

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR



**FACULTAD DE INGENIERÍA
MAESTRÍA EN REDES DE COMUNICACIÓN**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
MASTER EN REDES DE COMUNICACIÓN**

TEMA:

**“IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA
GESTIÓN DE SISTEMAS DE PARQUEO”**

FLORES SUAREZ ALDRIN ISMAEL

Quito, Noviembre 2015

DEDICATORIA

A mi Dios, mi Creador, mi Padre Celestial quien me ha dado todo lo que tengo y me ha traído hasta aquí.

“Toda buena dádiva y todo don perfecto descende de lo alto, del Padre de las luces, en el cual no hay mudanza, ni sombra de variación.”

Santiago 1:17

AGRADECIMIENTO

Agradezco a mi esposa Marisol por estar junto a mí en esta etapa de mí vida, a mis padres, hermanos y sobrinos que siempre quieren lo mejor para mí. A mi tutor Ing. Carlos Egas por su guía en la realización de este proyecto. Finalmente quiero agradecer a todo el personal Docente y Administrativo de la PUCE quienes me abrieron las puertas de esta prestigiosa institución y así poder cumplir este ansiado anhelo.

RESUMEN

El uso de Redes de Sensores Inalámbricos (WSN) va en continuo aumento, existen empresas que ofrecen soluciones basadas en estas tecnologías. Hoy en día es fácil encontrar sistemas embebidos diseñados para soportar diferentes protocolos de comunicaciones. Por otro lado la tendencia es que todas las personas posean y lleven consigo un teléfono inteligente.

Esta investigación tiene por objeto usar tanto WSN como teléfonos inteligentes en un modelo de gestión de parqueaderos. Cada plaza de parqueo dispone de un nodo sensor para detectar el vehículo. Cuando el auto llega/sale de la plaza de parqueo, el nodo sensor transmite este cambio de estado y el usuario indica al servidor por medio de su teléfono inteligente la utilización de este recurso físico. En el servidor corre una aplicación web que muestra en tiempo real el estado de ocupación de las plazas de parqueo y los registros de los clientes.

En resumen el objetivo del presente trabajo es implementar un sistema de monitoreo inalámbrico de los sitios de parqueo utilizando tecnología 802.11, y su gestión vía red utilizando teléfonos inteligentes.

Palabras clave: Gestión de parqueaderos, Red de sensores inalámbricos, IEEE 802.11, teléfonos inteligentes, ESP8266.

ABSTRACT

The use of Wireless Sensor Networks (WSN) is continuously increasing; there are companies that offer solutions based on these technologies. Today is easy to find embedded systems designed to support different wireless communication protocols. On the other hand the tendency is that all people possess and carry with them a smartphone.

This research aims to use both WSN and smartphones in a parking management model. Each parking site has a sensor node to detect the vehicle. When a car arrives/leaves the parking site, the sensor node transmits this state change and the user indicates to the server that he is using this physical resource by using its smartphone. In the server runs a web application that shows in real time the parking sites occupation state and the records of the clients.

In short, the objective of this work is to implement a wireless monitoring system parking sites using 802.11 technology, and management via network using smartphones.

Keywords: Parking management, Wireless Sensor Network, IEEE 802.11, smartphones, ESP8266.

CONTENIDO

1.	CAPÍTULO 1: INTRODUCCION	8
1.1	Introducción	8
1.2	Antecedentes	9
1.3	Justificación	10
1.4	Objetivos	12
1.4.1	Objetivo General	12
1.4.2	Objetivos Específicos	12
1.5	Alcance y Limitaciones del Prototipo	14
1.5.1	Alcance	14
1.5.2	Limitaciones	14
1.6	Resumen de contenido de los capítulos	15
2.	CAPÍTULO 2: ESTADO DEL ARTE	16
2.1	Antecedentes	16
2.2	Redes de Sensores Inalámbricos (WSN)	16
2.2.1	Topologías de red	20
2.2.2	Aplicaciones.....	22
2.3	Sensores y actuadores.....	23
2.3.1	Sensores para detectar metales	26
2.3.2	Sensores de presencia	27
2.4	Plataformas de Hardware para implementación de los Nodos	29
2.4.1	Plataforma WaspMote de Libelium	32
2.4.2	Plataforma TinyNode.....	34
2.4.3	Plataforma Arduino	34
2.5	Protocolos en WSN.....	36
2.5.1	IEEE 802.11.....	37
2.5.2	IEEE 802.15.4.....	39
2.5.3	Protocolos de capa Red	40
2.5.4	Protocolos de capa Transporte.....	41
2.6	Modelos de Gestión de Parqueaderos	42

2.6.1	Tipos de modelos de gestión de parqueaderos	42
2.6.2	Sistemas de gestión automáticos actualmente en funcionamiento en la ciudad de Quito.....	44
3.	CAPÍTULO 3: DISEÑO DE LA RED.....	46
3.1	Identificación de las necesidades y Objetivos del cliente	46
3.2	Caracterización de la red existente	50
3.3	Diseño lógico de la Red	52
3.3.1	Elección de la topología de Red.....	54
3.3.2	Elección del protocolo de comunicación a utilizar.....	55
3.4	Diseño físico de la Red.....	58
3.4.1	Nodos Sensores	59
3.4.2	Características de los Access Point	64
3.4.3	Características del equipo para el servidor.....	64
3.4.4	Características de los smartphones.....	65
4.	CAPÍTULO 4: DISEÑO Y CONSTRUCCION DEL SISTEMA PROTOTIPO	67
4.1	Diseño del hardware del NODO SENSOR.....	67
4.2	Programación del Nodo.....	70
4.3	Diseño de la Plataforma del Servidor	73
4.4	Diseño de la Aplicación para Smartphone	75
4.5	Pruebas	76
4.5.1	Pruebas en laboratorio.....	76
4.5.2	Pruebas de campo	80
4.5.3	Resultados de pruebas de campo.....	81
4.6	Prospectiva de uso	84
5.	CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	85
5.1	Conclusiones	85
5.2	Recomendaciones	86
	ANEXO I PROGRAMA FUENTE DEL NODO SENSOR	87
	anexo ii comandos at (esp8266).....	96
	ANEXO III aplicación SERVIDOR	99
	ANEXO IV ESTIMACION DE COSTOS	103
	Bibliografía	105

INDICE DE FIGURAS

Figura 1 WSN de un solo salto.....	18
Figura 2 WSN de salto múltiple	19
Figura 3 Sistema sensor/actuador.....	23
Figura 4 Detector de metales basado en BFO	27
Figura 5 Interruptores de Contacto.....	28
Figura 6 Funcionamiento Sensor ultrasónico	29
Figura 7 Esquema de un Nodo sensor parte de WSN	31
Figura 8 Elementos principales de una placa WaspMote.....	33
Figura 9 Escalabilidad de Hardware en Plataforma Arduino	36
Figura 10 Pila de Protocolos genérica para Redes de Sensores.....	37
Figura 11 Modelo Básico de gestión de parqueaderos	42
Figura 12 Sistema de Gestión con cámaras IP y reconocimiento de placa	44
Figura 13 Señalética electrónica para guía de usuarios de los parqueaderos	45
Figura 14 LEDs indicadores de ocupación en las plazas de parqueo	45
Figura 15 Gráfico comparativo entre oferta y demanda de parqueaderos	49
Figura 16 Topología de la red para gestión de parqueaderos	54
Figura 18 Módulo WiFi ESP8266	57
Figura 19 Formas de instalar un sensor ultrasónico para detector presencia de vehículos ..	61
Figura 20 Diagrama de bloques del NODO SENSOR	67
Figura 21 Esquema de conexiones de los componentes del NODO SENSOR.....	68
Figura 22 Diagrama Esquemático del hardware del Nodo Sensor	69
Figura 23 Diagrama de flujo del programa en el Nodo Sensor	72
Figura 24 Esquema lógico de la Aplicación Servidor.....	74
Figura 25 Pruebas en laboratorio	76
Figura 26 Formato de transmisión del nodo sensor al servidor	77
Figura 27 Esquema de pruebas	78
Figura 28 prueba de comunicación Nodo - Servidor	79
Figura 29 Aplicación Android - Parqueaderos	80
Figura 30 Prueba de campo	81
Figura 31 Captura de tráfico en Wireshark	82
Figura 32 Historial de ocupación del Parqueadero 5.....	83
Figura 33 Historial del cliente “aldrin”	83
Figura 35 Pantalla principal de la Aplicación Servidor	100
Figura 36 Pantalla de autenticación para ingreso a la aplicación servidor	100
Figura 37 Pantalla de monitoreo de los Nodos Sensores en la Aplicación servidor	101
Figura 38 Estadísticas de los Nodos Sensores	101
Figura 39 Clientes inscritos en el servidor a través del teléfono inteligente	102
Figura 40 Registro de uso de parqueaderos por los clientes	102

INDICE DE TABLAS

Tabla 1 Comparación entre una red tradicional y una red WSN	17
Tabla 2 Topologías de red	20
Tabla 3 Transductores de diferentes magnitudes físicas.....	25
Tabla 4 Protocolo 802.11	39
Tabla 5 Demanda actual de parqueaderos instalaciones PUCEQ.....	47
Tabla 6 Categorización de usuarios.....	48
Tabla 7 Utilización de las plazas de parqueo	48
Tabla 8 Comparación entre los diferentes estándares de comunicaciones inalámbricas	56
Tabla 9 Precios para los módulos de comunicaciones	57
Tabla 10 Cuadro comparativo entre las diferentes alternativas de sensores.....	60
Tabla 11 Comparación entre distintas plataformas de hardware usadas para nodos WSN..	62
Tabla 12 Lista de Comandos AT para ESP8266	98
Tabla 13 Costos de los Nodos Sensores	103
Tabla 14 Costos Puntos de Acceso WiFi	103
Tabla 15 Costos equipo servidor	103
Tabla 16 Costos de Programación, configuración e instalación.....	104
Tabla 17 Costo total estimado.....	104

1. CAPÍTULO 1: INTRODUCCIÓN

1.1 Introducción

Este trabajo presenta La IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA GESTIÓN DE UN SISTEMA DE PARQUEO, el prototipo utiliza una red de sensores inalámbricos (WSN) usada para el monitoreo del estado de ocupación de las plazas de los parqueaderos de la PUCE. El fin principal es que tanto los administradores como los usuarios de las plazas de parqueo tengan una información en línea del estado de ocupación, lo cual tiene por objeto dar un mejor servicio a los usuarios y disminuir el tráfico dentro de las instalaciones. El prototipo SE COMPONE de: un sensor de presencia, un sistema con microcontrolador, que comanda una Interface de comunicaciones WiFi. Este prototipo de nodo funciona de la siguiente forma: Cuando el auto esta sobre la plaza de parqueo se activa el sensor de presencia, esta señal de activado se codifica y se envía al servidor donde se almacena en una base de datos junto con el tiempo de lectura, esta información se estará actualizando todo el tiempo. Este equipo estará conectado vía WiFi a la estación base donde esta un servidor y por medio de este a la Intranet o al Internet. Cualquier dispositivo móvil conectado a la red en el cual se haya instalado previamente una aplicación podrá revisar el estado de ocupación de la plaza de parqueo.

Actualmente el monitoreo de las plazas de parqueo se lo realiza de tres formas que son: en forma física, usando cámaras y a través de sensores. La constatación de su

estado de ocupación es por medio de un operario, por indicadores visuales, letreros que indican el número de plazas libres, etc., las cuales todas son soluciones cableadas que requieren trabajos de instalación engorrosos, onerosos y no proporcionan el estado de ocupación fuera de las instalaciones. Este trabajo proyecta usar tecnologías existentes en monitoreo, microcontroladores, red inalámbrica de sensores WSN, para acoplarlas y programarlas para funcionar como nodos de monitoreo de ocupación de plazas de parqueo cuyo estado podrá ser constatado desde cualquier sitio a través de una aplicación para dispositivos móviles.

1.2 Antecedentes

Conociendo que el crecimiento del parque automotor de Quito bordea el 11% anual y actualmente en la ciudad circulan más de 445 mil autos privados. (1). Tomando en cuenta esta estadística y aplicándola a la población de la PUCE-Quito que concurre diariamente a las instalaciones que se encuentran en la Av. 12 de Octubre, se puede decir que la proporción de plazas de parqueo respecto a las solicitudes de parqueadero cada año será menor, lo cual va en desmedro del bienestar de la población estudiantil.

Conceptualizando a la movilidad, como un elemento del sistema de inclusión y equidad social que busca el buen vivir y siendo una política común para todas las instituciones “la incorporación de las nuevas tecnologías de información y comunicación en la gestión y control, la modernización de los servicios públicos y privados” (2).

Tomando en cuenta estos antecedentes y considerando que la Universidad es la llamada a dar soluciones se ha presentado este proyecto el cual tiene como innovación la forma como se informa al usuario del estado de ocupación del parqueadero, esto agiliza el movimiento vehicular dentro de las instalaciones de la PUCE-Q y sus alrededores ya que disminuye el tráfico recurrente.

La solución propuesta en este proyecto de graduación de maestría consiste en el diseño y construcción del prototipo de un nodo de red WSN (Wireless Sensor Network) que será probado en un modelo de gestión de parqueaderos.

1.3 Justificación

Uno de los problemas que enfrenta la ciudad de Quito todo el tiempo es la movilidad vehicular y esto deriva en problemas tales como la escasez de plazas de parqueo en sitios de alta afluencia. Esta situación de escasez provoca que los usuarios pierdan tiempo buscando un parqueadero libre y que generen más tráfico en el sector al seguir dando vueltas para conseguir una plaza (3). Debido a que Quito es una ciudad patrimonial no es posible la construcción de parqueaderos a gran escala, por lo que una buena administración de los existentes es de gran ayuda. El problema de escasez de plazas de parqueo lo podemos observar en horas pico principalmente en centros comerciales, mercados, universidades y cerca de instituciones públicas de gran demanda.

Aprovechando el desarrollo tecnológico y la infraestructura de red existente precisamente en estos sitios, podemos implementar una solución que ayude a administrar de una mejor manera este recurso limitado que son las plazas de parqueo.

Este trabajo busca utilizar la tecnología de red inalámbrica de sensores WSN para transmitir la información previamente obtenida a través de los sensores de presencia utilizando como intermediario un microcontrolador y la plataforma de hardware de comunicaciones.

En nuestro medio las soluciones comerciales presentadas hasta hoy para monitorear plazas de parqueo son con indicadores visuales y no permiten conocer el estado de ocupación cuando la señalización del parqueadero no es visible al usuario.

La solución que se presenta en este proyecto de grado, tiene como innovación el hecho de que los Nodos de monitoreo (sensor-microcontrolador-hardware de comunicaciones) están conectados inalámbricamente a un servidor al que podemos acceder en cualquier momento para conocer el estado de ocupación de las plazas de parqueo. Esta comunicación se hará vía WiFi, lo cual ahorra el tener que realizar trabajos civiles y además permite la instalación dinámica de los equipos cuando existan cambios en las Instalaciones.

Las soluciones comerciales que existen en el mercado tienen instalaciones cableadas con indicadores de ocupación visuales estáticos que no proporcionan ninguna observación cuando no se tiene visibilidad de la señalización del parqueadero. Además tienen tecnologías cerradas de alto costo y hasta hoy han sido implementadas solo en centros comerciales, y no están al alcance de Instituciones tales como universidades. Este proyecto se enfoca en ser una solución de bajo costo, con tecnología adaptada cuya configuración y programación son propias. La importancia de este proyecto radica en la utilización de la tecnología para agilizar los

procesos, dar un bienestar a los usuarios de las instalaciones y disminuir el tráfico recurrente.

1.4 Objetivos

1.4.1 Objetivo General

Diseñar un prototipo de GESTIÓN DE PARQUEADEROS en base a una red de sensores inalámbricos.

1.4.2 Objetivos Específicos

- Analizar el estado del arte respecto a WSN y los sensores disponibles que mejor se ajusten para la consecución del proyecto.

Para esto se revisara los catálogos en línea de los fabricantes y vendedores de Nodos de Monitoreo y Sensores más representativos. Se expondrán los principios de funcionamiento y características principales de los equipos que mejor se ajusten para la consecución del proyecto.

Se visitara las instalaciones de parqueaderos referenciales donde se encuentren instalados sistemas automáticos de gestión para realizar un análisis del estado del arte de los mismos.

Se revisará la información disponible de plataformas de hardware para WSN, respecto a los equipos, métodos y procedimientos que mejor se ajusten con los diferentes módulos de este proyecto de graduación.

Todo este análisis se lo realizara a nivel técnico es decir respecto a la tecnología usada, ventajas y desventajas de cada uno.

- Diseñar un PROTOTIPO de monitoreo de ocupación de una plaza de parqueo adaptando, configurando y programando equipos de la plataforma de hardware escogida para hacerlos funcionar como parte de una red de sensores WSN.

Se realizara un análisis de la gestión actual de una plaza de parqueo de la PUCE –Quito, en base a la información recopilada, se diseñará un modelo de gestión basado en WSN para este sector para lo cual se diseñará y construirá un prototipo de nodo que se podrá conectar vía WiFi a la futura red WSN o a la INTRANET actual.

Este prototipo de Nodo será construido usando la plataforma de hardware seleccionada para lo cual se estudiará detenidamente el hardware que tenga relación con nuestro proyecto.

Se programará en el entorno de desarrollo proporcionado por el fabricante que implementa el lenguaje de programación y el cargador de arranque que es ejecutado en la placa.

- Diseñar la aplicación que me permita ADMINISTRAR el Parqueadero (Nodo de monitoreo) y a los usuarios, esta aplicación correrá en un equipo al que se denominará servidor.
- Diseñar una aplicación para dispositivos móviles que tengan el sistema operativo Android. Esta aplicación permitirá visualizar el estado de ocupación de los parqueaderos y confirmar la ocupación de cualquiera de ellos, en el servidor se llevará el registro del tiempo de ocupación del parqueadero por el cliente.

- Probar el funcionamiento de este nodo en las Instalaciones de un parqueadero.

Se instalara el prototipo del nodo en un punto de la plaza de parqueo escogida y se realizará las pruebas controladas de monitoreo. Se comprobará desde cualquier sitio dentro de la cobertura de la red inalámbrica vía WiFi la conectividad con el nodo y a través de la aplicación se obtendrán los datos en línea del estado de ocupación así como los registros de tiempo de ocupación en el servidor.

1.5 Alcance y Limitaciones del Prototipo

1.5.1 Alcance

- El prototipo del hardware a construir llegará solo hasta el nivel de pruebas es decir no se hará trabajos civiles de instalación en los parqueaderos.
- Las pruebas se realizarán solo en el entorno de un servidor local en una red independiente de la red de la PUCE.
- El prototipo está orientado al estudio de una nueva tecnología en una aplicación en particular, no está orientado a la elaboración de un producto para uso comercial.

1.5.2 Limitaciones

- Por ser un prototipo de pruebas tanto el hardware como el software no están diseñados para trabajar en condiciones extremas o críticas.

- El hardware y software están diseñados para funcionar en un entorno controlado, por lo tanto es sujeto de errores de funcionamiento cuando nos salimos de este entorno.

1.6 Resumen de contenido de los capítulos

En el capítulo 2 se hace una descripción del estado del arte, primeramente de redes inalámbricas de sensores WSN y luego del estado del arte relacionado a la gestión de parqueaderos.

En el capítulo 3 se realiza el diseño lógico y físico de la red de sensores basada en el protocolo IEEE 802.11, al final se presenta una descripción técnica de las características del hardware a utilizarse.

En el capítulo 4 se presenta el diseño y construcción del sistema prototipo en lo que corresponde al hardware del nodo sensor, la programación del Nodo, la descripción de sistema del servidor y la aplicación Android desarrollados. Así mismo en este capítulo se presentan los esquemas y resultados de las pruebas de laboratorio y de campo.

Al final en el capítulo 5 se presentan las conclusiones y recomendaciones que se pudieron extraer como producto de la realización del presente proyecto.

2. CAPÍTULO 2: ESTADO DEL ARTE

2.1 Antecedentes

En la actualidad el avance tecnológico es vertiginoso. Nada está acabado en tecnología, a cada momento aparecen nuevos equipos con más funciones y mejores características que traen un abanico de posibilidades para mejorar un diseño, hacer una innovación y aun para reinventar la manera de realizar una actividad.

En este capítulo para describir el estado del arte en red de sensores WSN y sus componentes primeramente se detalla el fundamento teórico para luego describir el hardware o plataformas de hardware que se encuentran comercialmente difundidas y los protocolos de comunicaciones usados por estas.

En las dos últimas secciones se enfoca el estado del Arte en lo referente a la gestión de parqueaderos que servirá como caso de estudio para la aplicación de una solución en base a una red de sensores.

2.2 Redes de Sensores Inalámbricos (WSN)

Definición:

“Una red de sensores es una infraestructura compuesta de elementos de monitoreo (medición), procesamiento y comunicación que proporcionan al administrador la capacidad de observar y reaccionar a eventos y fenómenos en un ambiente específico. El administrador puede ser una entidad civil, gubernamental, comercial o industrial. El ambiente puede ser el mundo físico, un sistema biológico o una estructura de tecnologías de la información” (4)

Los componentes básicos de una red de sensores son:

- Un conjunto de sensores distribuidos o localizados.
- Una red de interconexión (usualmente inalámbrica)
- Un punto central de información
- Un conjunto de recursos informáticos en el punto central de información para manejar la correlación de datos, tendencias de los eventos, consulta de estados y minería de datos.

REDES TRADICIONALES	REDES DE SENSORES INALAMBRICAS
Diseño de propósito general; sirven a muchas aplicaciones	Diseño de propósito único; sirven a una única aplicación
Las consideraciones principales de diseño son la red, rendimiento y latencias; la energía no es una consideración importante	La energía es la restricción principal en el diseño de todos los nodos y componentes de la red.
Las redes están diseñadas y construidas de acuerdo a planes.	El despliegue, estructura de la red y el uso de recursos son a menudo ad-hoc (sin planificación).
Dispositivos y redes operan en entornos moderados y controlados	Las redes de sensores a menudo operan en entornos con condiciones hostiles.
El mantenimiento y reparación son ordinarios y las redes son de fácil acceso	El acceso físico a los nodos sensores es a menudo difícil o incluso imposible
El fallo de un componente se hace a través de mantenimiento y reparación	La falla de los componentes es tomada en cuenta en el diseño de la red.
El conocimiento global de la red es normalmente factible y es posible una gestión centralizada	La mayoría de las decisiones se toman en forma localizada sin el apoyo de un gestor central.

Tabla 1 Comparación entre una red tradicional y una red WSN

Las Redes de Sensores Inalámbricos son usadas como medio para intercambiar información entre una aplicación informática y uno o más sensores, la practicidad y utilidad de las redes de sensores (WSN) es que estas proveen una comunicación máquina – máquina flexible y efectiva a bajo costo.

Una red de sensores elemental se puede observar en el esquema de la figura 1 este tipo de WSN tiene un enlace de un solo salto (single-hop network link).

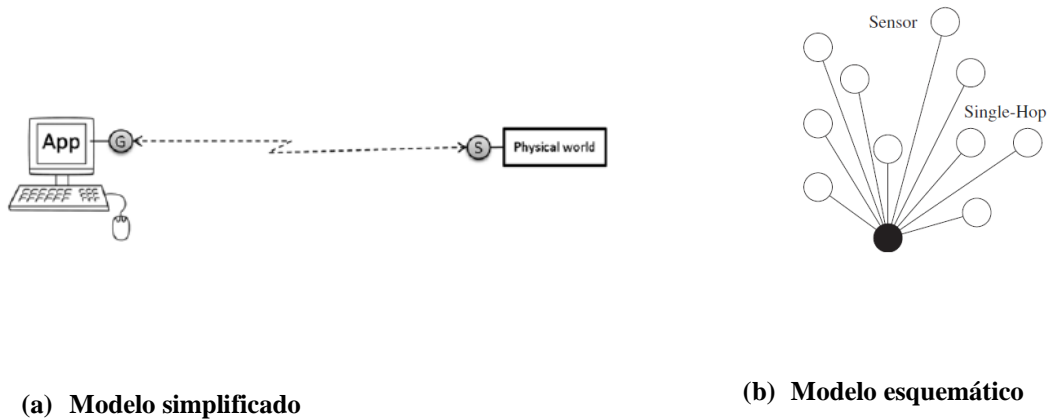


Figura 1 WSN de un solo salto¹

Los componentes elementales de esta red son el servidor, el Gateway y el nodo sensor (leaf node).

Para ampliar la cobertura de la red y tener la posibilidad de que los datos tengan más de un único camino para llegar a su destino se añade entre el Gateway y el nodo sensor un nodo de retransmisión inalámbrico (relay node) como se muestra en la figura 2.

¹ Fuente (6)

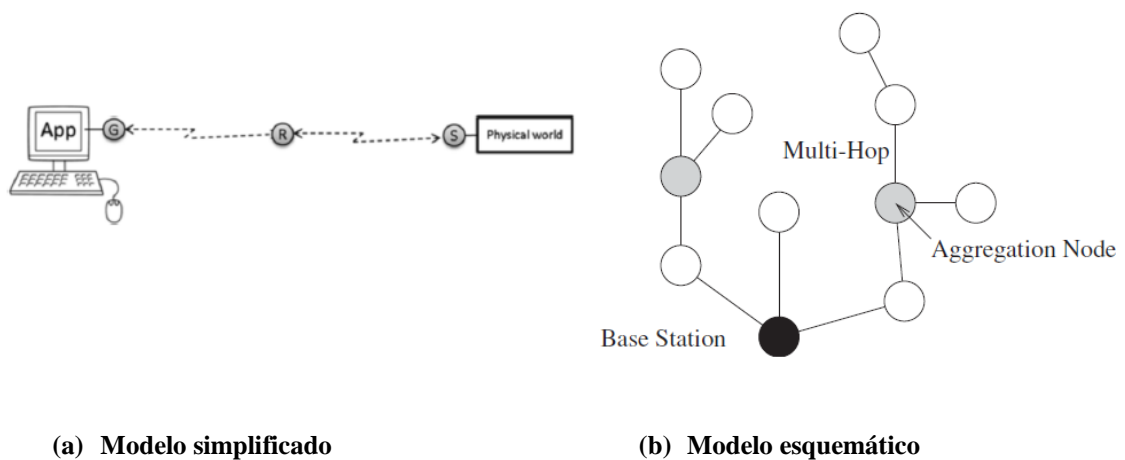


Figura 2 WSN de salto múltiple²

El Gateway proporciona la interface entre la plataforma informática donde corre la aplicación que recoge los datos y los nodos inalámbricos, esta aplicación puede correr en un servidor local o en un servidor conectado en red.

El Gateway puede ser un computador o un sistema embebido el cual sirve tanto para comunicar datos y comandos entre los nodos y transferir datos a una aplicación u otra red, como Internet.

Además los Gateways se encargan de realizar la conversión de protocolos lo que permite a la red de sensores inalámbricos comunicarse con dispositivos que no manejan el mismo protocolo de comunicaciones.

Las funciones principales de los nodos de retransmisión (relay nodes) son de extender el área de cobertura de la red y cambiar el camino de los datos en caso de congestión o de falla, es decir se comportan como “routers”, además tienen la

² Fuente (6)

posibilidad en caso de requerirlo de actuar simultáneamente como nodo sensor es decir son dispositivos completos y se identifican como FFD (full-function device).

Los nodos sensores (leaf node) a veces llamados endpoint están diseñados para proveer la interfaz física entre la red de sensores inalámbricos y el sensor/actuador al que está conectado, estos dispositivos tienen funcionalidad reducida y se identifican como RFD (reduced-function device).

2.2.1 Topologías de red

Ya que una WSN puede estar compuesta por decenas, cientos o miles de dispositivos, las topologías de red deben ser consideradas en su diseño, las topologías de red más comunes en WSN son: estrella, árbol, malla, o redes híbridas que combinan las anteriores. Cada una de estas topologías tiene sus propios retos, ventajas y desventajas.

Topología	Uso de Energía	Rango de cobertura	Requiere sincronización de tiempo
Estrella	baja	corta	no
Árbol	baja	larga	si
	Alta	larga	no
Híbrida	Baja (típicamente)	larga	(depende de la configuración)

Tabla 2 Topologías de red

Topología de estrella

Características:

- Es un sistema de salto único
- Los nodos sensores se comunican directamente con el Gateway

- Si un sensor falla el resto sigue funcionando.
- Para transmitir datos no se necesita sincronización con los otros nodos
- Tiene el más bajo consumo de energía promedio.
- La distancia de transmisión va desde 10 a 100m

Topología de árbol

Características:

- Es un sistema de salto múltiple
- Tiene nodos de retransmisión en niveles y jerarquías
- Optimiza el consumo de energía
- Amplia el rango de cobertura de la red
- Para transmitir datos se necesita sincronización entre todos los nodos

Topología de malla

Características:

- Es un sistema de salto múltiple
- Cada nodo está conectado a los nodos que están en su radio de cobertura
- Tiene bajo consumo de energía para largas distancias
- Es necesario que todos los nodos estén siempre activos
- Necesita sincronización

2.2.2 Aplicaciones

Las aplicaciones de WSN solo dependen de la imaginación ya que la infraestructura de red ya está desplegada y lo único que se necesita es diseñar la WSN de modo que se adapte o conecte a ella. A continuación se presentan algunos campos donde se están usando redes de sensores.

- Aplicaciones comerciales
- Monitoreo ambiental (tierra, aire, mar) y sensores inalámbricos para la agricultura
- Automatización de viviendas, incluyendo alarmas
- Aplicaciones de seguridad nacional: red de sensores para monitorear posibles eventos catastróficos.
- Control y monitoreo industrial
- Operaciones Metropolitanas (tráfico, peajes automáticos, etc.)
- Aplicaciones militares

En Ecuador aunque los nodos sensores pasen desapercibidos para el ciudadano común estos están presentes en gran parte de las principales ciudades, Un ejemplo de una red WSN a gran escala en Ecuador es el EQU 911 que es un Sistema Integrado de Seguridad donde una de sus funciones es monitorear a 55.000 unidades de transporte de todo el país, cada unidad tiene un GPS, dos cámaras de video y tres botones de auxilio. Otros ejemplos a citar son:

- Monitoreo de actividad de los volcanes

- Monitoreo de parámetros ambientales del INAMHI
- Red de cámaras para el sistema de fotomultas en el DMQ.

2.3 Sensores y actuadores

En el cuerpo humano, los órganos de los sentidos son los medios a través de los cuales el cerebro recibe información del medio externo, el cerebro es el que se encarga de dar el significado a esta información y según sea su patrón de respuesta efectuar una acción. Haciendo una analogía del cuerpo humano con una de una red de sensores, los nodos sensores vendrían a ser los órganos de los sentidos y los microprocesadores serían las extensiones inteligentes del cerebro.

Los sensores y actuadores son los dispositivos de hardware que se usan para interactuar con el medio físico al cual pertenecen el parámetro o parámetros que se desean monitorear y/o controlar.

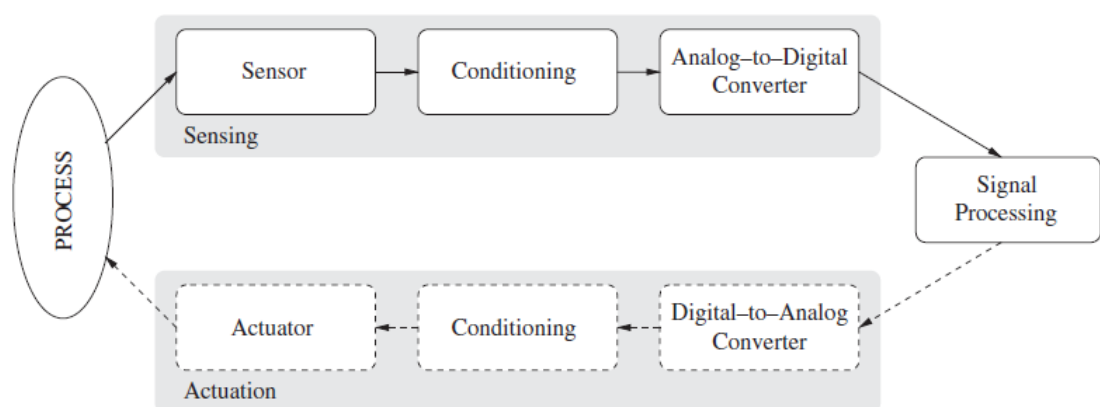


Figura 3 Sistema sensor/actuador³

³ Fuente (12)

Un sensor también se conoce como transductor cuya definición ampliamente difundida dice así:

“Un transductor es un dispositivo capaz de convertir el valor de una magnitud física en una señal eléctrica codificada, ya sea en forma analógica o digital” (5)

El sensor (transductor) es conectado a los puertos de lectura del microprocesador a través de circuitos electrónicos que se encargan de “adecuar” la señal a los niveles de voltaje estandarizados que el microcontrolador es capaz de discriminar. (6)

La señal eléctrica que es entregada por el sensor al microprocesador puede ser analógica o digital siendo por lo tanto necesario definir en el programa grabado en el microprocesador el tipo de lectura que se va a realizar.

- Señal eléctrica digital: en este caso solo se tienen dos niveles lógicos 0L y 1L, un ejemplo de este sensor puede ser un sensor de presencia.
- Señal analógica: el microcontrolador tiene internamente circuitos conversores analógico/digital que codifican el nivel de señal analógica en un número binario equivalente de 8 bits si el microcontrolador trabaja con datos de este tamaño. Un ejemplo de este tipo de sensor es un sensor de temperatura.

En la tabla a continuación se muestran transductores de diferentes magnitudes físicas.

MAGNITUD DETECTADA	TRANSDUCTOR	CARACTERISTICAS
Posición lineal o angular	Potenciómetro	Analógico
	Encoders	Digital
Pequeños desplazamientos o deformaciones	Transformador diferencial	Analógico
	Galga extensométrica	Analógico
Velocidad lineal o angular	Dinamo tacométrico	Analógico
	Detector inductivo u óptico	Digital
Fuerza y par	Medición indirecta (galgas o trafo diferenciales)	analógicos
Presión	Membrana + detector de desplazamiento	Analógicos
Caudal	De turbina	Analógico
Temperatura	Termopar	analógico
Sensores de presencia o proximidad	Inductivos	Todo-nada
	Capacitivos	Todo-nada
	Ópticos / ultrasónicos	Todo-nada o analógicos
Sistemas de visión artificial	Cámaras de video	Procesamiento digital por puntos o píxeles
	Cámaras CCD	

Tabla 3 Transductores de diferentes magnitudes físicas⁴

En el caso del diseño del prototipo de nodo de red WSN aplicado a la gestión de parqueaderos el único parámetro que por el momento interesa monitorear es la presencia o no de un auto en la plaza de parqueo por lo cual el sensor deberá ser un detector de presencia o un detector de metales que entregará al microcontrolador un valor lógico 1L ó 0L. con los valores de voltaje y corriente concordantes a los parámetros permitidos por el microcontrolador los cuales son especificados en su respectiva documentación técnica.

⁴ Fuente (5)

2.3.1 Sensores para detectar metales

Los detectores de metales tienen diferentes usos que van desde la detección del cableado eléctrico en las paredes hasta la búsqueda de objetos arqueológicos enterrados en la tierra, en el caso del presente proyecto el detector de metales será usado para detectar la presencia o no de un vehículo sobre la plaza de parqueo.

Los detectores de metales los podemos clasificar de acuerdo a su principio de funcionamiento en dos principales tipos:

- Detectores que se basan en el principio de un oscilador de barrido de frecuencia BFO (Beat Frequency Oscillator) por sus siglas en inglés. Los detectores basados en este principio están formados por dos osciladores como se muestra en la figura 3, en ausencia de metales se calibra para que los dos osciladores oscilen a una misma frecuencia fija por lo que la salida del mezclador será una salida de frecuencia y amplitud mínima y en el parlante no se escuchará ningún sonido.

Cuando se acerca un metal a la bobina de búsqueda que forma el circuito resonante del oscilador 1 este circuito modifica su frecuencia de resonancia porque el metal es un elemento inductivo, en este caso a la salida del mezclador se tendrá una señal de frecuencia audible y amplitud aceptable que ya puede escucharse en un parlante.

Existen muchos circuitos para detectores de metales que se basan en este principio, la diferencia entre unos y otros es por los elementos usados para construir los osciladores y el mezclador de frecuencias para lograr mayor precisión y sensibilidad

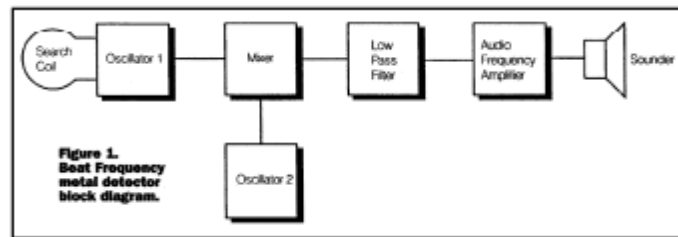


Figura 4 Detector de metales basado en BFO⁵

- Detectores basados en magnetómetros:

“Cuando una carga eléctrica se mueve en un campo magnético experimenta una fuerza perpendicular a su movimiento por lo tanto si estas cargas se desplazan en un alambre conductor, serían impulsadas hacia un lado del alambre. Debido a esto se produce una separación de carga en el alambre denominada efecto Hall. El efecto Hall nos proporciona un método adecuado para medir campos magnéticos” (7)

Los sensores basados en este principio se calibran en ausencia de metales de acuerdo al campo magnético terrestre en ese lugar y luego cuando un metal entra en su rango de detección este cambia el valor del campo magnético produciendo una señal indicadora.

2.3.2 Sensores de presencia

Son sensores que detectan objetos que se encuentran dentro del rango de cobertura del sensor. Según el principio físico los más usados son:

- **Interruptores de posición**

Son interruptores electromecánicos mecánicos en base a contactos normalmente abiertos o cerrados que cambian su estado al tener contacto con un objeto, se utilizan comúnmente en puertas para indicar que la puerta esta abierta o cerrada y también en la industria en bandas transportadoras para

⁵ Fuente (10)

indicar la posición de fin de carrera. La mayoría de estos sensores son pasivos es decir no necesitan energía para funcionar.



Figura 5 Interruptores de Contacto⁶

- **Detectores capacitivos**

Este tipo de detectores basan su funcionamiento en la afectación de un campo eléctrico ante la presencia de un objeto. Se determina el valor de voltaje en ausencia de objetos externos dentro del campo eléctrico, al ingresar un objeto en el campo eléctrico del sensor existe una variación de voltaje la misma que se usa como parámetro indicador. Se usan en pantallas táctiles y detectores de diversos materiales en base a su constante dieléctrica, son de bajo alcance (aprox 60mm).

- **Detectores fotoeléctricos**

Basan su funcionamiento al hecho de que todos los objetos reflejan la luz que incide sobre ellos, específicamente los sensores infrarrojos trabajan en el rango de 700 nm (430 THz) a 1mm (300 GHz).

Se tiene dos tipos los sensores pasivos y los sensores activos.

⁶ Fuente (13)

Sensores pasivos: están formados únicamente por un fototransistor que detecta las radiaciones provenientes de los objetos.

Sensores activos: necesitan de elemento emisor y uno receptor

Las aplicaciones son muy diversas, en lo relacionado a la detección de objetos se usan principalmente para detectar seres vivos ya que estos radian energía electromagnética en forma de luz infrarroja.

- **Detectores ultrasónicos**

Este tipo de sensores consta de un elemento emisor y un receptor. El emisor emite un impulso acústico de alta frecuencia (40 KHz) y corta duración (aproximadamente 10 us) el cual viaja por el aire a la velocidad del sonido (aproximadamente 340 m/s) y al encontrar un objeto se refleja como eco al sensor receptor. Este sensor necesita de un circuito integrado que se encarga de calcular la distancia hacia el objeto utilizando la formula $d = v \cdot t$ (distancia es igual a velocidad por tiempo).

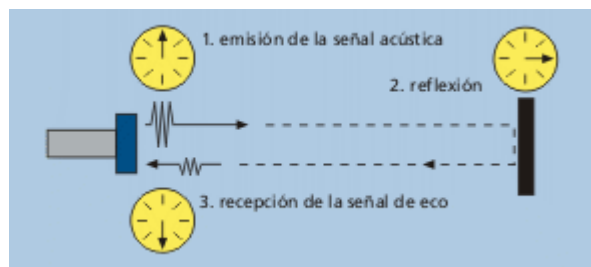


Figura 6 Funcionamiento Sensor ultrasónico⁷

2.4 Plataformas de Hardware para implementación de los Nodos

Los Nodos son los elementos diferenciadores de una red estándar de datos con una red WSN y en forma general las funciones de los Nodos son:

⁷ Fuente (14)

- Recibir las señales enviadas por los sensores y acondicionarlas a los niveles de voltaje y corriente estándar del microprocesador.
- Almacenar de datos obtenidos en el nodo mismo.
- Capacidades de procesamiento de los datos.
- Análisis de los datos para generación de alertas.
- Capacidad de generar señales actuadoras.
- Realizar las tareas de medición (monitoreo) de una manera programada
- Permitir la reconfiguración del Nodo en cuanto a la programación de las tareas de monitoreo y algoritmos de procesamiento de datos.
- Recepción, transmisión y envío (forwarding) de paquetes de datos.
- Programación y ejecución de tareas de comunicación y networking (para que el nodo se comunique con otros nodos dentro de la misma red o con otras redes).

Básicamente el hardware de un nodo de una red WSN tiene los siguientes componentes principales: Sensores/actuadores, un microcontrolador, un sistema de comunicaciones y un sistema de alimentación de energía.

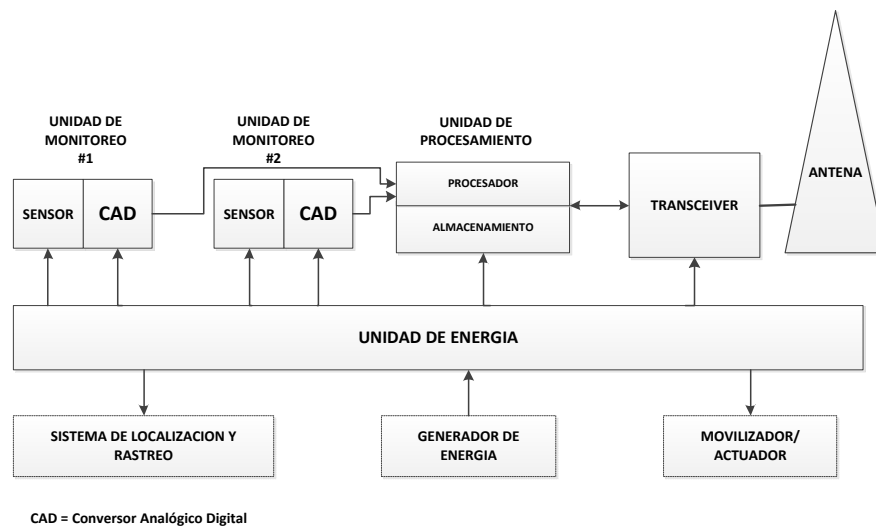


Figura 7 Esquema de un Nodo sensor parte de WSN⁸

En la sección 2.2 del presente capítulo se clasifico a los Nodos en Nodos de Retransmisión y Nodos Sensores propiamente dichos.

Un nodo de retransmisión podría ser un dispositivo integrado ó una estación de trabajo ó un poderoso servidor el cual sería capaz de proveer todas las funciones arriba descritas. Un nodo sensor propiamente dicho es típicamente un dispositivo integrado que provee solo parte de estas funciones, básicamente dos funciones que este tipo de nodo no implementa es el forwarding de paquetes de datos y las tareas de networking.

Dado que la mayoría de las aplicaciones de redes de sensores exigen condiciones de ubicación y alimentación de energía los dispositivos integrados son los más adecuados para la construcción de nodos.

⁸ Fuente (11)

Los dispositivos integrados vienen en tamaños pequeños, tienen limitación de recursos y pueden ser alimentados por baterías.

En la actualidad se dispone de una gran variedad de plataformas de hardware en base a las cuales se han diseñado dispositivos inalámbricos con el propósito de funcionar como nodos para redes WSN. Como ejemplo de plataformas de hardware se puede citar: Wasmote de libelium, Tiny node, Arduino, PSoC, etc.

2.4.1 Plataforma WaspMote de Libelium

Un WaspMote es un nodo sensor manufacturado por una empresa española llamada Libelium su diseño está basado en una arquitectura modular lo que le permite cambiar los diferentes módulos y expandirlos de acuerdo a las necesidades.

Los módulos que dispone WaspMote están categorizados así:

- Módulo ZigBee/802.15.4 (2.4GHz, 868MHz, 900MHz).
- Módulo GSM/GPRS (4 bandas: 850MHz/900MHz/1800MHz/1900MHz).
- Módulo 3G/GPRS (3 bandas UMTS 2100/1900/900MHz y 4 bandas GSM/EDGE, 850/900/1800/1900 MHz).
- Módulo WiFi.
- Módulos Bluetooth: Bluetooth Low Energy and Bluetooth Pro
- Módulo GPS
- Módulos NFC/RFID
- Módulos sensores
- Módulo de almacenamiento: SD Memory Card

Dentro de la categoría de módulos sensores Libelium pone a disposición una gran cantidad de sensores que se pueden usar para la mayor parte de aplicaciones.

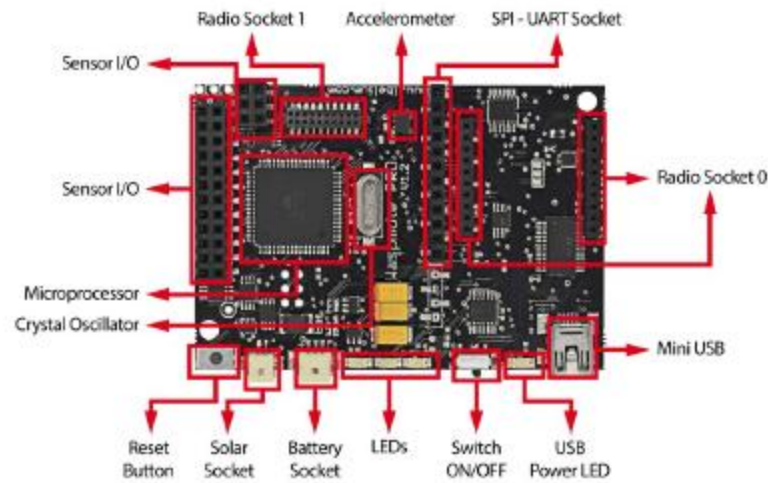


Figura 8 Elementos principales de una placa WaspMote⁹

El firmware del WaspMote que se carga en el microcontrolador es desarrollado por el programador usando una IDE (Integrated Development Environment) propietaria. El control y monitoreo de los sensores también es también a través de software propietario.

Tiene una API (Application Programming Interface) abierta, pero el hardware y el software de monitoreo son código propietario.

En los últimos años Libelium junto con IBM han puesto a disposición de los desarrolladores e investigadores el WaspMote Mote Runner que es un WaspMote modificado que trabaja en el ambiente desarrollado por el laboratorio de Investigación de IBM con el objeto de que la comunidad de investigadores aporte al desarrollo de 6LowPAN (IPv6 over Low power Wireless Personal Area Networks).

⁹ Fuente (15)

2.4.2 Plataforma TinyNode

La filosofía de diseño de TinyNode es que los componentes que son el núcleo de todas las aplicaciones estén en una placa base y las funciones adicionales se ejecuten en tarjetas de extensión. La placa base es un nodo inalámbrico versátil de baja potencia que tiene un arreglo de hardware que permite una amplia variedad de opciones de conectividad, almacenamiento, energía e interfaces.

Una diferencia con otras “motas” es su amplio rango de bit rate que va desde 1,2kbps hasta 152 kbps, además esta plataforma tiene el soporte total del sistema operativo TinyOS.

TinyOS está escrito en nesC que es una extensión del lenguaje C. Este soporta una programación basada en componentes dirigidos por eventos (event-driven). El concepto básico de este tipo de programación es descomponer el programa en componentes con una funcionalidad contenida en si mismos, estos componentes interactúan intercambiando mensajes a través de sus interfaces. La principal ventaja de este tipo de programación es la reusabilidad de los componentes.

TinyOS es la herramienta estándar de facto para programación en WSN.

2.4.3 Plataforma Arduino

Arduino es una plataforma de hardware abierta basada en los microcontroladores ATMEGA8 y ATMEGA168 de Atmel y un ambiente de desarrollo (basado en Wiring) que permite grabar y depurar los programas directamente en el hardware.

La plataforma Arduino puede interactuar con su entorno físico a través de su sistema de entrada salida que permite el ingreso y salida de señales tanto digitales como

analógicas y los proyectos con Arduino pueden ser autónomos o se pueden comunicar con un software en ejecución.

Las características principales de la Plataforma Arduino son:

- Bajos costos: se puede encontrar placas desde 22\$
- Multiplataforma: El software de Arduino corre sobre los sistemas operativos más comunes (ejemplo: Windows, Linux, etc)
- Entorno de programación simple y claro: El software de programación esta basado en el entorno Processing el cual es fácil de usar para principiantes y es flexible para que los avanzados puedan aprovecharlo.
- Código abierto y software extensible: El software de Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++
- Código abierto y hardware extensible: los diagramas del hardware están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados pueden hacer su propia versión del circuito extendiéndolo y mejorándolo.

Dado los bajos costos del hardware y su arquitectura completamente abierta, Arduino es muy popular y para empezar a desarrollar proyectos con él basta tener conocimientos mínimos de electrónica.

“Con Arduino se aprende haciendo”

La plataforma Arduino no fue diseñada específicamente para construir Nodos Sensores para WSN, pero debido a su Arquitectura modular puede ser usado con este propósito, ya que a la placa base se pueden acoplar módulos sensores y módulos de comunicación según sea la necesidad, así se puede ver en la figura 9 un ejemplo de un nodo sensor en base a la plataforma Arduino.



Figura 9 Escalabilidad de Hardware en Plataforma Arduino¹⁰

2.5 Protocolos en WSN

La meta en WSN es implementar una red inalámbrica normalizada de bajo costo que soporte tasas de transmisión de datos bajas y medias, tenga bajo consumo de potencia garantizando seguridad y confiabilidad. Una condición importante en WSN es que los nodos deben tener características de movilidad, trabajar en condiciones extremas que pueden cambiar en cualquier momento esto significa que los protocolos y los algoritmos de una red de sensores deben poseer características de auto-organización.

¹⁰ Fuente (16)

Por ejemplo en aplicaciones militares los dispositivos sensores deben ser de fácil despliegue y deben tener la posibilidad de conectarse a la red en modo ad hoc en un ambiente altamente dinámico.

En la figura. 10 se propone una abstracción en un modelo de capas para una red de sensores, todas las capas deben proveer funciones particulares para redes de sensores como son: administración de potencia, movilidad y programación de tareas.

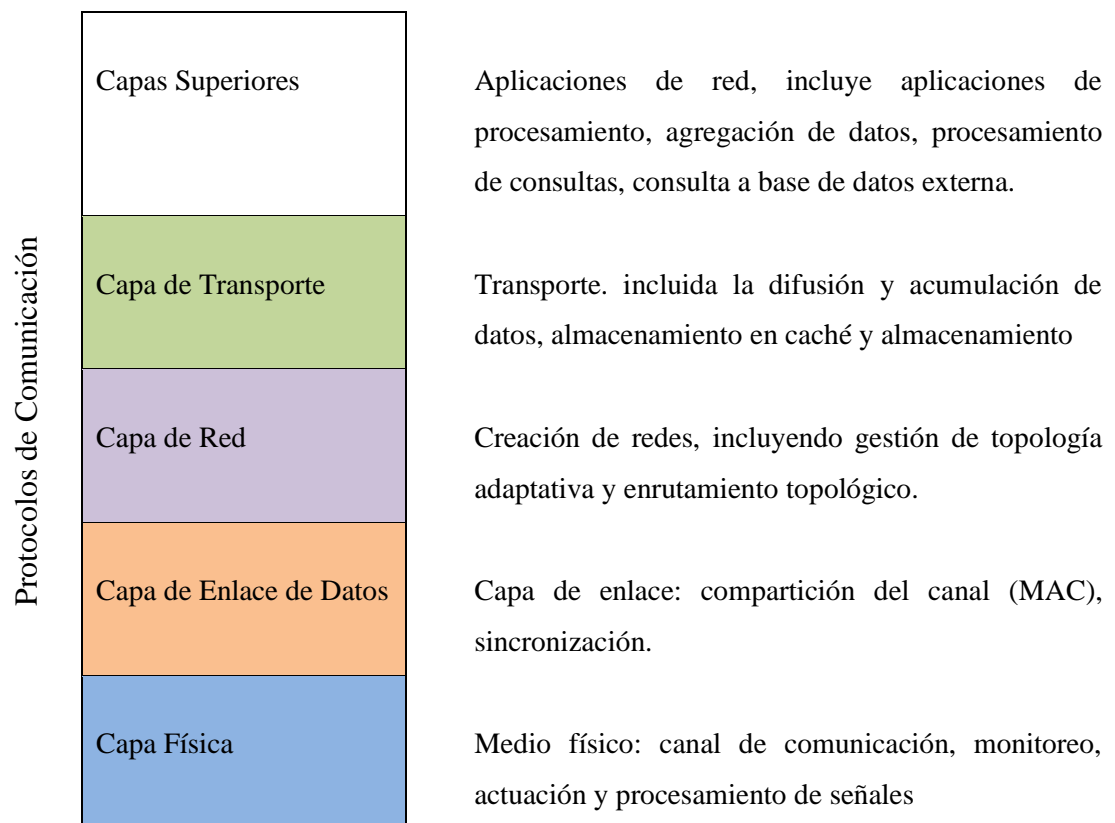


Figura 10 Pila de Protocolos genérica para Redes de Sensores

2.5.1 IEEE 802.11

IEEE.11 es un conjunto de especificaciones de capa física y capa enlace para la implementación de redes inalámbricas de área local (WLAN) en las bandas de

frecuencia 2.4GHz (ISM band) y 5.8 GHz (UNII band). El rango de cobertura es hasta 100 metros.

IEEE 802.11 tiene dos modos de operación:

- Función de punto de coordinación (PCF)

La comunicación a través de los dispositivos es a través de una entidad central llamada punto de acceso (AP) o una estación base.

- Función de coordinación distribuida (DCF)

Los dispositivos se comunican directamente unos con otros.

IEEE 802.11 se basa en CSMA/CA (carrier-sense multiple access with collision avoidance) cuyo objetivo es proveer un equitativo acceso al medio con soporte de alto rendimiento y movilidad. Sin embargo debido a que los dispositivos emplean una gran cantidad de tiempo escuchando el medio se incurre en costos de tiempo y energía, para solucionar este problema de consumo de energía IEEE 802.11 ofrece un modo de ahorro de energía para los dispositivos en modo PCF.

Protocolo	Características Principales
802.11	Se aplica a WLANs, provee 1 o 2 Mbps en la banda de 2.4 GHz utilizando salto de frecuencia en espectro ensanchado (FHSS) o Secuencia Directa en espectro ensanchado (DSSS).
802.11b	También referido como 802.11 de alta velocidad o Wi-Fi Es una extensión de 802.11 que provee 11 Mbps en la banda de 2.4 GHz, usa solamente DSSS
802.11g	Se aplica a WLANs y es usado para transmisión en cortas distancias de hasta 54 Mbps en la banda de 2.4 GHz.
802.11n	Además de las características de los estándar 802.11 se añade múltiple-input múltiple output (MIMO). Las antenas transmisoras y receptoras adicionales permiten mayores velocidades de transmisión de datos a través de

	multiplexado espacial. La velocidad real podría ser 100 Mbps (hasta 250 Mbps en capa física).
--	--

Tabla 4 Protocolo 802.11

2.5.2 IEEE 802.15.4

IEEE 802.15.4 es un estándar abierto cuyo principal objetivo es dar soporte a la conectividad inalámbrica de un vasto número de aplicaciones industriales, médicas y de hogar, estas aplicaciones requieren bajos costos, larga vida de la batería (meses y años), instalación automática y semiautomática. El estándar IEEE 802.15.4 soporta todos estos requerimientos.

El estándar IEEE 802.15.4 ha sido adoptado por la alianza de tecnologías para redes inalámbricas de área personal ZigBee, que es una asociación de cientos de miembros alrededor del mundo que trabajan con el objetivo de lograr un estándar global abierto para redes de sensores.

Principales características

- Tasas de transmisión de datos de 20Kbps, 40Kbps, 250 Kbps.
- Rango de cobertura de 10 a 30 metros.
- Modo de operación en estrella o per to per.
- Soporte para dispositivos de bajo tiempo de acceso.
- Protocolo de acceso al canal CSMA-CA.
- Provee un acceso dinámico a los dispositivos.
- Protocolo totalmente coloquial para fiabilidad de transferencia de información.

- Bandas de frecuencia de operación:
 - 16 canales en la banda ISM de 2.4 GHz.
 - 10 canales en la banda ISM de 915 MHz.
 - 1 canal en la banda europea de 868 MHz.

2.5.3 Protocolos de capa Red

La responsabilidad fundamental de la Capa red es encontrar caminos desde las fuentes de datos a los gateways o estaciones base.

En el modelo de comunicación de salto único todos los sensores son del tipo RFD y están en posibilidad de comunicarse directamente con el Gateway aquí los datos en un solo salto alcanzan su destino y es el modelo más simple de manejar.

En el modelo de comunicación de salto múltiple, que es el que se usa en proyectos reales, la tarea crítica de la capa red en cada uno de los nodos es identificar un camino desde el sensor al Gateway a través de los múltiples nodos FFD que están actuando como nodos de retransmisión.

Debido a las características únicas de WSN como son la falta de recursos (procesamiento, almacenamiento, ancho de banda, energía) o la falta de fiabilidad del medio inalámbrico (cambios en la topología debido a fallas en los nodos, etc) se requiere de soluciones de enrutamiento adaptativas y flexibles. Además, a diferencia de los protocolos de enrutamiento tradicionales para redes cableadas, los protocolos para redes de sensores podrían no ser confiables en esquemas de direccionamiento globales como lo son las direcciones IP en Internet.

Ejemplos de Protocolos de capa red para WSN son:

- Sensor Protocols for Information via Negotiation (SPIN).
- Low Energy Adaptive Clustering Hierarchy (LEACH)
- Greedy Perimeter Stateless Routing for Wireless (GPSR)
- Stateless Nondeterministic Geographic Forwarding (SGNF), etc.

2.5.4 Protocolos de capa Transporte

Todos los nodos FFD tienen al menos las tres primeras capas (Física, Enlace y Red) del modelo OSI. La capa Transporte y las capas superiores están presentes exclusivamente en los puntos finales o los servidores y tienen las funciones de un protocolo de extremo a extremo.

Aunque TCP y UDP son protocolos de transporte populares y ampliamente desplegados en Internet, no son una buena opción para WSN, debido principalmente a que estos están diseñados para el transporte de grandes volúmenes de datos lo cual no es el caso de WSN.

Ejemplos de protocolos de capa Transporte diseñados para WSN son:

- ConGESTIÓN Detection and Avoidance (CODA)
- Event-to-Sink Reliable Transport (ESRT)
- Reliable Multisegment Transport (RMST), etc

2.6 Modelos de Gestión de Parquaderos

2.6.1 Tipos de modelos de gestión de parquaderos.

Comercialmente hablando se tiene tres modelos de gestión de parquaderos

- Modelo Básico

Consta de un control de ingreso y salida mediante ticket el mismo que se entrega a la entrada del auto al parquadero y para poder salir el usuario tiene que cancelar el tiempo de uso en un punto de pago. No tiene monitoreo de las plazas de parqueo.

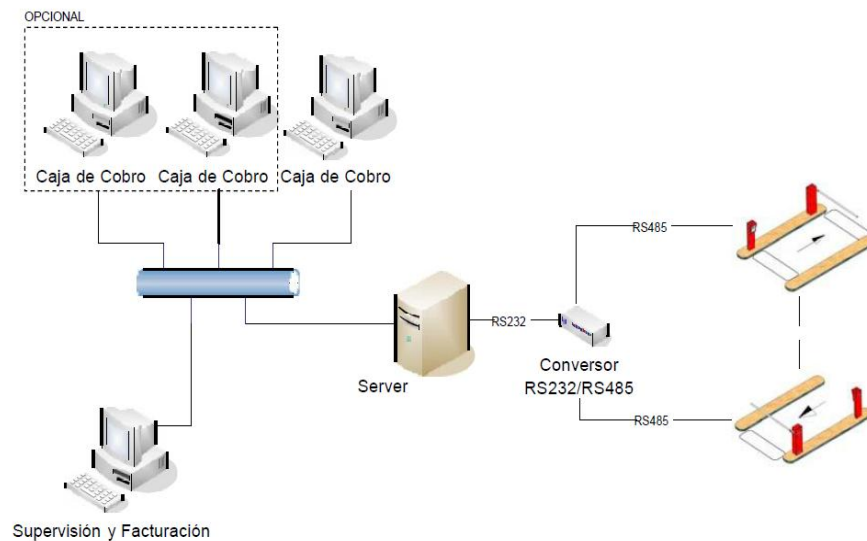


Figura 11 Modelo Básico de gestión de parquaderos¹¹

- Modelo de gestión con señalización de plazas de parqueo

¹¹ Fuente (17)

Maneja la entrada y salida de la misma forma que el anterior es decir mediante ticket, este añade un sistema de monitoreo y señalización de las plazas de parqueo y sistema de señalética electrónica para guiar al usuario.

Para el monitoreo generalmente se usa sensores ultrasónicos y para la señalización se usa indicadores led sobre el sitio de estacionamiento color rojo para ocupado, verde para libre y azul para personas con capacidades especiales. También dispone de pantallas informativas con el número de plazas libres para guiar a los usuarios. La solución comercial que se presenta en el mercado es una solución cableada que usa protocolo de comunicaciones CAN bus (Controller Area Network).

- Modelo de Gestión con cámaras IP y reconocimiento de Placa.

En cada plaza de estacionamiento se instala una cámara IP la cual captura imágenes en tiempo real. Esta información permite en primera instancia determinar la ocupación o no de la plaza y mediante indicadores LED realizar la señalización visual correspondiente.

Las imágenes se transmiten al servidor via red, este las procesa obteniéndose la información de estado de ocupación, placa del vehículo, imagen del vehículo, hora de llegada y salida, tiempo de ocupación para utilizarlos en administración, estadística y monitoreo.

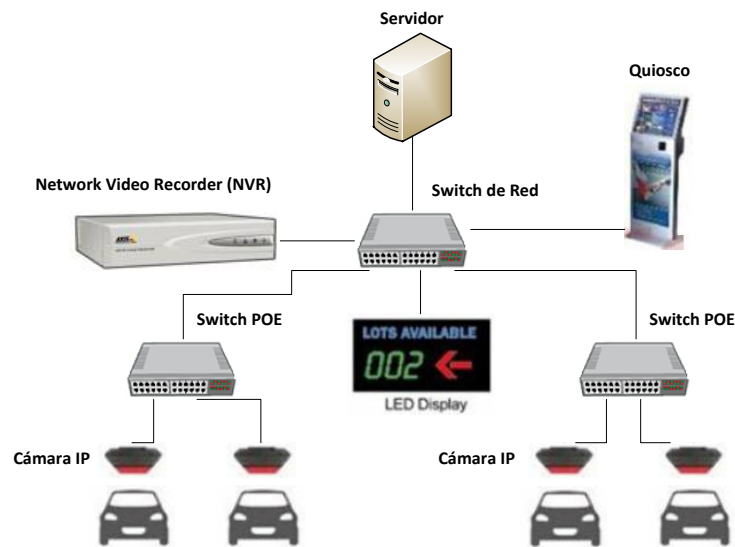


Figura 12 Sistema de Gestión con cámaras IP y reconocimiento de placa¹²

2.6.2 Sistemas de gestión automáticos actualmente en funcionamiento en la ciudad de Quito.

Al momento de realización de este proyecto de graduación se tiene que el modelo de gestión de parqueaderos más avanzado se encuentra en el Aeropuerto de Tababela allí se tiene un monitoreo y administración por medio de cámaras IP, no se usa señalética electrónica ya que siempre existe disponibilidad de plazas.

En el centro comercial CORAL HIPERMERCADOS localizado en el norte de la ciudad de Quito en la Av. 6 de Diciembre 5297, se dispone de un sistema de parqueaderos con monitoreo de las plazas de parqueo y señalética electrónica para guiar a los usuarios.

¹² Fuente (18)



Figura 13 Señalética electrónica para guía de usuarios de los parqueaderos

Los LED indicadores tienen 3 colores, verde que indica plaza libre, rojo para plaza ocupada y azul para plaza reservada para usuarios con capacidades especiales.



Figura 14 LEDs indicadores de ocupación en las plazas de parqueo

3. CAPÍTULO 3: DISEÑO DE LA RED

Se ha escogido como metodología de diseño la propuesta en Top-Down Network Design (Diseño Descendente de Redes) de Priscilla Oppenheimer. (8)

3.1 Identificación de las necesidades y Objetivos del cliente

La Visión de la Pontificia Universidad Católica del Ecuador textualmente señala:

“En los próximos años, la PUCE, fundamentada en el pensamiento y en las directrices pedagógicas ignacianas, se consolidará como un sistema nacional integrado competitivo y autosostenible, con infraestructura tecnológica de vanguardia

Será reconocida por su gestión ética en servicio de la comunidad, y por su estructura académica moderna para la formación de profesionales con responsabilidad social.

Será también reconocida por los resultados de la investigación científica desarrollada en sus unidades académicas, por realizar su gestión con el apoyo de un sistema técnico, innovador y efectivo, con procesos eficientes y recursos humanos capacitados y comprometidos con la misión institucional.” (9)

Como se indica en el primer párrafo se plantea a la PUCEQ como un sistema nacional integrado competitivo y autosostenible, con infraestructura tecnológica de vanguardia, por lo tanto la innovación para mejora en los servicios a la comunidad universitaria es un objetivo prioritario.

En el segundo párrafo al indicar que su gestión es de servicio a la comunidad, se puede extrapolar que el sistema de gestión de parqueaderos debe ser un sistema moderno autosustentable que preste un servicio de calidad a los miembros de la comunidad universitaria.

Con el objetivo de tener un conocimiento de la situación actual del modelo de gestión de parqueaderos se entrevistó a un funcionario de la administración de parqueaderos de la PUCEQ. A continuación se presentan las preguntas y las respectivas respuestas:

- **P.** ¿Cuántas plazas de parqueo existen en las instalaciones de la PUCE?

R. El total de plazas destinadas a parqueaderos es de 800, distribuidas en todo el campus y en la iglesia el Girón.

- **P.** ¿Cuál es la demanda existente de parqueaderos?

R.

Población	Valores aproximados demandados
Estudiantes	3000
Docentes	900
Administrativos	400
TOTAL	4300

Tabla 5 Demanda actual de parqueaderos instalaciones PUCEQ

- **P.** ¿En qué forma se administra el uso de las plazas de parqueo existentes?

R. Los docentes y administrativos tienen derecho al uso de plazas de parqueo solo con la firma laboral.

Los estudiantes que es la población más grande y con mayor demanda envían sus solicitudes y entran en un sorteo para acceder al uso de parqueadero.

La forma utilizada para ampliar el número de cupos disponibles es la asignación de stickers para los usuarios de los parqueaderos:

DOCENTES	TP	Tiempo parcial
	MT	Medio Tiempo
	TC	Tiempo Completo
ESTUDIANTES	N	Nocturno, a partir de las 18hoo
	G	Parqueadero Girón
	PG	Posgrado, solo fines de semana

Tabla 6 Categorización de usuarios

El personal administrativo se encuentra dentro de las instalaciones desde las 8:00H hasta las 16:00H. Esto libera las plazas para el uso de estudiantes en modalidad nocturna.

Este tipo de gestión requiere que el personal encargado de controlar los accesos revise el sticker antes de permitir el ingreso a las instalaciones.

Con este modelo de gestión se logra optimizar el uso de las 800 plazas de parqueo para atender a un máximo de 2050 usuarios:

Estudiantes	Abierto	250
	Nocturno	100
	Girón	100
	Posgrado	300
Docentes		900
Administrativos		400
TOTAL		2050

Tabla 7 Utilización de las plazas de parqueo

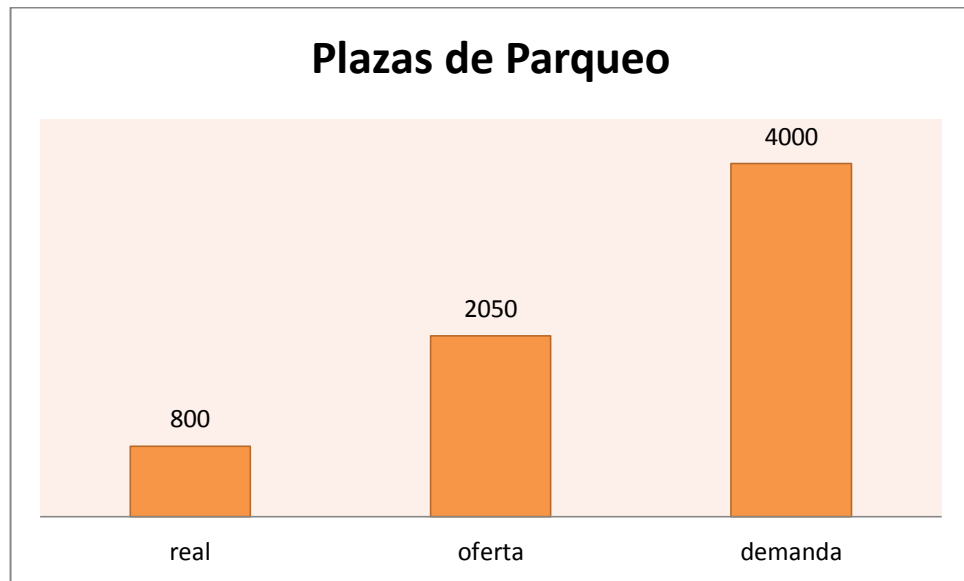


Figura 15 Gráfico comparativo entre oferta y demanda de parqueaderos

- **P.** ¿Cuáles son los problemas de la forma actual de administración de parqueaderos en la PUCE?

R.

- Demasiada demanda de parqueaderos para tan poca oferta

- Se basa en el nivel de compromiso de la comunidad universitaria para hacer buen uso de este recurso manteniendo el respeto y la consideración por el semejante.

- Depende en gran parte del personal encargado de permitir el ingreso a las instalaciones.

- **P.** ¿Se ha pensado en forma alternativa de administración de los parqueaderos?

R. Por el momento no se ha pensado un cambio al modelo de gestión de parqueaderos

Después de esta entrevista con el personal encargado de la gestión de parqueaderos se concluye que debido a la poca oferta de parqueaderos no es posible una mayor optimización de su uso que la existente con stickers.

En este caso lo que se plantea es el diseño de un sistema de gestión que complemente al existente. Aprovechando que en cada plaza de parqueo se tendría instalado un nodo sensor se puede saber en tiempo real cuántas y qué plazas están ocupadas a lo largo del tiempo, pudiéndose manejar estadísticas y patrones de uso, esto serviría como base para aplicar políticas dinámicas de distribución de las plazas de parqueo.

Al tener un conocimiento de cuáles son los usuarios que están ocupando las plazas y el tiempo que las están usando permite hacer una asignación de cupos de horas a cada usuario.

También es posible saber en tiempo real cuantos estudiantes, cuantos administrativos y cuantos docentes se encuentran en un momento dado parqueados dentro de las instalaciones de la universidad, esto es útil para analizar patrones de comportamiento de los usuarios y aplicar políticas al respecto.

3.2 Caracterización de la red existente

Debido a que la red de sensores inalámbricos a diseñarse tiene como opción la interconexión con la red existente es necesario conocer ciertos detalles al respecto

para lo cual se entrevistó al funcionario encargado de la Red de la PUCEQ. A continuación se presentan las preguntas y las respectivas respuestas:

- **P.** ¿Al momento es factible la disponibilidad de puntos de acceso inalámbrico en los diversos parqueaderos?

R. Los accesos WiFi están direccionados a la población académica por lo tanto cubren el 100% de las aulas, Bibliotecas y áreas verdes.

Los parqueaderos aparecen en el plano de cobertura como perimetrales y para poder incorporarlos requiere de la instalación de una red exclusiva para este hecho.

Esta red consistiría de Switch, Puntos de Acceso inalámbricos (APs) y cableado por lo tanto al momento no es factible tener cobertura WiFi en el área de parqueaderos.

- **P.** ¿Es factible la instalación de una red inalámbrica independiente dentro de las instalaciones de la PUCE, destinada únicamente a la gestión de parqueaderos?

R. La red actual está en capacidad de recibir el nuevo flujo de datos pero es necesaria la instalación de un Firewall entre la red de gestión de parqueaderos y la red PUCE actual.

No es necesario instalar una red independiente.

- **P.** ¿Entre las alternativas 1 y 2 cuál es la más factible?

R. Siendo la alternativa 1 que la nueva red use infraestructura de la red PUCE, esta es la más factible para posteriores necesidades.

- **P.** ¿Es posible un incremento de 2000 potenciales nuevos usuarios que usen la red en periodos cortos y al mismo tiempo?

R. la red PUCE dispone de AP con capacidad de 200 usuarios en modo navegación por lo tanto para un flujo de datos pequeño como el enviado por los sensores y los smartphones es posible de ser atendida por la red actual.

3.3 Diseño lógico de la Red

Luego de identificar las necesidades y objetivos del “cliente” se establecen los siguientes requerimientos para el sistema a diseñarse:

- El sistema se encargará del monitoreo de los parqueaderos y de los usuarios dentro de la PUCEQ.
- En cada plaza de parqueo se instalará un NODO SENSOR el cual será encargado de monitorear el estado de ocupación y transmitir esta información a la ESTACIÓN BASE (Servidor).
- Los usuarios de los parqueaderos estarán registrados en el Sistema como “clientes” a través de una aplicación en SO Android que se instalará en un teléfono inteligente (con acceso a red WiFi) de propiedad del usuario.
- El ciclo del cliente comienza al inicio de cada periodo académico cuando es aceptado para usar las plazas de parqueo dentro de las instalaciones. El cliente deberá acercarse con su identificación y teléfono inteligente ante el

funcionario asignado por la PUCE quien será el encargado de instalar la aplicación en el teléfono del usuario con la clave UNICA y verificar su activación en el sistema

- Esta clave única solo será conocida por el funcionario de la PUCE y el administrador del sistema y será cambiada al inicio de cada periodo.
- La aplicación instalada en el teléfono le permitirá al cliente conectarse al Servidor dentro del área de cobertura de la Red y conocer el estado de ocupación de las plazas de parqueo.
- Cuando un cliente estacione su auto en la plaza de parqueo, el NODO SENSOR correspondiente notificará al servidor que la plaza está ocupada y el cliente a través de la aplicación en su teléfono confirmará la ocupación. Una vez que el cliente desocupe la plaza de parqueo el NODO SENSOR notificará al servidor la liberación de la plaza.
- En caso de que el usuario ocupe una plaza y no confirme mediante la aplicación, en el servidor se conocerá que plaza está ocupada y no confirmada, un funcionario se dirigirá a la plaza y tomará nota de la placa del usuario y se penalizará al usuario
- En la base de datos del servidor se dispondrá de un historial de la actividad de los clientes y de los parqueaderos.

Para el diseño lógico de la red no se considerará los puntos de ingreso/salida a la institución ya que se pretende integrarlos en posteriores investigaciones.

3.3.1 Elección de la topología de Red

Se ha escogido la topología tipo estrella por las siguientes características:

- Si un NODO SENSOR falla no afecta al resto del sistema.
- No es necesario que los sensores o clientes se comuniquen entre sí.
- Las distancias de transmisión al punto de acceso WiFi son inferiores a 100m por lo que no es necesario hacer múltiples saltos.
- Este tipo de topología simplifica la programación de los nodos.

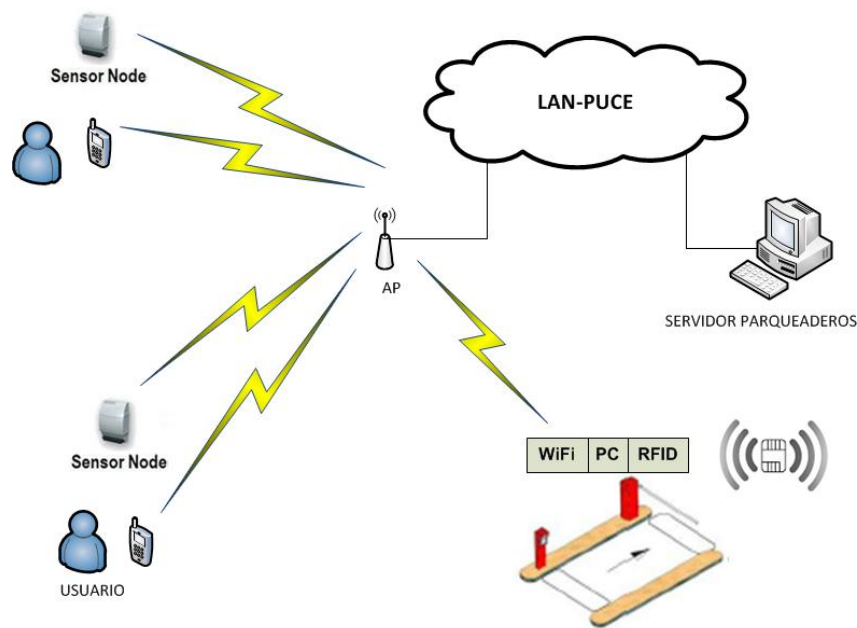


Figura 16 Topología de la red para gestión de parqueaderos

3.3.1.1 Modelos de direccionamiento y de numeración

La Estación Base donde está instalada la aplicación Servidor tiene dirección IP fija, los NODOS SENSORES y teléfonos “clientes” se comunicaran con el servidor a través de esta dirección.

Los nodos sensores y dispositivos inalámbricos no usan IP estáticas sino una asignación dinámica de direcciones IP a través de los punto de Acceso WiFi (Access Point) mediante el protocolo DHCP.

En el servidor cuando se crea una plaza de parqueo, el sistema le asigna un número único HASH y un ID. Los nodos sensores se autentican en el servidor a través de este número único e ID.

Los usuarios “clientes” se autentican en el servidor a través del identificador de dispositivo (Device ID) de su teléfono inteligente.

3.3.2 Elección del protocolo de comunicación a utilizar

Al tener una topología de salto único se elimina las tareas de enrutamiento

En la tabla 8 se presentan los estándares de comunicaciones inalámbricas y sus principales características.

	GPRS/GSM 1xRTT/CDM A	IEEE 802.11 b/g	IEEE 802.15.1	IEEE 802.15.4
Nombre comercial	2.5G/3G	Wi-Fi	Bluetooth	ZigBee
Tipo de red	WAN/MAN	WLAN y puntos de red inalámbrica	PAN y DAN	WSN
Aplicación principal	Amplia cobertura en voz y datos	Aplicaciones empresariales (datos y VoIP)	En lugar de cables para la conexión de periféricos	Monitoreo y Control
Ancho de Banda (Mbps)	0,064 – 0,128+	11 - 54	0,7	0,020-0,25
Rango de transmisión	10000	1 a 1000	1 a 100	1 a 1000

n (m)				
Factores de Diseño	Alcance y calidad de transmisión	Soporte de la empresa, escalabilidad y costo	Costo, facilidad de uso	Confiabilidad, consumo de energía y costo

Tabla 8 Comparación entre los diferentes estándares de comunicaciones inalámbricas

Como se puede ver en la tabla 8, el estándar idóneo para red de sensores es IEEE 802.15.4 conocido comercialmente como ZigBee el cual fue desarrollado específicamente para aplicaciones de redes de sensores inalámbricos, XBee es el nombre que la empresa Digi ha dado a sus módulos de comunicaciones que usan el protocolo IEEE 802.15.4

Los XBee los podemos encontrar en dos series:

- XBee Serie 1

Son los más fáciles de usar, no necesitan ser configurados previamente, se usan para comunicaciones punto a punto.

- XBee Serie 2

Deben ser configurados antes de ser usados. Con ellos se puede armar una red de topología mallada, estos módulos son altamente configurables permitiendo que los nodos de una WSN puedan estar conectados en cualquier topología. El costo aproximado en Estados Unidos es 50 dólares cada módulo de comunicación.

Para poder acoplar una red WSN a una red existente es necesario un Gateway que haga el cambio de protocolos, como ejemplo se tiene el Meshlium de Libelium cuyo

costo es aproximadamente 1000 dólares americanos, se puede encontrar otros Gateway desde 500 dólares.

El presente proyecto pretende proponer una alternativa económica que pueda adaptarse a la red existente por lo que se ha escogido para el montaje del prototipo módulos de comunicaciones con el estándar IEEE 802.11 b/g (WiFi).

Una de las razones de esta decisión fue la aparición en el mercado del módulo WiFi ESP8266, el cual tiene un costo en Estados Unidos menor a 5 dólares.

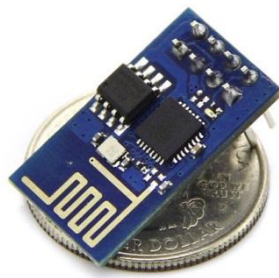


Figura 17 Módulo WiFi ESP8266¹³

	XBee (IEEE 802.15.4)	WiFi (IEEE 802.11 b/g)
Costo módulo de comunicación para cada nodo	50 \$	5 \$
Gateway (XBee to WiFi)	desde 500 \$	ninguno

Tabla 9 Precios para los módulos de comunicaciones

A más del incremento en costos por usar módulos XBee en cada nodo se presenta mayores requerimientos de programación y configuración en el ambiente de red.

¹³ Fuente (19)

3.3.2.1 Estrategias de seguridad de la red

- Cada NODO SENSOR se autentica en la aplicación servidor a través de un número único el cual va cifrado en una petición POST. Si se desea aumentar la seguridad se puede hacer un filtrado por MAC en los puntos de acceso WiFi (Access Point).
- Los usuarios “clientes” se autentican en el servidor a través del identificador de dispositivo (Device ID) de su teléfono inteligente el cual es un código alfanumérico único lo que impide que otro teléfono pueda conectarse al sistema o el mismo teléfono tenga más de un usuario.
- Se sugiere que cada usuario inscriba su teléfono, en este paso se extrae el Device ID del dispositivo y se liga en la base de datos junto al nombre del usuario, se le envía la aplicación por correo y cuando este la instala en su dispositivo dentro de las instalaciones se autentica en la base de datos.
- Para evitar saturación de la red en horas pico y por mayor seguridad se aconseja la instalación de una red independiente solo para el sistema de gestión de parqueaderos.
- En caso de que la red para gestión de parqueaderos se monte sobre la estructura de la red PUCE se debe usar un Firewall entre las dos redes para que el tráfico de la red de sensores no ingrese a la red de la PUCE

3.4 Diseño físico de la Red

Los elementos constitutivos de la red son:

- Nodos Sensores

- Puntos de Acceso WiFi
- Estación Base
- Teléfonos Inteligentes

3.4.1 Nodos Sensores

3.4.1.1 Elección del sensor

El sistema sensor/actuador puede comprender uno o más sensores/actuadores dependiendo de cómo se haya conceptualizado el diseño del sistema de gestión en forma global.

Así por ejemplo si se desea que el sensor detecte la presencia de un vehículo sobre el parqueadero y envíe una foto de la placa al servidor se necesitaran dos sensores uno de presencia y otro de visión artificial en base a cámara CCD. Si a más de esto se desea que según sea la condición el servidor notifique al nodo que debe activar una alarma o visualizador en el parqueadero se necesita un actuador.

Para el caso del presente proyecto el objetivo principal es que tanto los administradores como los usuarios de las plazas de parqueo tengan una información en línea del estado de ocupación para esto se necesita únicamente un sensor de presencia del vehículo sobre la plaza de parqueo y no se ha considerado el uso de actuadores.

En la tabla 10 se hace un análisis comparativo entre los distintos tipos de sensores para detección de vehículos. Se considera que el sensor de ultrasonido es el más óptimo ya que es económico, fácil de instalar, fácil de realizar mantenimiento.

	DETECTORES DE METALES		SENSORES DE PRESENCIA	
	INDUCTIVOS	MAGNETOMETRO	INFRARROJO	ULTRASONIDO
Método de Detección	Cambio en frecuencia de resonancia	Variaciones en el campo magnético terrestre	Haz luminoso	Sonar (eco)
Rango de Detección	centímetros	centímetros	Decenas de metros	20mm a 10m
Costo	bajo	alto	medio	bajo
Material del objeto a detectar	Detecta solamente materiales ferromagnéticos	Detecta solamente materiales ferromagnéticos	El objeto no puede ser brillante o transparente	Detecta todo tipo de materiales
Tamaño del sensor	Bobina grande	mediano	Pequeño	Pequeño
Sensibilidad Ambiental	Es sensible a la interferencia eléctrica	Es necesario realizar un corte en el pavimento para su instalación	Es sensible a la interferencia luminosa	la temperatura y extrema turbulencia del aire pueden afectar su desempeño
Tiempo de respuesta	milisegundos	microsegundos	microsegundos	milisegundos

Tabla 10 Cuadro comparativo entre las diferentes alternativas de sensores

En la figura 18 se muestra alternativas para la instalación física del sensor ultrasónico.

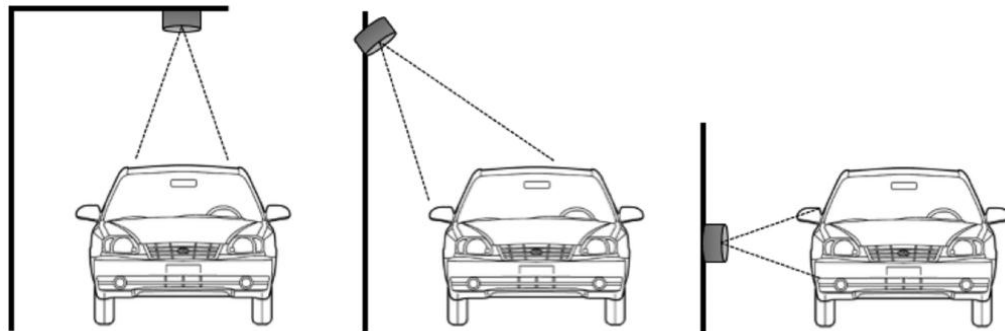


Figura 18 Formas de instalar un sensor ultrasónico para detectar presencia de vehículos¹⁴

3.4.1.2 Alternativas de hardware para la implementación de nodos WSN

En la Tabla 11 se comparan las siguientes plataformas de hardware: Wasmote de Libelium; Wasmote Mote runner de Libelium-IBM; Raspberry Pi®; Arduino.

	Wasmote	Wasmote Mote Runner	Arduino	Raspberry Pi®
Módulos inalámbricos	XBee, GPRS, 3G, WiFi, Bluetooth	Módulos de radio 6LoWPAN	XBee, GPRS, 3G, WiFi, Bluetooth	XBee, GPRS, 3G, WiFi, Bluetooth
Sensores	Si	Si	Si	Si
Programación inalámbrica OTA	Si por software de Libelium	Si por software de IBM	Si (XinoRF)	limitado
SD Card	Si	No	Si	Si
IDE/SDK	Si	Si	Si	Si
Lenguaje de Programación	Similar a C/C++	Java o C#	Similar a C/C++	Python
Modos de ahorro de energía	Si (sleep, deep sleep, hibernate)	Si pero controlados por el S.O	Si (sleep, deep sleep, hibernate) ArduinoPro	No
Tipo de Programación	Secuencial	Reactiva	Secuencial	Multi paradigma
Sistema operativo	No	Si	No	Si

¹⁴ Fuente (20)

Sistema de arranque	preinstalado	Necesita ser instalado	preinstalado	preinstalado
Soporte técnico	Libelium	Libelium e IBM	No	No
Recomendado con 6LoWPAN	No	Si	No	No
Recomendado para proyectos industriales/comerciales	Si (Módulos Plug&Sense)	No	No	No
Recomendado para nuevos usuarios	Si	No	Si	Si
Manejable para usuarios con bajo nivel de programación	Si	No	Si	Si
Costo (dólares USA)	170	300	40	70
Disponibilidad	Inmediata	mediana	Inmediata	Inmediata
Recomendada para IoT	Si	Si	Si	No

Tabla 11 Comparación entre distintas plataformas de hardware usadas para nodos WSN

Se han escogido estas cuatro alternativas de hardware por ser las más conocidas en el medio ecuatoriano, de estas cuatro alternativas waspmote de libelium y waspmote moterunner son las únicas específicamente diseñadas para uso en IoT (Internet de las cosas).

Waspote de libelium tiene dos líneas: una línea para desarrolladores conocida solo como waspmote y una línea comercial con sensores listos para ser instalados en campo conocida como Plug&Sense.

Waspote moterunner se encuentra en etapa de experimentación bajo la dirección de Libelium e IBM, se espera se convierta en el estándar para 6LoWPAN.

En este proyecto se ha descartado Waspote por las siguientes razones:

- Los precios son altos comparando con Arduino.
- La tecnología NO es totalmente abierta.
- No existe mucha información en la red.
- Los sensores son caros y se venden bajo ciertas políticas comerciales.

En el presente proyecto se ha considerado el uso de la plataforma Arduino porque:

- Los precios son bajos.
- La tecnología es totalmente abierta
- Existe mucha información en la red
- Existen variedad de sensores.

Aunque Arduino no fue específicamente diseñado para IoT, existen líneas como XinoRF y Arduino Pro totalmente compatible con Arduino que se proyectan en ese sentido.

Al ser Arduino una plataforma barata y totalmente abierta permite experimentar con diversas combinaciones y configuraciones de hardware, logrando así encontrar una configuración de hardware apropiada a las necesidades.

Una desventaja de Arduino es que los módulos están diseñados netamente para experimentación y no para ser instaladas directamente en procesos industriales o comerciales.

Para que estos módulos de hardware puedan ser comercializados necesitan ser pasados por un proceso de diseño industrial.

Ya que el presente proyecto es a nivel de experimentación o prototipo no se ha puesto cuidado en la presentación o ergonomía del producto.

3.4.2 Características de los Access Point

El protocolo de comunicaciones escogido para los Nodos Sensores es 802.11 b/ g / n (2.4 GHz) comercialmente conocido como WiFi lo cual simplifica la red ya que los teléfonos inteligentes también tienen conexión WiFi, por lo tanto se debe usar Access Point que trabajen con este estándar.

Una segunda característica para los Access Point es que deben ser para instalación en exteriores ya que los nodos generalmente estarán ubicados en lugares a campo descubierto.

Como tercera característica se puede citar el hecho de que los Access Point tengan sistema de alimentación de energía POE (Power Over Ethernet), para minimizar los requerimientos de cableado.

Dos características principales son que el punto de acceso WiFi maneje asignación de direcciones IP mediante DHCP y además tenga opción de filtrado por MAC con el fin de aumentar la seguridad.

3.4.3 Características del equipo para el servidor.

El servidor de una red WSN es una computadora que suele ser más potente que una computadora promedio. Está diseñado específicamente para atender los requerimientos de cientos a miles de Nodos Sensores en cortos intervalos de tiempo. Los servidores poseen el hardware para administrar redes inalámbricas o cableadas por Ethernet.

Debe estar diseñado para manejar cargas de trabajo más grandes en intervalos cortos de tiempo, utilizan software y hardware específico para visualización y monitoreo de los Nodos en tiempo real.

Debe contar con herramientas de administración remota, lo que significa que una persona del área de TI puede controlar el uso y diagnosticar problemas desde otra ubicación. Esto también significa que puede realizar el mantenimiento de rutina como la adición de nuevos usuarios o el cambio de contraseñas, el hardware mínimo necesario consiste de:

- 4 GB de memoria RAM
- 300 GB de disco duro
- 3 GHZ en procesador
- Servidor Web con soporte para PHP 5 y SSL
- Base de datos MySQL 5
- Puertos USB 2.0
- Tarjeta de red Ethernet 10/100
- WLAN, WiMax/Wi-Fi abgn/agn

3.4.4 Características de los smarthphones.

- Debe tener Sistema Operativo Android 4.0 en adelante
- Debe tener conexión a red WiFi
- Permite la instalación de programas de terceros
- Utiliza cualquier interfaz para el ingreso de datos, como por ejemplo teclado QWERTY, pantalla táctil
- Memoria disponible de al menos 10MB para instalación de la aplicación

- Interfaz de comunicación Bluetooth
- Aplicación de correo electrónico
- Aplicación de gestor de archivos.

4. CAPÍTULO 4: DISEÑO Y CONSTRUCCION DEL SISTEMA PROTOTIPO

4.1 Diseño del hardware del NODO SENSOR

Una vez determinado el tipo de sensor, la plataforma de hardware y el protocolo de comunicación, se puede estructurar la configuración del Nodo Sensor a implementarse.

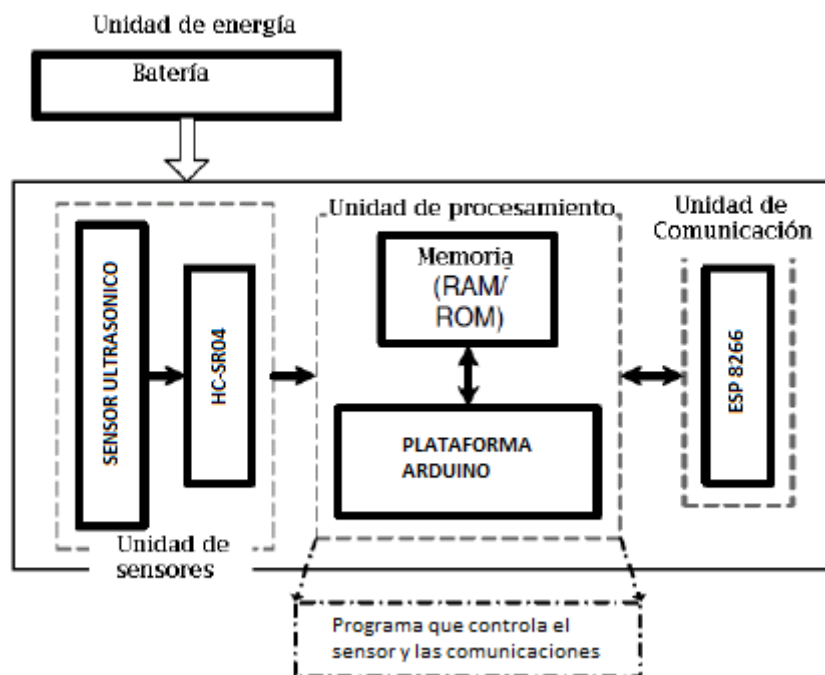


Figura 19 Diagrama de bloques del NODO SENSOR

El primer problema a solucionar es la alimentación de energía de cada uno de los módulos ya que la plataforma Arduino UNO funciona a 7.5 voltios, el módulo de comunicaciones a 3.3 V y el sensor ultrasónico a 5V.

Arduino tiene internamente un regulador que transforma el voltaje de entrada a 5V, por lo tanto esta solucionada la alimentación de energía para el sensor ultrasónico HC-SR04.

La alimentación de energía del módulo ESP8266 requiere el uso de un regulador de 3.3 V externo a la placa Arduino y además “adecuar” los niveles de voltaje de 5V ↔3.3V en la comunicación entre la placa Arduino y el módulo ESP8266.

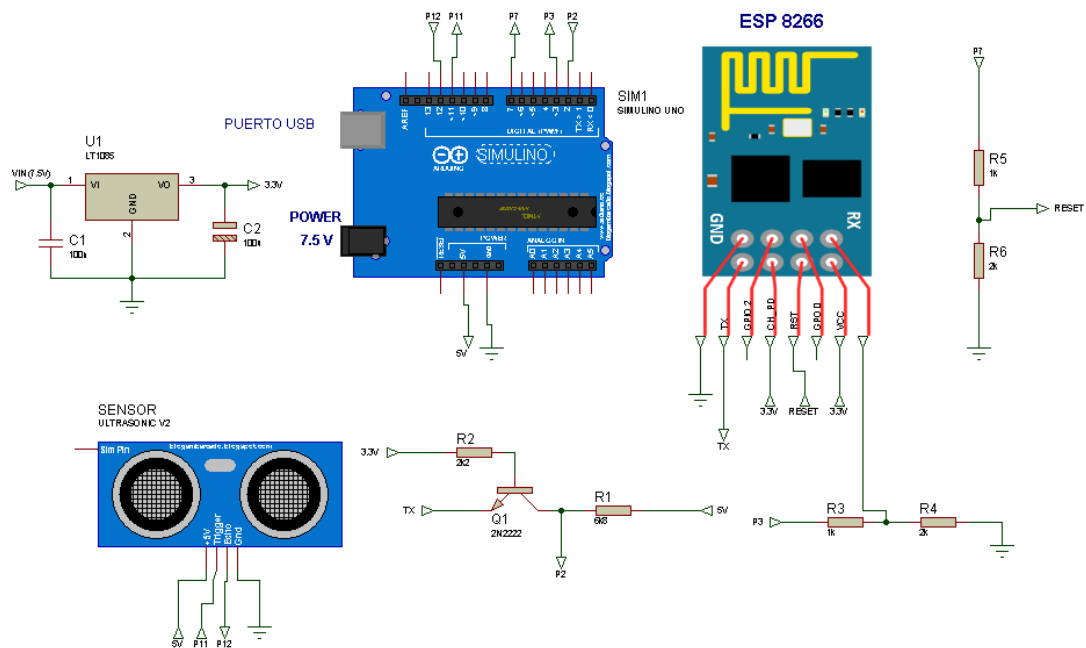


Figura 20 Esquema de conexiones de los componentes del NODO SENSOR

La plataforma Arduino posee una salida regulada de 3.3V pero el nivel de corriente es menor a 50mA. por lo tanto no puede ser usada para alimentar al módulo de comunicación ESP8266 que requiere 3.3V y al menos 100mA. la solución es obtener

los 3.3V a partir de la fuente de alimentación general de 7.5V mediante el circuito regulador LM1086CT-3.3 que es un regulador de 3.3V.

La plataforma Arduino se comunica con los pines de TX y RX del módulo ESP8266 a través de los pines 2 y 3, por esto es necesario que la salida de TX del ESP8266 cambie los niveles 0 – 3.3v a 0 – 5v esto se lo hace a través de la conexión en base común del transistor Q1 como se observa en la figura 20.

Las demás adaptaciones de voltaje se hacen a través de divisores de tensión que cambian los niveles lógicos 0 – 5v a 0 – 3.3v.

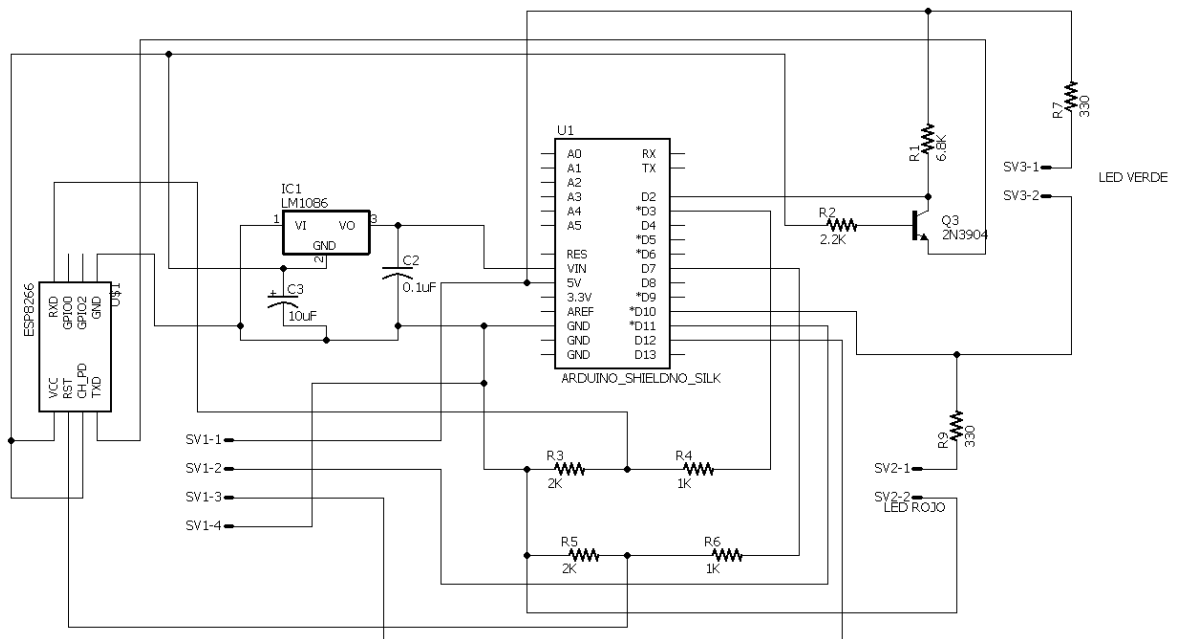


Figura 21 Diagrama Esquemático del hardware del Nodo Sensor

4.2 Programación del Nodo

El ambiente integrado para desarrollo provisto por Arduino (IDE) contiene un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comúnmente usadas y una serie de menús. Este IDE se conecta al hardware de Arduino para cargar programas y comunicarse con él.

Este ambiente de desarrollo provee de la opción Serial Monitor la cual se usa en el programa en el Nodo como una herramienta de depuración en tiempo real.

El lenguaje de programación de Arduino (basado en Wiring) es implementado en C/C++ y consta de una estructura básica de tres partes:

- Área de declaración de librerías y variables globales, se ejecuta una sola vez al inicio.
- Función de Configuración (Setup), se ejecuta una sola vez al inicio.
- Función bucle (Loop) contiene el código que se ejecutará continuamente

En el caso del presente proyecto en la primera parte se declara las librerías, se personaliza los nombres de los puertos de Arduino, y se inicializan las variables globales.

En la segunda parte se configura las funciones de los diferentes puertos, se conecta el módulo ESP8266, se lo inicializa y configura en modo estación. Cabe indicar que la plataforma Arduino se comunica con el Modulo ESP8266 por medio de comandos AT.

La primera y segunda parte solo se ejecutan cuando se enciende el Nodo Sensor.

En la función bucle están todas las instrucciones que permiten de manera repetitiva leer el estado del parqueadero y transmitir este estado al servidor.

En el diagrama de flujo de la figura 22 se presenta la estructura lógica del programa en el Nodo Sensor.

La variable global “val” almacena el estado presente de ocupación del sitio de parqueo y la variable “antVal” almacena el estado anterior. Cuando se resetea el nodo se parte con la condición de plaza desocupada es decir antVal=0 y val=0. El nodo monitorea el estado de ocupación por medio del sensor ultrasónico cada minuto y carga este estado en “val” este valor se compara con “antVal”, si los valores son iguales el nodo entra en reposo hasta la próxima lectura pero si los valores son diferentes se transmite este nuevo estado al servidor donde se actualizan los campos de la base de datos y además se carga en “antVal” este nuevo valor posteriormente el nodo entra en reposo hasta la próxima lectura y así sucesivamente se repite el ciclo.

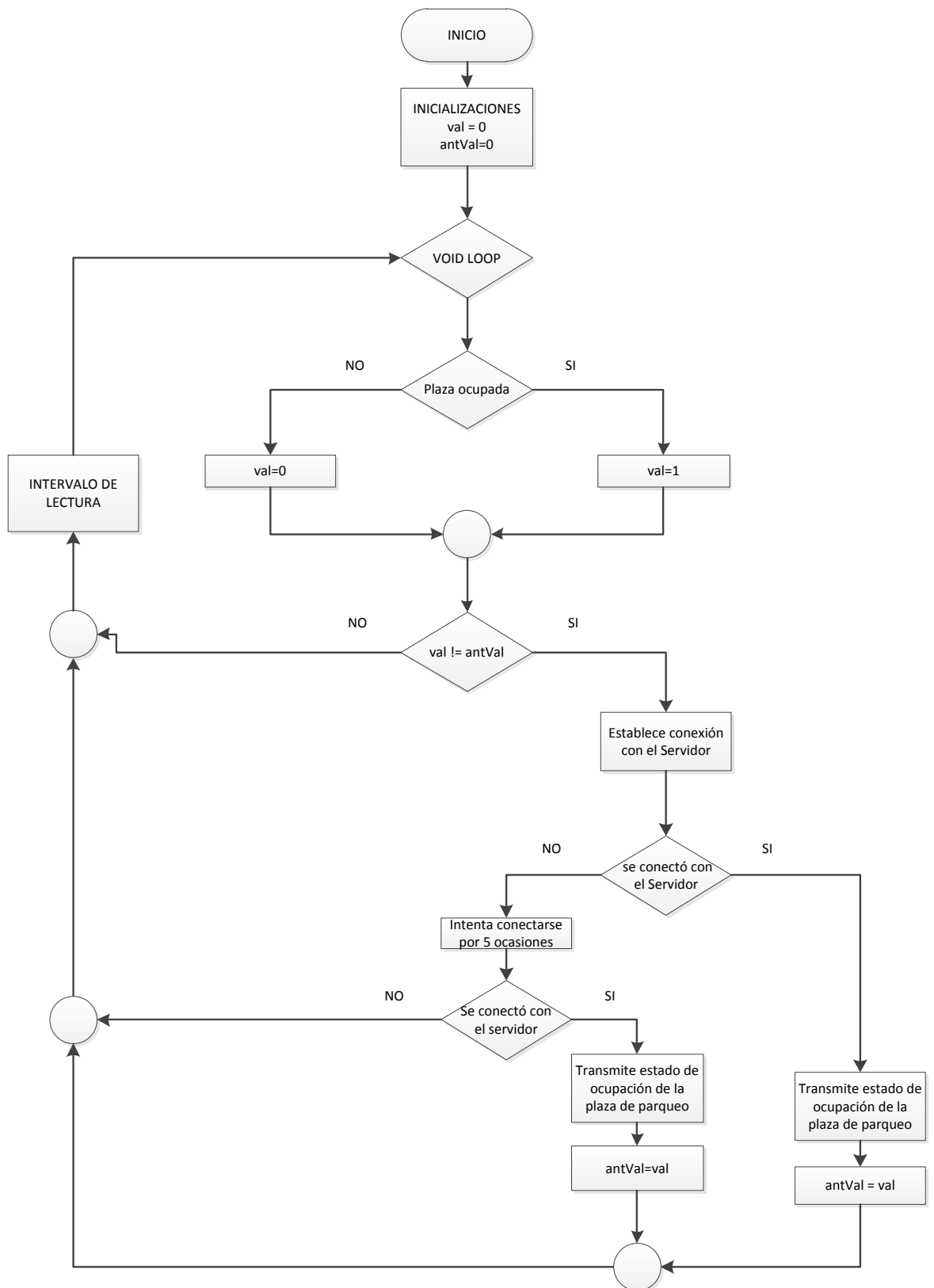


Figura 22 Diagrama de flujo del programa en el Nodo Sensor

Para evitar datos aleatorios del sensor ultrasónico, se ha incorporado una rutina en la cual cada minuto el sensor ultrasónico dispara por 8 veces el pulso ultrasónico en intervalos de 0.5 segundos lo que equivale en total a algo más de 4 segundos, si el número de rebotes que esta dentro del rango de 1.2 metros de distancia es mayor o igual a 4, entonces se acepta como presencia de un auto.

Una forma de solucionar en mayor parte este tipo de problemas es instalando en cada Nodo Sensor 4 dispositivos ultrasónicos (emisor/receptor) en hilera horizontal y se acepta como presencia ó no de vehículo cuando las respuestas de los 4 sensores coincidan. La desventaja de esta alternativa es que el consumo de energía se multiplica por 4 llegando a 100 mA. en cada intervalo de lectura.

4.3 Diseño de la Plataforma del Servidor

Consta de una Aplicación Web desarrollada usando BitNami (Bitrock Inc n.d.) que es un instalador multiplataforma con licencia GPL.

La aplicación Web tiene una Interfaz moderna, responsiva que recibe datos (estado) de los nodos sensores y los presenta en una interfaz web.

Almacena un historial de estados (eliminable) con tiempo (time interval) de estados de los dispositivos.

Permite el registro, edición, eliminación, consulta de clientes y dispositivos arduino (parqueaderos).

Seguridad para verificar el dispositivo (hash) antes de aceptar datos.

Para comunicarse con los dispositivos utiliza la Api RESTful para get/post con JSON.

Visualiza en tiempo real del estado actual de dispositivos (último estado enviado).

Desarrollada con php con el uso de frameworks Laravel, Huga, y Slim para el api RESTful.

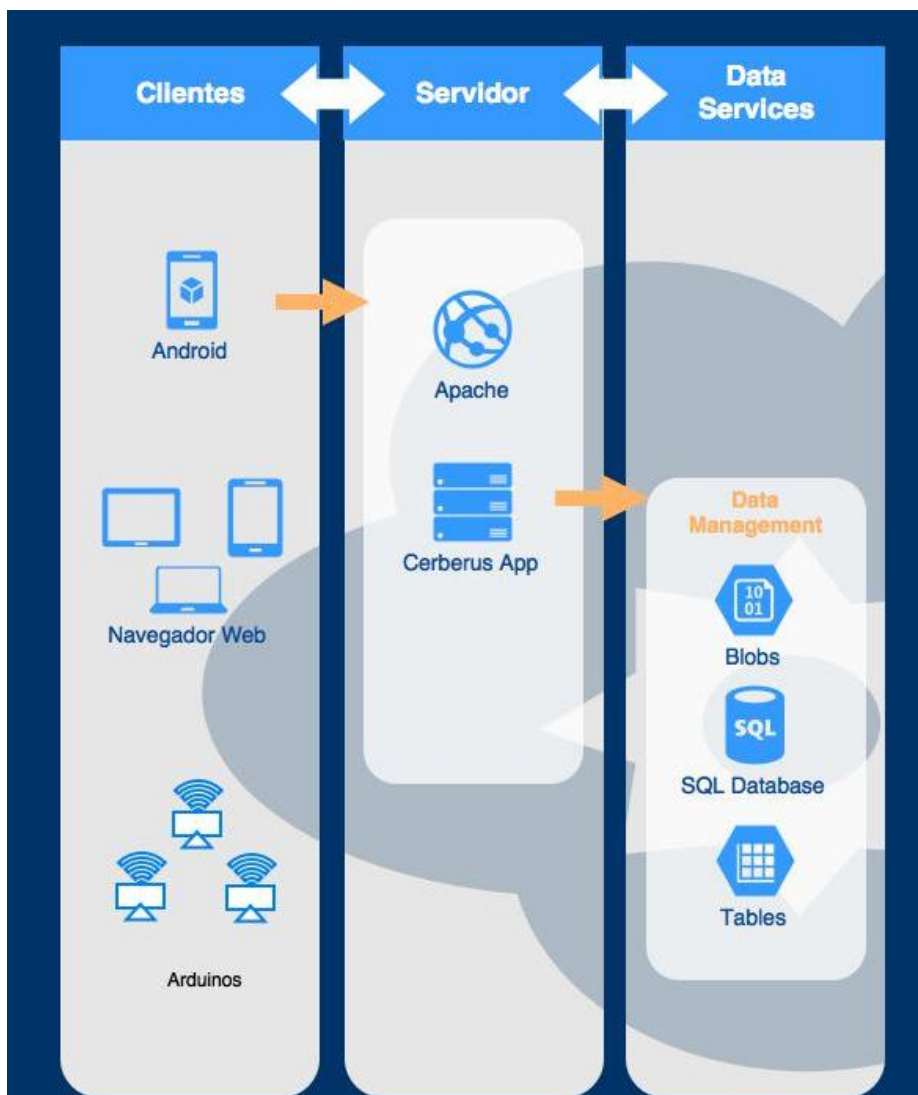


Figura 23 Esquema lógico de la Aplicación Servidor

4.4 Diseño de la Aplicación para Smartphone

Consta de una Aplicación Cliente en Android con material design que permite el registro de un cliente desde el dispositivo y usa el Device ID para su identificación única.

Permite visualizar los parqueaderos libres y ocupados.

Permite que el cliente confirme su estado de parqueadero ocupado mediante el dispositivo. Permite visualizar el tiempo transcurrido del parqueadero que el cliente está ocupando.

Desarrollado en Java usando IDE Android Studio usando el patrón de diseño MVC con gradle y Git

Estructura de folders

Controls: Controles modificados y adaptados.

Helper: Validadores y Clases de conexión y consume de RestFul

Model: Modelos y Entidades para las conexiones y datos de presentación (ListItem, etc)

View: Controladores para manejo de las Vistas

Res: Archivos XML de definición de las Vistas y otros recursos.

4.5 Pruebas

4.5.1 Pruebas en laboratorio

El sistema completo requiere de los siguientes elementos para ser probado:

- Hardware del nodo sensor
- Access Point o Router inalámbrico
- Un computador donde se instalará la aplicación servidor
- Teléfono inteligente

Inicialmente el hardware del nodo sensor se armó en una tabla de proyectos, esto se hace debido a que en etapas iniciales siempre existen muchos cambios.

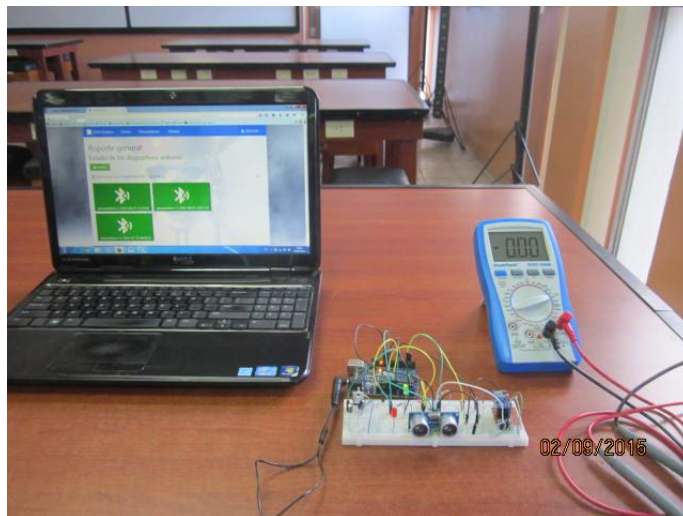


Figura 24 Pruebas en laboratorio

En estas pruebas se pudo definir el tipo de sensor a usar debido a que el sensor en base a bobina de inducción es muy inestable y no paso las pruebas de laboratorio.

Aquí se pudo establecer un método práctico para minimizar los disparos en falso del sensor ultrasónico.

Otra de las situaciones que se resolvió en las pruebas es el formato de transmisión del paquete desde el nodo sensor al servidor para transmitir los datos.

```
POST /cerberus/arduinios/receiveData_action HTTP/1.1\r\n
Host: 192.168.1.2\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length:44

ard_hash=20150614114434&ard_id=8&ard_state=0
```

Figura 25 Formato de transmisión del nodo sensor al servidor

El procedimiento de pruebas consiste en:

- Levantar el servidor en la estación base
- Conectar el teléfono inteligente a la red WiFi
- Activar la aplicación de parqueaderos en el teléfono
- Encender el nodo sensor de modo que el sensor ultrasónico tenga cobertura despejada.
- Conectar el nodo vía cable USB a un computador que tenga el IDE de Arduino.
- Recargar el programa en el Nodo servidor y activar el monitor serial del IDE de Arduino.

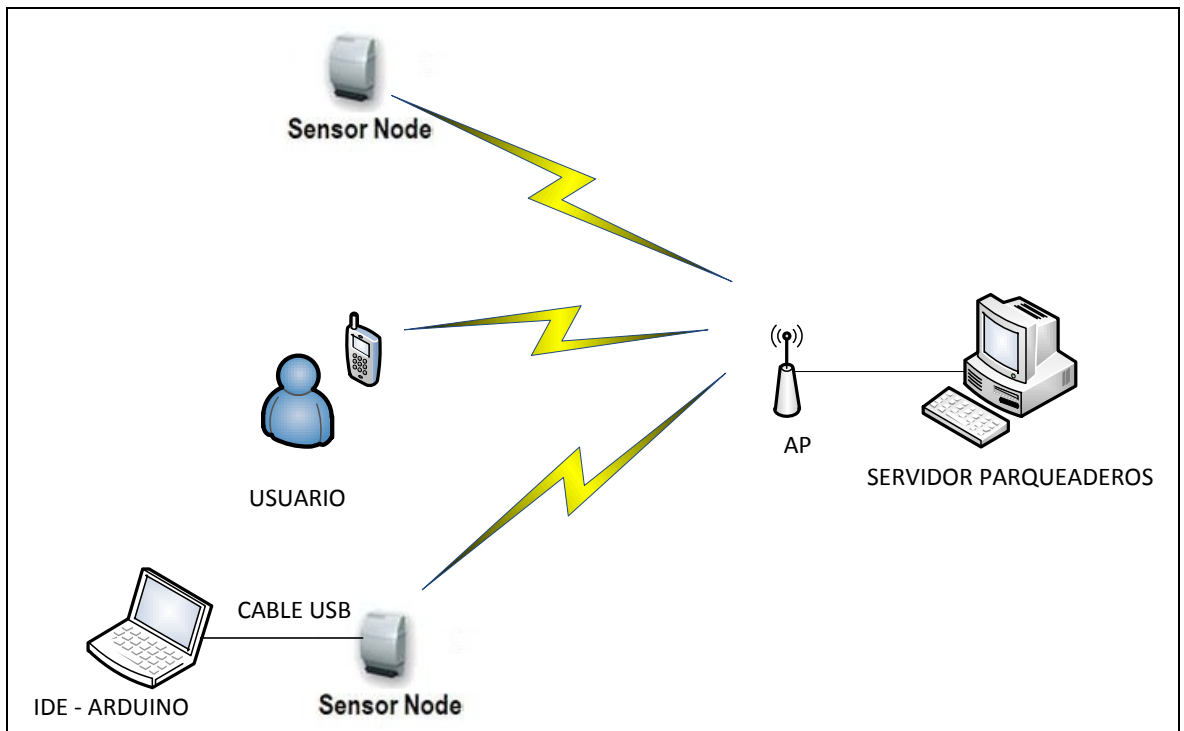


Figura 26 Esquema de pruebas

Luego de realizar estos pasos en el monitor serial del IDE de Arduino se puede visualizar la configuración del módulo WiFi del nodo sensor, el establecimiento de la comunicación con el servidor y las trasmisión de datos. Esto es posible debido a que en el programa grabado en el Nodo sensor se establecen líneas de código que hacen un eco al monitor serial del IDE.

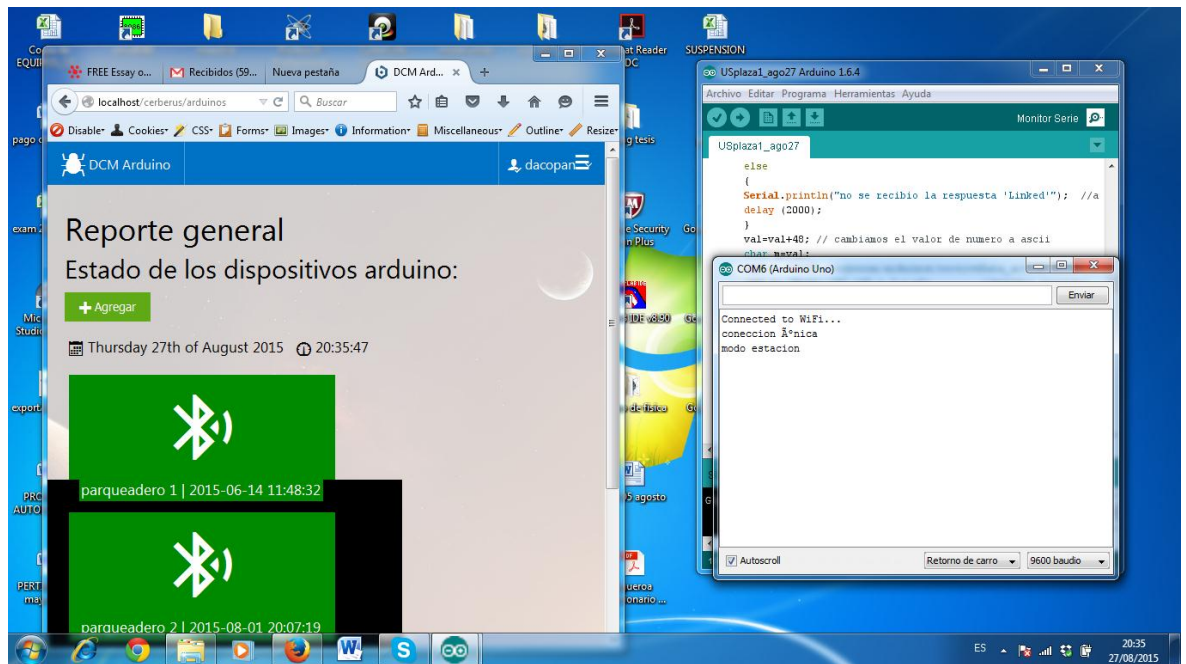


Figura 27 prueba de comunicación Nodo - Servidor

El siguiente paso consiste en “simular” la presencia de un vehículo interponiendo un objeto frente al sensor ultrasónico del nodo. La próxima vez que el nodo sensor efectúe una lectura detectara un cambio en el estado de ocupación, transmitiendo este dato al Servidor produciendo así una actualización en la visualización de los parqueaderos tanto en el servidor como en la aplicación Android del teléfono inteligente.

Finalmente en el teléfono se confirma la ocupación de la plaza de parqueo recibiendo la confirmación del servidor y pudiendo visualizarse la actualización en la base de datos del servidor.

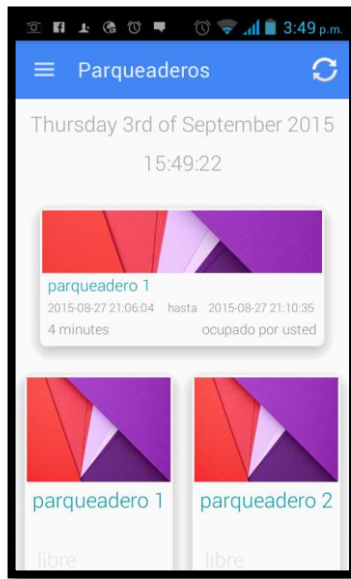


Figura 28 Aplicación Android - Parqueaderos

4.5.2 Pruebas de campo

En base al diagrama esquemático de la figura 22 se diseñó un circuito impreso compatible pin a pin con el hardware de Arduino UNO. Esta placa provee al microcontrolador de Arduino un sistema de comunicaciones WiFi basado en el módulo ESP8266 y un sensor ultrasónico.

Una vez diseñada y construida la placa se armó en un receptáculo de plástico duro el nodo sensor tal como se muestra en la figura 30.

Se realizó pruebas de campo en un área de parqueadero de 50m². En base al esquema de la figura 27 se armó una red de área local con un router inalámbrico con un PIRE de 20dbm, se colocó los nodos sensores en bases provisionales en dos plazas de parqueo como se muestra en la figura 30.



Figura 29 Prueba de campo

4.5.3 Resultados de pruebas de campo

- Existe conectividad entre los elementos de la Red.
- Se cumple a cabalidad con la lógica del diseño que se propuso en la sección 3.3 del presente documento
- Se probó el servidor con 2 nodos sensores y 2 teléfonos inteligentes, existiendo conectividad completa y cumpliéndose a cabalidad la lógica del diseño descrita en la sección 3.3.

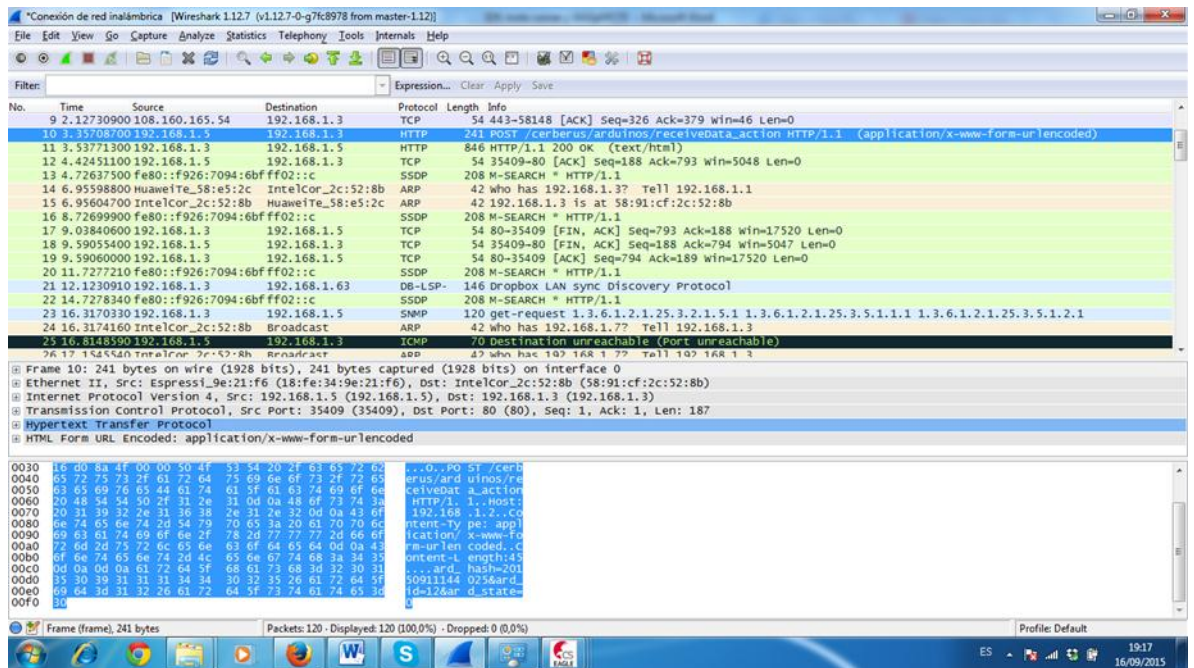


Figura 30 Captura de tráfico en Wireshark

En la captura de tráfico de la figura 31 se puede visualizar el establecimiento de la comunicación entre el Nodo (ip 192.168.1.5) y el Servidor (ip 192.168.1.3), los datos transmitidos y la liberación de la conexión. En el contenido del paquete 10 seleccionado se puede apreciar el formato en que se transmiten los datos desde el nodo al servidor.

En la Figura 32 se muestra como el servidor presenta el historial de ocupación del parqueadero 5 (nodo sensor). En la Figura 33 se muestra el historial de uso de parqueaderos del cliente “aldrin” (teléfono inteligente), se puede cruzar la información y comprobar el correcto funcionamiento del modelo de gestión propuesto.

Desde	Hasta	Tiempo	Estado	Cliente
2015-09-16 16:04:26	2015-09-16 16:05:45	1 minute	1 : ocupado sin confirmar	
2015-09-16 16:05:45	2015-09-16 16:25:24	19 minutes	0 : libre	
2015-09-16 16:25:24	2015-09-16 16:31:15	5 minutes	1 : ocupado sin confirmar	
2015-09-16 16:31:15	2015-09-16 16:49:39	18 minutes	1 : ocupado sin confirmar	
2015-09-16 16:49:39	2015-09-16 17:33:15	43 minutes	0 : libre	
2015-09-16 17:33:15	2015-09-16 18:58:57	1 hour	2 : ocupado confirmado	aldrin
2015-09-16 18:58:57	2015-09-18 10:58:24	1 day	0 : libre	
2015-09-18 10:58:24	2015-09-18 11:34:12	35 minutes	2 : ocupado confirmado	aldrin

Showing 31 to 40 of 41 entries

Cerrar

Figura 31 Historial de ocupación del Parqueadero 5

parqueadero 5	2015-09-18 10:58:24	2015-09-18 11:34:12	35 minutes
parqueadero 5	2015-09-16 17:33:15	2015-09-16 18:58:57	1 hour
parqueadero 5	2015-09-11 22:53:44	2015-09-11 23:01:27	7 minutes
parqueadero 4	2015-09-11 11:57:59	2015-09-11 12:02:30	4 minutes
parqueadero 4	2015-09-05 22:06:38		

Cerrar

Figura 32 Historial del cliente "aldrin"

4.6 Prospectiva de uso

- Gestión de parqueaderos

El bajo volumen de datos, las bajas tasas de transmisión y la propiedad de escalabilidad que poseen las redes hace que el número de nodos y teléfonos inteligentes que se pueden conectar a la red sea grande. Esto hace que el modelo de gestión propuesto tenga grandes posibilidades de comercialización.

- Seguridad antirrobo de vehículos

El usuario del parqueadero posee una aplicación en tiempo real que le permite monitorear si el vehículo se encuentra en el sitio de parqueo. En caso que este apareciera libre se tendría un procedimiento de alerta para que se impida la salida del vehículo de las instalaciones del parqueadero.

- Minería de Datos

Las estadísticas de ocupación de los distintos parqueaderos y las estadísticas de uso de los clientes almacenados en la base de datos posibilitan la aplicación de la teoría de Minería de datos para obtener patrones de uso del recurso y patrones de comportamiento de los usuarios. Esta información ayudaría a optimizar el uso de parqueaderos en una manera científica.

5. CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Debido a la escasa disponibilidad de espacio para parqueaderos en el campus actual de la PUCEQ no es posible realizar mejoras sustanciales a la actual existente con stickers.
- El sistema de gestión diseñado en base a WSN requiere que el usuario confirme el uso de la plaza de parqueo mediante el uso de su teléfono inteligente, de no hacerlo el sitio de parqueo aparece como “ocupado y no confirmado” sin embargo tomando en cuenta que el usuario registro su ingreso en el punto de entrada en el sistema se registra que el usuario ingreso a las instalaciones de la Universidad.
- Al usar asignación de direcciones por DHCP y debido a que los Nodos y los smartphones están configurados para encontrar al servidor en una IP fija, se hace necesario hacer una reserva de dirección por MAC para el servidor en el Access Point (router).
- Utilizando los datos recolectados en la red de sensores se pueden aplicar la teoría de minería de datos para descifrar patrones de uso de un recurso y patrones de comportamiento de los usuarios.
- Este modelo de gestión aumenta la seguridad ante el robo de vehículos ya que si el vehículo sale de la plaza de parqueo este estado se actualiza en el servidor y en los teléfonos inteligentes en tiempo real por lo tanto el dueño puede monitorear si su auto continua o no en la plaza de parqueo.

5.2 Recomendaciones

- Aprovechando que los nodos para el sistema de gestión de parqueaderos se desplegaran en lugares donde siempre tendrán acceso a la red de energía eléctrica se recomienda utilizarla para alimentar los nodos mediante un sistema de alimentación ininterrumpida de energía.
- De ser instalado este sistema en forma real se recomienda realizar una etapa informativa hacia los usuarios en la que se les indique el plus de seguridad que se tiene al confirmar la plaza con el Smartphone.
- Se recomienda que en una investigación futura se diseñe una solución en la cual cada nodo disponga de una cámara que tome foto de la placa y que en el servidor exista un sistema de reconocimiento de caracteres, en este caso el Smartphone sería exclusivamente para información de plazas libres y seguridad.
- Se recomienda usar un sistema de respaldo para los datos del servidor.
- Se recomienda resetear el sistema cada 6 meses.
- Se recomienda que en cada Access point se incorpore la opción de filtraje por MAC, esto aumentara la seguridad del sistema.

ANEXO I PROGRAMA FUENTE DEL NODO SENSOR

El programa del nodo sensor fue editado y compilado en el IDE de Arduino versión 1.6.4 para Windows.

```
/*
programa de prueba para esp8266 EN EL PARKEADERO 5
ard_hash=20150911144025&ard_id=12&ard_state=?
*/
#include <SoftwareSerial.h>
#define SSID "ALDRIN" //nombre del punto de acceso inalámbrico para conectarse
#define PASS "1001662186a" //password de mi red WiFi
#define DST_IP "192.168.1.3" //mi sitio web
#define trigPin 11 // ultrasonido, pin de envío
#define echoPin 12 // ultrasonido, pin de recepción
#define led2 10 // indicador visual de parqueadero ocupado
#define LED 9 // indicador de sistema detenido
#define RESET 7 //utilizamos el pin 7 para hacer el reset del ESP8266 por hardware
desde el arduino
SoftwareSerial Serial1(2,3);
//int loops = 0; //contador para pruebas verificar si es posible eliminar esta variable
int val = 0;
int antVal=0;
//-----
void setup() //inicialización del equipo y conexión al AP
{
  pinMode(RESET,OUTPUT);
  pinMode(trigPin, OUTPUT); //ultrasonido, config pin
  pinMode(echoPin, INPUT); //ultrasonido, config pin
  pinMode(led2, OUTPUT); // pin indicador ocupado, configuración
  pinMode(LED, OUTPUT); // pin indicador sistema detenido, configuración
  reset();
}
```

```

Serial1.begin(9600); // hardware de comunicación serial que conecta el arduino al
módulo ESP8266

Serial.begin(9600); // conexión serial usb que se conecta el arduino al PC
delay(4000); //tiempo de espera hasta que se establezcan los puertos
//if(!cwmode3()) Serial.println("ESP8266 no se configuró en modo 3"); //se envía
este mensaje al monitor serial del Arduino - IDE

boolean wifi_conectado=false; //no se conecta todavía...
for(int i=0;i<5;i++) //intentamos 5 veces conectarnos a la red WiFi
{
  if(connectWiFi()) //verificamos la conexión a la red WiFi
  {
    wifi_conectado = true; //confirmación de conexión a la red WiFi
    break; //
  }
}
if (!wifi_conectado) hang("no esta conectado a la red WiFi");
delay(250);
if(!cipmux0()) hang("falló conección única");
delay(250);
if(!cipmode0()) hang("falló modo estación");
delay(250);
}

void loop()
{
  int flag=0; //esta bandera indica si se pudo enviar datos o no
  //-----
  //SW ULTRASONICO
  //
  long duration, distance;

```

```

int mediana = 0; //este es un contador
//se trasmite 8 veces el pulso ultrasónico y se lee la respuesta
//para evitar pulsos falsos
for (int i=0; i <= 7; i++){
digitalWrite(trigPin, LOW); // Added this line
delayMicroseconds(2); // Added this line
digitalWrite(trigPin, HIGH);
delayMicroseconds(10); // Added this line
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;

if (distance < 120 && distance > 0){
    mediana=mediana+1;
}
else {
    digitalWrite(led2,LOW);
}
delay(500);
}
if (mediana > 3){
    val=1;
    digitalWrite(led2,HIGH);//parquadero ocupado
}
else {
    val=0; //parquadero libre
}
//-----

if (val!=antVal){
    //si ha cambiado el estado de ocupación notificamos al servidor. configuramos

```

esp8266 para transmisión

```
reset(); //reseteamos el modulo ESP8266 cada vez que hagamos una lectura  
delay(5000); //esperamos hasta que se reinicialice
```

```
String cmd = "AT+CIPSTART=\"TCP\",\""; //hacemos el siguiente comando:  
AT+CIPSTART="TCP", "192.168.1.2",80
```

```
cmd += DST_IP;
```

```
cmd += "\",80";
```

```
Serial1.println(cmd); //enviamos el comando al modulo ESP8266
```

```
Serial.print(cmd);
```

```
delay(4000); //esperamos un tiempo por la respuesta 'Linked' esto es importante
```

```
if(Serial1.find("Linked")) //mensaje que devuelve cuando se establece la  
conexión
```

```
{
```

```
Serial.print("Connected to server at "); //mensaje de control de errores
```

```
Serial.println(DST_IP);
```

```
}
```

```
else
```

```
{
```

```
Serial.println("no se recibió la respuesta 'Linked'"); //a veces se recibe esta  
respuesta pero realmente esta conectado
```

```
delay (2000);
```

```
}
```

```
val=val+48; // cambiamos el valor de numero a ascii
```

```
char m=val;
```

```
cmd = "POST /cerberus/arduinios/receiveData_action HTTP/1.1\r\n";
```

```
//construimos el http POST request
```

```
cmd += "Host: 192.168.1.2\r\n"; //nuestro servidor
```

```
cmd += "Content-Type: application/x-www-form-urlencoded\r\n";
```

```
cmd += "Content-Length:45";
```

```
cmd += "\r\n"; // necesitamos insertar dos líneas
```

```

cmd += "\r\n";
cmd += "ard_hash=20150911144025&ard_id=12&ard_state="; //enviamos el
número único como nos reconoce nuestro servidor, el id con el que estamos
mostrados en pantalla
cmd += m; // el estado de ocupación del parqueadero
Serial1.print("AT+CIPSEND="); //arduino configura al ESP8266 para
enviar una cadena de caracteres hacia el servidor
Serial1.println(cmd.length()); //el esp8266 necesita conocer la longitud del
mensaje a enviar

if(Serial1.find(">")) //prompt ofrecido por el esp8266 aceptando la petición
para enviar el mensaje
{
Serial1.println(cmd); //aquí esta nuestro http POST request
Serial.println("empieza transmisión"); // monitoreo desde el monitor serial del
arduino-IDE
Serial.println(cmd); //monitoreo desde el monitor serial del Arduino-IDE
flag=1;
}
else
{
Serial1.println("AT+CIPCLOSE"); //si el ESP8266 no nos permite transmitir
cerramos la conexión
Serial.println("No '>' prompt received after AT+CPISEND");
flag=0;
}

//una vez enviados los datos cerramos la conexión
Serial1.println("AT+CIPCLOSE");

if(Serial1.find("Unlink")) //mensaje que ESP8266 devuelve cuando se

```

desconecta de la red WiFi

```
{
  Serial.println("la Conexión se cerró correctamente...");
}
else
{
  Serial.println("fallo el cierre de la conexión");
}
val=val-48;
if (flag=1){
  antVal= val;//actualizamos el estado de ocupación del parqueadero para la
siguiente lectura
  digitalWrite(LED,HIGH);//se apaga la alerta
}
if (flag=0){
  digitalWrite(LED,HIGH);//se enciende alerta, que no hay conexión con el
servidor
}
}

//intervalo de lectura del estado de ocupación de un minuto
for(int count = 0; count < 60 ; count++)
{
  delay(1000);
}
}
//-----
boolean connectWiFi()
{
  String cmd="AT+CWJAP=\""; //esta función establece conexión entre el ESP8266
y la red WiFi
  cmd+="SSID;
```

```

cmd+="\","";
cmd+=PASS;
cmd+="\","";
Serial1.println(cmd);
delay(5000); //como estamos manejando comandos AT - debemos esperar un
tiempo hasta que se establezca la comunicación
if(Serial1.find("OK")) //la conexión se estableció correctamente
{
  Serial.println("Connected to WiFi...");
  delay(1000);
  char c = Serial1.read(); // read the next character.
  Serial.print(c);
  return true;
}
else
{
  Serial.println("Not connected to WiFi.");
  return false;
}
}
//-----

//Reset por hardware del modulo ESP8266
void reset()
{
  digitalWrite(RESET,LOW);
  digitalWrite(LED,HIGH);
  delay(100);
  digitalWrite(RESET,HIGH);
  digitalWrite(LED,LOW);
}
//-----

```

```

// configuración del ESP8266 en conexión única comando AT+CIPMUX=0
boolean cipmux0()
{
  Serial1.println("AT+CIPMUX=0");
  if (Serial1.find("OK"))
  {
    Serial.println("conexcion única");
    return true;
  }
  else
  {
    return false;
  }
}

//-----
// configuramos el ESP8266 en modo estación comando AT+CIPMODE=0
boolean cipmode0()
{
  Serial1.println("AT+CIPMODE=0");
  if (Serial1.find("OK"))
  {
    Serial.println("modo estacion");
    return true;
  }
  else
  {
    return false;
  }
}

//-----

void hang(String error_String) //para depuración

```

```
{
  Serial.print("proceso detenido... ");
  Serial.println(error_String);
  while(1)
  {
    digitalWrite(LED,HIGH);
    delay(100);
    digitalWrite(LED,LOW);
    delay(100);
  }
}
//-----
void hangreset (String error_String) //para depuración
{
  Serial.print(error_String);
  Serial.println(" - reseteando");
  reset();
}
//-----
```

ANEXO II COMANDOS AT (ESP8266)

Commands	Description	Type	Set/Execute
AT	general test	basic	-
AT+RST	restart the module	basic	-
AT+GMR	check firmware version	basic	-
AT+CWMODE	wifi mode	wifi	AT+CWMODE=<mode>
AT+CWJAP	join the AP	wifi	AT+ CWJAP =<ssid>,<pwd >
AT+CWLAP	list the AP	wifi	AT+CWLAP
AT+CWQAP	quit the AP	wifi	AT+CWQAP
AT+ CWSAP	set the parameters of AP	wifi	AT+CWSAP=<ssid>,<pwd>,<chl>,<ecn>
AT+CWLIF	check join devices' IP	wifi	AT+CWLIF
AT+ CIPSTATUS	get the connection status	TCP/IP	AT+ CIPSTATUS

AT+CIPSTART	set up TCP or UDP connection	TCP/IP	1)single connection (+CIPMUX=0) AT+CIPSTART= <type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>
AT+CIPMODE	set data transmission mode	TCP/IP	AT+CIPMODE=<mode>
AT+CIPSEND	send data	TCP/IP	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND= <id>,<length>
AT+CIPCLOSE	close TCP or UDP connection	TCP/IP	AT+CIPCLOSE=<id> or AT+CIPCLOSE
AT+CIFSR	Get IP address	TCP/IP	AT+CIFSR
AT+ CIPMUX	set mutiple connection	TCP/IP	AT+ CIPMUX=<mode>

AT+ CIPSERVER	set as server	TCP/IP	AT+ CIPSERVER= <mode>[,<port>]
AT+ CIPSTO	Set the server timeout	AT+CIPSTO =<time>	AT+CIPSTO?
+IPD	received data		

Tabla 12 Lista de Comandos AT para ESP8266¹⁵

¹⁵ Fuente (21)

ANEXO III APLICACIÓN SERVIDOR

Consta de una Aplicación Web con Interfaz moderna, responsiva que recibe datos (estado) de dispositivos Arduino y los presenta en una interfaz web.

Almacena un historial de estados (eliminable) con tiempo (time interval) de estados de los dispositivos. La aplicación web permite el registro, edición, eliminación, consulta de clientes y dispositivos Arduino. Seguridad para verificar el dispositivo (hash) antes de aceptar datos.

La comunicación con los dispositivos Arduino es a través de la Api RESTFUL para get/post con JSON.

Visualizador en tiempo real del estado actual de dispositivos (ultimo estado enviado).

Desarrollada con php con el uso de frameworks Laravel, Huga, y Slim para el api RESTFul.

Detalle de carpetas de proyecto modelo vista controlador (MVC)

Htdocs: Configuración de apache para la aplicación

Config: Archivos de configuración: conexión a Base de datos, settings, strings, params, etc.

Controllers: Controladores de navegación (request)

Libs: Auxiliares para render, DB Conecciones. Etc

Model: Modelos para cada una de las entidades con su CRUD y otras acciones.

View: Vistas y Templates.

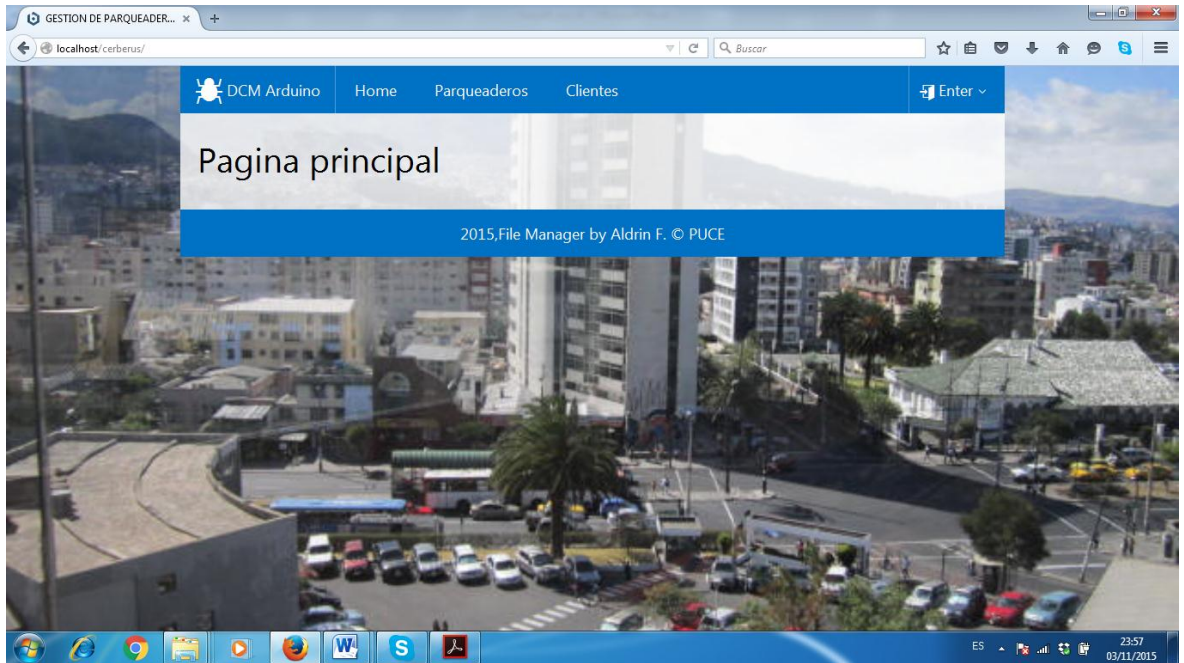


Figura 33 Pantalla principal de la Aplicación Servidor

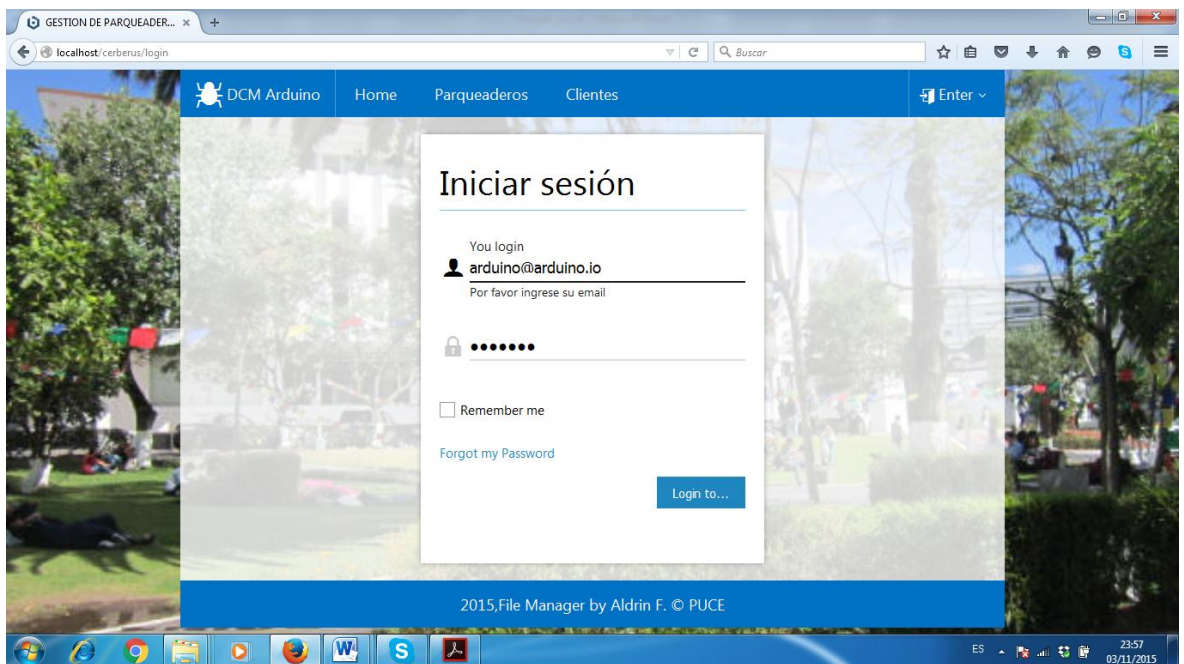


Figura 34 Pantalla de autenticación para ingreso a la aplicación servidor

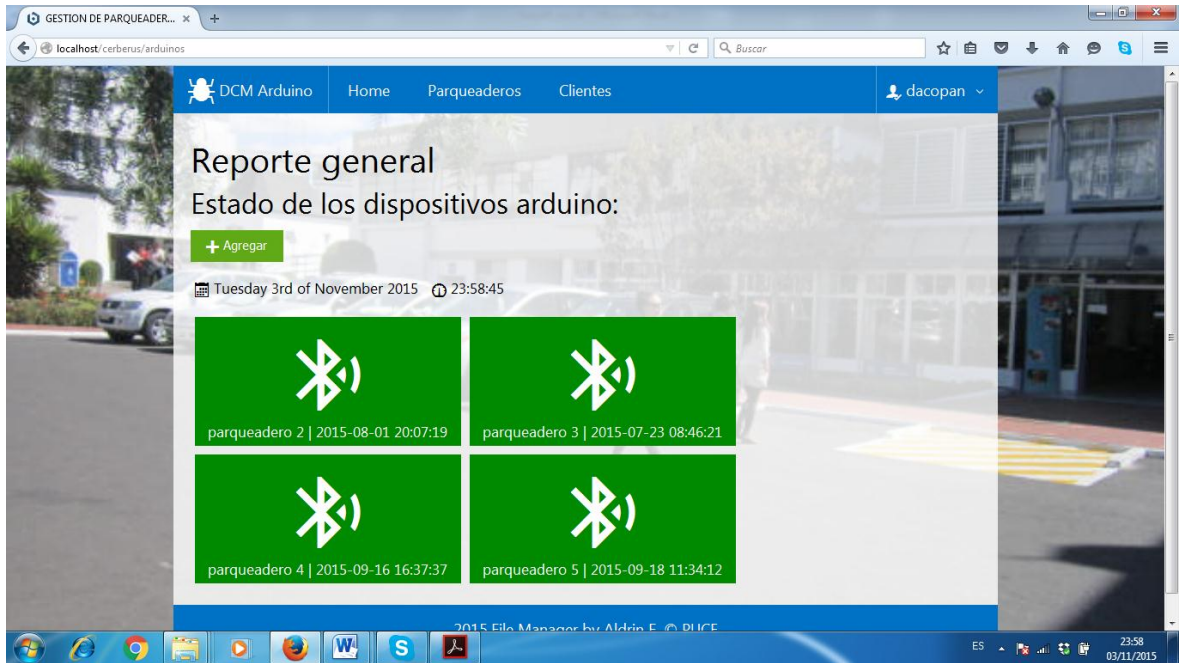


Figura 35 Pantalla de monitoreo de los Nodos Sensores en la Aplicación servidor

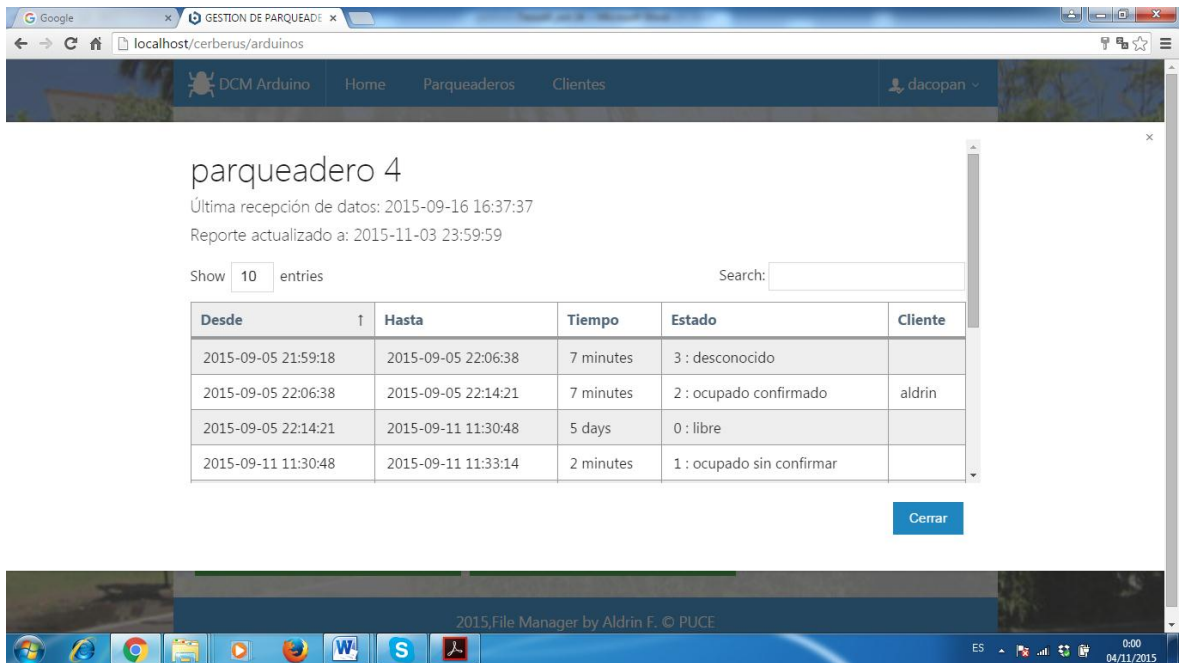


Figura 36 Estadísticas de los Nodos Sensores

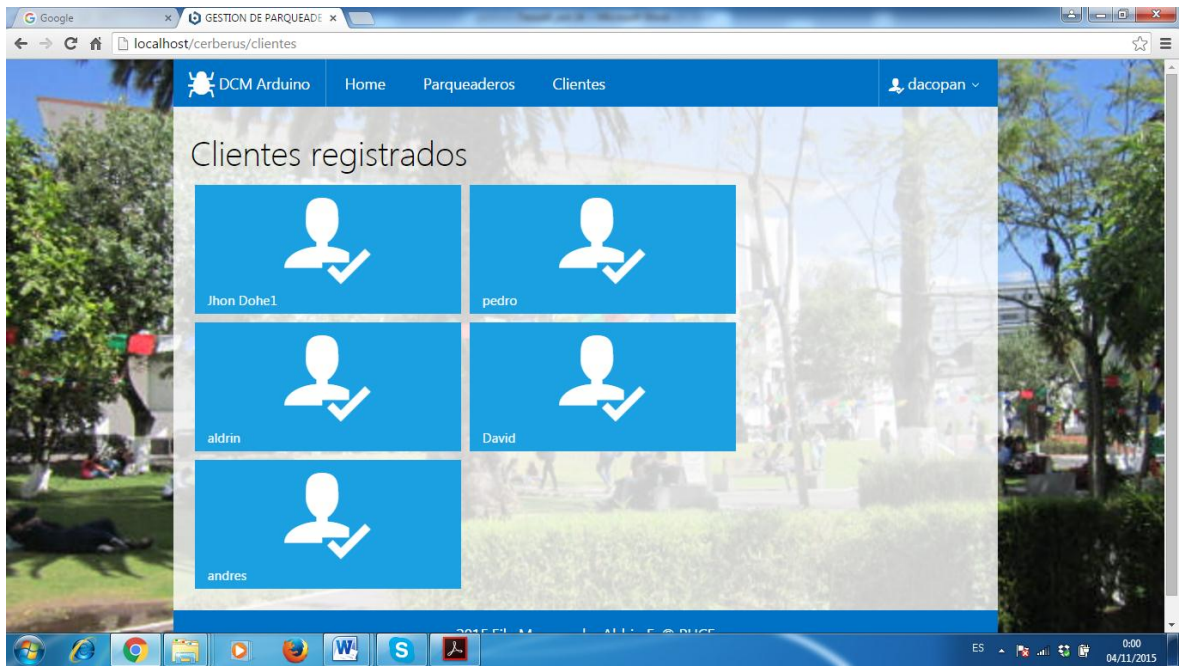


Figura 37 Clientes inscritos en el servidor a través del teléfono inteligente

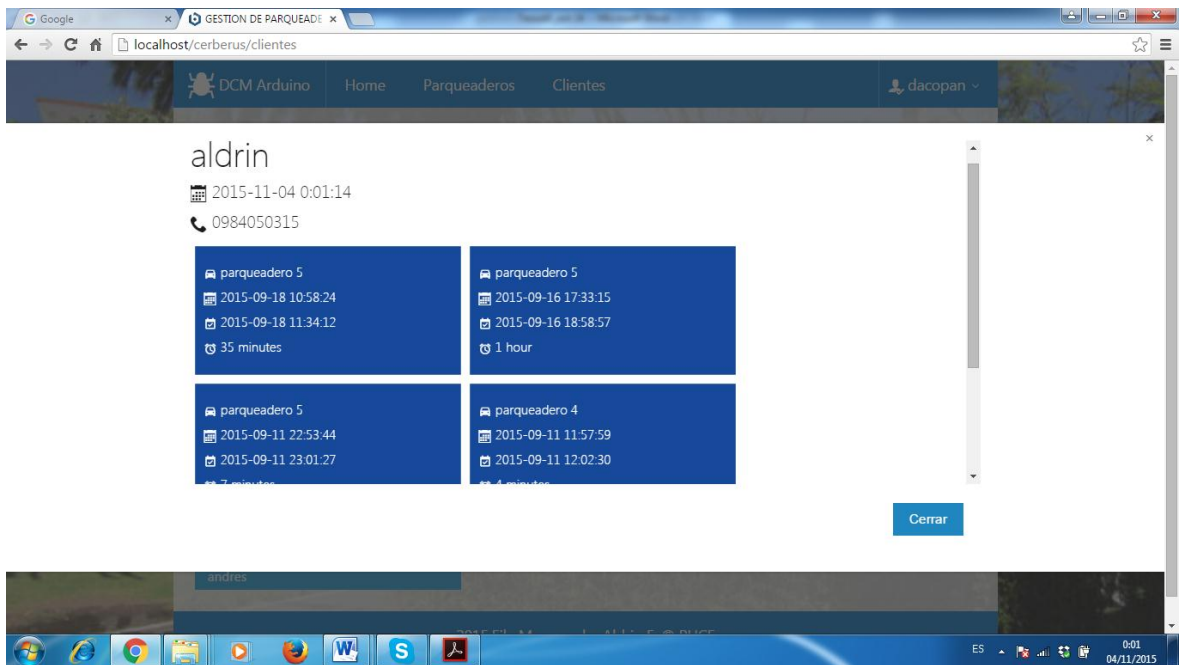


Figura 38 Registro de uso de parqueaderos por los clientes

ANEXO IV ESTIMACION DE COSTOS

- El Sistema esta diseñado para acoplar infraestructura de ingreso/salida RFID con que cuenta la institución.
- Esta concebido su conexión a la red LAN existente’.

Con las consideraciones señaladas los costos del sistema se detallan en las tablas a continuación.

Cantidad	Descripción	Precio Unitario	Total
100	Arduino UNO	22	2200
100	Placas PUCE-WiFi	10	1000
100	Sensores Ultrasónicos	5	500
100	ESP 8266	5	500
100	Baterías recargables	14	1400
100	Costo de fabricación	10	1000
		TOTAL	6600

Tabla 13 Costos de los Nodos Sensores

Cantidad	Descripción	Precio Unitario	Total
3	Puntos de Acceso WiFi (TL-WA5210G)	100	300
		TOTAL	300

Tabla 14 Costos Puntos de Acceso WiFi

Cantidad	Descripción	Precio Unitario	Total
1	Computador core i3	800	800
		TOTAL	800

Tabla 15 Costos equipo servidor

Cantidad	Descripción	Precio Unitario	Total
1	Programación de Módulos	150	150
1	Diseño del software y aplicaciones	500	500
100	Instalaciones de los sensores	30	3000
1	Instalaciones eléctricas	500	500
		TOTAL	4150

Tabla 16 Costos de Programación, configuración e instalación

Total 1	6600
Total 2	300
Total 3	800
Total 4	4150
Total Proyecto	11850

Tabla 17 Costo total estimado

BIBLIOGRAFÍA

1. **El Telégrafo.** <http://www.telegrafo.com.ec>. *http://www.telegrafo.com.ec*. [En línea] 8 de Agosto de 2013. [Citado el: 14 de noviembre de 2014.]
<http://www.telegrafo.com.ec/noticias/quito/item/el-parque-automotor-crece-mas-que-la-poblacion.html>.
2. **Distrito Metropolitano de Quito.** <http://www.quito.gob.ec>. *Secretaría de Movilidad*. [En línea] 2012. [Citado el: 9 de 11 de 2014.] <http://www.quito.gob.ec/secretarias/secretaria-de-movilidad#pol%C3%ADtica>.
3. **El Telégrafo.** 2.614 sitios de parqueo no son suficientes en el Centro Histórico. *www.telegrafo.com.ec*. [En línea] 31 de Agosto de 2012. [Citado el: 9 de 11 de 2014.]
<http://www.telegrafo.com.ec/noticias/quito/item/2614-sitios-de-parqueo-no-son-suficientes-en-el-centro-historico.html>.
4. **Sohraby Kazem, Minoli Daniel, Znati Taieb.** *WIRELESS SENSOR NETWORKS Technology, Protocols, and Applications*. Hoboken, New Jersey : John Wiley & Sons, Inc, 2007.
5. **Balcells Josep, Romeral José Luis.** *Sensores y Actuadores. Autómatas programables*. Barcelona (España) : MARCOMBO, S.A., 2000, pág. 450.
6. **Cecilio, José y Furtado, Pedro.** *Wireless Sensors in Heterogeneous Networked Systems: Configuration and Operation Middleware*. Switzerland : Springer, 2014.
7. **Tipler Paul, Mosca Gene.** *Efecto Hall. Física para la ciencia y la tecnología, Volumen 2*. Barcelona (España) : Reverté S.A, 2005, pág. 1258.
8. **Oppenheimer, Priscilla.** *Top-Down Network Design*. Indianapolis : Cisco Press, 2010.
9. **PUCE.** <http://www.puce.edu.ec/portal/content/Universidad/104?link=oln30.redirect>.
<http://www.puce.edu.ec/portal/content/Universidad/104?link=oln30.redirect>. [En línea] [Citado el: 20 de Marzo de 2015.]
<http://www.puce.edu.ec/portal/content/Universidad/104?link=oln30.redirect>.
10. *Metal Detectors.* **Gavin, Cheeseman.** 1999, Electronics and Beyond.
11. **Sohraby Kazem, Minoli Daniel.** INTRODUCTION AND OVERVIEW OF WIRELESS SENSOR NETWORKS. *Wireless Sensor Networks: Technology, Protocols, and Applications*. Hoboken, New Jersey : John Wiley & Sons, Inc, 2007, pág. 303.
12. **Dargie, Waltenege y Poellabauer, Christian.** *WIRELESS SENSOR NETWORKS - Theory and Practice*. Singapore : Markono, 2010.

13. **shoptropica.** www.shoptropica.com. [En línea] [Citado el: 15 de febrero de 2015.] <http://www.shoptronica.com/interruptor-de-palanca/2268-micro-interruptor-final-de-carrera.html>.
14. **microsonic.** www.microsonic.de. <http://www.microsonic.de/es/Interesting-facts.htm>. [En línea] [Citado el: 21 de febrero de 2015.] <http://www.microsonic.de/es/Interesting-facts.htm>.
15. **Libelium Comunicaciones Distribuidas S.L.** *waspmote technical guide v5.6*. Zaragoza : s.n., 2014.
16. **nkcelectronics.** www.nkcelectronics.com. www.nkcelectronics.com. [En línea] [Citado el: 03 de marzo de 2015.] www.nkcelectronics.com.
17. **transtec.** <http://transtec.com.ec/>. <http://transtec.com.ec/>. [En línea] [Citado el: 7 de marzo de 2015.] <http://transtec.com.ec/>.
18. **ezsistemo.** <http://ezsistemo.com/services/parking-automation-solutions>. <http://ezsistemo.com/services/parking-automation-solutions>. [En línea] [Citado el: 7 de marzo de 2015.] <http://ezsistemo.com/services/parking-automation-solutions>.
19. **aliexpress.** <http://www.aliexpress.com/item/Free-shipping-ESP8266-serial-WIFI-wireless-module-wireless-transceiver/2031529724.html>. <http://www.aliexpress.com/item/Free-shipping-ESP8266-serial-WIFI-wireless-module-wireless-transceiver/2031529724.html>. [En línea] [Citado el: 20 de marzo de 2015.] <http://www.aliexpress.com/item/Free-shipping-ESP8266-serial-WIFI-wireless-module-wireless-transceiver/2031529724.html>.
20. **jo, Youngtae. y Jung, Inbum.** *Analysis of Vehicle Detection with WSN-Based Ultrasonic Sensors*. s.l. : Multidisciplinary Digital Publishing Institute, Multidisciplinary Digital Publishing Institute, 2014.
21. **NURDspace.** *NURDspace*. [En línea] [Citado el: 20 de abril de 2015.] <https://nurdspace.nl/ESP8266>.
22. **Arduino.** *Arduino*. [En línea] [Citado el: 20 de Abril de 2015.] <https://www.arduino.cc/>.
23. **IEEE STANDARDS ASSOCIATION.** <https://standards.ieee.org>. [En línea] [Citado el: 20 de febrero de 2015.] <https://standards.ieee.org/about/get/802/802.15.html>.
24. **NETWORK Platforms.** *NETWORK Platforms*. [En línea] [Citado el: 15 de marzo de 2015.] <http://www.networkplatforms.co.za/wired-vs-wireless>.
25. **Google Inc.** <https://developer.android.com/index.html>. [En línea] [Citado el: 25 de marzo de 2015.] <https://developer.android.com/sdk/index.html>.

26. **Bitrock Inc.** <https://bitnami.com/>. [En línea] [Citado el: 20 de marzo de 2015.]
<https://bitnami.com/support>.