan grandes entre el objeto a detectar y el sensor muchos objetos pueden generar interferencias que generarían falsas alertas [14] [15].

Para el caso de las zonas de parqueo las distancias que el sensor debe medir son pequeñas ya que el objetivo es detectar la distancia entre el suelo y la carrocería inferior del vehículo.

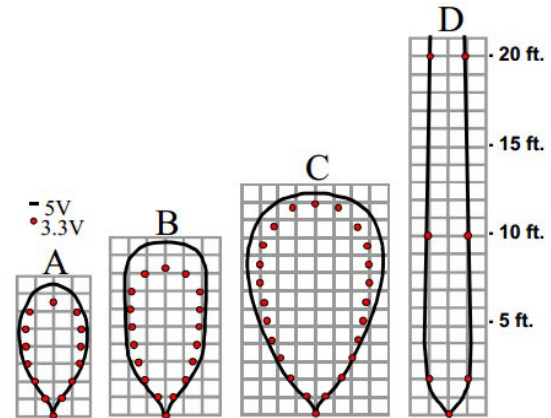


Fig. 9. Lóbulos teóricos que generan el sensor

### B. Módulos Wasmote

Estos módulos son fabricados por la empresa española Libellium, consisten en pequeños dispositivos modulares capaces de medir cualquier tipo de variable física y enviar estos datos mediante diferentes protocolos de comunicaciones.

La idea de estos sensores es llevar el mundo real al internet, para saber que sucede y en donde sucede. Sus creadores se basaron para el desarrollo de los sensores en la comunicación y apoyo que muestran los insectos al trabajar, es así como estos sensores pueden comunicarse entre ellos y apoyarse entre ellos para enviar la información a su destino.

Utilizando un sensor Geiger implementado en los módulos Wasmote se pudo crear un mapa de las zonas afectadas por la radiación después del desastre en la central nuclear de Fukushima en Japón.

Los módulos tienen un chip central de procesamiento que es el cerebro del módulo, dependiendo del tipo de sensor que se le coloque en el puerto de sensores este puede procesar dicha información y enviarla mediante la interfaz inalámbrica que se haya colocado. En la Figura 10 se puede observar las diferentes interfaces del sensor. Entre los módulos de comunicaciones que se puede utilizar tenemos WiFi, GPRS o ZigBee. El objetivo es visualizar y monitorizar la información en la nube; las principales aplicaciones en que se usan estos sensores son Ciudades Inteligentes, Agricultura Inteligente y Seguridad [16] [17].

Algunas de las principales características técnicas de los módulos Wasmote son:

- Microcontrolador: ATmega1281

- Frecuencia: 14MHz
- Memoria flash: 128Kb

Los niveles de consumo de batería del módulo es muy bajo a pesar de ello se puede conectar un panel solar al módulo para recargar la batería. Sus niveles de consumo energético son:

- Encendido: 15mA
- Dormido: 55uA

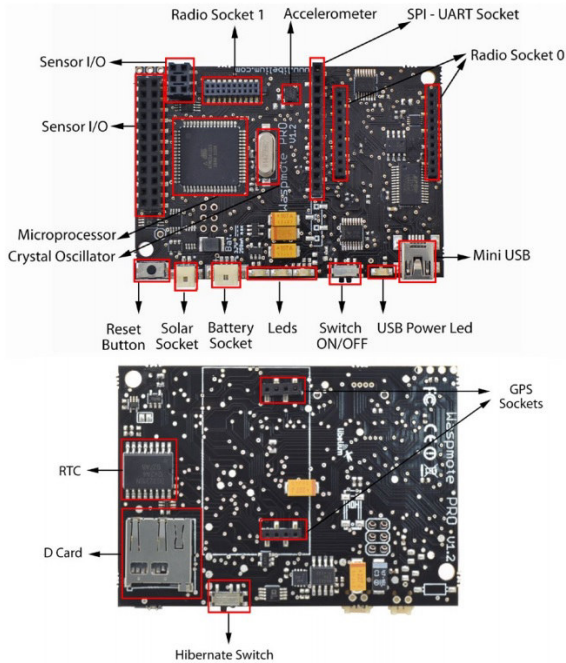


Fig. 10. Partes del Waspote

Los módulos cuentan con muchos puertos para poder desarrollar diferentes aplicaciones y usos, entre estos podemos destacar los siguientes:

- 7 entradas analógicas
- 8 entradas/salidas digitales
- 2 UART
- 1 I2C
- 1 conector mini USB

En el prototipo de parqueo lo que se está haciendo es conectar el sensor ultrasónico en el Sensor I/O del Waspote mediante la adaptación que se puede ver en la Figura 11.

Mediante el IDE de programación proporcionado por el fabricante lo que se hace es activar el pin 5V SENSOR POWER que nos da el voltaje necesario para energizar el sensor ultrasónico, después se mide el valor del pin AN del sensor por medio del pin ANALOG 1 del Waspote [18].

Se realizó varias pruebas para determinar la distancia entre el suelo y un vehículo de modo que este sea detectado por el sensor, se determinó que con un voltaje de 20mV en la salida AN del sensor se tiene una distancia aceptable para no tener falsos positivos o negativos. Este valor se compara mediante una sentencia en el Waspote que se encarga de entregar un alto en el pin DIGITAL 4.

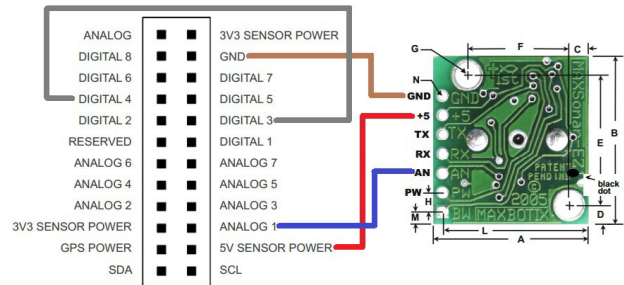


Fig. 11. Conexión del sensor al Waspote

El valor que se obtuvo en el pin DIGITAL 4 se envía al pin DIGITAL 3, este valor es impreso por el Waspote, es decir este valor es el que será enviado por el módulo inalámbrico al Gateway conectado a la PC. Parte de la sentencia que lee el sensor ultrasónico se muestra a continuación:

```
void loop() {
  value=analogRead(ANALOG1);
  if(value > 20){
    digitalWrite(DIGITAL4, HIGH);
  }
  else{
    digitalWrite(DIGITAL4, LOW);
  }
}
```

### C. Módulos XBee

Estos módulos son fabricados por la empresa Digi; estos dispositivos tienen embebidos módulos RF de diferentes protocolos de conexión inalámbrica. Estos módulos tienen una buena relación costo – beneficio ya que son relativamente baratos y pueden operar con dispositivos de otros fabricantes.

Los módulos XBee son fáciles de usar, no requieren programación adicional ya que se deben configurar requerimientos muy básicos para su funcionamiento [19].

Para el caso de nuestro prototipo se utilizó módulos XBee Pro Series 2 que se muestran en la Figura 12 que utilizan el protocolo ZigBee para comunicarse entre sí, la gran ventaja de estos módulos es que pueden crear redes en malla que permiten comunicar todos los módulos entre sí, esto es muy importante ya que si por algún motivo un nodo falla otros nodos podrán suplir la deficiencia de este.

Para configurar los módulos de Digi se utiliza el Gateway que se muestra en la Figura 12, de este modo se conecta el módulo XBee a una PC para configurarlo; otro elemento necesario para configurar el módulo es el programa X – CTU mostrado en la Figura 13 que es provisto por el fabricante; las variables que deben ser configuradas para el correcto funcionamiento de los módulos son:

- Function Set: Modo en que se comporta el módulo
- ID-PAN ID: Identificador de red
- Baud Rate: Velocidad de transmisión de datos

### D. Interfaz

Para presentar la información de los sensores se utilizó Java de modo que podemos observar gráficamente que plazas de



Fig. 12. Módulo XBee PRO Series 2 con su gateway USB

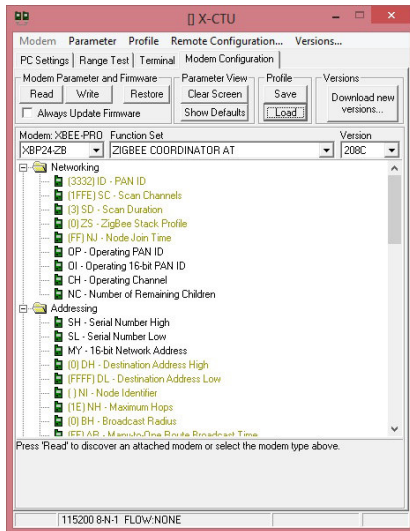


Fig. 13. Programa X-CTU

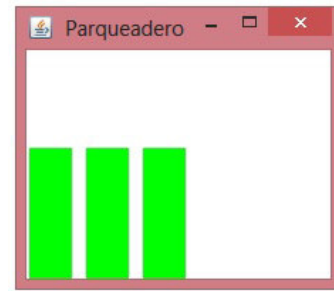


Fig. 14. Presentación gráfica



Fig. 15. Wasmotes y gateway listos

parqueo están desocupadas.

Lo que se hizo es leer el puerto serial en Java que genera el Gateway de modo que se compara el estado de la información que envía el sensor, de este modo sabemos cuándo el sensor detecta la presencia de un vehículo como se muestra en la Figura 14.

#### E. Funcionamiento

Se debe colocar el módulo XBee y el sensor ultrasónico en el Wasmote, por otro lado se debe conectar el Gateway en la PC para empezar a recibir los datos en la misma como se muestra en la figura 15.

El funcionamiento del sistema se puede describir en tres etapas; la primera es cuando el sensor detecta o no un vehículo, estos datos son procesados por el Wasmote que dentro de su programación compara los datos del sensor con sentencias determinadas para saber si está un vehículo en la plaza de parqueo.

La segunda etapa es la del envío de la información, después que el Wasmote procesó los datos que obtuvo del sensor ultrasónico, dependiendo de la programación se envía la información por medio del módulo XBee que en nuestro caso

se comunica con el Gateway mediante el protocolo ZigBee; el Gateway en la computadora se comporta como un puerto serial donde llegan los datos de forma ordenada de cada una de las plazas de parqueo.

La tercera etapa es el procesamiento de los datos en la PC, en Java se realizó un programa que despliega una ventana gráfica que muestra las plazas de parqueo, muestra en verde las plazas de parqueo desocupadas y en rojo las ocupadas como se muestra en la Figura 14. El programa en Java lo que hace es leer el puerto serial que crea el Gateway conectado a la PC y compara la secuencia de datos que llegan en formato ASCII, primero lee el identificador del sensor y después lee su estado para saber si el espacio está ocupado o desocupado. En las siguientes líneas de código se puede observar el dato que se envía en la programación del Wasmote y el dato que se lee en el programa en Java.

Línea de código en el Wasmote:

```
sprintf(aux, "3%d",
```



```
digitalRead(DIGITAL3, DEC));
```

Líneas de código en Java:

```
if(array1[j]==51){
    j++;
    if(array1[j]==49){
        System.out.println("Plaza libre 2");
        s3=1;
        ventan.repaint();
    }
    if(array1[j]==48){
        System.out.println("Plaza ocupada 2");
        s3=0;
        ventana.repaint();
    }
}
```

#### IV. EVALUACIÓN DEL SISTEMA

El pilar fundamental en el que se basa un servicio es en los usuarios, por este motivo se realizó un recorrido que se puede observar en la Figura 16, buscando plazas de parqueo en el campus de la ESPE de modo que se pueda tomar muestras de tiempo y sacar conclusiones de la eficiencia del servicio; se realizó pruebas para cuatro casos diferentes:

- El usuario no conoce el estado de las plazas de parqueo y no encuentra una plaza de parqueo libre
- El usuario no conoce el estado de las plazas de parqueo y encuentra una plaza de parqueo libre
- El usuario conoce el estado de las plazas de parqueo y no encuentra una plaza de parqueo libre
- El usuario conoce el estado de las plazas de parqueo y encuentra una plaza de parqueo libre

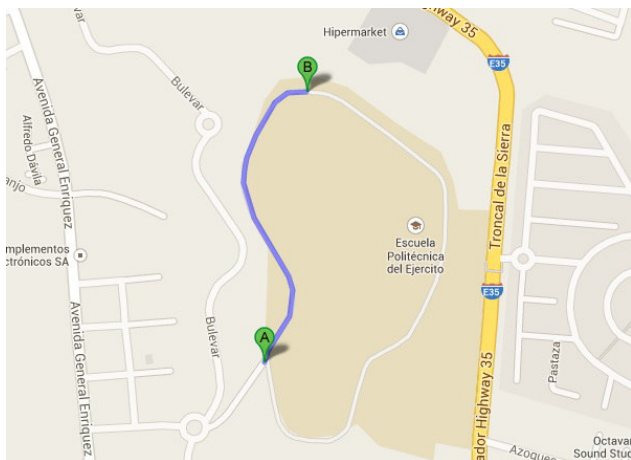


Fig. 16. Recorrido de pruebas

*A. El usuario no conoce el estado de las plazas de parqueo y no encuentra una plaza de parqueo libre*

En este caso los usuarios tomaron mucho tiempo en encontrar una plaza de parqueo libre, debido que primero el usuario debe ir lentamente buscando un espacio y después porque al llegar al lugar de parqueo todos estaban ocupados por lo que tenía

que regresar a buscar una plaza de parqueo nueva en otro sitio, en la Figura 17 se muestra una gráfica con el número de pruebas que se realizaron y los tiempos que tomó en cada caso.

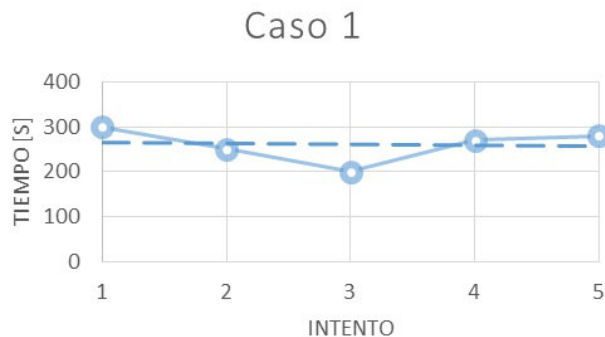


Fig. 17. Intentos y tiempo para el caso 1

*B. El usuario no conoce el estado de las plazas de parqueo y encuentra una plaza de parqueo libre*

Para este caso los usuarios tienen la suerte de encontrar una plaza de parqueo libre pero desconocen de ella hasta que la ven, el tiempo que toman en encontrar el espacio sigue siendo alto. En la Figura 18 se puede observar los intentos y el tiempo que demoró cada intento.



Fig. 18. Intentos y tiempo para el caso 2

*C. El usuario conoce el estado de las plazas de parqueo y no encuentra una plaza de parqueo libre*

En este caso el tiempo en que se demora el usuario en encontrar una plaza de parqueo puede ser muy variante debido que el usuario sabe que las plazas de parqueo están ocupadas por lo que usa espacios cercanos, en la Figura 19 se muestra la gráfica del número de intentos con el tiempo que tomo para cada intento.

*D. El usuario conoce el estado de las plazas de parqueo y encuentra una plaza de parqueo libre*

Este caso fue el mejor ya que fue el que menor tiempo tomó para encontrar una plaza de parqueo, esto se debe a que los usuarios saben que existe una plaza de parqueo libre y pueden ir más rápido a esta. Para este caso los tiempos de

## Caso 3

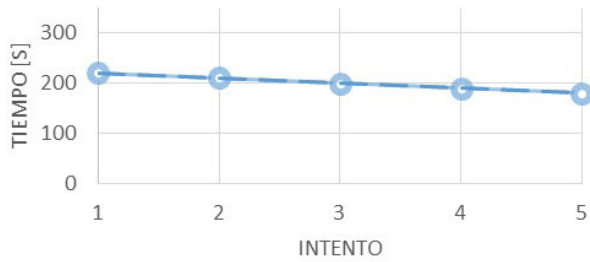


Fig. 19. Intentos y tiempo para el caso 3

búsqueda de la plaza de parqueo son casi iguales en todos los intentos debido que siempre se hace el mismo recorrido en el mismo tiempo, en la Figura 20 se puede observar la gráfica del número de intentos y el tiempo para el caso 4.

## Caso 4

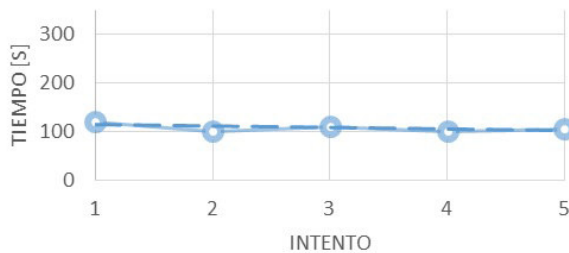


Fig. 20. Intentos y tiempo para el caso 4

## E. Análisis

Si se analizan los cuatro casos podemos ver claramente que el sistema de parqueadero inteligente soluciona el problema de hallar una plaza de parqueo reduciendo claramente el tiempo de búsqueda, reducción del consumo de combustible y tráfico que pueda generar la búsqueda, en la Figura 21 se puede observar las gráficas de los cuatro casos.

## Casos de Evaluación



Fig. 21. Casos de evaluación del sistema

## V. CONCLUSIONES

Es necesario tomar medidas respecto a las zonas de estacionamiento en ciudades donde el parque automotor

aumenta rápidamente.

El sensor ultrasónico es de fácil implementación y en las pruebas realizadas no dio muchos problemas detectando los vehículos en las plazas de parqueo.

En las pruebas que se realizaron los Wasmote no consumieron casi nada de energía, a pesar de ello se utilizó los paneles solares para asegurar su correcto funcionamiento. Los módulos XBee son muy confiables ya que se han hecho pruebas de alcances de hasta 200 metros sin línea de vista, además son muy fáciles de configurar.

La interfaz propuesta para el prototipo es muy sencilla de usar y entender por cualquier persona. Se pudo reducir el tiempo de búsqueda de una plaza de parqueo con el prototipo, esto es gracias a que los usuarios sabían en qué lugar había espacios vacíos de antemano.

## REFERENCES

- [1] LA BITÁCORA. Los Parqueaderos en la Ciudad. Obtenido de <http://www.scasantander.org/bitacoras/bitacora%2083.pdf>.
- [2] MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO, Empresa Municipal de Movilidad y Obras Públicas, Gerencia de Planificación de Movilidad. Plan Maestro de Movilidad para el Distrito Metropolitano de Quito 2009 – 2025.
- [3] WIKIPEDIA. Red de Sensores. Obtenido de [http://es.wikipedia.org/wiki/Red\\_de\\_sensores](http://es.wikipedia.org/wiki/Red_de_sensores).
- [4] WIKIPEDIA. SOSUS. Obtenido de <http://es.wikipedia.org/wiki/SOSUS>.
- [5] WIKIPEDIA. Smartdust. Obtenido de <http://en.wikipedia.org/wiki/Smartdust>.
- [6] GORDON MARIO, VÁSQUEZ BRYAN. Análisis de Desempeño de Redes de Sensores Inalámbricas en Tiempo Real Aplicadas a Monitorización Volcánica.
- [7] WIKIPEDIA. Network Time Protocol. Obtenido de [http://es.wikipedia.org/wiki/Network\\_Time\\_Protocol](http://es.wikipedia.org/wiki/Network_Time_Protocol).
- [8] WIKIPEDIA. IEEE 802.15.4. Obtenido de [http://es.wikipedia.org/wiki/IEEE\\_802.15.4](http://es.wikipedia.org/wiki/IEEE_802.15.4).
- [9] WIKIPEDIA. ZigBee. Obtenido de <http://es.wikipedia.org/wiki/ZigBee>.
- [10] LIBELIUM. Smart Parking Sensor Platform enables city motorist save time and fuel. Obtenido de [http://www.libelium.com/es/smart\\_parking](http://www.libelium.com/es/smart_parking).
- [11] ASÍN ALICIA, GASCÓN DAVID. 50 Sensor Applications for a Smarter World.
- [12] WIKIPEDIA. Sensor Ultrasónico. Obtenido de [http://es.wikipedia.org/wiki/Sensor\\_ultras%C3%B3nico](http://es.wikipedia.org/wiki/Sensor_ultras%C3%B3nico).
- [13] MAXBOTIX. LV-MaxSonar-EZ0. Obtenido de [http://www.maxbotix.com/Ultrasonic\\_Sensors/MB1000.htm](http://www.maxbotix.com/Ultrasonic_Sensors/MB1000.htm).
- [14] MAXBOTIX. LV-MaxSonar-EZ0 High Performance Sonar Range Finder Datasheet. Obtenido de [http://www.maxbotix.com/documents/MB1000\\_Datasheet.pdf](http://www.maxbotix.com/documents/MB1000_Datasheet.pdf).
- [15] LIBELIUM. Smart Cities Technical Guide. Obtenido de [http://www.libelium.com/downloads/documentation/smart\\_cities\\_sensor\\_board.pdf](http://www.libelium.com/downloads/documentation/smart_cities_sensor_board.pdf).
- [16] TVE. I+ - Wasmote. Obtenido de <http://www.rtve.es/alicarta/videos/i/wasmote/1693629>.
- [17] LIBELIUM. Hardware. Obtenido de <http://www.libelium.com/es/products/wasmote/hardware>.
- [18] LIBELIUM. Wasmote Technical Guide. Obtenido de [http://www.libelium.com/downloads/documentation/wasmote\\_technical\\_guide.pdf](http://www.libelium.com/downloads/documentation/wasmote_technical_guide.pdf).
- [19] DIGI INTERNATIONAL. XBee & XBee-Pro ZB Datasheet. Obtenido de [http://www.digi.com/pdf/ds\\_xbeezbmodules.pdf](http://www.digi.com/pdf/ds_xbeezbmodules.pdf).