

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL



CARRERA DE INGENIERÍA ELECTRÓNICA

DOCUMENTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO ELECTRÓNICO

TRABAJO DE TITULACIÓN:

**DISEÑO E IMPLEMENTACIÓN DE UN PARQUEO INTELIGENTE UTILIZANDO
ARDUINO YUN BASADO EN INTERNET DE LAS COSAS (IoT)**

AUTOR:

LEANDRO NELSON ROSALES LINDAO

TUTOR:

ING. RICARDO CAJO

Guayaquil – Ecuador

Diciembre de 2016

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

“La responsabilidad del contenido de este Proyecto de Titulación, conceptos plasmados en el texto, desarrollo, análisis y conclusiones del presente trabajo son de exclusiva responsabilidad de Leandro Nelson Rosales Lindao y el patrimonio intelectual del mismo a la UNIVERSIDAD POLITÉCNICA SALESIANA”.

Guayaquil, noviembre del 2016

Leandro Nelson Rosales Lindao

C.C.# 091923596-0

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN

Yo, Leandro Nelson Rosales Lindao expreso mi decisión de conceder el derecho patrimonial consagrado en la ley de Propiedad Intelectual del Ecuador dedicado a la Universidad Politécnica Salesiana el derecho en calidad de autor del proyecto de titulación escogido “DISEÑO E IMPLEMENTACION DE UN PARQUEO INTELIGENTE UTILIZANDO ARDUINO YUN BASADO EN INTERNET DE LAS COSAS (IoT)” el que se desarrolló para la obtención del título de Ingeniero Electrónico en la Universidad Politécnica Salesiana sede Guayaquil para tener el poder de ejercer completamente los derechos cedidos por nosotros.

Como autor me reservo los derechos morales del trabajo previamente citado. Por medio del documento me suscribo para hacer la entrega final de este trabajo de forma impresa y digital en la biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, noviembre del 2016

Leandro Nelson Rosales Lindao
CC. #0919235960

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN SUSCRITO POR EL TUTOR

En calidad de CO-TUTOR DEL TRABAJO DE TITULACION “Diseño e implementación de un parqueo inteligente utilizando Arduino Yun basado en internet de las cosas (IoT)”, elaborado por Leandro Nelson Rosales Lindao, declaro y certifico la aprobación del presente trabajo de titulación basándome en la supervisión y revisión de su contenido.

Guayaquil, noviembre del 2016

Ing. Ricardo Cajo Díaz

CC. #0924770282

DEDICATORIA

Dedico en especial a mi madre Angela Esperanza Lindao mi gran ejemplo de aguante y paciencia, mi apoyo constante durante toda mi vida, a mi novia Charlotte Potman que me ayudo a superar este escalón y me motiva a seguir superándome como persona y profesional.

A todos aquellos profesores de los cuales aprendimos más que una materia para superar el curso, sino aprender que somos capaces de realizar cualquier cosa que nos podamos imaginar, lo cual se refleja en el presente trabajo

Leandro Nelson Rosales Lindao

RESUMEN

AÑO	TÍTULO	ALUMNO	DIRECTOR DE TESIS	TEMA DE TITULACIÓN
2016	Ingeniería Electrónica	Leandro Nelson Rosales Lindao	Ing. Ricardo Cajo.	"Diseño e implementación de un parqueo inteligente utilizando Arduino Yun basado en internet de las cosas (IoT)"

El manejo apropiado de uso de espacios y lugares para estacionar vehículos en la ciudad Guayaquil es algo muy importante que colabora con el desarrollo constante de la urbe.

Encontrar lugares para estacionar a determinadas horas y en sectores de la ciudad resulta una tarea estresante que genera una pérdida considerable de tiempo, energía y contaminación ambiental. Existen varios parqueaderos en la UPS-G, pero ninguno de ellos nos brinda información anticipada, para saber si existen lugares libres o no y solo obtenemos esta información cuando estamos en el sitio.

Con el objetivo de optimizar el proceso en la búsqueda de lugares para parquear se elabora el siguiente Trabajo de Titulación, el cual está basado en la utilización de sensores ultrasónicos que servirán para detectar objetos, en nuestro caso vehículos medianos los cuales enviarán señales a nuestra tarjeta receptora de datos y esta a su vez transmitirá datos de manera digital a nuestra Arduino Yun, que con la ayuda de la plataforma Temboo la cual va a ser la encargada de administrar y enviar la información por medio de internet, para poder visualizar la disponibilidad de parqueos disponibles mediante una cuenta Twitter.

La información actualizada y anticipada ayudará a la toma de decisiones a los conductores antes de acudir a un lugar en búsqueda de un parqueo, de tal manera que colabore con un mejor uso del tiempo y recursos.

PALABRAS CLAVES

Arduino Yun, Diseño, Implementación, Internet de las Cosas (IoT), Parqueo Inteligente, Internet, Twitter.

ABSTRACT

YEAR	DEGREE	STUDENTS	TESIS DIRECTOR	TITLE TOPIC
2016	Ingeniería Electrónica	Leandro Nelson Rosales Lindao	Ing. Ricardo Cajo.	Desing and implementation of a Smart Parking System using Arduino Yun based on the Internet of Things (IoT)

The proper management of the use of parking spaces in Guayaquil, is something very important that comes with the constant development of the city. Finding a parking spot at specific hours, in specific sectors of the city, results in stress, a loss of time and energy, and environmental pollution.

There are various parking lots in the city. However, none of them provides real-time information on whether there are or aren't any available spots. This graduation project has the objective of optimizing the process of searching a parking spot. It is based on the use of ultrasonic sensors that detect objects, in this specific case cars.

The sensors will send a signal to a data receiving card that will digitally transmit these data to an Arduino Yun and Temboo. This Arduino Yun and Temboo manage and send the information through the internet, thereby making it possible to visualize the availability of parking spots on a specially created Twitter account.

This real-time information will help decision making in drivers before they arrive at the actual parking spot. In that way it assist in a better use of time and resources.

KEYWORDS

Design, Implementation, Internet of Things (IoT), Smart Parking, Arduino Yun, Data acquisition, Internet, Twitter.

ÍNDICE GENERAL

RESUMEN.....	VI
ABSTRACT	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS	XII
INTRODUCCIÓN.....	0
CAPÍTULO 1	
EL PROBLEMA	1
1.1. DESCRIPCIÓN DEL PROBLEMA.....	1
1.2. ANTECEDENTES.....	1
1.3. IMPORTANCIA	2
1.4. ALCANCE	3
1.5. DELIMITACIÓN.....	3
1.5.1. Espacial.....	3
1.5.2. Temporal.....	4
1.5.3 Académica	4
1.6. EXPLICACIÓN DEL PROBLEMA	4
1.6.1. Causas del problema	4
1.6.2. Formulación del problema científico	5
1.6.3. Solución	5
1.7. OBJETIVOS.....	6
1.7.1. Objetivo general	6
1.7.2. Objetivos específicos.....	6
CAPÍTULO 2	
MARCO TEÓRICO	7
2.1. FUNDAMENTOS TEÓRICOS.....	7
2.1.1. Definición de Internet de las Cosas (IoT).....	7
2.1.2. Beneficios del Internet de las Cosas (IoT)	8
2.1.3. Tendencias de la Internet de las Cosas (IoT).....	9
2.1.4. Los riesgos, privacidad y la seguridad del IoT.....	12
2.1.5. IoT en sistemas de Parqueo Inteligente.....	14
2.1.5.1 Sistemas de Parqueo Inteligente.....	14
2.1.5.2. Los beneficios de un sistema de Parqueo Inteligente.....	14

2.1.6. Software Libre.....	19
2.1.7. Hardware Libre	20
2.1.8. Arduino Yun	20
2.1.8.1 Hardware de Arduino Yun.....	201
2.1.8.2 Microprocesador Linux Atheros AR9331.....	202
2.1.8.3 Energía y Alimentación	24
2.1.9. Plataforma Temboo.....	24
2.1.10. Twitter	24
2.1.11. Sensor SRF02	25
2.1.12. Microcontrolador Atmega328.....	26
2.1.13. Cristal Oscilador de Cuarzo.....	287
2.1.14. Resistencia Eléctrica	28
2.1.15. Diodos Led Dual	29
2.1.16. Borneras de Conexión	29
2.1.17. Proteus.....	30
2.2. METODOLOGÍA.....	30
2.2.1. Método Experimental	30
2.2.2. Método Teórico	31
2.2.3. Método Inductivo	31
2.3. TÉCNICAS	31
2.3.1. Técnica Experimental.....	31
2.3.2. Técnica Documental	31
2.4. POBLACIÓN Y MUESTRA	32
2.4.1. Población	32
2.4.2. Muestra	32
2.5. DESCRIPCIÓN DE LA PROPUESTA.....	32
2.6. BENEFICIARIOS	33
2.7. IMPACTO.....	33

CAPITULO 3

RESULTADOS.....	34
3.1. IMPLEMENTACIÓN Y DESARROLLO	34
3.1.1. Diseño de Circuito Impreso (PCB)	34
3.1.2. Configuración de Microcontrolador ATmega328	35
3.1.3. Configuración de Arduino Yun & Temboo para envío de Tweets y página Web	38
3.1.3.1 Diseño Web e integración con Twitter.....	43
3.1.4. Diseño de Infraestructura para el sistema.....	44

CAPITULO 4

ANÁLISIS DE RESULTADOS.....	46
4.1. <i>Desarrollo de diagramas de funcionamiento y bloques del sistema</i>	46
4.2. <i>Análisis del desarrollo de Tarjeta Electrónica de Adquisición de Datos.....</i>	49
4.3. <i>Análisis de Instalación de Infraestructura del Sistema</i>	53
4.4. <i>Análisis de configuración de Arduino Yun para envío de Tweets y visualización de Página Web</i>	54
4.4.1. <i>Resultados del diseño de la página Web e integración con Twitter.....</i>	62
4.4.2. <i>Luces de estado (Puesto ocupado o libre).....</i>	62
4.4.3. <i>Resultado y Análisis de las encuestas realizadas a docentes y personal administrativo. 53</i>	53
5. CONCLUSIONES	67
6. RECOMENDACIONES	68
7. REFERENCIAS BIBLIOGRÁFICAS.....	69
8. ANEXOS.....	72
8.1. <i>Presupuesto de desarrollo del sistemaa</i>	73
8.2. <i>Encuestas.....</i>	73

ÍNDICE DE TABLAS

Tabla 1. Características de AVR Arduino microcontrolador.....	222
Tabla 2. Microprocesador Linux.....	22
Tabla 3. Características del SRF02	26
Tabla 4. Características Atmega328.....	27
Tabla 5. Cotos de desarrollo del sistema Parquadero Inteligente	727

ÍNDICE DE FIGURAS

Figura 1: Área del bloque B donde serán instalados el Parqueadero Inteligente.....	3
Figura 2: Número de usuarios de Internet en porcentaje a la población.....	11
Figura 3: Visión general Parqueo Inteligente	16
Figura 4: Placa Arduino Yun	21
Figura 5: Identificación de Hardware Placa Arduino Yun	21
Figura 6: Sensor Ultrasónico SRF02.....	266
Figura 7: Microcontrolador ATmega328.....	277
Figura 8: Cristal 16Mhz.....	288
Figura 9: Resistencia	299
Figura 10: Diodo Led Dual	299
Figura 11: Bornera.....	30
Figura 12: Diagrama de funcionamiento del sistema desarrollado.	33
Figura 13: Diseño de la PCB realizado en Proteus 8 Professional.....	34
Figura 14: Diagrama de pistas impresa de la PCB diseñada	35
Figura 15: Interface hardware de programación Arduino UNO.....	36
Figura 16: Portal Temboo	39
Figura 17: Portal Twitter.....	39
Figura 18: Portal App Twitter	40
Figura 19: Diagrama de ductería instalada	44
Figura 20: Diagrama de funcionamiento del sistema desarrollado.	45
Figura 21: Aplicación Twitter.....	46
Figura 22: Diagrama de bloques del sistema desarrollado.....	47
Figura 23: PCB de la Tarjeta Electrónica para recepción de Datos.....	48
Figura 24: Placa Receptora de datos sin obstáculo en el sensor	49
Figura 25: Placa Receptora de datos con obstáculo en el sensor.....	52
Figura 26: Infraestructura Instalada en el parqueadero del bloque B	553
Figura 27: Tarjetas y Sensores Instalados en el parqueadero del bloque B.....	53
Figura 28: Cajas reflectoras Instaladas en el parqueadero del bloque B.....	54
Figura 29: Cuenta creada en Temboo	54
Figura 30: Cuenta creada en Twitter.....	55
Figura 31: Portal Twitter dev console.....	55
Figura 32: Portal Twitter dev console permisos de lectura y escritura.....	56
Figura 33: Cuenta Twitter @smart_parking_b.....	57
Figura 34: Cajas reflectoras Instaladas en el parqueadero del bloque B.....	61
Figura 35: Página Web de Parqueo Inteligente.....	62

Figura 36: Cajas reflectoras Instaladas en el parqueadero del bloque B.....	63
Figura 37: Conocimiento sobre Parqueos Inteligentes	64
Figura 38: Usuarios dispuestos a usar nuestro sistema	65
Figura 39: Usuarios que piensan que el sistema optimiza recursos	65
Figura 40: Usuarios que recomiendan instalar el sistema en otros lugares.....	65
Figura 41: Usuarios que utilizan Twitter y Portal WEB	66

INTRODUCCIÓN

En los últimos años el número de vehículos ha crecido exponencialmente. Las calles o infraestructura vial, no siempre se han desarrollado a la misma velocidad. En las grandes ciudades en crecimiento, la congestión vehicular es un problema que detiene el crecimiento y desarrollo. Guayaquil también está luchando con este problema.

Aunque el crecimiento y desarrollo en ciudades hoy en día resulta en algunos problemas, también hay iniciativas que proporcionan posibles soluciones. En especial en el campo tecnológico, el desarrollo va al menos a la misma velocidad que el desarrollo urbano. La tecnología forma cada vez más parte de la vida diaria en la ciudad y sus habitantes. Esta tecnología se refiere al término Ciudades Inteligentes o, Smart Cities. Con la ayuda de la Internet de las Cosas, IoT, se ofrece más y mejores servicios a los usuarios. A los habitantes de las ciudades les resulte más efectivo el uso de su tiempo. Al mismo tiempo se consigue optimizar recursos naturales, utilizándolos más efectivamente.

Uno de los ámbitos en los que puede aplicarse este concepto es en los llamados Parques Inteligentes o Smart Parking, que permita al usuario obtener información anticipada y ayude en la toma de decisiones oportunas al momento de buscar un lugar de parqueo.

La propuesta de este trabajo consiste en la instalación de un parqueadero inteligente con capacidad para 12 vehículos que informará de manera anticipada la disponibilidad de lugares libres para estacionar vía Twitter. Esto se logrará a través de una red de sensores ultrasónicos que detectarán la presencia del vehículo en una plaza de estacionamiento, cuyos datos serán receptados por la Arduino Yun, la cual con la ayuda de la plataforma Temboo se encargará de enviar la información a la nube, internet, donde los usuarios podrán informarse sobre los espacios disponibles por medio de la vía antes mencionada. De esta forma se consigue que los conductores pasen la menor cantidad de tiempo posible en la búsqueda de un lugar de parqueo.

CAPÍTULO 1

EI PROBLEMA

1.1. Descripción del problema

Actualmente la ciudad de Guayaquil tiene un problema de congestión vehicular, el cual también se refleja en los alrededores de la Universidad Politécnica Salesiana sede Guayaquil. Donde la actual infraestructura vial fue diseñada para manejar poco volumen de tránsito, y la falta de información anticipada hacia conductores sobre la disponibilidad de parqueos libres en las distintas zonas asignadas causa que se genere congestión, y pérdida de tiempo a determinadas horas pico.

1.2. Antecedentes

En los últimos años el número de vehículos ha crecido dentro de la ciudad de Guayaquil. El desarrollo urbanístico no siempre se desarrolla a la misma velocidad que la tecnología. Hablando de la Universidad Politécnica Salesiana, las calles en los alrededores fueron diseñadas para manejar volúmenes de circulación menos densos de los que se genera hoy en día a determinadas horas pico. Debido a que los conductores no saben dónde pueden encontrar un estacionamiento libre, manejan despacio para buscar un lugar desocupado. El resultado es un gran congestión provocado por vehículos buscando estacionamiento. Además, de causar frustración a los conductores, el congestión trae algunos problemas:

- Pérdida de tiempo y energía – los conductores deben buscar mucho tiempo antes de que puedan encontrar un aparcamiento libre, pierden tiempo que sin el congestión podrían utilizar de una forma más efectiva. Además, que afecta a otros conductores que usan las calles de los alrededores de la universidad, siendo afectados por esta demora.

- Contaminación – los vehículos utilizan gasolina o diesel y emiten gases nocivos de CO₂. Cuanto más tiempo un vehículo está en movimiento, más larga serán las emisiones de CO₂. Cada minuto que el vehículo pasa con el motor en marcha, causa daños innecesarios al medio ambiente.
- Situaciones peligrosas – Mientras que los conductores están buscando un estacionamiento libre, su atención está dividida entre su medio ambiente, el manejo y la búsqueda. Porque están distraídos, la probabilidad de un accidente crece y la seguridad disminuye.

En la ciudad de Guayaquil según cifras de la Comisión de Tránsito del Ecuador, el parque automotor de la ciudad en el 2011 alcanzó las 370.000 unidades y para el año 2012 se proyecta que sea superior a los 400.000 (Neumann, 2012). Eso implica un crecimiento anual que oscila entre el 10% y 15%. Cuando nos fijamos en otros países que tienen problemas similares, algunas cifras aparecen. Según un informe realizado en el 2006 por el Departamento de Planificación Urbana de la Universidad de los Ángeles, EEUU, un 30% de conductores se encuentran en búsqueda activa de un lugar para estacionarse. Tardan un promedio de 8 minutos en encontrar dicha plaza, como consecuencia se genera pérdida de tiempo y combustible, y son responsables de una fracción considerable de las emisiones de gases por parte de los vehículos. En otro lado, se ha reportado que, en un distrito pequeño del centro de los Ángeles, los conductores que deambulan buscando estacionamiento, durante un año realizaron el equivalente a 38 viajes alrededor del mundo, consumiendo casi 200.000 litros de combustible y emitiendo 730 toneladas de dióxido de carbono (Formoso, Mazzilli & Sotelo, 2014)

La solución propuesta para el problema de falta de información anticipada y congestión provocados por vehículos buscando estacionamiento, es importante porque no solo se ahorra tiempo y energía, sino también se puede reducir la contaminación y aumentar la seguridad en la carretera.

1.3. Importancia

Este trabajo pretende desarrollar un sistema de parqueadero inteligente, el cual es capaz de brindar al usuario información en tiempo real sobre los estados del parqueo y que este pueda tomar una decisión antes de ir al lugar, de tal forma que esta ayude

a mejorar el flujo vehicular en los alrededores del campus y hacer un uso más efectivo de su tiempo.

1.4. Alcance

Con este trabajo se busca brindar información anticipada sobre plazas de parqueo disponible en un sector del bloque B asignado para esta actividad y colaborará en un mejor uso del tiempo y en parte con la disminución del congestionamiento provocado por vehículos buscando estacionamiento en las calles u otros parqueaderos en los alrededores de la Universidad Politécnica Salesiana en Guayaquil.

1.5. Delimitación

1.5.1. Espacial

Este trabajo, se centra en la instalación de 12 parqueaderos del bloque B de la Universidad Politécnica Salesiana como lo muestra la Figura 1. Y será desarrollado e instalado con el fin de brindar una herramienta para ayudar en la búsqueda de parqueos a los profesores y demás personas que laboran en el área.



Figura 1: Área Bloque B donde se instala el Parqueo Inteligente

(Google, 2016)

1.5.2. Temporal

Este trabajo de titulación tiene una duración 12 meses.

1.5.3 Académica

En este trabajo se pondrá en práctica conocimientos adquiridos durante el periodo de formación como Ingeniero Electrónico con mención Telecomunicaciones en la UPS-G, dando énfasis a temas relacionados con Telecomunicaciones y Electrónica.

1.6. Explicación del problema

1.6.1. Causas del problema

Las ciudades en crecimiento, como Guayaquil, presentan cada vez más problemas con su infraestructura de tránsito. La mayoría de residentes tienen un vehículo y cada vez hay más de ellos en la carretera. El resultado es un gran congestionamiento provocado por vehículos buscando estacionamiento. Se ha identificado tres posibles causas del problema:

1. No hay suficiente aparcamiento para todos los usuarios.
2. Las calles en los alrededores fueron diseñadas para manejar volúmenes de circulación menos densos, de los que hoy en día se genera a determinadas horas pico.
3. Los conductores no saben dónde pueden encontrar un estacionamiento libre, dado que hay poca o nada de información que les puede ayudar en su búsqueda.

La primera posible causa del problema, se puede rechazar. Ya que la Universidad Politécnica Salesiana cuenta con 427 lugares para parquear. De esto se puede concluir que hay suficientes lugares para estacionar vehículos, ya que algunos de estos lugares aún tienen capacidad disponible.

La segunda posible causa, calles demasiadas pequeñas, es un problema reconocido por una ciudad en crecimiento como Guayaquil. Es importante abordar este problema. Sin embargo, el círculo de influencia en este aspecto es limitado ya que no se puede influir directamente en cambios de la infraestructura vial. Uno puede preguntarse si el desarrollo vial-urbanístico del sector, es una solución concreta a corto plazo y que sea realista.

La última posible causa del problema, es algo que se puede resolver relativamente de una manera simple y para lo cual no se necesita cambiar cosas a nivel administrativo, físico a nivel de la infraestructura vial, solicitándolo al municipio o autoridad competente. Por eso, este Trabajo de Titulación se centra en desarrollar una solución tecnológica que proporcione información anticipada sobre disponibilidad de parqueo y que ayude a los conductores a encontrar espacio libre.

1.6.2. Formulación del problema científico

¿Cómo optimizar recursos tales como el tiempo, combustible y mejorar el flujo de tránsito con la ayuda de una solución tecnológica que muestre al conductor en tiempo real los espacios libres para parquear?

1.6.3. Solución

Este trabajo consiste en el desarrollo e instalación de un parqueadero inteligente utilizando sensores ultrasónicos que conectados a una circuitería electrónica proporcionan información a través de internet a los usuarios sobre la disponibilidad de estacionamientos en un sector del parqueadero del bloque "B". De esta forma se logra que los conductores puedan tomar una decisión oportuna antes de ir al lugar y pasen la menor cantidad de tiempo en la búsqueda de un lugar libre.

La solución es importante para los usuarios ya que podrían colaborar con la disminución del tráfico, reducir el consumo de combustible y emisiones de gases por parte de los vehículos, se hace un mejor uso de los lugares disponibles, mejoraría el uso del tiempo y acceso a ciertos puntos del campus universitario.

1.7. Objetivos

1.7.1. Objetivo general

Desarrollar e implementar un sistema Parquadero Inteligente utilizando Arduino Yun, basado en Internet de las Cosas.

1.7.2. Objetivos específicos

- Desarrollar una tarjeta para la adquisición de datos, basada en microcontroladores que sirvan para interpretar la señal recibida por los sensores ultrasónicos.
- Instalar los sensores ultrasónicos con su respectiva tarjeta de adquisición de datos en el parqueadero del bloque B de la UPS-G.
- Conectar dispositivos externos como luces para indicar el estado de cada puesto de parqueo.
- Configurar y cargar librerías de Arduino Yun que sirvan para la visualización de la información desde cualquier dispositivo con acceso a internet vía Twitter.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Fundamentos teóricos

En esta sección se describe el Estado del Arte con respecto a la Internet de las Cosas y Sistemas de Parqueaderos Inteligentes. Se iniciará con una definición de la Internet de las Cosas (IoT), los beneficios, tendencias que apoyan la IoT y los posibles riesgos con respecto a la privacidad y la seguridad. Se continúa con IoT en Sistemas de Parqueo Inteligente, cinco tipos de estos sistemas y sus beneficios, además de todos los elementos utilizados en el desarrollo de este trabajo de titulación.

2.1.1. Definición de Internet de las Cosas (IoT)

En la actualidad no existe una definición única sobre Internet de las Cosas. En este trabajo se utiliza la definición propuesta por la Unión Internacional de Telecomunicaciones (2012) un organismo de las Naciones Unidas especializada para la Información y comunicaciones tecnológicas (ICTs).

Internet de las cosas es una infraestructura global para la sociedad de la información, lo que permite servicios avanzados mediante la interconexión de las cosas (físicas y virtuales) en base a cosas existentes y evolución de las tecnologías de la información y comunicación interoperables.

- Nota 1.- A través de la explotación de la identificación, captura de datos, capacidades de procesamiento y comunicación, la IoT hace un uso completo de las cosas que ofrecen los servicios a todas las aplicaciones, garantizando al mismo tiempo que se cumplan los requisitos de seguridad y privacidad.
- Nota 2.- Desde una perspectiva más amplia, la IoT puede ser percibido como una visión con implicaciones tecnológicas y sociales.

2.1.1.1. Cosas

Con respecto a la Internet de las Cosas, esto es un objeto del mundo físico (cosas físicas) o en el mundo de la información (cosas virtuales), que son capaces de ser identificados e integrados en redes de comunicación (ITU Telecommunication Standardization Sector, 2012).

2.1.1.2 Dispositivos

Con respecto al internet de las cosas, se trata de una pieza de equipo con las capacidades obligatorias de comunicación y las capacidades opcionales de la detección, captura de datos, almacenamiento de datos y procesamiento de datos (ITU Telecommunication Standardization Sector, 2012).

2.1.2. Beneficios del Internet de las Cosas (IoT)

Con una definición más clara del Internet de las Cosas y los dispositivos, se centrará en los beneficios para la sociedad. El objetivo principal de la IoT es hacer la vida más fácil. IoT debe hacer una “cosa” más fácil, no más complejo.

La interacción entre el usuario y el dispositivo IoT debe ser natural y no significativamente diferente de cuando la “cosa” era sólo una “cosa. Un coche, por ejemplo, parece en gran medida el mismo que hace 50 años. Todavía puedes conducir desplazando las ruedas y acelerando. En el interior del coche, sin embargo, muchas cosas han cambiado: los sistemas de frenado con sensores, sensores de lluvia en el parabrisas, etc. (Coursera - University of California, 2016).

Además de beneficio general de hacer la vida más fácil, la IoT también hace que la gente sea menos dependiente de otras personas. Un ejemplo muy sencillo de esto, son los dispositivos médicos que miden todas las funciones vitales del cuerpo. Las personas con problemas de salud ya no tienen que visitar al médico solo para medir la presión sanguínea y el ritmo cardíaco. Llevan un dispositivo especial que supervisa constantemente sus funciones vitales y es capaz de comunicar con la oficina de los médicos. Solo cuando ciertos parámetros no se cumplen, la señal de alarma se

apaga y el médico se pone en contacto con el paciente. De una manera esto es una amenaza para las personas que cumplan las profesiones que son relativamente fáciles de reemplazar por la IoT o robótica (Coursera - University of California, 2016). En el otro extremo, en casi todo el mundo los gastos médicos son muy altos y los recortes financieros en el sector de la salud son necesarios. IoT ofrece una manera de hacer de la salud menos costosa y aun ofreciendo el mismo cuidado que los pacientes necesitan.

Como último beneficio se gustaría señalar, es el hecho de que la IoT nos permite conectar con el “mundo entero”; incluso más de lo que ya lo hicimos (Coursera - University of California, 2016). Por ejemplo, no es necesario volar hasta el otro extremo del mundo para estar en una reunión, pero hay dispositivos IoT muy avanzados que te pueden proyectar allí, cierran los micrófonos de personas que no están hablando y mover la cámara hacia la persona que si está hablando en ese momento. Esto ofrece enormes ventajas para la globalización y los negocios internacionales.

Con todo, la IoT ofrece muchos beneficios a la sociedad. Todos ellos vienen a hacer la vida más fácil y más eficiente.

2.1.3. Tendencias de la Internet de las Cosas (IoT)

IoT no es algo de los últimos años. El New York Times ya ha publicado un ejemplo interesante de la IoT en 1999.

Ejemplo:

La casa clase media de Scott Smyers, vicepresidente de los laboratorios de investigación de Sony Electronics en San José, California, se concentra en gran medida en el uso de microprocesadores para la electrónica de consumo, por ejemplo. La lavandería del Sr. Smyers tiene una lavadora Maytag de \$1.000 llamada Neptune, lo que elimina el efecto exorcista (vibraciones excesivas) que se apodera de la mayoría de las lavadoras cuando van fuera de equilibrio. Antes de que la máquina tenga la oportunidad de estar fuera de control, su procesador detecta la vibración anormal y frena el tambor lo suficiente para reorganizar la lavandería. En la cocina

del Sr. Smyers, la estufa Thermidor produce una cocción perfecta, encendiendo y apagando la llama. Si una ráfaga desde una ventana abierta extingue la llama, la estufa la enciende sí mismo. La tostadora Accu-Sense (100\$), también llamada ‘tecnología de tostar’, del Sr. Smyers, detecta la temperatura de la superficie del alimento insertado. Si está congelado, la tostadora se mantiene a fuego lento por un tiempo. Para el día de la Madre, el Sr. Smyers dio a su esposa un horno de convección térmica doble con tantos botones para empujar como un VCR (Hafner, 1999).

Por supuesto, esto no es un ejemplo de alguien que tiene una casa de lo más normal “automático” en el momento, pero ilustra que la IoT ha estado por un largo tiempo. En 1999, los hogares de la clase media, en promedio, ya tenían alrededor de 40 a 50 microprocesadores en ella (Hafner, 1999). Sin embargo, la última década ha habido varias tendencias o desarrollos que apoyan la creciente aplicación de la IoT. Las grandes empresas han pronosticado el aumento en la IoT (Gartner, 2013; Cisco, 2013; EMC, 2014). Una predicción es que el número de cosas conectadas en el mundo tendrá un aumento de treinta veces entre el 2009 y 2020 y que para este año habrá 26 millones de cosas que estarán conectadas al internet. Por un lado, la tendencia general para automatizar los procesos, hay varios desarrollos que apoyan el aumento de la IoT. Los más importantes se enumeran a continuación.

- Tecnología Computacional se ha vuelto mucho más barato que antes. Ahora se encuentra en un rango de precios en el que es razonable agregarlo a los costos básicos de la “cosa” que se supone mejorar (Coursera - University of California, 2016).
- El hardware se ha convertido más pequeño y ligero, por lo tanto, más fácil de construir en “cosas”. La ley de Moore establece que el número de transistores en un chip se duplica aproximadamente cada dos años (Moore, 1998). Esto ha permitido a la gente desarrollar ordenadores más potentes con el mismo tamaño del chip. Intel, un fabricante de chips semiconductores notoriamente conocida, haya tenido durante 1971, 2300 transistores en un procesador y para el año 2012 sus procesadores contenían 1.4 millones de transistores. Este es un aumento de aproximadamente 610.000% y se espera que esta

tendencia continúe (Henderson, 2014). Al girar el gráfico para ley de Moore se puede interpretar como el tamaño de un ordenador (de una capacidad fija): se divide por dos cada dos años.

- La capacidad computacional ha crecido enormemente. IoT necesita una alta velocidad de procesamiento de datos. La ley de Koomey explica que el número de cálculos por kilovatio/hora se duplica aproximadamente cada año y medio (Greene, 2011). Al girar el gráfico de la ley de Koomey al revés, se puede interpretar como la cantidad de energía necesaria para realizar un cálculo está cayendo o decreciendo a un ritmo rápido (Ashton, 2014).
- Acceso al Internet ha mejorado drásticamente como lo muestra la Figura 2. En casi todas las partes del mundo desarrollado se tiene acceso a internet. En algunas partes del mundo todavía hay una falta de acceso fácil, pero las grandes empresas como Google (Project Loon, 2016) y Facebook (Internet.org, 2016) están trabajando en eso.

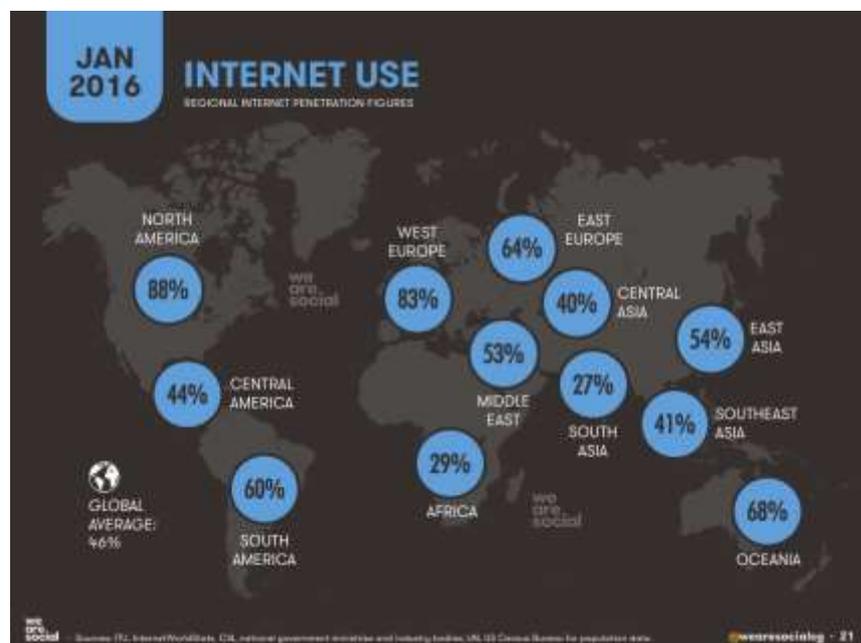


Figura 2: Número de usuarios que usan Internet en porcentaje a la población (Rosabel, 2016)

- Internet Inalámbrico facilita la creación de redes menos costosas debido a que no es necesario tirar cables por todas partes para conectarse.
- Costos de datos son bastante bajos (dependiendo de donde vivas). Enviar y recibir grandes cantidades de datos está dentro del alcance de muchas personas.
- Ancho de Banda de datos es alta por lo que son capaces de procesar datos muy rápido, mejorando la capacidad computacional.

Hay varios desarrollos tecnológicos que han impulsado el ya creciente campo de la IoT. Con el acceso a Internet y la disponibilidad de dispositivos inteligentes que crecen en una base diaria, el final de su desarrollo ya no está a la vista. Siendo un desarrollo tan importante, también es necesario tomar en consideración los posibles riesgos.

2.1.4. Los riesgos, privacidad y la seguridad del IoT

A pesar de la capacidad de la IoT para hacer la vida más fácil y aumentar la seguridad, también hay algunos riesgos potenciales que deben tenerse en cuenta al aplicar o usar la IoT, aunque se puede prevenir los riesgos, es importante identificarlos y tomar tantas medidas de seguridad como sea posible. Se identifican 5 riesgos importantes (Coursera - University of California, 2016).

1.- Aunque evidentemente es económicamente atractivo tener dispositivos IoT para reemplazar a los humanos en algunas tareas, también es un riesgo en el aislamiento social. Por ejemplo, muchas personas mayores no tienen una gran cantidad de contactos sociales y son siempre felices con las visitas de enfermería. Si la enfermera no tiene que visitar más, el aislamiento social está al acecho (Coursera - University of California, 2016).

2.- Otro riesgo es la dependencia de la tecnología y las infraestructuras que van con ello. Mientras más se usa la IoT, más se puede “perder” cuando la energía o el internet se caen, no pensar en “errores” que podrían estropear un dispositivo IoT. Hay informes de los marcapasos que fallan debido a problemas de software, lo que podría producir situaciones que amenazan la vida (Sandler, Ohrstrom, Moy & McVay, 2010)

3.- La privacidad es una parte importante cuando se habla de IoT, estos dispositivos tienden a registrar una gran cantidad de información, por ejemplo: la ubicación de su teléfono, la salud en un Health Watch y los hábitos de compras en su tarjeta de crédito. Se utiliza toda esta información, en la base, para hacer la vida más fácil. Sin embargo, es obvio que en las manos equivocadas esta información se puede utilizar contra nosotros. Un ejemplo de lo relativamente fácil que es acceder a la información almacenada por las aplicaciones en los teléfonos inteligentes, se analizó en agosto 2015, cuando alguien “accidentalmente” descubrió como obtener acceso a todos los datos de carácter personal en Facebook (Griffin, 2015). Con sólo el uso de una herramienta relativamente simple, los hackers fueron capaces de descubrir el número de teléfono que es necesario introducir en una cuenta sin contraseña. Datos como la ubicación, las actividades de diferentes dispositivos, etc., todos fueron expuestos.

4.- Otro tema de la privacidad en la IoT, previene de las empresas que lo venden. Muchas empresas que venden dispositivos IoT o aplicaciones, tienen acuerdos de usuario de gran tamaño. No tantas personas la leen de la A a la Z. A veces en las letras pequeñas afirman que sus datos pueden ser recogidos y transmitidos a terceros. La mayoría de las veces esto se utiliza para la comercialización del Internet, por ejemplo, los anuncios que muestran exactamente las cosas que has estado buscando en google sobre el día antes. Sin embargo, una situación más preocupante se presenta cuando se basa en la información combinada de diferentes fuentes, es posible que ya no sea capaz de obtener un seguro de salud o un préstamo para su casa. Hoy en día, las grandes empresas se han especializado en exactamente eso: recopilación, análisis y la venta de los llamados grandes datos sobre las personas (Boutin, 2016).

5.- El último tipo de riesgo, riesgos de seguridad, está relacionado a la privacidad. Los datos que están siendo recogidos se almacenan en un lugar. Usted asume que se mantienen de una manera segura, cifrada en un servidor bien protegido en algún lugar de la nube. Sin embargo, usted como usuario no tiene idea si la seguridad es buena. En los últimos años ha habido muchos hacks de seguridad en las empresas e instituciones muy conocidas que dejaron datos personales expuestos al mundo. Un ejemplo reciente es el de la Samsung Smart Refrigeradora que usa g-mail para comunicarse con el calendario (Leyden, 2015). Aunque la nevera implementa SSL, no valida los certificados SSL, permitiendo de este modo los ataques contra la mayoría de las conexiones y poner las cuentas g-mail en riesgo. Cuando los datos no se almacenan localmente en exclusiva, nunca se está seguro de que posiblemente pueda acceder a ellos.

2.1.5. IoT en sistemas de Parqueo Inteligente

Aunque hay muchas aplicaciones del IoT, ahora se centrará en la aplicación específica de la IoT en este proyecto de titulación: la IoT en Sistemas de Parqueadero Inteligente. Primero se describe el concepto de Sistemas de Parqueadero Inteligente y se introduce cinco sistemas diferentes. A continuación, se enumera varios beneficios que surgen del uso de este tipo de sistemas. Se termina con un breve resumen.

2.1.5.1 Sistemas de Parqueo Inteligente

El objetivo de los sistemas de parqueo inteligente es ayudar a los conductores a encontrar plazas de aparcamiento de manera eficiente a través de tecnologías de la información y comunicaciones. Si bien la idea inicial se ha propuesto hace diez años, los últimos cinco años se ve cada vez más sistemas de parqueo inteligente. Debido al rápido aumento del uso de teléfonos inteligentes, ahora muchos son capaces de conectarse a Internet y buscar información cuando están fuera de casa (Lin, 2015).

Los sistemas de parqueo inteligente se proponen como la primera solución para aliviar la congestión vehicular cada vez mayor, muchos investigadores y ciudades han iniciado con el despliegue de servicios de acuerdo con diferentes perspectivas.

Un servicio esencial de parqueo inteligente incluye dos corrientes: Información y el tráfico. El flujo de tráfico que pasa sobre la vía donde se desea buscar parqueo. Se muestra en el triángulo inferior azul de la Figura 3, por lo general los conductores de vehículos reciben información de disponibilidad de estacionamiento, luego se dirigen al área deseada y después parquean (Lin, 2015).

Sobre la base de la infraestructura de la ciudad y tecnología de la comunicación, los conductores solo pueden obtener información parcial o anticuada y tendrían que repetir la búsqueda de estacionamiento hasta conseguir una. Cuando hay muchos conductores que buscan estacionamiento, se produce una competencia y como resultado hay, conflictos de estacionamiento acumulativo. El comportamiento de estacionamientos también varía teniendo en cuenta la información que poseen los conductores y el tiempo que conducen en la calle. Una vez que un vehículo llegue o salga de un lugar de estacionamiento, los cambios de información sobre la disponibilidad de estacionamiento tienen que mostrarlo inmediatamente a los conductores de vehículos que estén en busca de un lugar. El flujo de información se muestra en el triángulo rosa superior de la Figura 3.

Con el fin de obtener el estado de las plazas de parqueo, se instalan sensores fijos o móviles en calles, parqueaderos cubiertos, etc., para detectar los eventos vehiculares. Se puede utilizar una amplia variedad de sensores, entre ellos: infrarrojos, magnetómetros, tubo neumático de carretera, sensores magnéticos, sensores ultrasónicos, sensores de peso en movimiento, RFID, procesadores de imagen de video, etc. (Idna Idris, Leng, Tamil, & Razak, 2009)

Los sensores forman una red y envían la información más actualizada a los dispositivos de almacenamiento de datos. Los conductores pueden obtener la información actualizada en sus dispositivos inteligentes, que intercambian mensajes con la infraestructura de caminos, estaciones bases transceptoras.

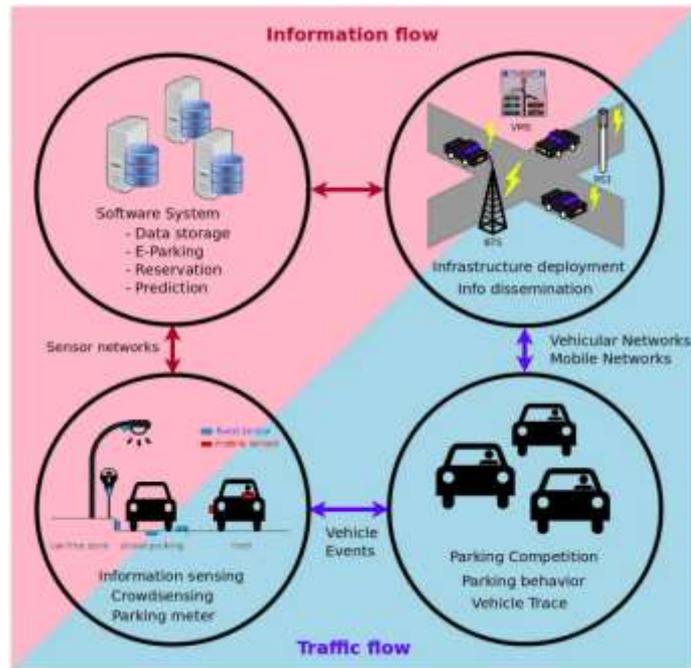


Figura 3: *Visión general Parqueo Inteligente*

(Lin, 2015)

Sistema de información de guía de parqueo (IGP)

Uno de los primeros ejemplos de parqueo inteligente, son los sistemas de información de guía de parqueo (IGP). Sistemas de IGP intentaron minimizar la búsqueda de aparcamiento mediante el control de estacionamiento disponible de forma dinámica y dirigiendo los automovilistas. Pueden incluir ya sea en toda la ciudad o simplemente la instalación de aparcamiento (Rodier, Shaheen & Kemmerer, 2008). En IGP los sensores de detección de vehículos se instalan habitualmente en las entradas, salidas y/o espacio de estacionamiento individual para detectar ocupación de los vehículos. Las lecciones aprendidas mediante la evaluación y el modelado de estos sistemas muestran que, con el fin de ser eficaz, los mensajes deben mostrar información adecuada que se adapte a las necesidades del conductor. Los sistemas de IGP se encuentran para reducir la longitud de las colas de instalación de aparcamiento; Sin embargo, las reducciones de todo el sistema en tiempo de viaje y desplazamiento del vehículo y los beneficios económicos pueden ser relativamente pequeños (Waterson, Hounsell & Chatterjee, 2001)

Sistemas basados en tránsito (TB)

Basándose en las lecciones aprendidas de los sistemas de IGP, los sistemas basados en tránsito tienen como objetivo aumentar el uso del transporte y de los ingresos y reducir la marcha del vehículo, el uso de combustible y la contaminación del aire. Muchos sistemas basados en tránsito (TB) implementados en países como Francia, Alemania, Irlanda, Japón, Suiza, el Reino Unido y Estados Unidos, son muy parecidos a los de la IGP (Rodier, Shaheen & Kemmerer, 2008). Sin embargo, los sistemas de TB tienen por objeto proporcionar información sobre las opciones de Park-and-Ride y cómo pueden transitar los usuarios del coche al transporte público lo más suave posible. Una revisión de la literatura sugiere que la escasez de aparcamiento en las estaciones de tren cercanas puede limitar significativamente el uso del transporte. Además, los conductores pueden responder a información pre-viaje e información sobre la disponibilidad de estacionamiento en su destino final. Por último, los viajeros regulares parecen estar más sensibles a la información de estacionamiento en combinación con sistemas de transporte IGP más básicas porque este tipo de información en tiempo real tiene mayor relevancia para ellos (Rodier, Shaheen & Eaken, 2004)

Sistemas de pago inteligente

Ambos sistemas IGP y sistemas TB, proporcionan a sus usuarios de información valiosa acerca de la disponibilidad de estacionamiento y, en el caso de la TB, su viaje. Un tipo de sistemas inteligente de estacionamiento que no se centra en la disponibilidad, pero que apuntan a la optimización de los procesos de estacionamiento, es un sistema de pago inteligente. En estos sistemas, el proceso de pago es completamente automatizado para evitar los retrasos que se producen durante los métodos de pago convencionales. Algunos sistemas utilizan un registro de entrada y salida con unos dispositivos móviles. Por lo que es innecesario que el usuario busque un parquímetro y llevar monedas o tarjetas bancarias. Otros sistemas funcionan con una pequeña caja colocada en el coche, automatizando completamente el proceso de pago, simplemente porque se detecta cuando se conduce dentro y fuera del garaje. A pesar de que existen sistemas de pago inteligente, el público en general todavía tiene sus preocupaciones sobre la privacidad y la seguridad de los datos necesarios a implementar en sistemas de este tipo (Idna Idris et al., 2009).

Sistemas E-parking

Un cuarto tipo de sistema de aparcamiento inteligente, se llama E-parking. En un E-parking los usuarios pueden consultar la disponibilidad y/o reservar una plaza de estacionamiento. El sistema se puede acceder a través de numerosos métodos, tales como SMS, Teléfonos Inteligentes y otros dispositivos conectados a internet. Esto permite a los usuarios ver información actualizada antes y durante su viaje. Cuando también está equipado con un sistema de reserva, un sistema E-parking es capaz de evitar que dos conductores lleguen al mismo momento en el mismo lugar. Añadir a estas características un sistema de pagos eficiente y automatizar todo el proceso de aparcamiento (Idna Idris et al., 2009).

Sistemas Parqueo Automatizado

La forma final de un parqueo inteligente que se discute aquí, es un sistema Parqueo Automatizado. En esos sistemas, cada paso del proceso de aparcamiento está automatizado. El conductor se detiene en una plataforma, deja su coche bloqueado y las máquinas se encarguen del resto. Este tipo de aparcamiento, obviamente permite almacenar muchos coches en un espacio relativamente pequeño ya que los conductores no necesitan el espacio adicional para conducir, para entrar y salir de sus vehículos. Además, ya que los conductores ya no se pasean en medio del espacio de estacionamiento, se aumenta la seguridad. Un estacionamiento automatizado puede ser implementado tanto en las instalaciones de aparcamiento existente o construir desde cero (Idna Idris et al., 2009).

2.1.5.2. Los beneficios de un sistema de Parqueo Inteligente

El propósito general de cualquier dispositivo IoT es hacer la vida más fácil. Al ser una aplicación de la IoT, se puede traducir este fin de uno-a-uno para sistemas de parqueo inteligente: apuntan a hacer la vida más fácil. Para ser más específicos, los sistemas de parqueadero inteligente tienen varias ventajas; no exclusivamente para los conductores, sino también para los propietarios de estacionamiento y el público en general. Una lista de los tres beneficios más importantes.

La primera y más relevante, los conductores pueden acortar el tiempo de búsqueda de estacionamiento (Rodier et al., 2008). De lo contrario debería conducir alrededor con la esperanza de encontrar un lugar de aparcamiento disponible, con un SPI, pueden conducir directamente a una ubicación o lugar libre. Y con los sistemas más avanzados, incluso pueden hacer una reserva y pagar de forma automática. Esto no sólo ahorra tiempo, combustible, sino también libera un poco la frustración de los conductores.

Esto nos lleva al segundo beneficio de SPI: una mejora del medio ambiente. SPI típicamente resulta en un alivio de la congestión del tráfico debido a que se evita que el conductor maneje a los alrededores con el fin de encontrar una plaza de aparcamiento. Menos vehículos en conducción en torno a la búsqueda de un lugar, significa una reducción de la contaminación del medio ambiente. Además, el ruido de la calle será menor y la seguridad mejorará.

Un tercer beneficio de un SPI, reside en una capacidad para registrar datos sobre el aparcamiento. Todos los datos son cuidadosamente registrados y se pueden utilizar para los análisis. Por ejemplo: ¿Cuándo la mayoría de los coches entra en el garaje? ¿Cuánto dura la estancia media? ¿Cuándo tenemos más disponibilidad? Sobre la base de estos datos, el dueño del garaje puede reconocer patrones y tratar de influir en ellos, por ejemplo, el uso de diferentes tasas por hora. Las estrategias de precios pueden ser aplicados con el fin de optimizar los beneficios de la empresa (Idna Idris et al., 2009).

2.1.6. Software Libre

El software libre ofrece al usuario cuatro libertades: libertad de uso, de estudio y modificación, de distribución y redistribución de las versiones modificadas (Free Software Foundation Europe, 2016). Adicionalmente existen licencias para software libre que lo garantizan y que le dan una cobertura legal, como por ejemplo la GPL, Licencia Pública General (GNU, 2016)

2.1.7. Hardware Libre

De acuerdo a la asociación de Hardware de fuentes abiertas (Open Source Hardware Association / Hardware Libre) es aquel hardware cuya disponibilidad de diseño es de acceso público con el objetivo de que cualquier persona se sienta en la libertad de analizar, modificar, distribuir o crear nuevos diseños a partir del diseño original (Open Source Hardware Association, 2016)

2.1.8. Arduino Yun

Arduino es una plataforma electrónica abierta para la elaboración de prototipos basados en software y hardware flexibles. Arduino puede obtener información de su entorno a través de sus pines de entrada de toda una extensa gama de sensores y puede controlar aquello que le rodea tales como motores, sensores, luces. El microcontrolador en Arduino se programa mediante el lenguaje de programación Arduino basado en Wiring y el entorno de desarrollo basado en Processing (Arduino, 2016)

La tarjeta electrónica Arduino Yun es una placa electrónica basada en ATmega32u4 y el Atheros AR9331 como lo muestra la Figura 4. El procesador Atheros es compatible con una distribución Linux basada en OpenWrt llamado OpenWrt-Yun. La Yun se distingue de otras placas Arduino ya que puede comunicarse con la distribución de Linux.

Entre las características principales de la tarjeta electrónica se encuentra que viene incorporado puerto Ethernet y WiFi, 20 pines de entrada y salida digitales, un cristal oscilador de 16 Mhz, una conexión USB, una cabecera ICSP y 3 botones de reinicio. Comunicación con la distribución Linux cargada, ofrece un poderoso ordenador en red con la facilidad de Arduino las características se muestran en la Tabla 1.

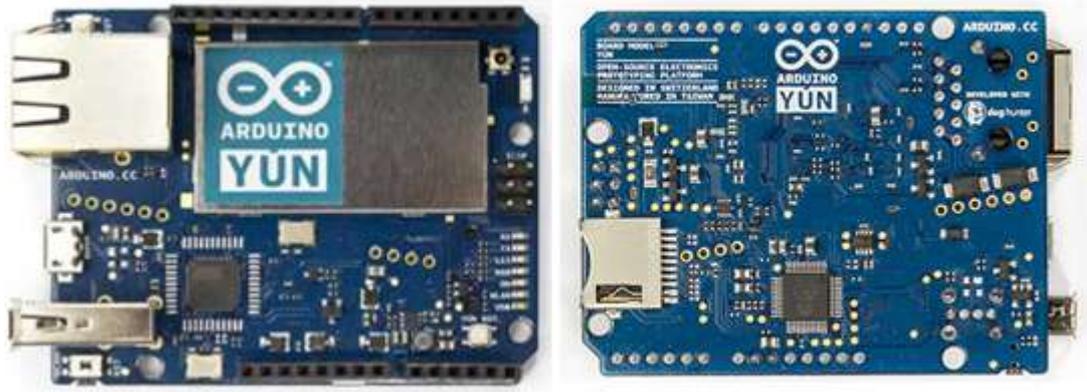


Figura 4: *Placa Arduino Yun*
(El Autor)

2.1.8.1 Hardware de Arduino Yun

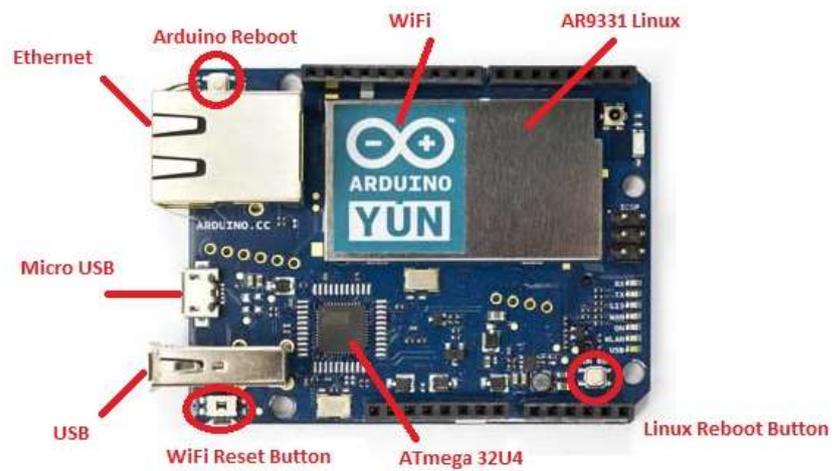


Figura 5: *Hardware de placa Arduino Yun*
(El Autor)

Tabla 1. Características de AVR Arduino microcontrolador

Microcontrolador	ATmega32U4
Voltaje de Operación	5V
Voltaje de Entrada	5V
Entradas/Salidas Digitales	20
Canales PWM	7
Entradas Analógicas	12
Corriente DC por Entrada/Salida	40 mA
Corriente DC para el Pin 3.3V	50 mA
Memoria Flash	32 KB de los cuales 4 son usados para el bootloader
SRAM	2.5 KB
EEPROM	1 KB
Velocidad del Reloj	16 Mhz

Nota: Rangos de operación del 1er. procesador que tiene el Arduino Yun
(Arduino, 2016)

2.1.8.2 Microprocesador Linux Atheros AR9331

El Atheros AR9331 es un poderoso integrado y rentable IEEE 80211n 1x1, 2.4GHz (SoC) para red de área local inalámbricas (WLAN), AP y plataformas ruteables, algunas de sus características se muestran en la Tabla 2.

Tabla 2. Microprocesador Linux

Procesador	Atheros AR9331
Arquitectura	MIPS @400MHz
Voltaje de Operación	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB tipo A	2.0 Host
Lector de Tarjeta	Solo Micro-SD
RAM	64 MB DDR2
Memoria Flash	16 MB
SRAM	2.5 KB
EEPROM	1KB

Velocidad Reloj	16 MHz
PoE compatible 802.3af soporta tarjeta	

Nota: Rangos de operación del Atheros AR9331

(Arduino, 2016)

Wi-Fi

La Arduino Yun utiliza el estándar IEEE 802.11n WiFi / WLAN utiliza tecnologías como OFDM y MIMO para lograr altas velocidades de transmisión

Puerto Ethernet

Otra manera de proporcionar conectividad de red la Arduino Yun posee un conector RJ45 que va integrado y brinda conectividad a 10/100 Mbps

Botones de Reset

Yun RST

Restablece el microprocesador AR9331, hará el reinicio del sistema Linux, borrara todos los datos almacenados en la memoria RAM y todos los programas que se estén ejecutando se darán por terminado.

32U4 RST

Resetea el microcontrolador ATmega32Ua. Normalmente se añade un botón de reinicio para escudos que bloquean la tarjeta.

WLAN RST

Este botón tiene una función doble. Primero sirve para restablecer la conexión WiFi a la configuración de fábrica, la cual consiste en poner el WiFi del Yun en modo punto

de acceso (AP) y asignarle la dirección IP por defecto 192.168.240.1, en esta condición se puede conectar con el computador a la red WiFi que aparece con el nombre SSID "XXXXXXXXXXXXXXXXXXXX". Una vez conectado se puede ingresar al entorno web de la Yun. Tener en cuenta que esto provocará el reinicio del entorno Linux. Para realizar esto debe tener presionado el botón por 5 segundos.

La segunda función es restaurar la imagen de Linux a la imagen de fábrica por defecto. Para lograr esto se debe presionar el botón por 30 segundos, tenga en cuenta que esto hará perder todos los archivos guardados y los programas informáticos instalados en la Flash del AR9331

2.1.8.3 Energía y Alimentación

La tarjeta se energiza a través de la conexión micro-USB con 5VDC regulado, dado que esta no posee un regulador de voltaje para picos más altos que pudieran dañar la tarjeta. Estos 5VDC se utilizan para alimentar los microcontroladores y otros componentes en la placa.

2.1.9. Plataforma Temboo

Temboo es un servicio que sirve de intermediario para conectarse con otras aplicaciones Web. Permite integrar diversas plataformas para que se comuniquen entre sí. Con Temboo se puede usar desde Arduino las APIs de Twitter, Facebook, Instagram, YouTube, Xively, MySQL o incluso PayPal (Temboo, 2016)

2.1.10. Twitter

Es un servicio de microblogging gratuito de red social que permite a los usuarios registrados transmitir mensajes cortos llamados tweets, es importante saber que solo puedes enviar 1000 tweets por día. Los miembros pueden transmitir tweets y seguir los tweets de otros usuarios mediante el uso de múltiples plataformas y dispositivos, están limitados a 140 caracteres (Rouse & McMahon, 2015)

2.1.11. Sensor SRF02

Se denomina sensor a un “sistema de medida” formado por, transductor, acondicionador de señal y sensor, que permite la medida de valores de ciertas características tales como: físicas, químicas, mecánicas, etc. Para tal objetivo se adapta la señal obtenida desde el transductor (mV).

Se denomina transductor a todo dispositivo que convierte una señal de una forma física en una señal correspondiente, pero de otra forma física distinta. Es decir, que convierte un tipo de energía en otro. Esto significa que la señal de entrada es siempre energía o potencia, pero al medir una de las componentes de la señal suele ser tan pequeña que puede despreciarse, y se interpreta que se mide solo la otra componente.

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida transfusible que es función de la variable medida.

Sensor y transductor se emplean a veces como sinónimos, pero sensor sugiere un significado más extenso: la ampliación de los sentidos para adquirir un conocimiento de cantidades físicas que, por su naturaleza o tamaño, no pueden ser percibidas directamente por los sentidos. Transductor, en cambio, sugiere que la señal de entrada y de salida no debe ser homogéneas (Arney, 2003).

Para el trabajo de graduación se utiliza el sensor SRF02 tiene en su placa de circuito impreso el Transductor Piezoeléctrico terminales de conexión para polarización, comunicación y un circuito integrado que sirve como interfaz de comunicaciones I²C, con una capacidad máxima total de 16 dispositivos diferentes bajo este protocolo. El sensor se muestra en la figura 6 y sus características se detallan en la Tabla 3.



Figura 6: *Sensor Ultrasonico SRF02*
(El autor)

Tabla 3. *Características del SRF02*
(MSE Microsystems Engineering, 2016)

Características del Sensor Ultrasonico SRF02
Tensión de alimentación – 5Vcc
Consumo típico 4mA
Frecuencia ultrasónica de 40KHz
Rango de medidas de 15 cm hasta 6 m
Control automático de ganancia
Interface serie estándar o protocolo bus I ² C
Sintonía automática
Medida expresada en uS, cm o pulgadas
Reducido tamaño de 24 x 20 x 17 mm

2.1.12. Microcontrolador Atmega328

Es un circuito integrado que contiene mucha de las mismas cualidades de una computadora de escritorio, tales como la CPU, la memoria, etc., pero no incluye ningún dispositivo de “comunicación con humanos”, como monitor, teclados o mouse. Los microcontroladores son diseñados para aplicación de control de máquinas más que para interactuar con humanos (Parallax, 2003).

Para el proyecto de graduación se utiliza el Microcontrolador Atmega328, el cual es de alto rendimiento de 8 bit con arquitectura RISC de la fábrica ATmel, con características similares al PIC de la fábrica microchip sin embargo este Atmega puede ser configurado haciendo uso de la plataforma de Arduino, ATmega328 se

muestra en la Figura 7 y algunas de sus características más importantes están en la Tabla 4.

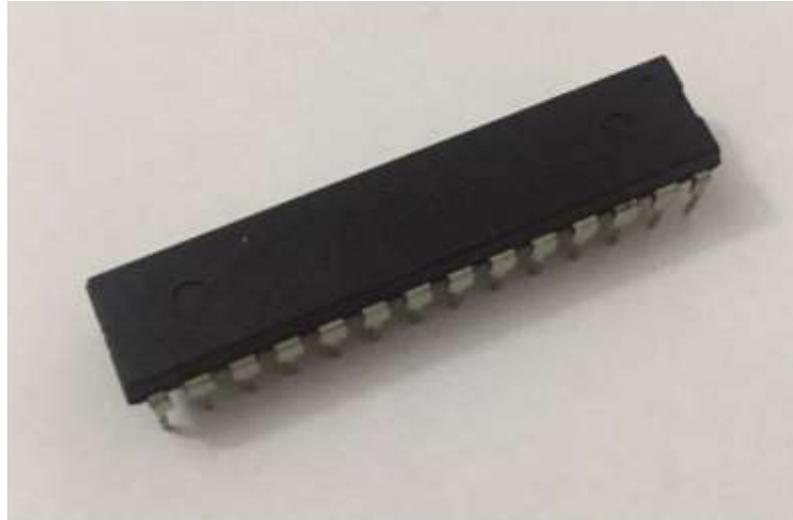


Figura 7: *Microcontrolador ATmega328*
(El autor)

Tabla 4. *Características Atmega328*

Características	ATmega328
Número de Pines	28/32
Flash (Bytes)	32K
SRAM (Bytes)	2K
EEPROM (Bytes)	1K
Propósito General Líneas I/O	23
SPI	2
TWI (I ² C)	1
USART	1
ADC	10-bit 15kSPS
ADC Canales	8
8-bit Timer/Couters	2
16-bit Timer/Counters	1

2.1.13. Cristal Oscilador de Cuarzo

Es un elemento que trabaja mediante el efecto de piezoelectricidad, el cual cuando se aplica una diferencia de potencial entre sus terminales de conexión se deforma. Este principio es aprovechado en la electrónica, dado que la frecuencia de deformación de dicho elemento, depende de su forma, tamaño y longitud.

Cristal de cuarzo de alta calidad con encapsulado metálico como lo muestra la Figura 8 y una frecuencia de operación de 16Mhz. Es ideal como oscilador externo para un microcontrolador.



Figura 8: *Cristal 16Mhz*
(El Autor)

2.1.14. Resistencia Eléctrica

La resistencia eléctrica es toda oposición a la circulación de corriente en un circuito eléctrico cerrado. Para este trabajo de graduación se utiliza resistencias de cerámica con un valor de 270 Ohm como lo muestra la Figura 9, la cual son utilizadas para limitar la corriente de circulación en la conexión hacia los Diodos Leds.

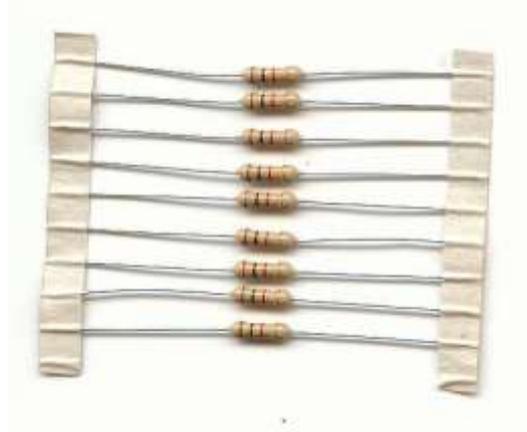


Figura 9: Resistencia
(El Autor)

2.1.15. Diodos Led Dual

Dispositivo semiconductor que emite luz incoherente de espectro reducido cuando se polariza de forma directa. La intensidad de la luz es directamente proporcional dependiendo de la cantidad de corriente aplicada en el mismo.

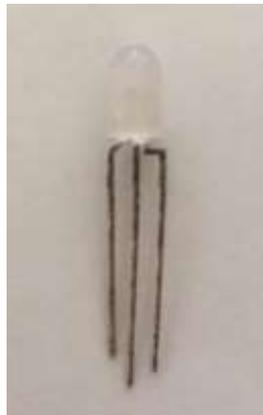


Figura 10: Diodo Led Dual
(El Autor)

2.1.16 Borneras de Conexión

Accesorio plástico con terminales metálicos que sirven para conectar distintos dispositivos por medio de cables.



Figura 61: *Bornera*
(El Autor)

2.1.17. Proteus

Es un software que sirve para diseño y realización de proyectos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción.

2.2. Metodología

Para el desarrollo del presente proyecto de graduación se aplicaron dos métodos de investigación:

2.2.1. Método Experimental

Realizando pruebas de funcionamiento del sistema (hardware-software) y el **método de campo**.

- Método Experimental
- Método de Campo

2.2.2. Método Teórico

Se realiza levantamiento de información tales como:

- Funcionamiento de Arduino Yun
- Funcionamiento de sensor ultrasónico
- Funcionamiento de microcontrolador Atmega328
- Búsqueda de información en sitios web

2.2.3. Método Inductivo

Se hace observaciones tratando de obtener conclusiones de carácter general desde la acumulación de datos específicos, este se aplica para la definición de los parámetros a sensor y tecnología a utilizar.

2.3. Técnicas

Las técnicas que se utilizaron para el desarrollo de la investigación fueron las siguientes:

2.3.1. Técnica Experimental

Se realizarán análisis a nivel experimental para generar ideas con el objetivo de dar solución o reducir la problemática diagnosticada. La evaluación experimental se va a analizar mediante la implementación real para detectar la factibilidad de un funcionamiento adecuado, verificando los resultados deseados.

2.3.2. Técnica Documental

Para llevar a cabo este proyecto se utilizará varios tipos de investigación, como investigación bibliográfica, donde se tomarán en cuenta otros sistemas similares para tener un soporte del desarrollo del mismo con otras tecnologías. Investigación de

campo, donde se determinará las características físicas y medidas de seguridad del sistema al momento de su implementación.

2.4. Población y Muestra

2.4.1. Población

El proyecto de titulación está dirigido a los usuarios profesores y del área administrativa que hacen uso del parqueadero del bloque “B” de la Universidad Politécnica Salesiana.

2.4.2. Muestra

Para este trabajo de titulación la muestra será tomada de 12 parqueaderos designados para profesores y personal administrativo en el bloque “B” de la Universidad Politécnica Salesiana.

2.5. Descripción de la Propuesta

El sistema de Parqueo Inteligente donde el autor pretende desarrollar un sistema que sea amigable, eficaz y sencillo, que busca principalmente brindar información anticipada sobre la disponibilidad de parqueos en un sector asignado del bloque “B” de tal manera que se logra optimizar el tiempo, mejorar el flujo vehicular, ahorro de energía de los usuarios, y que esta información sea de fácil acceso mediante el uso de Twitter y que el usuario pueda tomar decisiones oportunas antes de ir al sitio.

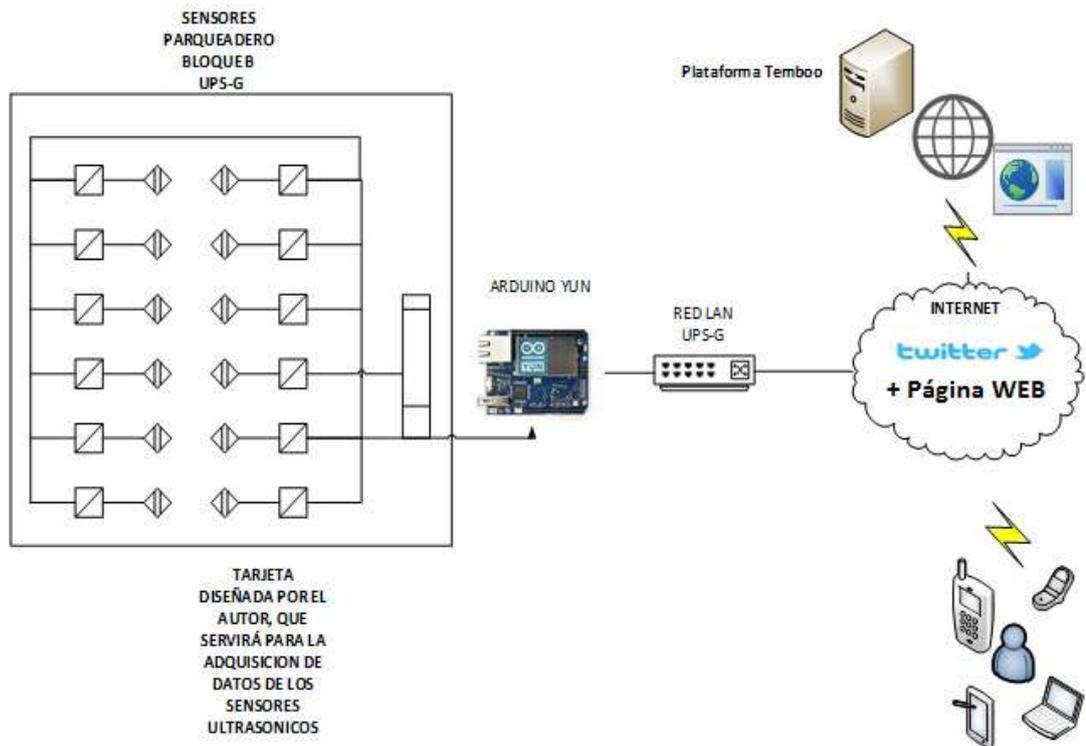


Figura 72: Diagrama de funcionamiento del sistema desarrollado.
(El Autor)

2.6. Beneficiarios

Los docentes de la UPS-G, docentes del Colegio Domingo Comín y personal administrativo que se encuentran dentro de los principales beneficiarios.

2.7. Impacto

Un aspecto a destacar es que al tener información anticipada, provoca impactos positivos en la optimización del tiempo, recursos, reducción de la densidad de tránsito que circula en los alrededores.

Tener un sistema centralizado facilita el acceso a la información de forma independiente al lugar donde el usuario se quiere dirigir y no será necesario ir al lugar para constatar si hay lugares de parqueo disponible.

CAPITULO 3

RESULTADOS

3.1. Implementación y Desarrollo

En esta sección se detallan los criterios de diseño del sistema desarrollado; se presentan los procesos de diseño y construcción de una tarjeta electrónica para recepción de señales transmitidas por cada uno de los sensores ultrasónicos y como se debe configurar nuestra tarjeta principal Arduino Yun & Temboo para que administre los datos y sean visualizados vía Twitter, así como la instalación de la infraestructura donde se instalará el sistema.

3.1.1. Diseño de Circuito Impreso (PCB)

Para poder fabricar una PCB (*Print Circuit Board*), en primer lugar se necesita dibujar las pistas y los pads que interconectarán los elementos, con la ayuda de un computador y un software que permita llevar a cabo esta tarea. En este caso se ha utilizado Proteus 8 Professional para el diseño, cuyo diagrama esquemático se muestra en la Figura 13.

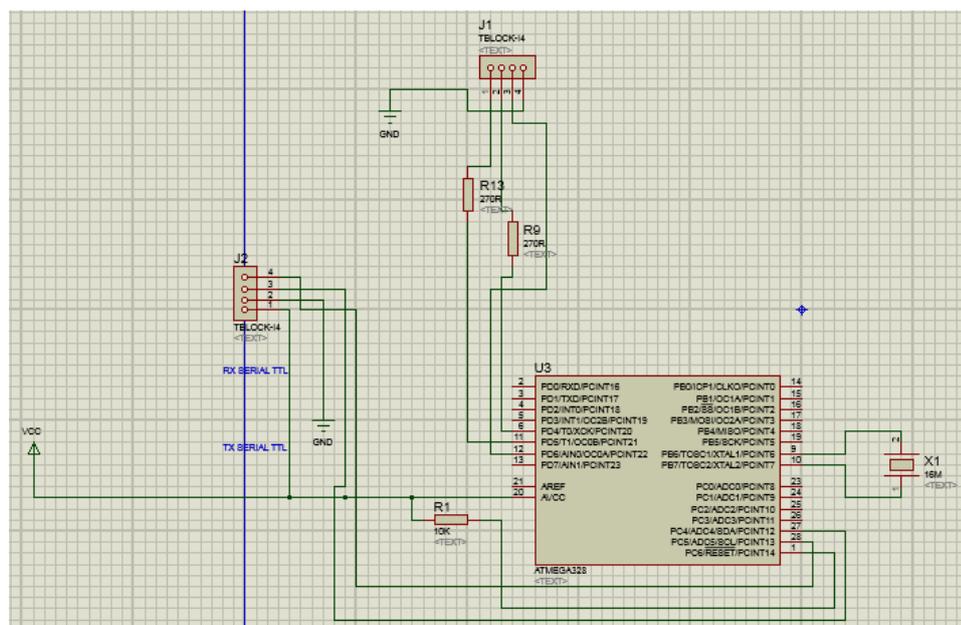


Figura 83: Diagrama esquemático realizado en Proteus 8 Professional
(El Autor)

El diagrama de pistas impresa sobre la placa conductora de cobre, como se muestra en la Figura 14. Para la elaboración de la PCB se siguió el proceso artesanal.

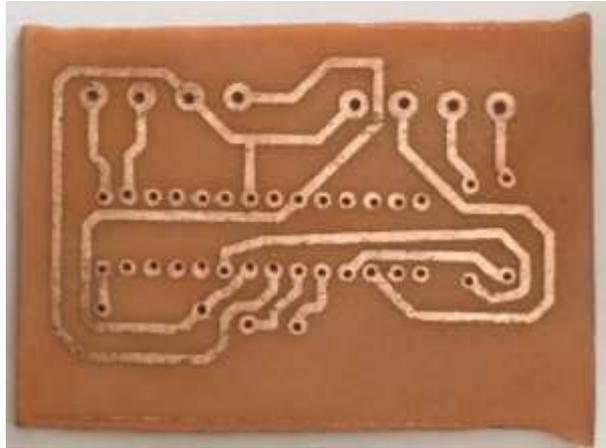


Figura 94: *Diagrama de pistas impresa de la PCB diseñada
(El Autor)*

3.1.2. Configuración de Microcontrolador ATmega328

Como paso inicial se utilizará una tarjeta Arduino UNO el cual será nuestra interface de programación como lo muestra la Figura 15. Se utilizará como modelo de programación el código que se encuentra en la siguiente dirección obtenida de internet: <https://www.arduino.cc/en/Tutorial/SFRRangerReader>, este es código clasificado como Software Libre, y este puede ser modificado para el proposito.

Este código debe ser modificado para que el sensor mida los vehículos existentes en su área de cobertura, si hay un vehículo la salida setea un 1 digital y si no hay vehículo será un 0 digital.



Figura 105: *Interface hardware de programación Arduino UNO*
(El Autor)

Código para el microcontrolador Atmega328:

```
// I2C SRF10 or SRF08 Devantech Ultrasonic Ranger Finder
// by Nicholas Zambetti <http://www.zambetti.com>
// and James Tichenor <http://www.jamestichenor.net>
// Demonstrates use of the Wire library reading data from the
// Devantech Ultrasonic Rangers SFR08 and SFR10
// Created 29 April 2006
// This example code is in the public domain.

#include <Wire.h>
void setup() {
  Wire.begin();          // join i2c bus (address optional for master)
  Serial.begin(9600);    // start serial communication at 9600bps
}
int reading = 0;
void loop() {
  // step 1: instruct sensor to read echoes
  Wire.beginTransmission(112); // transmit to device #112 (0x70)
  // the address specified in the datasheet is 224 (0xE0)
  // but i2c addressing uses the high 7 bits so it's 112
  Wire.write(byte(0x00)); // sets register pointer to the command register (0x00)
```

```

Wire.write(byte(0x50)); // command sensor to measure in "inches" (0x50)
// use 0x51 for centimeters
// use 0x52 for ping microseconds
Wire.endTransmission(); // stop transmitting

// step 2: wait for readings to happen
delay(70); // datasheet suggests at least 65 milliseconds
// step 3: instruct sensor to return a particular echo reading
Wire.beginTransmission(112); // transmit to device #112
Wire.write(byte(0x02)); // sets register pointer to echo #1 register (0x02)
Wire.endTransmission(); // stop transmitting

// step 4: request reading from sensor
Wire.requestFrom(112, 2); // request 2 bytes from slave device #112

// step 5: receive reading from sensor
if (2 <= Wire.available()) { // if two bytes were received
  reading = Wire.read(); // receive high byte (overwrites previous reading)
  reading = reading << 8; // shift high byte to be high 8 bits
  reading |= Wire.read(); // receive low byte as lower 8 bits
  Serial.println(reading); // print the reading
}

delay(250); // wait a bit since people have to read the output :)
}

/*

// The following code changes the address of a Devantech Ultrasonic Range Finder
(SRF10 or SRF08)
// usage: changeAddress(0x70, 0xE6);
void changeAddress(byte oldAddress, byte newAddress)
{
  Wire.beginTransmission(oldAddress);
  Wire.write(byte(0x00));
  Wire.write(byte(0xA0));
  Wire.endTransmission();
}

```

```

Wire.beginTransmission(oldAddress);
Wire.write(byte(0x00));
Wire.write(byte(0xAA));
Wire.endTransmission();

Wire.beginTransmission(oldAddress);
Wire.write(byte(0x00));
Wire.write(byte(0xA5));
Wire.endTransmission();

Wire.beginTransmission(oldAddress);
Wire.write(byte(0x00));
Wire.write(newAddress);
Wire.endTransmission();
}
*/

```

3.1.3. Configuración de Arduino Yun & Temboo para envío de Tweets y página Web

Esta tarjeta tiene la capacidad de interactuar con el mundo real haciendo uso de sus interfaces Ethernet y Wifi, equivalente a la tarjeta Leonardo o Arduino Uno con shield Ethernet. Puede ser utilizada para diversas aplicaciones, entre las cuales está el envío de emails, SMS y para esto proyecto de graduación: utilizar Twitter haciendo uso de una cuenta que se configura internamente.

Para que Arduino Yun pueda publicar Tweets, se usa la plataforma Temboo, donde es necesario utilizar, modificar y cargar unas librerías disponibles en internet que nos ayudaran a interactuar con Twitter y la tarjeta Arduino Yun. Para realizarlo se debe seguir los siguientes pasos:

Paso 1:

Crear una cuenta en la plataforma Temboo en la siguiente dirección <https://temboo.com/signup/> como lo muestra la Figura 16



Figura 116: Portal Temboo

(El Autor)

Paso 2:

Crear una cuenta en la plataforma Twitter en la siguiente dirección <https://twitter.com/> como lo muestra la Figura 17.



Figura 127: Portal Twitter

(El Autor)

Paso 3:

Crear la aplicación en Twitter ingresando a la siguiente dirección <https://apps.twitter.com/app/new> y copiar API Keys. La cual debe ser copiada en el Sketch de Arduino Yun al momento de programar.

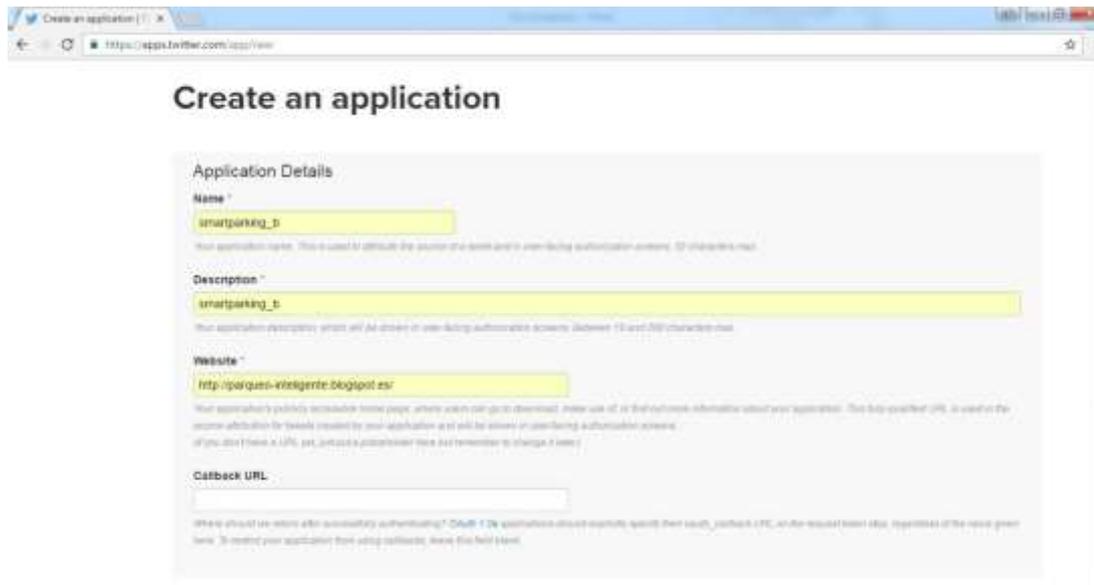


Figura 138: Portal App Twitter
(El Autor)

Paso 4:

Utilizar el modelo de Sketch existente en la plataforma Temboo que servirá para programar el Arduino Yun con la finalidad de enviar Tweets, y posterior a esto modificar este código fuente para el objetivo de este proyecto de graduacion.

Código fuente Temboo & Arduino Yun:

```
/*
```

```
SendATweet
```

Demonstrates sending a tweet via a Twitter account using the Temboo Arduino Yun SDK.

This example code is in the public domain.

```
*/

#include <Bridge.h>
#include <Temboo.h>
#include "TembooAccount.h" // contains Temboo account information

/** SUBSTITUTE YOUR VALUES BELOW: */

// Note that for additional security and reusability, you could
// use #define statements to specify these values in a .h file.
const String TWITTER_ACCESS_TOKEN = "your-twitter-access-token";
const String TWITTER_ACCESS_TOKEN_SECRET = "your-twitter-access-token-
secret";
const String TWITTER_API_KEY = "your-twitter-api-key";
const String TWITTER_API_SECRET = "your-twitter-api-secret";

int numRuns = 1; // execution count, so this sketch doesn't run forever
int maxRuns = 1; // the max number of times the Twitter Update Choreo should run

void setup() {
  Serial.begin(9600);

  // for debugging, wait until a serial console is connected
  delay(4000);
  while(!Serial);

  Bridge.begin();
}

void loop()
{
  // only try to send the tweet if we haven't already sent it successfully
  if (numRuns <= maxRuns) {

    Serial.println("Running SendATweet - Run #" + String(numRuns++) + "...");
```

```

    // define the text of the tweet we want to send
    String tweetText("My Arduino Yun has been running for " + String(millis()) + "
milliseconds.");

    TembooChoreo StatusesUpdateChoreo;

    // invoke the Temboo client
    // NOTE that the client must be reinvoiced, and repopulated with
    // appropriate arguments, each time its run() method is called.
    StatusesUpdateChoreo.begin();

    // set Temboo account credentials
    StatusesUpdateChoreo.setAccountName(TEMBOO_ACCOUNT);
    StatusesUpdateChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
    StatusesUpdateChoreo.setAppKey(TEMBOO_APP_KEY);

    // identify the Temboo Library choreo to run (Twitter > Tweets > StatusesUpdate)
    StatusesUpdateChoreo.setChoreo("/Library/Twitter/Tweets/StatusesUpdate");

    // set the required choreo inputs
    // see https://www.temboo.com/library/Library/Twitter/Tweets/StatusesUpdate/
    // for complete details about the inputs for this Choreo

    // add the Twitter account information
    StatusesUpdateChoreo.addInput("AccessToken", TWITTER_ACCESS_TOKEN);
    StatusesUpdateChoreo.addInput("AccessTokenSecret",
TWITTER_ACCESS_TOKEN_SECRET);
    StatusesUpdateChoreo.addInput("ConsumerKey", TWITTER_API_KEY);
    StatusesUpdateChoreo.addInput("ConsumerSecret", TWITTER_API_SECRET);

    // and the tweet we want to send
    StatusesUpdateChoreo.addInput("StatusUpdate", tweetText);

    // tell the Process to run and wait for the results. The
    // return code (returnCode) will tell us whether the Temboo client
    // was able to send our request to the Temboo servers
    unsigned int returnCode = StatusesUpdateChoreo.run();

```

```

// a return code of zero (0) means everything worked
if (returnCode == 0) {
    Serial.println("Success! Tweet sent!");
} else {

    // a non-zero return code means there was an error
    // read and print the error message
    while (StatusesUpdateChoreo.available()) {
        char c = StatusesUpdateChoreo.read();
        Serial.print(c);

    }
}

StatusesUpdateChoreo.close();

// do nothing for the next 90 seconds
Serial.println("Waiting...");
delay(90000);

}

}

```

3.1.3.1 Diseño Web e integración con Twitter

Como alternativa para los usuarios que no tengan una cuenta Twitter se procede al diseño de una página web utilizando protocolo de transferencia de hipertexto (HTTP), que es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador, Tablet o celular inteligente.

Con los siguientes pasos vamos a integrar la información de Twitter a nuestra página Web:

1. Dando por hecho que tenemos nuestro perfil de Twitter creado, accedemos a la siguiente URL: <https://twitter.com/settings/widgets> donde vamos a crear el widget que incluiremos en nuestra página web.

Hacemos clic en “**Crear nuevo**”.

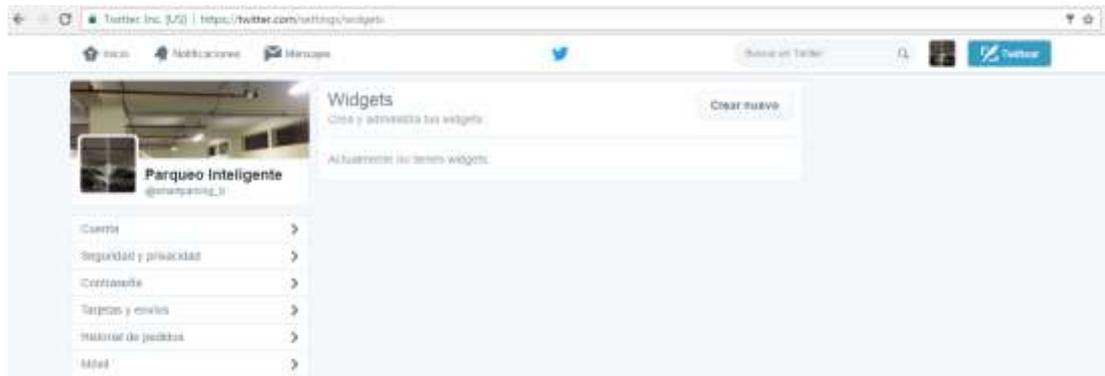


Figura 19: Creación de widget
(El Autor)

2. Observamos cuatro pestañas. Nosotros vamos a ver la primera. “**Cronología de usuario**” es la que nos interesa para incrustar los tweets en nuestra web. Podemos personalizar el widget según las opciones que nos muestran y una vez terminado hacemos clic en “**Crear Widget**” donde se nos ofrece un código que debemos pegar en el HTML de nuestra página, allí donde queremos que aparezca exactamente.

3.1.4. Diseño de Infraestructura para el sistema

Para el montaje de nuestras tarjetas receptoras y sensores se realizó el siguiente diseño como lo muestra Figura 20 que consisten en la instalación de tuberías metálicas, cajas de paso cuadradas metálicas, cajas de paso octogonales plásticas y cableado para cada dispositivo.

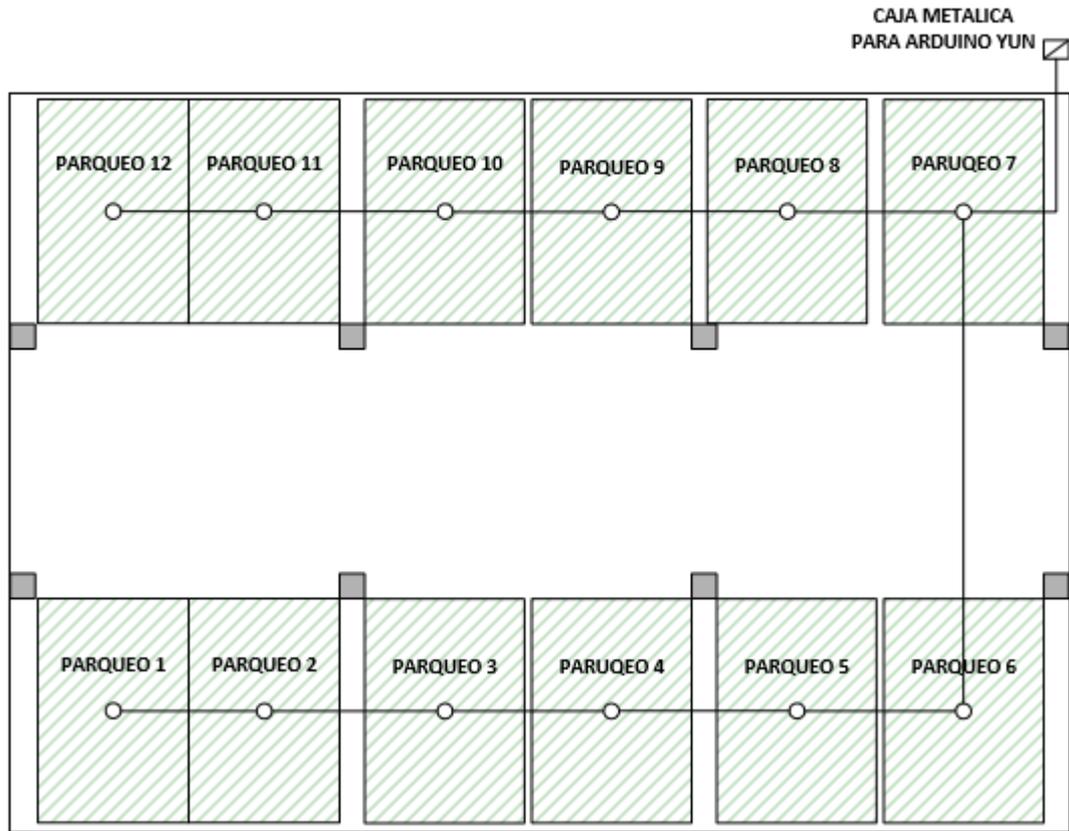


Figura 20: *Diagrama de ductería instalada*
(El Autor)

CAPITULO 4

ANÁLISIS DE RESULTADOS

En esta sección se analizan los resultados obtenidos del Trabajo de Titulación con los cuales se comprobó el funcionamiento del sistema. Para ello se realizaron diagramas de funcionamiento y diagramas bloques del sistema y se evaluaron los dispositivos instalados tanto a nivel de hardware y software, así como el de su infraestructura, para cumplir con cada uno de los objetivos planteados para este trabajo de graduación.

4.1. Desarrollo de diagramas de funcionamiento y bloques del sistema

La Figura 21 describe el diagrama de funcionamiento del sistema desarrollado. En primer lugar, hay el dispositivo Arduino Yun enviando información a Temboo constantemente y publicándolo vía Twitter, donde el usuario puede hacer su consulta respectiva usando cualquier dispositivo con acceso a internet, mediante la aplicación propia de Twitter como lo muestra la Figura 22.

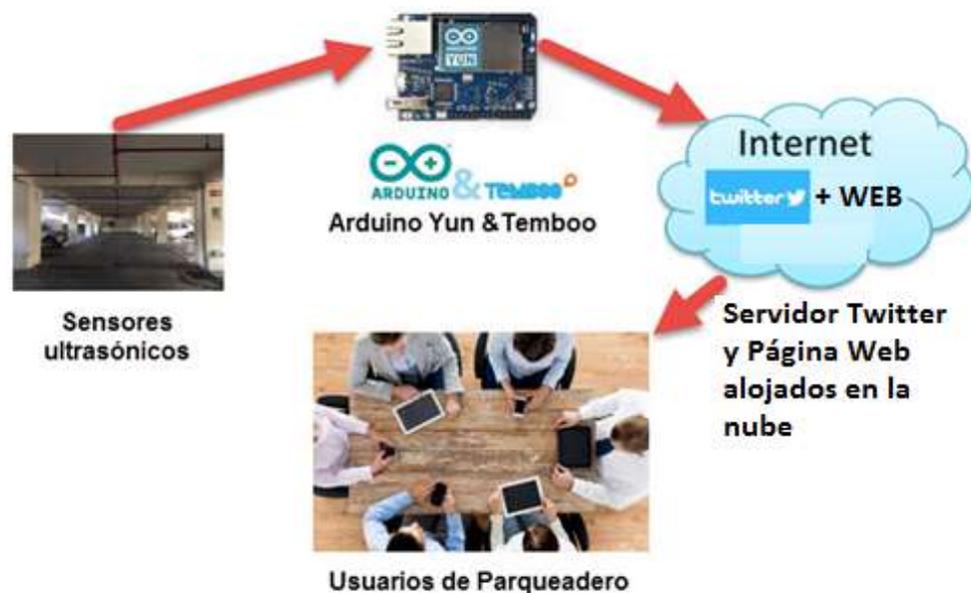


Figura 21: Diagrama de funcionamiento del sistema desarrollado.

(El Autor)



Figura 142: *Aplicación Twitter.*

(El Autor)

El diagrama de bloques del sistema desarrollado se muestra en la Figura 23. Según este diagrama, el sistema consta de tres bloques principales: el Bloque 1 corresponde al Sensor Ultrasónico y tarjeta electrónica receptora de datos del sensor, el Bloque 2 hace referencia a la Arduino Yun & Temboo que es un conjunto de Hardware Libre y Software Libre que tiene como función principal procesar la información obtenida por cada uno de los sensores, administrarla, almacenarla en Temboo y enviarla a la nube (Servidores Twitter) y el Bloque 3 corresponde a los diferentes métodos de visualización de la información haciendo uso de dispositivos inteligentes, la información corresponde a la disponibilidad de lugares disponibles para parquear. Cabe mencionar que cada parte y cada bloque, de los distintos diagramas, se describen y analizan sus resultados con mayor detalle en las secciones posteriores de este documento.

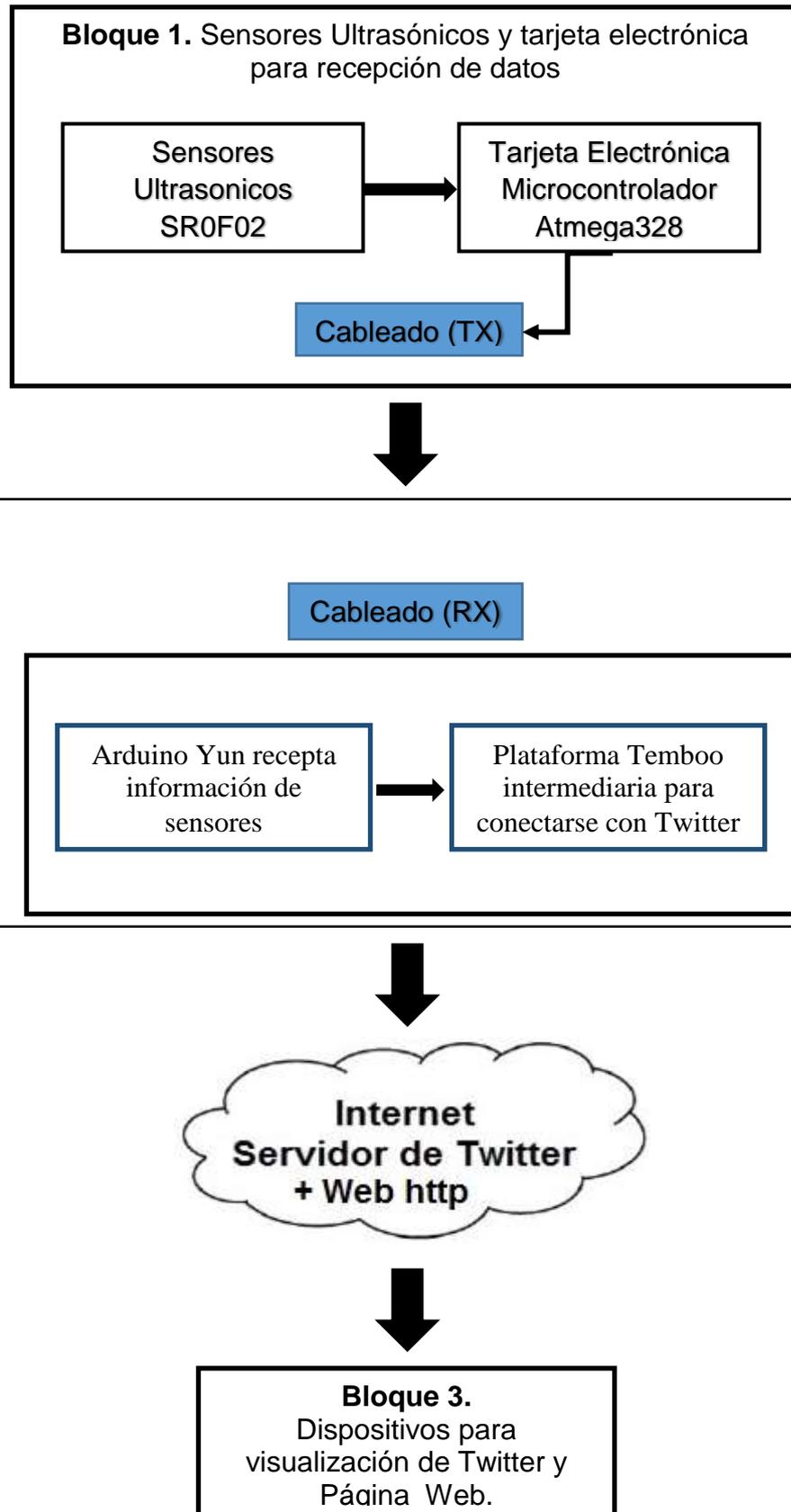


Figura 23: Diagrama de bloques del sistema desarrollado.

(El Autor)

4.2. Análisis del desarrollo de Tarjeta Electrónica de Adquisición de Datos

Como se describió en el diagrama de bloques de la Figura 23 el conjunto de dispositivos sensor ultrasónico y tarjeta electrónica de adquisición de datos corresponde al Bloque 1.

Se soldaron todos los elementos electrónicos en la PCB, terminando de esta manera la construcción de la Tarjeta Electrónica para recepción de Datos como se muestra en la Figura 24. Posterior a esto es necesario la modificación del código fuente (Software Libre) publicado en la red y cargado al microcontrolador ATmega328, el cual lo se muestra a continuación.

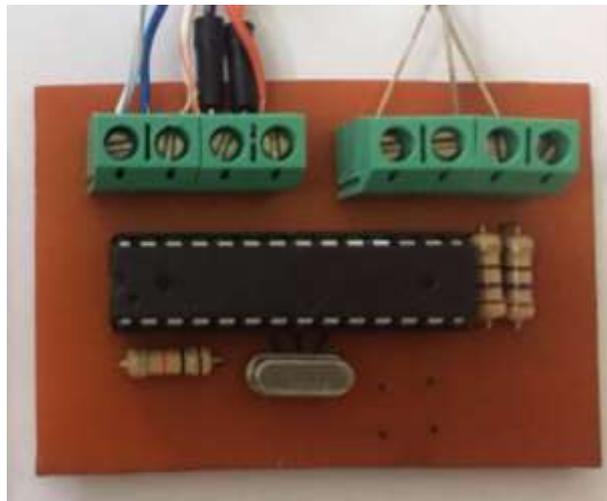


Figura 24: PCB de la Tarjeta Electrónica para recepción de Datos
(El Autor)

Código fuente modificado para microcontrolador ATmega328 (Software Libre)

Programación nueva:

/*

Sample code for test the SRF02 with the I2C mode based on Arduino UNO!

Command for reference:<http://robot-electronics.co.uk/html/srf02techI2C.htm>

Connection:

SRF02 Arduino

5v Vcc -> 5V

```

SDA    -> A4
SCL    -> A5
Mode   -> no connection
0v Ground -> GND
*/

#include <Wire.h>

const int led5 = 5;
const int led4 = 4;
const int led6 = 6;
void setup()
{
  Wire.begin();          // join i2c bus (direccion opcional para el dispositivo master)
  Serial.begin(9600);    // iniciar la comunicacion serial a 9600bps
  pinMode(led5, OUTPUT); // Definicion de la salida digital
  pinMode(led4, OUTPUT); // Definicion de la salida digital
  pinMode(led6, OUTPUT); // Definicion de la salida digital
}

int reading = 0;
void loop()
{
  // paso 1: indicarle al sensor como leer la rafaga de señal
  Wire.beginTransmission(112); // transmitir al dispositivo #112 (0x70)
  // la direccion especificada en el datasheet es 224 (0xE0)
  // pero el direccionamiento i2c usa los altos 7 bits asi esto es 112
  Wire.write(byte(0x00)); // coloca el registro puntero a el registro de comandos
(0x00)
  Wire.write(byte(0x51)); // comando del sensor para medir en "centimetros"
(0x51)
  // use 0x51 para centimetros
  Wire.endTransmission(); // finaizar la transmision

  // step 2: esperar la lectura de datos
  delay(70); // el datasheet sugiere aproximadamente 65 milisegundos
  // step 3: instruccion para que el sensor reciba la señal de rafaga de datos

```

```

Wire.beginTransmission(112); // transmitir al dispositivo #112
Wire.write(byte(0x02)); // coloca el registro puntero a el registro echo #1 (0x02)
Wire.endTransmission(); // para de transmision

// step 4: peticion de lectura del sensor
Wire.requestFrom(112, 2); // peticion de 2 bytes del dispositivo esclavo #112

// step 5: recibiendo lectura del sensor
if (2 <= Wire.available()) // if two bytes were received
{
    reading = Wire.read(); // recibir los bytes mas significativos (sobrescribir la
lectura previa)
    reading = reading << 8; // cambiar el byte mas alto a los 8 bits mas
significativos
    reading |= Wire.read(); // recibir el byte menos significativo como 8 bits menos
significativos
}
Serial.print(reading); // imprimir la lectura en el puerto serial
Serial.println("cm");

if(reading <=200){ // si la distancia <200, la salida es 1 digital
    digitalWrite(led5,HIGH); //salida digital alta
    digitalWrite(led4, LOW); //salida digital baja
    digitalWrite(led6,HIGH); //salida digital alta
}

if(reading >=200){ // si la distancia >200, la salida es 0 digital
    digitalWrite(led5,LOW); //salida digital baja
    digitalWrite(led4,HIGH); //salida digital alta
    digitalWrite(led6,LOW); //salida digital baja

// Serial.print(led5);
}
{
    delay(250); // espera para que se pueda realizar la lectura :)
}
}

```

De esta forma se puede manejar los umbrales de distancia de los sensores ultrasónicos SRF02 y obtener las señales digitales a la salida de nuestra tarjeta, dependiendo de la existencia de algún vehículo frente al sensor. Para el caso en este proyecto se configura que para una distancia mayor a 200 cm (2mts) y setee un 0 digital, en otras palabras un espacio libre y para una distancia menor a 200 cm (2mts) setee un 1 digital, indicando que el lugar para parquear esta ocupado.

Finalizado el ensamblaje de la tarjeta y configuración de cada uno de los microcontroladores ATmega328, se procedió a conectar sensor SRF02 a su respectiva tarjeta, se inician las primeras pruebas respectivas en laboratorio obteniendo los resultados deseados como lo muestran en las Figuras 25 y 26.

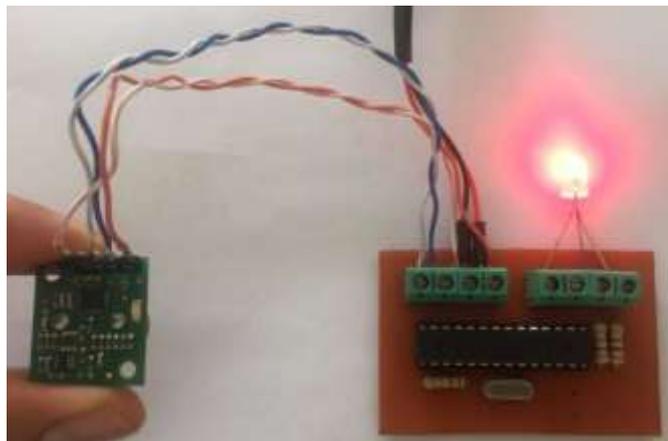


Figura 25: *Placa Receptora de datos sin obstáculo en el sensor*
(El Autor)

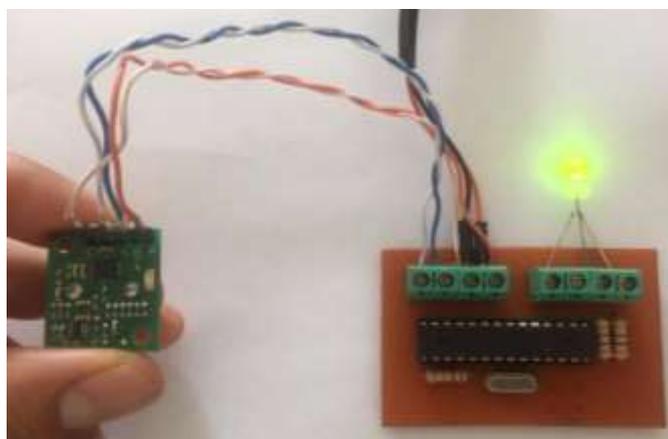


Figura 26: *Placa Receptora de datos con obstáculo en el sensor*
(El Autor)

Por cada sensor habrá una señal que ingresará al microcontrolador que convertirá la presencia de un vehículo de acuerdo a los umbrales configurados en la salida digital. Esta salida digital ingresará a un puerto de la tarjeta Arduin Yun, quien contará con la programación adecuada para que basado en las entradas digitales y su ubicación puede contabilizar la cantidad de puestos disponibles en el parqueo.

Los umbrales de distancia deben ser configurados basado al lugar en el que serán instalados los sensores.

4.3. Análisis de Instalación de Infraestructura del Sistema

Para la siguiente fase del trabajo fue necesario instalar una infraestructura metálica como lo muestra la Figura 27 que consta de tubería galvanizada, cajas de paso metálicas y plásticas, codos y cableado, donde se instalaron las 12 tarjetas receptoras de datos y sus respectivos sensores ultrasónicos como lo muestra la Figura 28 y 12 cajas reflectoras con leds rojos y verdes como otra alternativa para visualizar un lugar disponible Figura 29.



Figura 27: *Infraestructura Instalada en el parqueadero del bloque B*
(El Autor)

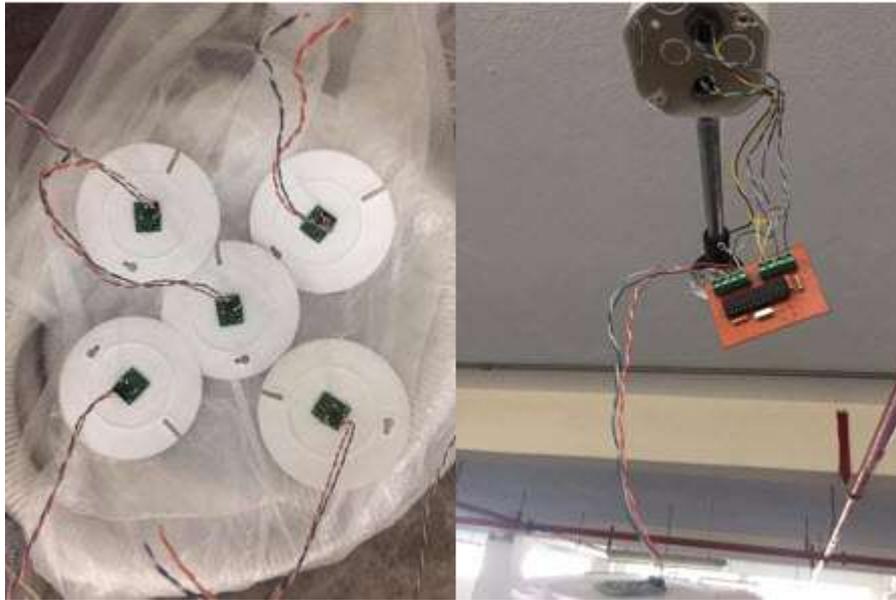


Figura 28: *Tarjetas y Sensores Instalados en el parqueadero del bloque B*
(El Autor)



Figura 29: *Cajas reflectoras Instaladas en el parqueadero del bloque B*
(El Autor)

4.4. Análisis de configuración de Arduino Yun para envío de Tweets y visualización de Página Web

Se obtuvieron los resultados deseados, creando las cuentas en Temboo, Twitter, y la modificación del código fuente para el envío de Tweets, que gracias a las facilidades que presentan el concepto de Software Libre, el cual nos permite copiar el código y adaptarlo a las necesidades del trabajo de titulación.

Se muestran los pasos a continuación:

Paso 1:

Cuenta creada en la plataforma Temboo

Crear una cuenta en la plataforma Temboo en la siguiente dirección <https://temboo.com/signup/> como lo muestra la Figura 30.



Figura 30: Cuenta creada en Temboo

Fuente: El Autor

Paso 2:

Cuenta creada en Twitter @smartparking_b como lo muestra la Figura 31.



Figura 31: Cuenta creada en Twitter

(El Autor)

Paso 3:

Crear la aplicación en Twitter ingresando a la siguiente dirección <https://apps.twitter.com/app/new>. Esta es una parte importante ya que es el nexo entre la plataforma Temboo y Twitter.

Paso 4:

Se registra la aplicación usando la **Twitter dev console** (<https://apps.twitter.com/>) como lo muestra la Figura 32. Después de crear la aplicación, se encontrará las claves API para esta aplicación, en las pestaña API Keys. Las cual deben ser copiada en el Sketch de Arduino Yun al momento de programar.

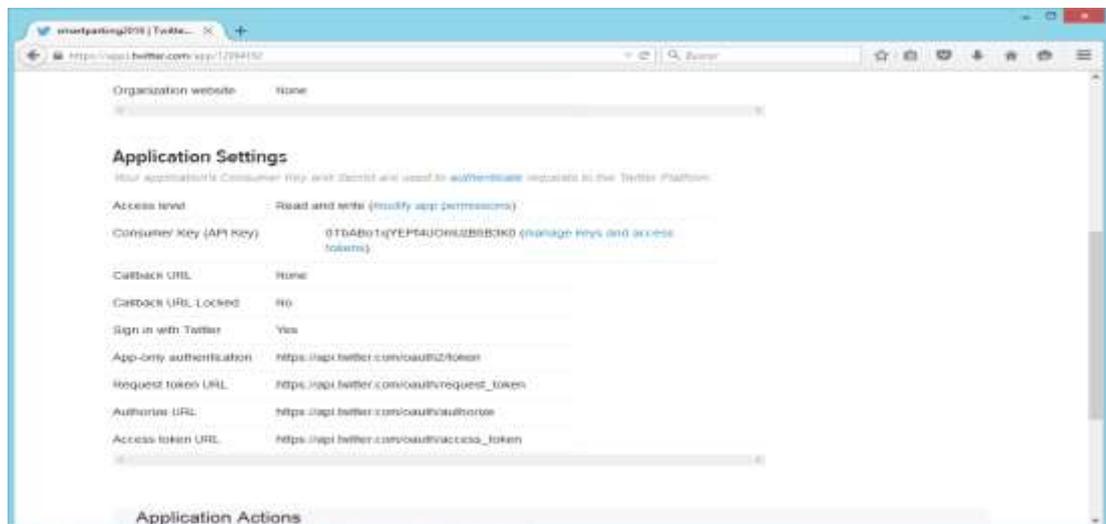


Figura 32: Portal Twitter dev console
(El Autor)

Desde este sketch se creara un nuevo Tweet, la aplicación necesita ser configurada con permisos de lectura + escritura. Esto se puede configurar en ajustes de la sección API Keys de la **Twitter dev console** mencionado anteriormente como lo muestra la Figura 33.

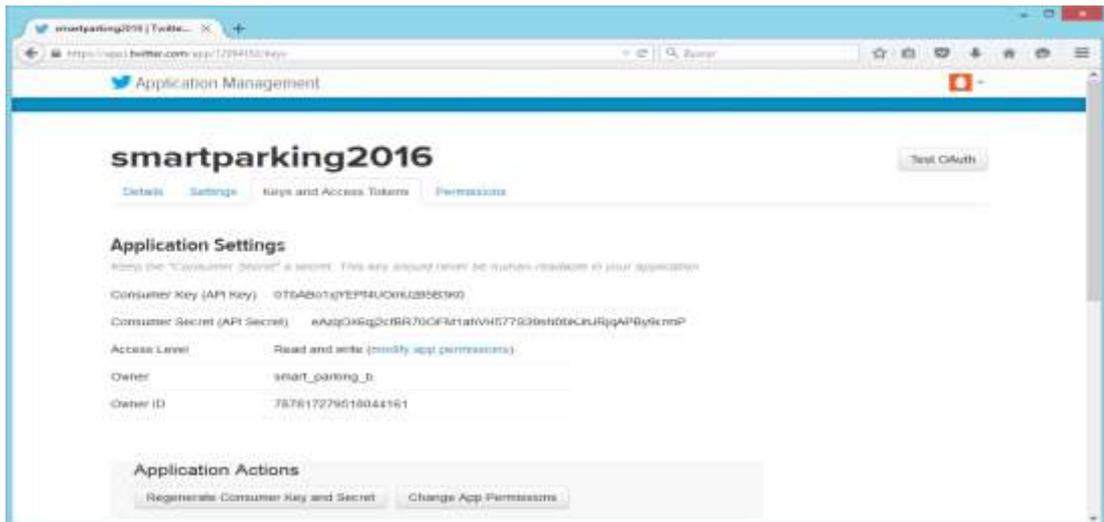


Figura 33: Portal Twitter dev console permisos de lectura y escritura
(El Autor)

Luego de copiar la información de la página de Temboo, de la página de Twitter Aplicación, se coloca el mensaje que quiere enviar.

Se modificó el modelo de Sketch existente en la plataforma Temboo que sirve para programar el Arduino Yun con la finalidad de enviar Tweets, a continuación se muestra el nuevo código acorde a nuestras necesidades.

Código fuente modificado Temboo & Arduino Yun

Programación nueva:

```
#include <Temboo.h>
#include <TembooCoAPEdgeDevice.h>
#include <TembooMQTTEdgeDevice.h>

#include <Bridge.h>
// #include <Temboo.h>
#include "TembooAccount.h" // Contiene la informacion de la cuenta de Temboo
```

```

const String TWITTER_ACCESS_TOKEN = "796157961383055361-
WpUbgTezCNwnAeHuLxQqh9uVvagDJjk";
const String TWITTER_ACCESS_TOKEN_SECRET =
"9oLNMnlzEqXk6pGELZ9AONPadYcxyVcr1y9NZX6syyqiE";
const String TWITTER_CONSUMER_KEY = "uU8MYkAPaSHc0BwAymkci4oiU";
const String TWITTER_CONSUMER_SECRET =
"RSZa16NwczeesiQQsYA3YHAFfaKhBqh17iqYPyhqSa3dCUqLy0";

unsigned long time;

int resultado,old_resultado;
int suma;
int Sensores[12];

void setup() {

    delay(2500);
    Serial.begin(9600);
    pinMode(2, INPUT);
    pinMode(3, INPUT);
    pinMode(4, INPUT);
    pinMode(5, INPUT);
    pinMode(6, INPUT);
    pinMode(7, INPUT);
    pinMode(8, INPUT);
    pinMode(9, INPUT);
    pinMode(10, INPUT);
    pinMode(11, INPUT);
    pinMode(12, INPUT);
    pinMode(13, INPUT);

    old_resultado = 0;
    // delay(10000);
    // while(!Serial);

    Bridge.begin();

```

```

}

void tweet() {
    Serial.println("Funcion envio de Tweet");
    Serial.print("Tiempo: ");
    time = millis (); // muestra el tiempo transcurrido desde que comenzó el
programa
    Serial.println(time/60000);

    // define the text of the tweet we want to send
    String tweetText("Los parqueos disponibles son " + String(12 - resultado) + ",
sistema activado" + " hace " + String(time/60000)+ " minutos");

    TembooChoreo StatusesUpdateChoreo;
    // invoca al cliente Temboo
    StatusesUpdateChoreo.begin();
    // se configura el nombre de las credenciales de acceso
    StatusesUpdateChoreo.setAccountName(TEMBOO_ACCOUNT);
    StatusesUpdateChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
    StatusesUpdateChoreo.setAppKey(TEMBOO_APP_KEY);
    // se identifica la libreria Temboo choreo para ejecutar (Twitter > Tweets >
StatusesUpdate)
    StatusesUpdateChoreo.setChoreo("/Library/Twitter/Tweets/StatusesUpdate");
    // se agrega la informacion de la cuenta de twitter
    StatusesUpdateChoreo.addInput("AccessToken", TWITTER_ACCESS_TOKEN);
    StatusesUpdateChoreo.addInput("AccessTokenSecret",
TWITTER_ACCESS_TOKEN_SECRET);
    StatusesUpdateChoreo.addInput("ConsumerKey",
TWITTER_CONSUMER_KEY);
    StatusesUpdateChoreo.addInput("ConsumerSecret",
TWITTER_CONSUMER_SECRET);
    // se configura el twett que se quiere enviar
    StatusesUpdateChoreo.addInput("StatusUpdate", tweetText);
    // tell the Process to run and wait for the results. The
    // return code (returnCode) will tell us whether the Temboo client
    // was able to send our request to the Temboo servers
    unsigned int returnCode = StatusesUpdateChoreo.run();

```

```

// a return code of zero (0) means everything worked
if (returnCode == 0) {
    Serial.println("Listo! Tweet enviado!");
} else {
    // a non-zero return code means there was an error
    // read and print the error message
    while (StatusesUpdateChoreo.available()) {
        char c = StatusesUpdateChoreo.read();
        Serial.print(c);

    }
}
// StatusesUpdateChoreo.close();
// do nothing for the next 90 seconds
Serial.println("Esperando...");

// delay(120000);
}

void loop() {

    for (int i=2;i<14;i++)
        Sensores[i-2] = digitalRead(i);

    resultado =
    Sensores[0]+Sensores[1]+Sensores[2]+Sensores[3]+Sensores[4]+Sensores[5]+Sen
    sores[6]+Sensores[7]+Sensores[8]+Sensores[9]+Sensores[10]+Sensores[11];

    for (int j=0;j<12;j++)
        Serial.println(Sensores[j]);

    if (resultado != old_resultado) {
        tweet();
    }

    old_resultado = resultado;
}

```

```

Serial.print("La Cantidad de Parqueos es: ");
Serial.print(12 - resultado);
Serial.println(" disponibles");
suma = (12 - resultado);
Serial.println(suma);

}

```

Esta programación se carga en el Arduino Yun la cual consiste en detectar los eventos cuando un vehículo es detectado por los sensores envía un Tweet informando sobre la cantidad de parqueos disponibles, se procede a conectar la tarjeta Arduino Yun a internet vía Ethernet a la Red LAN de la Universidad Politécnica Salesiana.

Se puede ingresar a la página de la cuenta Twitter @smartparking_b en la que se puede confirmar la funcionalidad de la tarjeta como lo muestra la Figura 34.



Figura 164: Cuenta Twitter @smartparking_b
(El Autor)

4.4.1. Resultados del diseño de la página Web e integración con Twitter

Los usuarios que no tenga una cuenta Twitter pueden ingresar a la siguiente dirección web como lo muestra la Figura 35, donde podrán consultar la información sobre el estado de los parqueos: <http://www.smartparkingb.260mb.net/>



Figura 35: *Página Web de Parqueo Inteligente*
(El Autor)

4.4.2. Luces de estado (Puesto ocupado o libre)

Como una alternativa de crear un tipo información visual se decidió instalar cajas reflectoras con un diodo dual dentro que cambia de color rojo cuando está ocupado y verde cuando el lugar de estacionamiento se encuentra libre tal como lo muestra la

Figura 36. Esta configuración se realizó previamente en cada uno de los microcontroladores Atmega328 con resultados exitosos.



Figura 36: *Cajas reflectoras Instaladas en el parqueadero del bloque B
(El Autor)*

4.4.3. Resultado y Análisis de las encuestas realizadas a docentes y personal administrativo.

Se realizaron 20 encuestas entre docentes y personal administrativo que utilizan el parqueadero subterráneo del bloque B, y arrojando los siguientes resultados:

En la Figura 37 se muestran que 10 de los 20 encuestados conocen o han oído hablar sobre parqueaderos inteligentes.

En la Figura 38 se muestra que un alto porcentaje está dispuesto a utilizar nuestro sistema ya que 8 encuestados respondieron afirmativamente y 9 con un probablemente sí. Mientras que un bajo índice estaba en duda.

En la Figura 39 se muestra que todos los encuestados están de acuerdo en que nuestro sistema de Parqueadero Inteligente les ayudará en la optimización de su tiempo y recursos, el cual es un dato muy interesante que confirma la utilidad del sistema instalado.

En la Figura 40 se muestra la necesidad de instalar un sistema similar en otros estacionamientos de la Universidad Politécnica Salesiana que suministre información anticipada a los usuarios sobre los estados de parqueo.

En la Figura 41 se muestra que un 50% de los usuarios utilizan Twitter y otro 50% utilizarían la página Web por lo que fue de gran utilidad tomar la iniciativa de crear un sitio web para que esta información pueda llegar a todos los usuarios y en especial a aquellos que no tengan una cuenta de Twitter y deseen hacer uso de nuestro sistema.

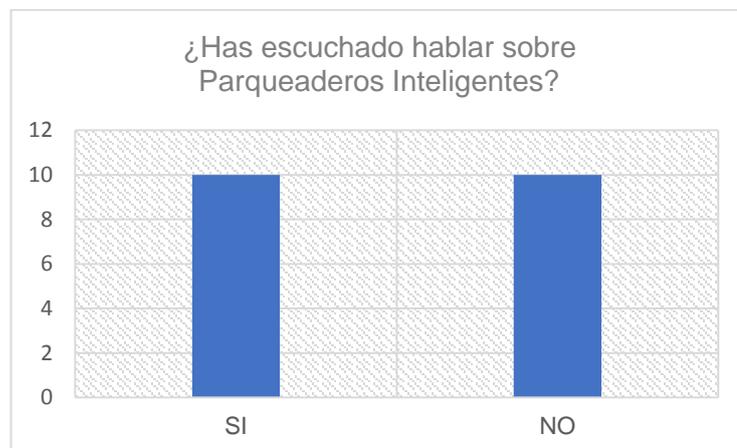


Figura 37: Conocimiento sobre Parquesos Inteligentes
(El Autor)

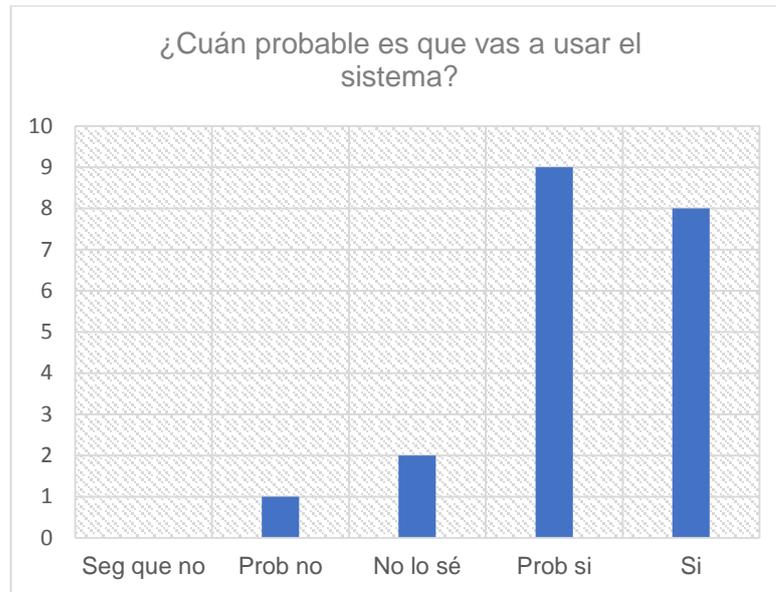


Figura 38: *Usuarios dispuestos a usar nuestro sistema*
(El Autor)

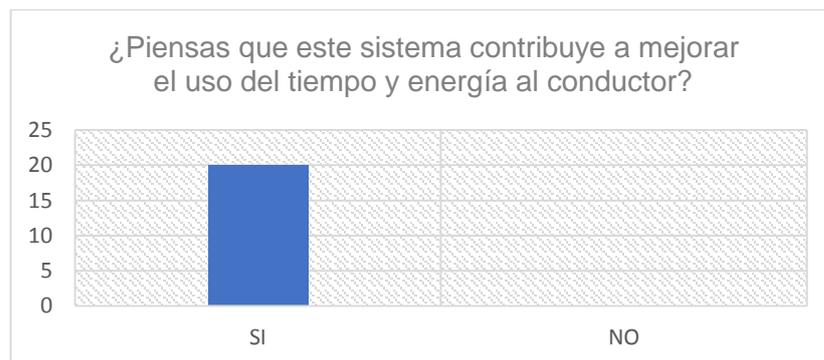


Figura 39: *Usuarios que piensan que el sistema optimiza recursos*
(El Autor)

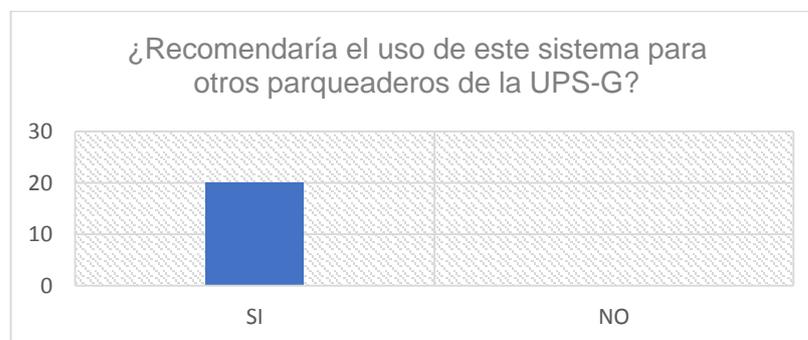


Figura 40: *Usuarios que recomiendan instalar el sistema en otros lugares*
(El Autor)

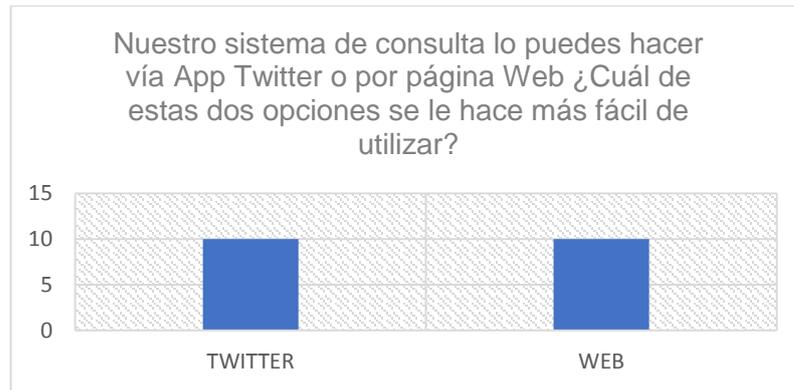


Figura 41: *Usuarios que utilizan Twitter y Portal WEB*
(El Autor)

5. Conclusiones

Con el desarrollo e implementación de un Sistema de Parqueadero Inteligente basado en Internet de las Cosas, se pudo comprobar que es factible diseñar un sistema que proporcione información anticipada a los usuarios que buscan un lugar libre para estacionar. Como resultado de la investigación se dio cumplimiento a los objetivos específicos de la investigación.

En base a la investigación realizada y a la información obtenida de libros, artículos, se determinó los equipos y tecnología necesarios para la elaboración de un Sistema de Parqueo Inteligente tales como: Tarjeta Arduino Yun y plataforma Temboo los mismos que fueron los principales actores del desarrollo del sistema (hardware y software).

Se configuró el aplicativo Twitter en conjunto con Temboo para que la información sobre la disponibilidad de parqueaderos para cualquier usuario que desee hacer una consulta. Así como también se diseñó una tarjeta electrónica capaz de interpretar las señales emitidas por los sensores ultrasónicos cuando detecta la presencia de algún vehículo u objeto en su rango previamente configurado. Adicional los datos y eventos son almacenados en la plataforma Temboo y el administrador puede hacer consultas de ellos para analizarlos si cree necesario.

Con los anteriores puntos se logró cumplir los objetivos y adicional brindar información en tiempo real a los usuarios sobre espacios libres de parqueo, ya que en la actualidad los estacionamientos existentes no brindan esta información al usuario y este solo puede llegar a saberlo cuando esta por ingresar al mismo.

6. Recomendaciones

Dada la amplia gama de opciones que nos brinda este sistema se recomienda que la institución o alguna entidad puedan invertir más en la investigación de la Internet de las Cosas y sus distintas aplicaciones para crear una Ciudades Inteligentes. Este proyecto fue orientado a Parqueaderos Inteligentes, un problema que va creciendo con el paso del tiempo en la ciudad de Guayaquil.

El proyecto propuesto puede ser instalado en parqueaderos cubiertos o parqueaderos al aire libre pero con variaciones en el esquema técnico para la adquisición de datos, ya que se puede usar otro tipo de sensores como por ejemplo: sensores de peso, sensores inalámbricos, etc. Dada las facilidades que presenta la tarjeta Arduino Yun.

En lo que respecta a la implementación del proyecto se realizaron algunas pruebas de transmisión de señal utilizando cables UTP, se observó que una señal de 5 voltios disminuyó a 4.7 voltios cuando se hizo pasar este voltaje por un tramo de cable de aproximadamente 40 metros. Esto nos asegura el funcionamiento correcto del proyecto al momento de su implementación.

Se eligió cable UTP para la transmisión de señales ya que al ser un tipo de cable entorchado está diseñado especialmente para la transmisión de señales de datos con un nivel mínimo de pérdida de la señal provocada por interferencias.

El sistema puede mejorarse ya que está realizado con tecnología open source, esto indica que es escalable y puede ser utilizado en un futuro como la base de un proyecto de innovación.

Con respecto al mantenimiento se recomendaría instalar un temporizador para apagar el sistema durante la noche desde las 23:00 hasta las 06:00 periodo en el cual no hay vehículos en el lugar.

7. Referencias Bibliográficas

- Arduino, "Introduction" [Online]. Recuperado de:
<https://www.arduino.cc/en/Guide/Introduction#>
- Areny, R.P. (2003). Sensores y acondicionadores de señal. Barcelona: marconbo.
- Ashton, K. [Kevin Ashton]. (2014, 19 de junio). *Kevin Ashton. "The Internet of Things." Seoul, June 19, 2014* [YouTube]. Recuperado de
https://www.youtube.com/watch?v=xSYkp8_Dn2E
- Boutin, P. (2016, 30 de mayo). The secretive world selling data about you.
Recuperado de: <http://europe.newsweek.com/secretive-world-selling-data-about-you-464789?rm=eu>
- Cisco (2013, 29 de julio). Connections Counter: The internet of everything in motion.
Recuperado de: <https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342>
- Coursera - University of California, " Introduction to the Internet of Things and Embedded Systems" [E-learning course]. Recuperado de:
<https://www.coursera.org/learn/iot>
- EMC, "The internet of things" [Online]. Recuperado de:
<http://www.emc.com/leadership/digital-universe/2014iview/internet-of-things.htm>
- Formoso, A., Mazzilli, A. & Sotelo, R. (2014). ParkIt - Plataforma inteligente de estacionamiento público. *Memoria Investigaciones en Ingeniería*, 12, 85-91.
- Free Software Foundation Europe, " ¿Qué es el Software Libre?" [Online].
Recuperado de: <https://fsfe.org/about/basics/freesoftware.es.html>
- Gartner (2013, 12 de diciembre). Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020. Recuperado de:
<http://www.gartner.com/newsroom/id/2636073>
- GNU, "Licencias" [Online]. Recuperado de GNU:
<https://www.gnu.org/licenses/licenses.es.html>
- Greene, K. (2011, 12 de septiembre). A new and improved Moores law. Recuperado de: <https://www.technologyreview.com/s/425398/a-new-and-improved-moores-law/>
- Griffin, A. (2015, 10 de agosto). Facebook security hole leaves personal data open to easy stealing. Recuperado de: <http://www.independent.co.uk/life-style/gadgets-and-tech/news/facebook-security-hole-leaves-personal-data-open-to-easy-stealing-10448147.html>
- Hafner, K. (1999, 27 de mayo). Honey I programmed the blanket. Recuperado de:

http://www.nytimes.com/1999/05/27/technology/honey-i-programmed-the-blanket.html?_r=1

- Henderson, R. (2014, 6 de enero). Processors will have as many transistors as there are neurons in a brain. Recuperado de: <http://www.pocket-lint.com/news/126289-intel-claims-that-by-2026-processors-will-have-as-many-transistors-as-there-are-neurons-in-a-brain>
- Idna Idris, M., Leng, Y., Tamil, E., & Razak, Z. (2009). Car Park System: A Review of Smart Parking System and its Technology. *Information Technology Journal*, 8 (2), 101-113. doi: 10.3923/itj.2009.101.113
- Internet.org, "Connecting the world" [Online]. Recuperado de: <https://info.internet.org/en/>
- ITU Telecommunication Standardization Sector (2012, 15 de junio). Overview of the Internet of things. Recuperado de : <http://handle.itu.int/11.1002/1000/11559-en?locatt=format:pdf&auth>
- Leyden, J. (2015, 24 de agosto). Samsung smart fridge leaves Gmail logins open to attack. Recuperado de: http://www.theregister.co.uk/2015/08/24/smart_fridge_security_fubar/
- Lin, T. (2015). *Smart parking: Network, infrastructure and urban service. Networking and Internet Architecture* (PHD Thesis). Recuperado de: <https://tel.archives-ouvertes.fr/tel-01249062v1/document>
- Moore, G. E. (1998). Cramming More Components onto integrated Circuits. *Proceedings of the IEEE*, 86 (1), 82-85.
- MSE Microsystems Engineering [Online]. Recuperado de: <http://www.microcontroladores.com/>
- Neumane, S. (2012, 19 de noviembre). Guayaquil, la ciudad de más de 400 mil autos, aumenta sus problemas de tránsito. Recuperado de El Universo: <http://especiales.eluniverso.com/otroguayaquil/transito/>
- Open Source Hardware Association, "Definition" [Online]. Recuperado de: <http://www.oshwa.org/definition/spanish/>
- Parallax (2003). Que es un Microcontrolador. Recuperado de: <https://www.parallax.com/sites/default/files/downloads/28123-Whats-A-Microcontroller-Espanol-v3.0.pdf>
- Project Loon, "What is project loon" [Online]. Recuperado de: <https://www.solveforx.com/loon/>

- Rodier, C., Shaheen, S., & Eaken, A. (2004). Transit-based smart parking in the San Francisco Bay area: an assessment of user demand and behavioral effects. Recuperado de:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.514.7121&rep=rep1&type=pdf>
- Rodier, C., Shaheen, S., & Kemmerer, C. (2008). Smart Parking Management Field Test: A Bay Area Rapid Transit (BART) District Parking Demonstration; Final Report. Recuperado de: <http://innovativemobility.org/wp-content/uploads/2015/09/Smart-Parking-Management-Field-Test.pdf>
- Rosabel, D. (2016, 27 de enero). Digital in 2016. Recuperado de We are social: <http://wearesocial.com/sg/special-reports/digital-2016>
- Rouse, M., & McMahon, D. (1 de Diciembre de 2015). Twitter. Recuperado de Whatis.com: <http://whatis.techtarget.com/definition/Twitter>
- Sandler, K., Ohrstrom, L., Moy, L., & McVay, R. (2010, 21 de julio). Killed by Code: Software Transparency in Implantable Medical Devices. Recuperado de: <https://www.softwarefreedom.org/resources/2010/transparent-medical-devices.pdf>
- Temboo, "Arduino + Temboo" [Online]. Recuperado de: <https://temboo.com/arduino/yun/>
- Waterson, B., Hounsell, N., & Chatterjee, K. (2001, 12 de octubre). Quantifying the potential savings in travel time resulting from parking guidance systems – a simulation case study. Recuperado de Semantic Scholar: <https://pdfs.semanticscholar.org/f809/6e5a7ca856fcd949ae506412423529de61f7.pdf>

8. Anexos

8.1. Presupuesto de desarrollo del sistema

Los costos por el desarrollo e implementación del sistema se detallan en la tabla XX.

Tabla 5. Cotos de desarrollo del sistema *Parqueadero Inteligente*

Ítem	Nombre del recurso	Tipo	Etiqueta de material	Precio Unitario	Precio Total
1	Arduino YUN board based on the ATmega32u4	Material	Placa Base / Modulo de Comunicación	\$100	\$100
2	Sensor SRF02	Material	Unidad x 12	\$55	\$660
3	Tarjeta fabricada por el autor para adquisición de datos para sensores	Material	Unidad x 12	\$40	\$480
4	Fuente de alimentación VDC regulada	Material	Unidad	\$20	\$20
5	Construcción e infraestructura para dispositivos de parqueadero	Material	Unidad	\$200	\$400
6	TOTAL				\$1660

8.2. Encuestas

Se anexan 20 encuestas realizadas a docentes y personal administrativo que hacen uso del parqueadero subterráneo del bloque B