



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

**“DISEÑO DE UN SISTEMA DE MONITOREO DE LA DISPONIBILIDAD DE
ESTACIONAMIENTOS PARA EL PARQUEADERO DE LA UNIVERSIDAD TÉCNICA
DEL NORTE”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA
EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

AUTOR: WILMER PATRICIO BEDOYA PUMA

DIRECTOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

Ibarra-Ecuador

2018



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA.

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

DATOS DE CONTACTO	
Cédula de identidad	1003474697
Apellidos y Nombres	Bedoya Puma Wilmer Patricio
Dirección	El Rosal de las Malvinas
E-mail	wpbedoyap@utn.edu.ec
Teléfono móvil	0981328123
DATOS DE LA OBRA	
Título	DISEÑO DE UN SISTEMA DE MONITOREO DE LA DISPONIBILIDAD DE ESTACIONAMIENTOS PARA EL PARQUEADERO DE LA UNIVERSIDAD TÉCNICA DEL NORTE.
Autor	Bedoya Puma Wilmer Patricio

Fecha	2018-02-08
Programa	Pregrado
Título	Ingeniero en Electrónica y Redes de Comunicación
Director	Ing. Jaime Roberto Michilena Calderón, MSc.

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD.

Yo, Bedoya Puma Wilmer Patricio, con cedula de identidad Nro. 1003474697, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad de material y como apoyo a la educación, investigación y extensión, en concordancia con la ley de Educación Superior Artículo 144.

3. CONSTANCIAS.

Yo, BEDOYA PUMA WILMER PATRICIO declaro bajo juramento que el trabajo aquí escrito es de mi autoría; y que este no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se presentan en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Universidad Técnica del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normatividad vigente de la Universidad Técnica del Norte.

En la ciudad de Ibarra, febrero de 2018.

EL AUTOR:



Bedoya Puma Wilmer Patricio

CI. 1003474697

Ibarra, febrero 2018



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

Yo, Bedoya Puma Wilmer Patricio, con cedula de identidad Nro. 1003474697, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador artículos 4, 5 y 6, en calidad de autor del trabajo de grado con el tema: DISEÑO DE UN SISTEMA DE MONITOREO DE LA DISPONIBILIDAD DE ESTACIONAMIENTOS PARA EL PARQUEADERO DE LA UNIVERSIDAD TÉCNICA DEL NORTE. Que ha sido desarrollado con propósito de obtener el título de Ingeniero en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

A handwritten signature in blue ink, appearing to read 'Bedoya Puma Wilmer Patricio', is written over a horizontal dotted line.

Bedoya Puma Wilmer Patricio

CI. 1003474697

Ibarra, febrero 2018



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN.

MAGISTER JAIME MICHILENA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación “DISEÑO DE UN SISTEMA DE MONITOREO DE LA DISPONIBILIDAD DE ESTACIONAMIENTOS PARA EL PARQUEADERO DE LA UNIVERSIDAD TÉCNICA DEL NORTE.” Ha sido desarrollado por el señor Wilmer Patricio Bedoya Puma bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

A handwritten signature in purple ink, appearing to read "Jaime Michilena", is written over a horizontal dotted line. The signature is stylized and cursive.

Ing. Jaime Michilena, MSc.

1002198438

DIRECTOR

Agradecimiento.

Agradezco,

En primer lugar a Dios por permitirme disfrutar de la vida, guiar mi camino y ayudarme a cumplir esta meta.

A los ingenieros Luis Suarez y Daniel Jaramillo que supieron ayudarme a culminar este trabajo de grado.

A mi director el Ingeniero Jaime Michilena por compartir su tiempo, sus conocimientos y toda la colaboración brindada durante la elaboración de este proyecto, lo cual me permitió culminar mi carrera de la mejor manera.

Finalmente agradezco a todos los docentes que tuve la oportunidad de conocer durante mi etapa de estudiante en la carrera de electrónica y redes de comunicación, los cuales supieron guiarme por el mejor camino y me ayudaron a crecer profesional y personalmente.

Wilmer Bedoya

Dedicatoria.

Dedicado,

A mis padres: Rosa y Galo, por su gran comprensión y ejemplo de bien que me han inculcado durante toda mi vida y que a pesar de los tropiezos que he tenido jamás han dejado de apoyarme.

A mis hermanos: Verónica, Alexandra y Alexis quienes siempre me ofrecieron su apoyo permanente para poder alcanzar esta meta.

A todas las personas, familiares y amigos que siempre estuvieron brindándome su ayuda, motivándome y dándome su cariño incondicional durante la realización de este Trabajo de Grado.

Wilmer Bedoya.

Índice.

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	I
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	IV
CERTIFICACIÓN.....	V
Agradecimiento.....	VI
Dedicatoria.....	VII
Índice.....	VIII
Índice de Ilustraciones.	XII
Índice de Tablas.....	XV
Resumen.....	XVII
Abstract.....	XVIII
Capítulo 1. Antecedentes.	1
1.1. Tema.....	1
1.2. Problema.....	1
1.3. Objetivos.	2
1.3.1 Objetivo general.	2
1.3.2 Objetivos específicos.....	2
1.4. Alcance.....	3
1.5. Justificación.....	4
Capítulo 2. Fundamentación Teórica.....	7
2.1. Introducción.	7
2.2. Sistema de monitoreo.....	8
2.2.1. Características de un sistema de monitoreo.....	8
2.2.2. Ventajas de un sistema de monitoreo.....	8
2.3. Red de sensores inalámbricos (WSN).....	9
2.3.1. Características de una WSN.....	9
2.3.2. Elementos de una WSN.....	10
2.3.3. Arquitectura de una WSN.	12
2.3.4. Topologías de una WSN.....	13

2.3.5. Estándares de comunicación inalámbrica utilizados en WSNs.	14
2.3.6. Almacenamiento de datos de una WSN.	16
2.4. Hardware.	18
2.4.1. Sensores.	19
2.4.2. Placas de procesamiento Arduino.	22
2.4.3. Transceptores inalámbricos.	24
2.4.4. Dispositivos para la presentación de información al usuario.	29
2.5. Software.	32
2.5.1. IDE Arduino.	32
2.5.2. Aplicación móvil o nativa.	32
2.5.3. Página Web.	33
2.5.4. Almacenamiento en la nube.	34
2.6. Descripción de sistemas de parqueo comerciales.	38
2.6.1. Libelium.	38
2.6.2. ParkHelp Mobility and Sustainability Solutions.	40
2.6.3. Parqueos Sistemas y Soluciones S.A.	42
2.7. Metodología.	44
2.7.1. Tipo de investigación.	44
2.7.2. Métodos de investigación.	44
2.7.3. Modelo en cascada.	45
2.7.4. Modelo en V.	47
Capítulo 3. Diseño del Sistema.	50
3.1. Análisis de la situación actual.	50
3.1.1. Técnicas de recolección de información.	50
3.1.2. Identificación de la población.	52
3.1.3. Universidad Técnica del Norte.	55
3.1.4. Campus de la UTN El Olivo.	56
3.1.5. Parqueaderos de la UTN.	57
3.1.6. Puertas de acceso vehicular.	66
3.1.7. Análisis de resultados.	68
3.2. Introducción al desarrollo del sistema.	69

3.2.1. Descripción general del sistema.....	70
3.3. Requerimientos del sistema.....	71
3.3.1. Requerimientos de stakeholders.....	72
3.3.2. Requerimientos funcionales.....	74
3.3.3. Requerimientos de arquitectura.....	76
3.4. Elección de hardware y software del sistema.....	78
3.4.1. Elección de Hardware.....	79
3.4.2. Elección de Software.....	94
3.5. Diseño del sistema.....	96
3.5.1. Diagrama de bloques.....	97
3.5.2. Arquitectura del sistema.....	98
3.5.3. Diseño de la WSN.....	99
3.5.4. Diagramas de flujo.....	115
3.5.5. Almacenamiento y visualización de la información en la nube.....	117
Capítulo 4. Pruebas de Funcionamiento.....	125
4.1. Implementación del prototipo.....	125
4.1.1. Colocación del nodo sensor.....	125
4.1.2. Colocación del nodo central.....	126
4.1.3. Dispositivos para la visualización de la información.....	127
4.2. Pruebas de funcionamiento del Prototipo.....	127
4.2.1. Pruebas del nodo sensor.....	128
4.2.2. Pruebas del nodo central.....	128
4.2.3. Pruebas de la plataforma Ubidots.....	129
4.2.4. Pruebas desarrolladas en el día.....	130
4.2.5. Pruebas desarrolladas en la noche.....	130
4.2.6. Resultados de las pruebas.....	131
4.3. Costos del sistema.....	132
4.3.1. Costos de hardware del nodo sensor.....	132
4.3.2. Costos de hardware del nodo central.....	133
4.3.3. Costos de la pantalla led.....	133
4.3.4. Costos de infraestructura.....	134

4.3.5. Costos de ingeniería.	135
4.3.6. Costos de software.....	135
4.3.7. Costos de implementación del sistema.....	136
Conclusiones y Recomendaciones.....	137
Referencias.....	139
Glosario de Términos y Acrónimos.....	143
Anexos.	146
ANEXO 1: Formato de encuesta aplicada a los usuarios de los parqueaderos de la UTN.....	146
ANEXO 2: Preguntas realizadas en la entrevista al jefe de seguridad de la UTN.	148
ANEXO 3: Tabulación de las encuestas aplicadas a los usuarios de los parqueadero de la UTN de los parqueaderos de la UTN.....	150
ANEXO 4: DataSheet Sensor SHARP 2Y0A21.	154
ANEXO 5: DataSheet Arduino UNO.	163
ANEXO 6: DataSheet Arduino Mega2560.....	171
ANEXO 7: DataSheet XBee S2C.	179
ANEXO 8: DataSheet Pantalla led MOD PC02.	183
ANEXO 9: Código Nodo Sensor – Arduino Uno.....	185
ANEXO 10: Código Nodo Central – Arduino Mega2560.....	186

Índice de Ilustraciones.

Ilustración 1. Componentes principales de una WSN.....	11
Ilustración 2. Arquitectura de una WSN.....	12
Ilustración 3. Topologías de una WSN.....	13
Ilustración 4. Diferencias entre usar o no Ubidots.....	37
Ilustración 5. Diagrama general del modelo en cascada.....	45
Ilustración 6. Diagrama del modelo en V.....	47
Ilustración 7. Vista superior de la UTN.....	55
Ilustración 8. Campus de la UTN.....	56
Ilustración 9. Ubicación de los parqueaderos en el campus de la UTN.....	57
Ilustración 10. Parqueadero A8 reservado para autoridades y visitas.....	60
Ilustración 11. Estacionamientos reservados para vicerrectores en el parqueadero A9. ..	61
Ilustración 12. Estacionamiento para personas con discapacidad.....	61
Ilustración 13. Estacionamientos para vehículos de la UTN.....	62
Ilustración 14. Lugar de aparcamientos de los buses de la UTN.....	63
Ilustración 15. Ubicación de los estacionamientos reservados.....	63
Ilustración 16. (a) Señal de estacionarse en reversa. (b) Límite de velocidad de circulación interna. (c) Señalización de los lugares prohibidos para estacionarse.....	66
Ilustración 17. Accesos vehiculares. (a) Puerta norte. (Puerta sur).....	67
Ilustración 18. Sensor infrarrojo Sharp 2Y0A21.....	80
Ilustración 19. Arduino UNO.....	82
Ilustración 20. Arduino Mega2560.....	84
Ilustración 21. Arduino shield Ethernet.....	86
Ilustración 22. XBee S2C.....	89
Ilustración 23. Pantalla led MOD PC-mini.....	92
Ilustración 24. XBee Shield.....	93
Ilustración 25. Interfaz del IDE de Arduino.....	94
Ilustración 26. Interfaz web de Ubidots.....	96
Ilustración 27. Diagrama de bloques del sistema.....	97
Ilustración 28. Arquitectura del sistema.....	99
Ilustración 29. Ubicación de los nodos sensores.....	102

Ilustración 30. Ubicación de los nodos centrales.	103
Ilustración 31. Componentes de los nodos sensor.	104
Ilustración 32. Diagrama de conexión del nodo sensor.	105
Ilustración 33. Diseño de la PCB - Shield para el sensor infrarrojo.	106
Ilustración 34. Montaje de las placas del nodo sensor. (a) Vista superior. (b) Vista frontal.	106
Ilustración 35. Caja de protección para el nodo sensor.	107
Ilustración 36. Vista interna del nodo sensor.	107
Ilustración 37. Nodo sensor finalizado.	108
Ilustración 38. Baterías Ultrafire.	110
Ilustración 39. Componentes del nodo central.	110
Ilustración 40. Diagrama de conexión del nodo central.	111
Ilustración 41. Montaje de las placas del nodo central. (a) Vista superior. (b) Vista frontal.	112
Ilustración 42. Caja de protección para el nodo central.	112
Ilustración 43. Vista interna del nodo central.	113
Ilustración 44. Nodo central finalizado.	113
Ilustración 45. Adaptador AC/DC para Arduino.	114
Ilustración 46. Diagrama de flujo del nodo central.	115
Ilustración 47. Diagrama de flujo del nodo central.	116
Ilustración 48. Diagrama de flujo de la plataforma de Ubidots.	116
Ilustración 49. Creación de dispositivos en Ubidots.	117
Ilustración 50. Creación de la variable en Ubidots.	118
Ilustración 51. Ingreso a las credenciales API de Ubidots.	118
Ilustración 52. Creación del Token en Ubidots.	119
Ilustración 53. Ingreso del Token Ubidots y su ID de variable.	119
Ilustración 54. Almacenamiento de datos en Ubidots.	120
Ilustración 55. Página principal de Ubidots.	120
Ilustración 56. Elección de la presentación de datos en Ubidots.	121
Ilustración 57. Elección del dato que va a mostrar Ubidots.	121
Ilustración 58. Interfaz de monitoreo en Ubidots.	122

Ilustración 59. Creación de eventos en Ubidots.....	123
Ilustración 60. Condición para que se genere la alerta en Ubidots.	124
Ilustración 61. Elección del medio por el cual Ubidots genera la alerta.	124
Ilustración 62. Colocación del nodo sensor.	125
Ilustración 63. Nodos sensores instalados.....	126
Ilustración 64. Instalación del nodo central	126
Ilustración 65. Interfaz de visualización en dispositivos móviles.	127
Ilustración 66. Datos que envía el nodo sensor al nodo central.	128
Ilustración 67. Valor que envía el nodo central a Ubidots.....	129
Ilustración 68. Numero de estacionamientos disponibles en Ubidots.....	129

Índice de Tablas.

Tabla 1. Comparación de los estándares de comunicación inalámbrica.....	16
Tabla 2. Características de los sensores.....	21
Tabla 3. Características de las placas Arduino #1.....	22
Tabla 4. Características de las placas Arduino #2.....	23
Tabla 5. Características del módulo WiFi ESP8266.....	26
Tabla 6. Características del módulo Bluetooth HC-05.....	27
Tabla 7. Características del XBee ZB S2C.....	28
Tabla 8. Plug & Sense! Smart Parking main characteristics.....	39
Tabla 9. Características principales del nodo sensor.....	41
Tabla 10. Distribución de los dispositivos electrónicos.....	53
Tabla 11. Número de usuarios a los que se les aplico la encuesta.....	55
Tabla 12. Capacidad de los parqueaderos de automóviles de la UTN.....	58
Tabla 13. Capacidad de los parqueaderos de motocicletas de la UTN.....	59
Tabla 14. Dimensiones de los parqueaderos.....	59
Tabla 15. Cantidad de estacionamientos disponibles para automóviles.....	64
Tabla 16. Términos abreviados y acrónimos.....	72
Tabla 17. Lista de Stakeholders del proyecto.....	72
Tabla 18. Requerimientos de Stakeholders del sistema.....	73
Tabla 19. Requerimientos funcionales del sistema.....	74
Tabla 20. Requerimientos de arquitectura del sistema.....	76
Tabla 21. Elección del sensor.....	79
Tabla 22. Especificaciones técnicas del sensor.....	81
Tabla 23. Elección de la placa de procesamiento de datos.....	81
Tabla 24. Especificaciones técnicas del Arduino UNO.....	83
Tabla 25. Especificaciones técnicas del Arduino Mega2560.....	85
Tabla 26. Especificaciones técnicas del Arduino Ethernet.....	87
Tabla 27. Elección del módulo de comunicación inalámbrica.....	88
Tabla 28. Especificaciones técnicas XBee ZB S2C.....	90
Tabla 29. Elección del dispositivo de visualización.....	91
Tabla 30. Especificaciones técnicas de la pantalla led.....	92

Tabla 31. Elección de la nube de almacenamiento.	95
Tabla 32. Cantidad de nodos centrales y sensores.	100
Tabla 33. Distancia máxima entre el nodo central y el nodo sensor.	103
Tabla 34. Consumo de corriente de los elementos del nodo sensor.	108
Tabla 35. Consumo de corriente de los elementos del nodo central.	114
Tabla 36. Pruebas del prototipo del sistema realizadas en el día.	130
Tabla 37. Pruebas del prototipo del sistema realizadas en la noche.	131
Tabla 38. Resultados de las pruebas.	131
Tabla 39. Costos de los elementos utilizados en el nodo sensor.	132
Tabla 40. Costos de los elementos utilizados en el nodo central.	133
Tabla 41. Costo de la pantalla led.	134
Tabla 42. Costos de infraestructura.	134
Tabla 43. Costos de ingeniería.	135
Tabla 44. Costos de software.	135
Tabla 45. Costos de la implementación del sistema.	136

Resumen.

El presente proyecto consiste en el diseño de un sistema de monitoreo de la disponibilidad de los estacionamientos para determinar la cantidad de estacionamientos que se encuentran libres en cada parqueadero, está basado en una red de sensores inalámbrica (WSN) y el uso de una plataforma en la nube para almacenar y visualizar los datos recolectados por el sistema.

Para el diseño de este sistema se realizó una revisión bibliográfica de los principales aspectos que intervienen en las redes de sensores inalámbricas, en las plataformas de almacenamiento en la nube, además se realizó un levantamiento de información de la situación actual de los parqueaderos para establecer el tamaño y la capacidad que tienen, lo cual permitió establecer de manera clara los requerimientos del sistema para poder ser desarrollado adecuadamente.

El prototipo del sistema fue realizado en base a las consideraciones del diseño, con el objetivo de verificar que el diseño propuesto funcione adecuadamente, para lo cual se utilizó sensores infrarrojo, placas Arduino, modulos XBee para la comunicación inalámbrica de la red de sensores y además un Shield Ethernet para tener acceso a internet y poder enviar lo información recolectada a la plataforma en la nube.

Se realizaron pruebas del sistema mediante la implementación de un prototipo que permitió verificar el correcto funcionamiento del sistema de monitoreo de los estacionamientos, que permite conocer la cantidad de estacionamientos que se encuentran disponibles para la comunidad universitaria, los datos obtenidos son subidos en tiempo real a una plataforma en Internet donde los usuarios pueden visualizar que parqueadero cuenta con plazas libres.

Abstract.

This project consists in the design of a parking availability monitoring system to determine the number of parking spaces that are free in each parking lot, is based on a wireless sensor network and the use of a cloud platform to store and visualize the data collected by the system.

For the design of this system, a bibliographic review of the main aspects that takes parts in the wireless sensor networks, the storage platforms of the cloud, was carried out. In addition, an information survey of the current situation about parking lots was made, to establish the size and the capacity they have, it allowed to organize clearly the requirements of the system to be able to be properly developed.

The prototype of the system was made based on the considerations of the design, in order to verify that the proposed design works correctly, for which infrared sensors, Arduino boards, XBee modules for wireless communication of the sensor network, were used. Ethernet Shield to access to the internet and send the information collected to the platform in the cloud.

System tests were carried out through the implementation of a prototype that allowed verifying the correct operation of the parking monitoring system, this lets to identify the number of parking spaces that are available to the university community; the data obtained are uploaded in real-time to an Internet platform where users can see which parking lot has free places.

Capítulo 1. Antecedentes.

En este capítulo se encuentra la descripción de las bases para el desarrollo del presente trabajo de titulación, lo cual consta de: el tema, la problemática, los objetivos, el alcance y la justificación, con la finalidad de determinar la importancia de la realización de un sistema de monitoreo de los estacionamientos de la Universidad Técnica del Norte.

1.1. Tema.

DISEÑO DE UN SISTEMA DE MONITOREO DE LA DISPONIBILIDAD DE ESTACIONAMIENTOS PARA EL PARQUEADERO DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

1.2. Problema.

En los últimos años, en los parqueaderos de la Universidad Técnica del Norte se ha percibido un incremento del parque automotor y los parqueaderos existentes dentro del campus universitario ya no son suficientes debido a la gran afluencia vehicular. Los conductores al desconocer la existencia de estacionamientos libres ingresan a las instalaciones de la UTN pero no se encuentra rápidamente un lugar vacío, por lo cual deben recorrer en su totalidad el parqueadero en busca de un lugar donde estacionarse, lo que representa una pérdida de tiempo y consecuentemente esto causa mal estar en los conductores, emisiones innecesarias de gases contaminantes afectando la salud y el ambiente.

Debido a la gran afluencia de estudiantes, docentes y personal administrativo que ingresan a la casona universitaria con sus automóviles, el problema del tráfico vehicular es una realidad que se enfrenta día a día y con ello la falta de espacios donde estacionar dicho conglomerado vehicular, razón por la cual se tiene como resultado que cada vez se vuelva más difícil encontrar un lugar donde estacionarse, especialmente en el horario de 8:00 a 8:30 en la mañana y de 14:30 a 15:00 en la tarde, siendo estos horarios considerados las hora pico por el departamento de seguridad de la institución.

Conociendo que el número de parqueaderos es fijo y que la cantidad de vehículos que ingresan va en aumento, es necesario encontrar métodos que nos ayuden a conocer la existencia de lugares de parqueo libres dentro de las instalaciones de la UTN, siendo una solución un sistema de monitoreo de la disponibilidad de plazas de estacionamiento, el cual permita informar al usuario la cantidad de estacionamientos vacantes.

1.3. Objetivos.

1.3.1 Objetivo general.

Realizar el diseño de un sistema de monitoreo para observar la disponibilidad de estacionamientos en los parqueaderos de la Universidad Técnica del Norte el cual permita visualizar la cantidad de lugares libres.

1.3.2 Objetivos específicos.

- Realizar una investigación bibliográfica acerca de los sistemas de parqueo y de una WSN.

- Recopilar información de los elementos que componen una red inalámbrica de sensores (WSN), los tipos de sensores inalámbricos disponibles en el mercado y la tecnología que utilizan para comunicarse, las plataformas de almacenamiento de datos y visualización de la información.
- Diseñar la red de sensores que permita la recolección de datos de cada estacionamiento, para lo cual es necesario determinar los requerimientos y el dimensionamiento del sistema, que permitirá una adecuada elección de los componentes de hardware y software.
- Desarrollar un prototipo del sistema el cual permitirá verificar el funcionamiento del proyecto y establecer si los componentes utilizados son los adecuados para el correcto desempeño del sistema.

1.4. Alcance.

El presente proyecto se refiere al diseño de un sistema de aparcamiento vehicular, para el monitoreo de la disponibilidad de los estacionamientos, dentro de los parqueaderos de la Universidad Técnica del Norte, con el fin de que los conductores puedan conocer la existencia de plazas disponibles donde se pueda dejar los vehículos.

Se realizará una revisión bibliográfica de sistemas de parqueo, de una WSN y los estándares de comunicaciones inalámbricas para identificar de manera clara cuáles son los elementos o componentes que conformaran este sistema.

Para el desarrollo de este diseño se va a realizar una recopilación de información de los tipos de sensores disponibles en el mercado, lo cual servirá para elegir los dispositivos adecuados, con los cuales se espera obtener información acerca de la disponibilidad de los estacionamientos.

El Parqueadero será optimizado al integrarlo con nuevas tecnologías como es el uso de una WSN, lo cual permite la comunicación de cada uno de los nodos que recolectan los datos de los sensores ubicados en los estacionamientos, procesar los datos obtenidos y proporcionar a los conductores que ingresan con su vehículo información en tiempo real de la existencia de plazas de parqueo libres permitiéndoles estacionar el automóvil de una manera más rápida.

Se diseñará una WSN, donde los nodos de los sensores deberán estar ubicados estratégicamente en cada estacionamiento, y en base a los requerimientos del sistema se realizará la elección de los dispositivos por los que estará conformado el sistema, y su funcionamiento se basará en sensores que se encargarán de obtener los datos y enviar a su controlador para que este se encargue de procesar la información y enviar a una plataforma de almacenamiento para su posterior tratamiento y visualización, la misma que se realizará a través de pantallas informativas colocadas en los accesos vehiculares.

Para comprobar que el sistema funcione adecuadamente, se realizará la implementación del prototipo del sistema en cinco estacionamientos del parqueadero de la FICA, el cual permitirá indicar cuantos estacionamientos se encuentran libres, esta información será mostrada en un dispositivo con pantalla.

1.5. Justificación.

Dentro de las instalaciones de la Universidad Técnica del Norte el uso adecuado de los parqueaderos y la circulación vial vehicular es muy importante, es por eso que el presente proyecto cuenta con una solución tecnológica, lo cual permite cumplir con cierta parte de la misión de la

universidad siendo esta “Generar, fomentar y ejecutar procesos de investigación, tecnológicos y de innovación.” (Universidad Técnica del Norte, 2008)

Debido al crecimiento del parque automotor, el problema del tráfico es una realidad que se enfrenta día a día y con ella la falta de espacio donde estacionar dicho conglomerado vehicular; razón por la cual, el sector público y privado se ha visto en la necesidad de adoptar un conjunto de acciones dentro de las cuales se contemplan medidas frente al problema de falta de parqueaderos y mejoramiento de la eficiencia de los ya existentes, por lo cual es necesario contar con un sistema que proporcione información e indicar si hay espacios disponibles dentro de las instalaciones de la universidad siendo esta una solución a la pérdida de tiempo ocasionada por la búsqueda de lugares de aparcamiento.

Es de mucha importancia contar con un sistema que nos permita conocer la disponibilidad de estacionamientos vehiculares ya que la tendencia tecnológica a nivel mundial está orientada a fomentar el máximo confort del usuario, y con este proyecto es posible que el personal administrativo, docentes y estudiantes de la Universidad Técnica del Norte, puedan aparcarse en un menor tiempo.

Un sistema de parqueo vehicular nos ayuda a obtener el máximo provecho de cada uno de los espacios de aparcamiento disponibles, disminuyendo el tiempo de búsqueda de estos lugares, ya que al proporcionar información actualizada a los conductores acerca de la ubicación de las plazas disponibles, estos pueden encontrar plazas de aparcamiento rápida y eficientemente. (Smart Parking Limited, s.f.)

Aparcamiento eficiente no sólo significa automovilistas más felices, sino que también reduce el consumo de combustible y por ende las emisiones de gases contaminantes, reduce el desgaste del vehículo, se puede ahorrar mucho tiempo, y ayuda al flujo del tráfico vehicular mediante una rápida y fácil manera de estacionarse. (PeopleWeb, 2014)

Capítulo 2. Fundamentación Teórica.

Este capítulo contiene una breve descripción de sistemas de parqueo, el análisis de la información bibliográfica de una WSN y sus diferentes componentes, así como sus principales características y topologías. Además de las herramientas de almacenamiento y visualización de la información del sistema.

2.1. Introducción.

Al contar con un sistema que permita detectar la cantidad de estacionamientos libres en un parqueadero convencional, se logra resolver los problemas de estacionamiento tradicional, ya que al conocer cuántos estacionamientos libres tiene cada parqueadero, los conductores pueden dirigirse directamente al parqueadero que tenga disponibilidad de estacionamientos y con esto se puede reducir la congestión del tráfico dentro de los parqueaderos, ya que los usuarios no tendrán que recorrer por todos los parqueaderos en busca de un estacionamiento libre, además ayudará a aumentar el uso de todos los estacionamientos.

Las redes inalámbricas de sensores han llegado a ser cada vez más aceptadas y esta evolución de las comunicaciones inalámbricas conlleva cambios importantes en la creación de redes de datos, y se está logrando que las redes integradas permitan liberar de cables al usuario y al profesional de las comunicaciones, es por eso que en este sistema se utilizara una red de sensores inalámbrica.

2.2. Sistema de monitoreo.

Realizar un monitoreo es una actividad de varias dimensiones y diferentes perspectivas, a través de puntos de sensores, puntos de atención y elementos donde se establecen prioridades y alertas para analizar las condiciones actuales frente a parámetros previamente establecidos, si son infringidos se crea un mensaje en forma de notificación y es enviado al administrador. (Sistembel, 2015)

2.2.1. Características de un sistema de monitoreo.

- Alertas de manera rápida y sencilla en respuesta a eventos establecidos, estas alertas pueden ser enviadas mediante un email, SMS y vía telefónica.
- Análisis en tiempo real.
- Gráficas interactivas.
- Historial de monitoreo.

2.2.2. Ventajas de un sistema de monitoreo.

Ayuda a conservar la credibilidad: un buen sistema de monitoreo siempre habla bien del funcionamiento de la organización.

Aumentar la eficiencia: el personal que se encarga de la administración del sistema y los usuarios pueden dedicarse a otras actividades ya que el sistema se encarga de avisar y alertar cuando sea necesario.

Un sistema de monitoreo posibilita el incremento del desempeño de la organización desde muchas perspectivas, ya que al perder el tiempo y afectar las operaciones de la organización no son opciones de efectividad. En pocas palabras un sistema de monitoreo es la solución para mejorar la calidad de los servicios e incrementar el orden.

2.3. Red de sensores inalámbricos (WSN).

Una red de sensores inalámbricos es una infraestructura que consta de múltiples estaciones llamadas nodos, cada nodo está compuesto de elementos de detección o medición (transductor), de procesamiento (microordenador) y de comunicación (transceptor), el cual se encargara de transmitir los datos de manera inalámbrica hacia el Gateway y posteriormente a la estación base que se encuentra conectada a un ordenador que puede comunicarse hacia el exterior a través de Internet o una red de área local (LAN), que le da al administrador la capacidad de monitorear, observar y reaccionar ante eventos y fenómenos en un entorno específico. (Rouse, TechTarget, 2006)

Los parámetros comúnmente monitoreados son temperatura, humedad, presión, dirección y velocidad del viento, intensidad de la iluminación, intensidad de vibración, intensidad del sonido, voltaje de la línea de alimentación, concentraciones químicas, niveles de contaminantes y funciones vitales del cuerpo entre otras.

2.3.1. Características de una WSN.

Algunas de las características que tiene una WSN son las siguientes:

- No se requiere una infraestructura de red. Debido a que una red WSN no tiene necesidad alguna de contar con una infraestructura física para su funcionamiento, ya que cada nodo puede operar como transmisor, receptor o ruteador.
- Se debe considerar que el consumo de energía y de memoria de cada nodo es muy reducido, debido a que la mayoría de este tipo de redes funciona con baterías y deben tener una larga autonomía de funcionamiento. Por lo cual pueden operar sin mantenimiento durante varios meses o años.
- Posee una topología dinámica, debido a que es una red escalable, la topología puede variar ya que los nodos son auto-configurables, tolerancia a fallos y presentan una elevada fiabilidad.
- Utilizan tecnologías inalámbricas de corto alcance, el encaminamiento entre dos nodos no necesita visión directa con el nodo destino porque pueden comunicarse mediante multisalto.
- Son capaces de integrarse con diferentes tecnologías para medir datos en ramas como la agricultura, biología, medicina, etc. En si tienen una alta gama de aplicaciones.
- Las WSNs presentan una elevada fiabilidad, tolerancia a fallos y bajo costo.
- Cada nodo está destinado a ser cada vez lo más pequeño posible.

2.3.2. Elementos de una WSN.

Las WSNs se componen principalmente de tres elementos, nodos, gateway y estación base como se puede observar en la *Ilustración 1*.

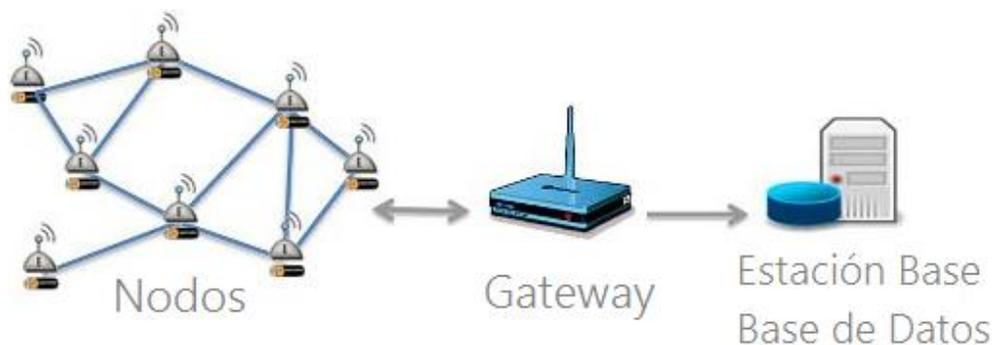


Ilustración 1. Componentes principales de una WSN

Fuente: <https://www.modpow.es/es/redes-de-sensores-inalambricas-wsn-wireless-sensor-network/>

2.3.2.1. Nodo.

El nodo sensor es una de las partes principales de una WSN. El hardware de un nodo sensor incluye generalmente cuatro partes: batería, un sensor, un micro-controlador y un transceptor inalámbrico.

La batería se encarga de alimentar de energía a todos los elementos del nodo. Un sensor se encarga de recoger y transformar las señales, tales como luz, vibración y señales químicas, etc. en señales eléctricas y luego transferirlas al micro-controlador. El micro-controlador recibe los datos del sensor y procesa los datos en consecuencia. El Transceptor Inalámbrico (módulo de RF) transfiere los datos, de manera que se puede lograr la realización física de la comunicación.

Es importante que el diseño de todas las partes de un nodo WSN se considere las características de tamaño pequeño y potencia limitada.

2.3.2.2. Gateway.

Es el dispositivo destinado a la interconexión entre la red de sensores y la estación base.

2.3.2.3. Estación base.

Actúa como recolector de la información de todos los Gateway basado en un ordenador común o un sistema integrado, puede ser un ordenador que tiene la capacidad de comunicarse hacia el exterior a través de Internet o una red de área local (LAN).

2.3.3. Arquitectura de una WSN.

El término arquitectura ha sido adoptado para describir la forma como está constituido cualquier tipo de sistema. La arquitectura de una WSN es la agrupación de varios elementos principales como se muestra en la *Ilustración 2*.

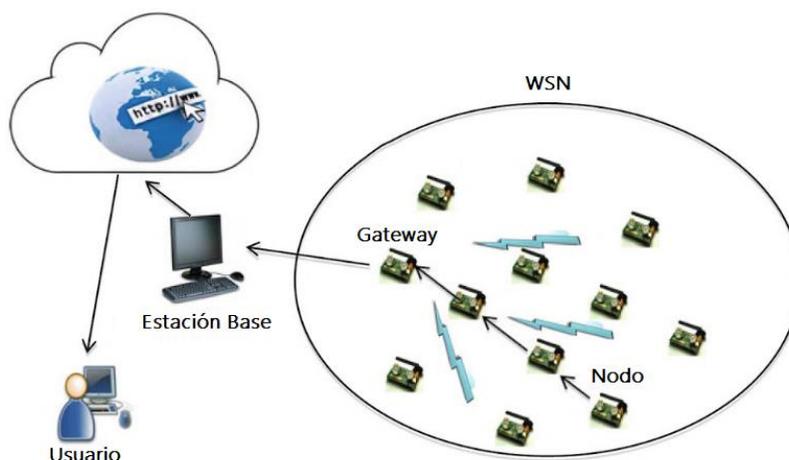


Ilustración 2. Arquitectura de una WSN
Fuente: (Fahmy, 2016)

Los nodos de la red de sensores tienen como objetivo principal realizar la detección de los eventos que se produzcan alrededor de ellos, los datos obtenidos son transformados y enviados al gateway a través de un medio inalámbrico, posteriormente el gateway envía los datos a la estación base, donde esta información puede ser almacenada, tratada y analizada para finalmente poder mostrarla al usuario a través de una LAN o Internet. (Fahmy, 2016)

2.3.4. Topologías de una WSN.

Existen tres tipos de topologías en las que pueden ser configuradas una WSN, en estrella, árbol y malla, tal como se muestra en la *Ilustración 3*.

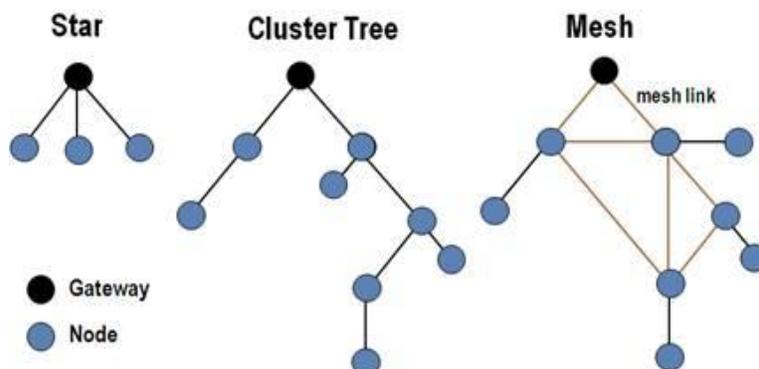


Ilustración 3. Topologías de una WSN

Fuente: <http://www.ni.com/white-paper/7142/es/#toc2>

2.3.4.1. Topología en estrella.

En esta topología, el Gateway está rodeado por un grupo de sensores finales, donde cada nodo se conecta directamente al gateway. Esta topología es muy llamativa debido a su simplicidad, pero al mismo tiempo tiene algunas desventajas. En caso de que el gateway deje de funcionar, toda la red deja de funcionar porque todo el tráfico debe viajar a través del centro de la estrella. Por la misma razón, el gateway podría fácilmente ser un cuello de botella para el tráfico dentro de la red, especialmente para una red a gran escala que puede tener más de un centenar de nodos. (Farej & Abdul-Hameed, 2015)

2.3.4.2. Topología en árbol.

En una red en árbol, el Gateway inicializa la red y es la parte de mayor jerarquía del árbol. El gateway ahora puede tener enrutadores o dispositivos finales conectados a él y en cada enrutador

puede haber más nodos secundarios conectados. Los nodos secundarios no pueden conectarse a un dispositivo final porque no tiene la capacidad de retransmitir mensajes. Esta topología permite diferentes niveles de nodos, con un coordinador que se conecta al nivel más alto. Para pasar mensajes a otros nodos de la misma red, el nodo de origen debe pasar el mensaje a su coordinador, que es el nodo más arriba por un nivel del nodo de origen, y el mensaje es retransmitido continuamente más arriba en el árbol hasta que alcanza el nodo de destino. Debido a que el mensaje puede tomar solamente una trayectoria, este tipo de topología no es la más confiable. Si un enrutador falla, entonces todos sus nodos pierden la comunicación con el resto de la red. (Farej & Abdul-Hameed, 2015)

2.3.4.3. Topología en malla.

La topología de malla es la más flexible y confiable entre las topologías. La flexibilidad está presente porque un mensaje puede tomar múltiples rutas desde la fuente hasta el destino. Si un enrutador en particular falla, entonces el nodo tiene la capacidad de buscar un camino alternativo para enviar el mensaje. (Farej & Abdul-Hameed, 2015)

2.3.5. Estándares de comunicación inalámbrica utilizados en WSNs.

El concepto de usar la comunicación inalámbrica para recolectar información o tareas de control no es nuevo. Existen varios estándares para redes inalámbricas de corto alcance, incluyendo la red inalámbrica de área local (WLAN) IEEE 802.11 e IEEE 802.15.1. Cada una de estas normas tiene sus ventajas en aplicaciones particulares. El estándar IEEE 802.15.4 se ha desarrollado específicamente para abordar la necesidad de implementación de redes inalámbricas de baja velocidad de datos con un consumo de energía bajo. (Farahani, 2008)

2.3.5.1. IEEE 802.11.

IEEE 802.11 (Wi-Fi) es una familia de estándares; IEEE 802.11b opera en la banda de 2,4 GHz, que es común con Bluetooth y ZigBee. IEEE 802.11b tiene una velocidad de datos de hasta 11 Mbps, y una de sus aplicaciones típicas es proporcionar conexión a Internet inalámbrica. El rango de cobertura de IEEE 802.11b es típicamente entre 30 y 100 metros.

2.3.5.2. IEEE 802.15.1.

IEEE 802.15.1 (Bluetooth), tiene una velocidad de datos más baja (menos de 3 Mbps) y su rango de cobertura es típicamente de 2 a 10 metros. Bluetooth no está diseñado para soportar la comunicación entre redes de varios nodos. Una aplicación popular de Bluetooth está en auriculares inalámbricos, donde Bluetooth proporciona los medios para la comunicación entre un teléfono móvil y un auricular manos libres.

2.3.5.3. IEEE 802.15.4.

IEEE 802.15.4 (ZigBee), tiene la tasa de datos más baja y la complejidad entre estos tres estándares y proporciona un consumo energético significativamente bajo. La tasa de datos muy baja de ZigBee significa que no es la mejor opción para implementar una conexión inalámbrica a Internet o un auricular inalámbrico de calidad donde se desee más de 1 Mbps. Sin embargo, si el objetivo de la comunicación inalámbrica es transmitir y recibir comandos simples para aplicaciones de monitoreo o control y/o recopilar información de sensores como sensores de temperatura, humedad, ZigBee proporciona buen potencial para hacerlo.

ZigBee es un estándar que define protocolos de comunicación para bajas velocidades de datos de redes inalámbricas de corto alcance. Los dispositivos inalámbricos basados en ZigBee operan en bandas de frecuencia de 868 MHz, 915 MHz y 2,4 GHz. La velocidad máxima de datos es de 250 Kbps. (Flores Carbajal, 2012)

Tabla 1. Comparación de los estándares de comunicación inalámbrica.

Estándar	IEEE 802.11	IEEE 802.15.1	IEEE 802.15.4
Característica			
Velocidad	54 Mbps	1 Mbps	250 Kbps
Consumo de Potencia	400 mA transmitiendo 20 mA en reposo	400 mA transmitiendo 0.2 mA en reposo	1.8 mA transmitiendo 5.1 uA en reposo
N° de nodos por master	32	7	64,000
Latencia	3 s	10 s	30 ms
Alcance	100	10	100
Ventajas	Gran ancho de banda.	Sustituto del cable.	Bajo consumo de energía.
Aplicaciones	Navegar en Internet, redes de ordenadores, transferencia de ficheros.	Wireless USB, móviles, informática.	Sistemas de control, productos dependientes de baterías, sensores.

Fuente: Adaptado de (Flores Carbajal, 2012)

2.3.6. Almacenamiento de datos de una WSN.

El almacenamiento de datos ha evolucionado rápidamente, desde tener dispositivos que almacenen bytes a dispositivos capaces de almacenar terabytes o peta bytes, donde el tamaño de los dispositivos se ha reducido significativamente, se debe tener en cuenta que el almacenamiento

de datos es un aspecto de mucha importancia que debe ser analizado desde distintos puntos de vista de acuerdo a las necesidades del usuario, empresa u organización. Además actualmente se tiene plataformas de almacenamiento en la nube, donde no se tiene almacenamiento físico local.

Existen tres métodos de almacenamiento y recuperación de datos para una WSN, denominados Almacenamiento Externo (ES), Almacenamiento Local (LS) y Almacenamiento Centrado en Datos (DCS).

2.3.6.1. Almacenamiento externo (ES).

En el almacenamiento externo los nodos envían datos a la estación base o gateway sin procesamiento previo o esperando cualquier consulta, y por lo tanto el tráfico generado es altamente dirigido desde muchos nodos hacia uno o varios nodos de frontera creando un potencial cuello de botella que llega a la estación base. El tráfico excesivo se crea debido a la notificación continua de datos de los sensores, lo cual produce un alto consumo de energía por nodo, reduciendo la vida útil total de la red. El almacenamiento externo también puede tener una desequilibrada tasa de consumo de energía entre los nodos debido a la variación de las distancias entre la estación base y el nodo sensor. Además, dado que el nodo de frontera es el único responsable de agregar / fusionar datos y responder a todas las consultas, el almacenamiento externo puede resultar un poco lento ante las consultas realizadas. (Ahmed & Gregory, 2012)

El almacenamiento externo se puede dar en una base de datos local, la cual es llevada a cabo en una computadora individual o en su servidor de almacenamiento. O en una base de datos en la nube o cloud, la cual se refiere a contar con un servidor virtual en el cual se puede alojar la

información y poder acceder a ella desde cualquier parte del mundo siempre y cuando tengamos conexión a internet.

2.3.6.2. Almacenamiento local (LS).

En el almacenamiento local cada nodo mantiene sus datos almacenados localmente y utiliza la inundación de peticiones para consultas de datos, los cuales consumen una cantidad significativa de recursos energéticos. Dado que el nodo que realiza la consulta no conoce el nodo objetivo que almacena los datos de interés, ejecuta una consulta a todos los nodos de la red para la obtención de los datos. (Ahmed & Gregory, 2012)

2.3.6.3. Almacenamiento centrado en datos (DCS).

En el almacenamiento centrado en datos, se selecciona un nombre de evento para encontrar la ubicación geográfica donde se envían los datos para ser almacenados. Los datos con el mismo nombre de evento se almacenan en el nodo más cercano a la misma ubicación geográfica. Por lo tanto, las consultas con un nombre determinado pueden ser enviadas directamente al nodo que almacena los datos con nombre evitando inundaciones de peticiones de datos. (Ahmed & Gregory, 2012)

2.4. Hardware.

Hardware se refiere a todas las partes físicas, es decir que son tangibles como por ejemplo: los componentes electrónicos, eléctricos, cajas de protección o cualquier elemento físico. Está definido por la RAE como: "Conjunto de los componentes que integran la parte material de una computadora", sin embargo, no necesariamente se aplica a computadoras.

2.4.1. Sensores.

Un sensor es un dispositivo que detecta y responde a algún tipo de entrada desde el entorno físico. La entrada específica podría ser luz, calor, movimiento, humedad, presión o cualquiera de una gran cantidad de otros fenómenos ambientales. La salida generalmente es una señal que se la procesa para convertirla en un formato legible para el ser humano, estos datos pueden ser mostrados en la ubicación del sensor o transmitidos a través de la red y tener acceso a toda la información recolectada mediante el uso de dispositivos que permitan la lectura de dicha información.

Los sensores que se presentan a continuación, se los ha tomado en cuenta debido a que estos sensores tienen un funcionamiento similar a los sensores utilizados en sistemas de parqueaderos comerciales y además se encuentran disponibles en el mercado nacional a un precio asequible.

2.4.1.1. *Ultrasónico HC-SR04.*

Un sensor ultrasónico es un dispositivo que puede medir la distancia a un objeto mediante el uso de ondas de sonido. Mide la distancia enviando una onda de sonido a una frecuencia específica y escuchando que la onda de sonido rebote. Al registrar el tiempo transcurrido entre la onda de sonido que se genera y la onda de sonido que rebota, es posible calcular la distancia entre el sensor de sonda y el objeto.

Ofrece una excelente detección de rango sin contacto con alta precisión y lecturas estables en un paquete fácil de usar. De 2 cm a 400 cm o de 1 a 13 pies. Su funcionamiento no se ve afectado

por la luz solar o el material negro, (aunque los materiales acústicamente blandos, como la tela, pueden ser difíciles de detectar). (Electronilab, s.f.)

2.4.1.2. Infrarrojo SHARP 2Y0A21.

Con sensor Sharp GY2Y0A21YK (comúnmente conocido como 2Y0A21) se puede detectar objetos y además medir la distancia a la que se encuentra. Esto lo consigue gracias a un emisor y un receptor Basados en luz infrarroja, este dispositivo es muy utilizado en proyectos donde se necesita realizar medidas de distancia con buena precisión.

Este sensor consta de tres pines: alimentación, tierra y salida. La tensión en esta última variará entre 0,3 y 3,1 voltios en función de la distancia medida, según la curva proporcionada por el fabricante en su hoja de datos.

La alimentación de este sensor debe estar entre 4,5 y 5,5 voltios, y se recomienda que sea lo más estable posible, ya que caso contrario se puede haber fallas de tensión en su pin de salida y por ende datos erróneos. Una manera de equilibrar las oscilaciones de tensión a la entrada es la colocando un condensador de 10 μ F entre los pines de alimentación y tierra a modo de filtro.

2.4.1.3. Magnetómetro HMC5883L.

Este componente basa su operación en la tecnología AMR (Anisotropic Magnetoresistive) y le permite medir tanto la dirección como la magnitud del campo magnético terrestre. Este magnetómetro HMC5883L tiene dentro 3 sensores magneto-resistivos dispuestos en tres ejes perpendiculares (los ejes cartesianos X, Y y Z) que pueden ser usados como brújula o compas digital.

El campo magnético afecta a estos sensores modificando de alguna manera la corriente que fluye a través de ellos. El componente HMC5883L se comunica con Arduino a través del protocolo I2C, un protocolo muy sencillo de usar, además es necesario contar con la librería HMC5884L para Arduino.

Posee un regulador de voltaje en placa, por lo que puede alimentarse directamente con 5V, del mismo modo el puerto I2C es compatible tanto con 3.3V como con 5V, sin necesidad de usar un conversor adicional. Es recomendable utilizarlo de modo que el módulo se encuentre paralelo al suelo y alejado de imanes, de otro modo las lecturas llegaran a ser erróneas.

Tabla 2. Características de los sensores.

Características	Ultrasónico	Infrarrojo	Magnetómetro
Voltaje de Operación:	5V	4.5V – 5.5V	3.3V – 6V
Interfaz de comunicación:	Input Trigger Output Echo	Salida Análoga	I2C
Resolución:	0.3 cm	-----	5 mili-Gauss
Distancia de medición:	2cm – 4m	10cm – 80cm	-----
Precisión:	<15°	-----	De 1 a 2 grados Sexagesimales
Data rate:	40 KHz	Depende del micro controlador	160 Hz
Temperatura de operación:	-----	-10 a 60 °C	-30 a 85 °C
Consumo de corriente:	15 mA	30 – 40 mA	(2 – 100)uA

Fuente: Adaptado de los Datasheet del sensores.

2.4.2. Placas de procesamiento Arduino.

Arduino es una plataforma de electrónica de código abierto que cuenta con hardware y software fácil de usar. Las placas Arduino pueden leer entradas digitales y analógicas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, la cual puede permitir activar un motor, encender un LED, publicar algo en línea, etc. Además se puede decirle a la placa qué hacer enviando un conjunto de instrucciones al micro controlador en la placa. Para hacerlo, utiliza el lenguaje de programación Arduino y el software Arduino (IDE). (Arduino, 2017)

Arduino también simplifica el proceso de trabajo con micro controladores, y además ofrece algunas ventajas como estas: el precio de sus placas es económico, son multiplataforma, tienen un entorno de programación sencillo y claro y su software y hardware es de código abierto y extensible. A continuación en la *Tabla 3* y *Tabla 4*, se puede observar las principales características de las placas de Arduino más conocidas.

Tabla 3. Características de las placas Arduino #1.

Características	Arduino Uno	Arduino Mega2560	Arduino Yun	Arduino Ethernet
Microcontrolador	ATmega328P	ATmega2560	ATmega32U4	ATmega328
Tensión de funcionamiento	5 voltios	5 voltios	5 voltios	5 voltios
Voltaje de entrada recomendando	7-12 voltios	7-12 voltios	5 voltios	7-12 voltios
Voltaje de entrada límite	6-20 voltios	6-20 voltios	-----	6-20 voltios
Pines de entrada/salida digitales	14 los cuales 6 proporcionan salida PWM	54 los cuales 15	20 los cuales 7	14 los cuales 4 proporcionan salida PWM

		proporcionan salida PWM	proporcionan salida PWM	
Pines de entrada analógicos	6	16	12	6
Corriente DC por pin de entrada/salida	20mA	20mA	40mA	40mA
Corriente DC del pin de 3.3 Voltios	50mA	50mA	50mA	50mA
Memoria Flash	32 KB de los cuales 0,5 KB utilizados por el gestor de arranque	256 KB de los cuales 8 KB utilizados por el gestor de arranque	32 KB de los cuales 4 KB utilizados por el gestor de arranque	32 KB de los cuales 0,5 KB utilizados por el gestor de arranque
SRAM	2 KB	8 KB	2,5 KB	2 KB
EEPROM	1 KB	4 KB	1 KB	1 KB
Velocidad de reloj	16 MHz	16 MHz	16 MHz	16 MHz
Precio aproximado	22 dólares	38,50 dólares	68,20 dólares	43,89 dólares

Fuente: Adaptado de (Arduino, 2017).

Tabla 4. Características de las placas Arduino #2.

Características	Arduino Micro	Arduino Nano	Arduino Mini-Pro	Arduino Lilypad
Microcontrolador	ATmega32U4	ATmega328	ATmega328	ATmega32U4
Tensión de funcionamiento	5 voltios	5 voltios	5 voltios	3.3 voltios
Voltaje de entrada recomendando	7-12 voltios	7-12 voltios	7-9 voltios	3.8-5 voltios
Voltaje de entrada límite	6-20 voltios	6-20 voltios	5-12 voltios	5 voltios
Pines de entrada/salida digitales	20 los cuales 7 proporcionan salida PWM	22 los cuales 6 proporcionan salida PWM	14 los cuales 6 proporcionan salida PWM	9 los cuales 4 proporcionan salida PWM

Pines de entrada analógicos	12	8	8	4
Corriente DC por pin de entrada/salida	20mA	40mA	40mA	40mA
Corriente DC del pin de 3.3 Voltios	50mA	50mA	50mA	-----
Memoria flash	32 KB de los cuales 4 KB utilizados por el gestor de arranque	32 KB de los cuales 2 KB utilizados por el gestor de arranque	32 KB de los cuales 2 KB utilizados por el gestor de arranque	32 KB de los cuales 4 KB utilizados por el gestor de arranque
SRAM	2.5 KB	2 KB	2 KB	2.5 KB
EEPROM	1 KB	1 KB	1 KB	1 KB
Velocidad de reloj	16 MHz	16 MHz	16 MHz	8 MHz
Precio aproximado	19,80 dólares	22 dólares	15,40 dólares	24,95 dólares

Fuente: Adaptado de (Arduino, 2017).

2.4.3. Transceptores inalámbricos.

Un transceptor es una combinación de transmisor / receptor en un solo paquete. El término se aplica a los dispositivos de comunicaciones inalámbricas, tales como teléfonos celulares, teléfonos inalámbricos, radios portátiles de dos vías y radios móviles bidireccionales.

Un transceptor inalámbrico consiste en un transmisor y un receptor. En el transmisor se realiza un proceso conocido como modulación, el cual convierte las señales digitales eléctricas dentro de una computadora en RF o luz, que son señales analógicas. Posteriormente los amplificadores aumentan la magnitud de las señales para ser transmitidos a través de una antena. En el destino, un receptor detecta las señales relativamente débiles y las demodula en tipos de

datos aplicables a la computadora de destino. El transceptor generalmente está compuesto de hardware que es parte de la NIC inalámbrica. (Digi International, 2017)

Los transceptores inalámbricos que se describen a continuación, pueden ser utilizados en redes inalámbricas de sensores y además hacen uso de los estándares de comunicación descritos anteriormente, por lo cual estos dispositivos posteriormente serán evaluados y elegidos.

2.4.3.1. WiFi.

El Módulo WiFi ESP8266 es autónomo con una pila de protocolos TCP / IP integrados que le permite a cualquier microcontrolador acceder a su red WiFi. El ESP8266 es capaz de alojar una aplicación o descargar todas las funciones de red WiFi de otro procesador de aplicaciones. Cada módulo ESP8266 viene pre-programado con un comando AT que configura el firmware, lo que significa que simplemente puede conectarlo a su dispositivo Arduino y obtener la mayor capacidad de WiFi que ofrece el Shield WiFi. El módulo ESP8266 es una placa extremadamente rentable con una gran comunidad en constante crecimiento. (EsparkFun Electronics, 2015)

Este módulo tiene una capacidad suficiente de procesamiento y almacenamiento a bordo que le permite integrarse con los sensores y otros dispositivos específicos de la aplicación a través de sus GPIO con un desarrollo mínimo inicial y carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite un circuito externo mínimo, incluido el módulo frontal, diseñado para ocupar un área mínima de PCB. A continuación en la *Tabla 5* se puede observar sus principales características.

Tabla 5. Características del módulo WiFi ESP8266.

Características	WiFi ESP8266
Procesador:	Tensilica L106 32-bit
Voltaje de operación:	3 V – 3.6 V
Consumo de corriente	80 mA
Consumo de corriente en modo sleep:	20 uA
Temperatura de operación:	-40 a 125 °C
Protocolos que soporta:	IPv4 – TCP/UDP/HTTP/FTP
WiFi:	802.11 b/g/n
Potencia de salida:	19.5 dBm en modo 801.11b
Memoria Flash:	Hasta 4 MB
Memoria RAM:	80 KB

Fuente: Adaptado de (EsparkFun Electronics, 2015)

2.4.3.2. *Bluetooth.*

El módulo HC-05 es un módulo Bluetooth SPP (Protocolo de puerto serie) fácil de usar, diseñado para la configuración de conexión en serie inalámbrica y transparente. Es ideal para aplicaciones inalámbricas, fácil de implementar con PC, microcontrolador o módulos Arduino. La tarjeta incluye un adaptador con 6 pines de fácil acceso para uso en protoboard.

El módulo bluetooth HC-05 se lo puede configurar para trabajar como maestro o esclavo. En el modo maestro puede conectarse con otros módulos bluetooth, mientras que en el modo esclavo debe esperar las peticiones de conexión. (Electronilab, s.f.) A continuación en la *Tabla 6* se presenta sus principales características:

Tabla 6. Características del módulo Bluetooth HC-05.

Características	Bluetooth HC-05
Protocolo Bluetooth compatible:	V2.0
Protocolo de comunicación:	UART RS232
Voltaje de alimentación:	3.3 V – 6 V
Voltaje de operación:	3.3 V
Corriente de operación:	< 40 mA
Corriente en modo sleep:	< 1 mA
Baud rate ajustable:	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
Baud rate por defecto:	9600
Tamaño:	1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)

Fuente: Adaptado del Datasheet Bluetooth HC-05

2.4.3.3. XBee.

XBee es el nombre comercial de Digi, de una familia de módulos de comunicación por radio y están basados en el estándar ZigBee, pero en esta ocasión se centrará en el XBee S2C, el cual es un módulo de bajo costo que permite realizar conexiones inalámbricas entre dispositivos electrónicos, trabaja con una frecuencia de 2.4Ghz y permite crear redes de conexión punto a punto, punto a multipunto, broadcast y mesh. Al utilizar el conjunto de características ZigBee PRO, el XBee S2C es interoperable con otros XBee, aunque el cambio más relevante está en su firmware, el cual abarca todos los seis firmwares de sus versiones anteriores en uno solo

(Coordinador AT, Coordinador API, Router AT, Router API, End Device AT y End Device API).
(Digi International, 2017)

El XBee S2C incorpora la comunicación SPI (serial peripheral interface) por lo cual es posible realizar el intercambio de datos a alta velocidad entre dispositivos, esto permite optimizar la conexión con microcontroladores. También posee mejoras en los rangos de comunicación y potencia de transmisión. A continuación en la *Tabla 7* se puede observar algunas características de este transceptor.

Tabla 7. Características del XBee ZB S2C.

Características	XBee ZB S2C
Alcance interior / Urbano	Hasta 200 pies (60 m)
RF al aire libre la línea de visión rango	Hasta 4000 pies (1200 metros)
La potencia de transmisión	6.3mW (+ 8dBm) modo Boost
Velocidad de datos de RF	250 Kbps
Tensión de alimentación	2,1 - 3,6 V
Corriente de transmisión (típico)	45 mA (+ 8dBm) Modo Boost
Corriente de apagado	<1uA
Frecuencia	ISM de 2,4 GHz
Temperatura de funcionamiento	-40 A 85 C
Opciones de antena	A través de orificios: PCB, integrado Whip, U.FL, RPSMA
Las topologías de red	Punto a punto, estrella, malla
Número de canales	16 canales de secuencia directa

Fuente: Adaptado de (Digi International, 2017)

2.4.4. Dispositivos para la presentación de información al usuario.

Existen diversos dispositivos que permiten mostrar información y que ayudan en la tarea de la difusión y presentación de la misma de una manera amigable para el usuario. Si bien la interfaz por la cual se presenta la información, debe integrarse adecuadamente a los dispositivos, en este contexto la interface podrá asistir al usuario del sistema a formular sus consultas, navegar, obtener la información disponible y comprender los resultados mostrados.

Los indicadores para informar que un parqueadero esta libre u ocupado cumplen una importante función: mantienen informados a los usuarios del estado de disponibilidad del aparcamiento. Los usuarios del parqueadero agradecen disponer de esta información para así poder decidir con suficiente antelación si ingresan o no al aparcamiento. Para el caso de este tema de titulación se analizaran varias formas de presentación de los datos del sistema hacia el usuario, entre las cuales tenemos: dispositivo móvil, rótulos electrónicos y pantalla led.

2.4.4.1. Dispositivo móvil.

Un dispositivo móvil es esencialmente una computadora de mano, tableta de mano, teléfono inteligente u otro dispositivo que está hecho para la portabilidad, por lo tanto, es compacto y liviano. Aunque la categoría de dispositivo móvil parezca incluir cualquier dispositivo electrónico lo suficientemente pequeño como para transportarse, el término implica la capacidad de comunicarse inalámbricamente. (Rouse, TechTarget, 2006)

Las nuevas tecnologías de almacenamiento, procesamiento y visualización de datos han permitido que estos dispositivos pequeños hagan casi cualquier cosa que tradicionalmente se haya

hecho tradicionalmente con computadoras personales más grandes. Muchos de estos comparten características comunes, como interfaces de pantalla táctil con pantallas a color, enlaces a programas de software de escritorio y correo electrónico, y acceso a plataformas inalámbricas. Los dispositivos móviles tienen características similares. Entre las cuales están:

- Wi-Fi o acceso a internet
- Una batería que alimenta el dispositivo durante varias horas
- Un teclado físico o en pantalla para ingresar información
- Tamaño y peso que permite llevarlo en una mano y manipularlo.
- Interfaz de pantalla táctil en casi todos los casos
- Un asistente virtual.
- La capacidad de descargar datos de Internet, incluidas aplicaciones y libros
- Funcionamiento inalámbrico.

2.4.4.2. *Rotulo electrónico.*

Las pantallas lineales de leds son el tipo de rótulo electrónico idóneo para facilitar toda clase de información de una forma dinámica y muy llamativa. Una de sus muchas aplicaciones es su utilización como pantalla publicitaria. Instalar una pantalla lineal de led permite tener un propio medio de comunicación exterior, y todo ello desde su PC de forma fácil, rápida y sencilla gracias a su software intuitivo. (Rotulos Electronicos, 2017)

La información exhibida en este tipo de rótulos electrónicos es limitada, pero suficiente para que un conductor pueda leerla sin descuidar la conducción. Localizar o no un parking o poder

informarse de los precios del combustible en una gasolinera pero dependerán en muy medida de que estos establecimientos sean llamativos. Dentro de sus principales características tenemos:

- Sus dimensiones suelen ser pequeñas.
- No es posible la reproducción de imágenes y videos.
- Su principal función es la proyección de textos y efectos.
- Su fabricación se estandariza con medidas determinadas.

2.4.4.3. Pantalla led.

Las pantallas led de exteriores son una herramienta de comunicación de alto impacto perfecta para hacer llegar a las personas todo tipo de información. Las llamativas pantallas led reproducen contenidos digitales en forma de animaciones y video, y atraen la atención de forma ineludible, ideales para señalar y facilitar información a los conductores. Es esta capacidad la que permite emitir mensajes en un lenguaje visual al que cada vez estamos más acostumbrados. Al reproducir contenidos en formato digital pueden ser manejados de forma remota, actualizando los anuncios desde un PC conectado a internet. (Rotulos Electronicos, 2017) Dentro de sus principales características tenemos:

- Pueden alcanzar grandes dimensiones.
- Permite la reproducción de imágenes y vídeos.
- Pueden incluir procesadores de vídeo para la reproducción en streaming.
- Suele fabricarse bajo pedido en función de las necesidades de cada cliente.
- Costo elevado.

2.5. Software.

Esta palabra se refiere a las partes intangibles de un sistema, es decir se lo llama software al conjunto de los componentes lógicos, los cuales ayudan a controlar el hardware y permiten hacer posible la realización de tareas específicas.

2.5.1. IDE Arduino.

El Arduino Software (IDE) hace que sea fácil de escribir código y subirlo a cualquiera de sus placas. Se ejecuta en Windows, Mac OS X y Linux. Dado que el Arduino es como un pequeño ordenador que ejecuta una serie de códigos que previamente le hemos introducido, necesitaremos un programa para poder meter estos códigos a la propia placa. Este programa se llama IDE, que significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). Este IDE estará instalado o ejecutado en nuestro PC, es un entorno muy sencillo de usar y en él escribiremos el programa que queramos que el Arduino ejecute. Una vez escrito, lo cargaremos a través del USB y Arduino comenzará a trabajar de forma autónoma. (Arduino, 2017)

Para cargar un programa en Arduino, el IDE de Arduino contiene un editor de texto para escribir nuestro sketch, una consola de error y un área con los menús y los botones que realizan las funciones más comunes como son abrir sketch, guardar sketch, compilar y cargar programa.

2.5.2. Aplicación móvil o nativa.

Las aplicaciones móviles son aplicaciones desarrolladas para pequeños dispositivos portátiles, tales como teléfonos móviles, teléfonos inteligentes, PDA, entre otros. Las aplicaciones móviles pueden venir precargado en el dispositivo, así como puede ser descargado por los usuarios

de las tiendas de aplicaciones o Internet. Las plataformas de teléfonos inteligentes más populares que soportan aplicaciones móviles son Android, iOS, Windows Phone, etc. (Rouse, TechTarget, 2013) Dentro de sus principales características tenemos las siguientes:

- Acceso rápido y sencillo a la información sin necesidad de autenticación en cada acceso.
- Mejor experiencia del usuario.
- Disponibilidad en App store.
- La actualización de la App es constante.
- Tienden a ser más caras de implementar.
- El código del cliente no es reutilizable entre las diferentes plataformas.
- Consumo de los recursos del dispositivo.

2.5.3. Página Web.

Una página Web es un documento comúnmente escrito o desarrollado en lenguajes como HTML, JavaScript y CSS, que es accesible a través de internet utilizando un navegador de internet. A una página web se accede mediante la introducción de una dirección URL y puede contener texto, gráficos y enlaces a otras páginas web y archivos. Para ver una página web se requiere un navegador. (Internet Explorer, Edge, Safari, Opera, Firefox, Chrome, etc.) (Computer Hope, s.f.) Dentro de sus principales características tenemos:

- El mismo código base es reutilizable en múltiples plataformas.
- No ocupa espacio de memoria en los dispositivos.
- Proceso de desarrollo más sencillo y económico.

- No necesitan ninguna aprobación externa para publicarse (a diferencia de las App nativas para estar visibles en App Store).
- La navegación, interacción y el tiempo de respuesta es menor que en una App nativa.

2.5.4. Almacenamiento en la nube.

El almacenamiento en la nube es un modelo de computación en la nube en el que los datos se almacenan en servidores remotos a los que se accede desde Internet, o "nube". Es mantenido, operado y administrado por un proveedor de servicios de almacenamiento en la nube en servidores de almacenamiento que se basan en técnicas de virtualización.

El almacenamiento en la nube funciona a través de la virtualización del centro de datos, proporcionando a los usuarios finales y las aplicaciones una arquitectura de almacenamiento virtual escalable de acuerdo con los requisitos de la aplicación. En general, el almacenamiento en la nube opera a través de una API basada en web que se implementa de forma remota a través de su interacción con la infraestructura de almacenamiento en la nube interna de la aplicación cliente para operaciones de entrada / salida (E / S) y de lectura / escritura (R / W). (Techopedia, s.f.)

Existen diferentes tipos de nubes, y cada una de ellas está orientada a las diversas necesidades de los usuarios, entre las cuales tenemos; nubes públicas, donde los servicios que brinda se alojan en servidores externos; nubes privadas, estas se encuentran a disposición de una sola empresa y no ofrecen servicios a terceros; nubes comunitarias, es la unión entre nubes privadas y son compartidas por varias organizaciones que trabajan en un mismo objetivo; nubes híbridas, son una combinación de las nubes públicas con las privadas de manera que comparten sus recursos entre ellas. (Microsoft Azure, s.f.)

De igual manera los servicios que ofrecen las diferentes nubes están divididas en; software como servicio (SaaS), consiste en el suministro de aplicaciones pero el usuario no necesita instalarla en su ordenador, ya que se hace uso de una interfaz web para acceder y se lo puede hacer desde cualquier dispositivo, pero el usuario no tiene control sobre la infraestructura; plataforma como servicio (PaaS), consiste en ofrecer a los usuarios un entorno de desarrollo donde los mismos usuarios puedan crear sus propias aplicaciones siempre y cuando estas sean soportadas por el proveedor; infraestructura como servicio (IaaS), es la disponibilidad de capacidad de almacenamiento y procesamiento, es donde se alojan físicamente toda la infraestructura física necesaria para ofrecer sus recursos a terceros.

A continuación se describe tres plataformas de almacenamiento orientadas al Internet de las cosas (IoT), las cuales se encargan de guardar los datos, que los dispositivos conectados a Internet constantemente recopilan y transmiten.

2.5.4.1. Xively.

Xively es una división de LogMeIn Inc (LOGM), una empresa pública global que es un proveedor líder de servicios remotos esenciales. Xively se ha convertido en una organización confiable de productos de atención al cliente SaaS, administración remota de TI, acceso y colaboración. Xively se basa en la plataforma en la nube LogMeIn Gravity, que maneja más de 255 millones de dispositivos, usuarios y clientes en siete centros de datos en todo el mundo. (Xively, 2007)

Xively ofrece una solución de aplicación y plataforma IoT para empresas que crean productos y servicios conectados. Xively permite que las empresas conecten sus productos de

forma segura y sólida, administren datos de esas conexiones y se relacionen más estrechamente con sus clientes. La solución da voz a los productos físicos en las relaciones que facilitan entre una empresa y todos sus componentes (clientes, socios, proveedores, instaladores, etc.) para desarrollar, comercializar, vender y soportar mejor la próxima generación de dispositivos conectados.

2.5.4.2. Exosite.

Exosite se define como una mezcla de profesionales obsesionados con IoT que ofrecen soluciones de IoT para empresas basadas en tecnología y listas para aplicaciones industriales específicas. Exosite proporciona plataformas de software de Internet de las cosas (IoT) impulsadas por análisis y de nivel empresarial que permiten a los fabricantes de equipos originales (OEM) aprovechar estratégicamente el revolucionario mundo de los dispositivos conectados a la red. Los servicios basados en la nube de Exosite, con oficinas en América, APAC y EMEA, permiten a los principales fabricantes mundiales de bienes duraderos en los mercados de consumo, comerciales e industriales construir e implementar rápidamente soluciones que aceleren la generación de IoT en sus negocios. (Exosite, s.f.) Las principales características que brinda la solución SaaS de Exosite son:

- Administrar sus activos y flotas de dispositivos.
- Recopilar y almacenar datos.
- Operacionalizar el análisis e información basada en datos.
- Proporcionar aplicaciones web y móviles para usuarios finales.

- Integrarse con los sistemas de TI existentes y los servicios de nube pública (p. Ej., Microsoft, AWS).
- Cultivar un ecosistema de IoT único con elementos intercambiables reutilizables.

2.5.4.3. *Ubidots.*

Ubidots es una plataforma que brinda un servicio en la nube el cual permite almacenar e interpretar información de diversos tipos de sensores en tiempo real, lo cual hace posible crear aplicaciones para el IoT de una manera rápida, fácil y divertida. Gracias a ésta plataforma, es posible ahorrar tiempo y dinero al momento de desarrollar aplicaciones como sistemas de telemetría GPS, sistemas de monitoreo, sistemas para contar vehículos en parqueaderos o calles, etc. (Ubidots, 2017)

En la *Ilustración 4* se puede apreciar las diferencias entre el ahorro en tiempo y esfuerzo al crear una aplicación de Internet de las Cosas con la plataforma Ubidots, o sin ella:

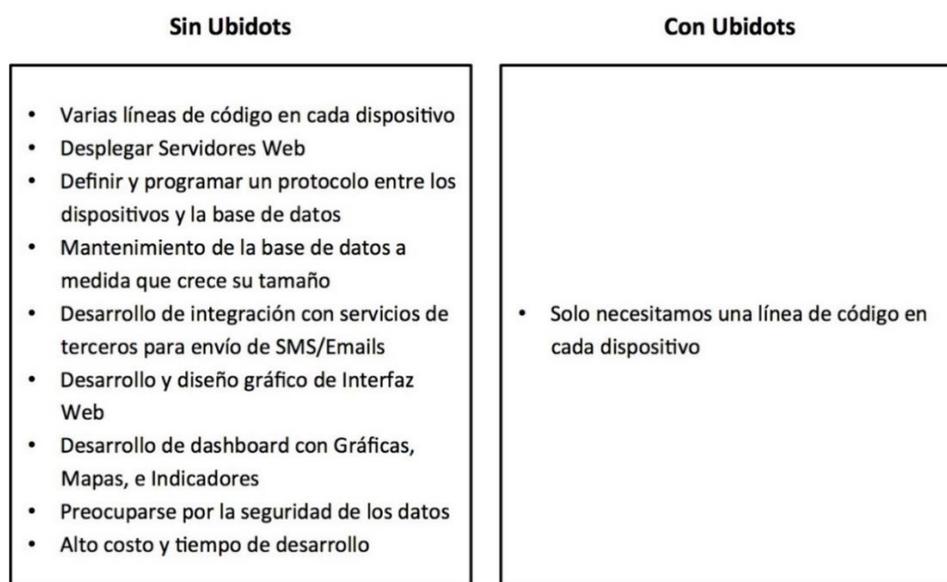


Ilustración 4. Diferencias entre usar o no Ubidots.

Fuente: https://ubidots.com/docs/es/get_started/introduccion.html

2.6. Descripción de sistemas de parqueo comerciales.

Existen varias empresas dedicadas a desarrollar y comercializar sistemas de parqueo inteligentes, ente las más relevantes tenemos las siguientes.

2.6.1. Libelium.

Libelium es una empresa de capital español, fundada en 2006 con una fuerte presencia en mercados que necesitan de comunicaciones inalámbricas. Se trata de un sensor de campo magnético que se incorpora a su conocido desarrollo WaspMote, tarjeta que incorpora tecnología ZigBee para las comunicaciones inalámbricas. La solución aportada por Libelium, es denominada WaspMote Plug & Sense Smart Parking, es una solución aplicada a Smart Cities que permite a los ciudadanos detectar lugares de estacionamiento disponibles.

Es un dispositivo fácil y barato de desplegar, ya que está desarrollado para ser colocado directamente en la superficie de la carretera, sin tener que cavar un agujero en el suelo como lo hacen la mayoría de otras versiones disponibles en el mercado, con esto se logra facilitar y reducir el tiempo de instalación y mantenimiento. Este sistema de sensores es totalmente compatible con tecnologías de radio que permiten un consumo de larga y baja potencia (LPWAN-LoRaWAN y Sigfox). Una característica importante del sistema es la posibilidad de usar ambas tecnologías de radio al mismo tiempo (868 MHz Europa y 900-930 MHz EE.UU/Canadá) o cambiar de una a otra utilizando el sistema gestor de la Nube.

En este sistema de sensores, una estación base es capaz de brindar servicio a miles de nodos alrededor de varios kilómetros en ambiente urbano. Esto ayuda a reducir los costos de instalación,

debido a la reducción del número de estaciones. Además, el consumo de energía del sensor ha sido optimizado, por lo que la batería puede durar hasta 10 años fácilmente. Hay que tener en cuenta que gracias a que los nodos cuentan con un software propio no hay que preocuparse por la programación de los nodos, solo basta con especificar los valores de los parámetros clave en el firmware para que los nodos empiecen a trabajar, además mediante la gestión remota es posible modificar los parámetros de los nodos, y así reprogramar miles de nodos simplemente accediendo al navegador web y estableciendo los valores correctos. (Libelium, 2016)

2.6.1.1. Descripción del hardware.

Tabla 8. Plug & Sense! Smart Parking main characteristics

Característica	Descripción
Frecuencia de operación	Radio Sigfox para Europa: 868,0 a 869,7 MHz Radio LoRaWAN para Europa: 863,0 a 870,0 MHz Radio Sigfox para Estados Unidos: 902.0 a 928.0 MHz Radio LoRaWAN para EE.UU.: 902,0 a 928,0 MHz
Fuente de alimentación	Baterías de litio incorporadas, vida útil esperada de 4-6 años.
Antena	Incluido
Detección	Magnético
Montaje	Sobre el piso
Dimensiones	230 mm de diámetro, 28 mm de altura
Protección	IP67, carcasa completamente sellada
Temperatura de funcionamiento	-20 a +65 °C

Fuente: adaptado de (Libelium, 2016)

2.6.1.2. *Funcionamiento del nodo sensor.*

El bucle básico del nodo consiste en la lectura del sensor y enviar un dato cuando el espacio de estacionamiento cambie de estado. Luego duerme durante tiempo determinado y vuelve a iniciar el bucle.

Algunos eventos pueden obligar al nodo a enviar una trama de información a la nube. Si transcurre el tiempo determinado desde la última transmisión de radio, el nodo enviará una trama Keep-Alive. Esta trama sólo contiene datos básicos del nodo (estado del espacio de estacionamiento y estado de la batería). Es útil saber que no hay cambios en el estacionamiento, pero el nodo sigue funcionando. El nodo también enviará una trama cada 24 horas con los datos de trabajo del día (número de mediciones, número de transmisiones). (Libelium, 2016)

2.6.1.3. *Modos de reposo.*

Plug & Sense! El estacionamiento inteligente tiene 2 modos de reposo: modo de día y modo de noche. El modo de día se usa cuando el estacionamiento tiene cambios con mayor frecuencia. El modo de noche se ha desarrollado para usar cuando se espera que el espacio del estacionamiento tenga menos cambios (por ejemplo, en la noche). Cada modo tiene sus propios parámetros de configuración. (Libelium, 2016)

2.6.2. ParkHelp Mobility and Sustainability Solutions.

ParkHelp Mobility and Sustainability Solutions es una empresa de capital español fundada en 2006. Cuenta con más de 310.000 plazas de aparcamiento en 47 países. Su objetivo principal es la de guiar el tráfico que se encuentre en búsqueda de aparcamiento hacia una plaza de

aparcamiento que se encuentre disponible y susceptible de ser usada por el vehículo guiado. Sus 2 líneas de trabajo son las denominadas Off Street y OnStreet, se basan en la instalación de sensores en el suelo en el caso de la línea OnStreet y en el techo en el modo OffStreet, siendo éstos capaces de detectar si la plaza de aparcamiento se encuentra libre u ocupada. (ParkHelp, s.f.)

ParkHelp's y su línea OnStreet es un sistema de dirección del estacionamiento al aire libre, el cual permite mejorar la gestión de aparcamientos. Los datos de ocupación recogidos por los sensores se alimentan automáticamente para mostrar señales ubicadas estratégicamente dentro de la instalación de estacionamiento para proporcionar a los conductores datos en tiempo real sobre el número de espacios disponibles en cada área. (ParkHelp, s.f.)

2.6.2.1. Descripción del nodo sensor.

Tabla 9. Características principales del nodo sensor.

Característica	Descripción
Frecuencia de operación	433/868 / de 915 MHz
Fuente de alimentación	Baterías incorporadas, vida útil esperada hasta 5 años.
Detección	Campo magnético de 98%
Montaje	Sobre el piso. Los sensores se colocan con pegamento bicomponente
Protección del nodo	IP67

Fuente: adaptado de (ParkHelp, s.f.)

2.6.2.2. Beneficios de la solución.

- Robusto e inmune contra ambientes ruidosos electromagnéticos.
- IP67: ultrarresistente y adecuado para ambientes salinos.

- Sensores y señalización de uso en todo el mundo.
- Fácil instalación sin obra civil.
- Mejora el funcionamiento del aparcamiento.
- Las unidades se encuentran a predisposición de la compra.

2.6.3. Parqueos Sistemas y Soluciones S.A.

Parqueos Sistemas y Soluciones S.A. es la única empresa ecuatoriana proveedora de soluciones inteligentes e integrales para parqueo tarifado público en el Ecuador. Es una empresa que está en continuo desarrollo de nuevas ideas, para ofrecer soluciones innovadoras que resuelvan problemas y mejoren la calidad de vida de las personas.

Sistema Integral de Parqueos es un sistema tecnológico moderno, ecológico, económico, creado con el objetivo principal de brindar una solución informática totalmente diferente a las opciones tradicionales de pago de parqueo tarifado, tanto para espacios públicos como para espacios privados. (Sip, 2015)

2.6.3.1. Descripción del funcionamiento del sistema.

- Estaciona el auto en un lugar disponible
- Envía un mensaje con la clave señalizada en el estacionamiento y el número de placa de tu vehículo al 20500
- El sistema dará de alta tu registro, marcará a ese estacionamiento como ocupado por tu automóvil y recibirás un mensaje SMS notificando tu registro, el cual será tu comprobante

- Si no traes tu celular, acércate al negocio autorizado más cercano para realizar tu pago, ahí te dan un comprobante de pago impreso
- Además si descargas la aplicación puedes consultar los lugares donde opera el sistema y los estacionamientos que se encuentren disponibles.

2.6.3.2. Beneficios de la solución.

- Disponibilidad de Parqueaderos: Información de parqueaderos disponibles por ciudad, sector y ubicación en tiempo real.
- Pagos mediante SMS: Compra de parqueo tarifado sin necesidad de terceras personas y enviando solamente un mensaje de texto SMS.
- Mejora la circulación vehicular: Aumenta la fluidez vehicular haciendo que los automóviles no generen doble o hasta triple fila en busca de un parqueadero libre.
- Elimina el posible cohecho: Toda transacción se genera mediante el sistema tecnológico disminuyendo el error humano y la posible corrupción.
- Maximiza ganancias: Ganancias que permiten realizar obras y apoyar programas sociales para el beneficio de la ciudadanía y el crecimiento de la ciudad.
- Reduce la contaminación: Menos vehículos a dos y tres filas a la espera de un parqueadero disponible significa menor contaminación ambiental y acústica.
- Estadísticas en tiempo real: Transacciones procesadas y realizadas en tiempo real para todos los usuarios del sistema. Recargas, compras y transferencias cuando y donde lo necesite.
- Moderno, ecológico, económico: Tecnología de vanguardia que permite modernizar una ciudad, sin desperdiciar ni utilizar papeles y todo esto a un precio justo e incomparable.

2.7. Metodología.

La metodología científica, está definida como el procedimiento investigativo utilizado principalmente en la creación de conocimiento basado en las ciencias. Se denomina científico porque dicha investigación se apoya en lo empírico y en la medición, ajustándose a los principios específicos de las pruebas de razonamiento.

2.7.1. Tipo de investigación.

Existen varios tipos de investigación, y dependiendo de los fines que se persiguen, se debe elegir uno. En este caso se utilizará la investigación aplicada, la cual se refiere al estudio y la investigación científica que busca resolver problemas prácticos. La investigación aplicada es utilizada para encontrar soluciones a problemas del día a día, curar enfermedades y desarrollar tecnologías innovadoras.

2.7.2. Métodos de investigación.

Se entiende por método al conjunto de procesos que se debe emprender en la investigación con el fin de demostrar la verdad. Para esta investigación se utilizará el método empírico, ya que este método posibilita revelar las características fundamentales del tema de estudio

La investigación empírica permite al investigador hacer una serie de investigaciones referente a su problemática, retomando experiencia de otros autores, para de ahí a partir con su exploración, también conlleva efectuar el análisis preliminar de la información, así como verificar y comprobar las concepciones teóricas. El método empírico que se utilizara es la medición y dentro

de este se aplicaran técnicas que posibiliten la recolección de información, entre las cuales se tiene: encuestas, entrevistas y cuestionarios.

2.7.3. Modelo en cascada.

El modelo en cascada se lo conoce como un modelo de ciclo de vida lineal-secuencial. En este modelo, cada fase debe completarse completamente antes de que la siguiente fase pueda comenzar. Este tipo de modelo de desarrollo de software se utiliza básicamente para proyectos pequeños y no hay requisitos inciertos. Al final de cada fase, se lleva a cabo una revisión para determinar si el proyecto está en el camino correcto y si desea continuar o descartar el proyecto. En este modelo, las pruebas comienzan sólo después de que el desarrollo se haya completado. (ISTQB Exam Certification, s.f.)

2.7.3.1. Diagrama del modelo en cascada.

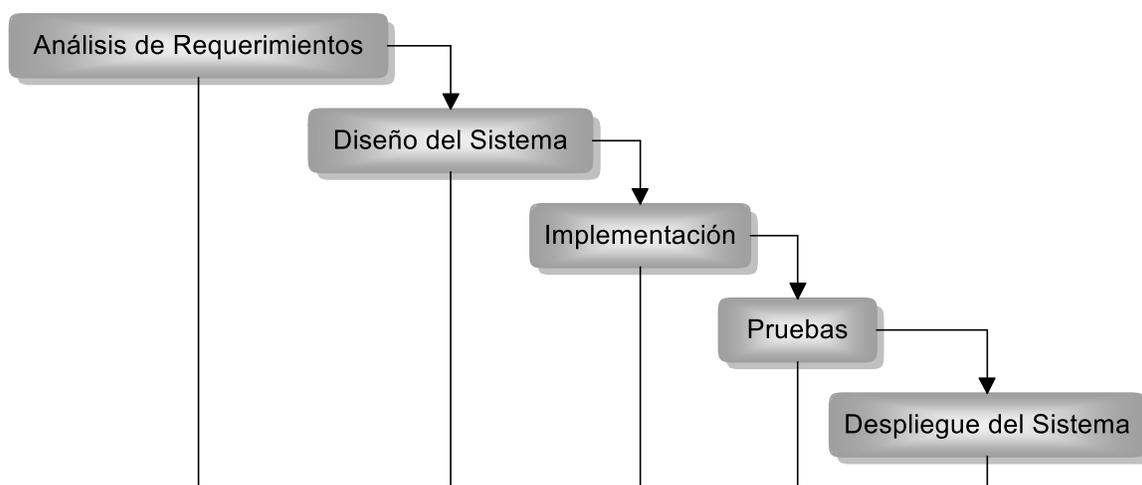


Ilustración 5. Diagrama general del modelo en cascada

Fuente: <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>

2.7.3.2. *Ventajas del modelo en cascada.*

- Este modelo es simple y fácil de entender y usar.
- Es fácil de manejar debido a la rigidez del modelo, cada fase tiene productos específicos y un proceso de revisión.
- En este modelo las fases son procesadas y completadas una a la vez. Las fases no se superponen.
- El modelo en cascada trabaja bien para los proyectos más pequeños donde los requisitos son muy bien entendidos.

2.7.3.3. *Desventajas del modelo en cascada.*

- Una vez que una aplicación está en la etapa de prueba, es muy difícil volver atrás y cambiar algo que no estaba bien pensado en la etapa de desarrollo.
- Grandes cantidades de riesgo e incertidumbre.
- No es un buen modelo para proyectos complejos y orientados a objetos.
- Modelo deficiente para proyectos largos y en curso.
- No es adecuado para los proyectos en los que los requisitos están en un riesgo de moderado a alto de cambiar.

2.7.3.4. *Cuando utilizar el modelo en cascada.*

- Este modelo se utiliza sólo cuando los requisitos son muy conocidos, claros y fijos.
- La definición del producto es estable.
- Se entiende la tecnología.

- No hay requisitos ambiguos y el proyecto es corto.

2.7.4. Modelo en V.

Es un medio de verificación y validación de un proceso. El ciclo de vida en forma de V es un camino secuencial de ejecución de los procesos, donde cada fase debe ser completada antes de que comience la siguiente fase. EL modelo en V es uno de los muchos modelos de desarrollo de software. La codificación forma el vértice de la V, con el análisis y el diseño a la izquierda y las pruebas a la derecha, la unión mediante líneas discontinuas entre las fases de la parte izquierda y la derecha representa que por un lado sirve para indicar en qué fase de desarrollo se deben definir las pruebas correspondientes y para saber a qué fase de desarrollo hay que volver si se encuentran fallos en las pruebas correspondientes. (ISTQB Exam Certification, s.f.)

2.7.4.1. Diagrama del modelo en V.

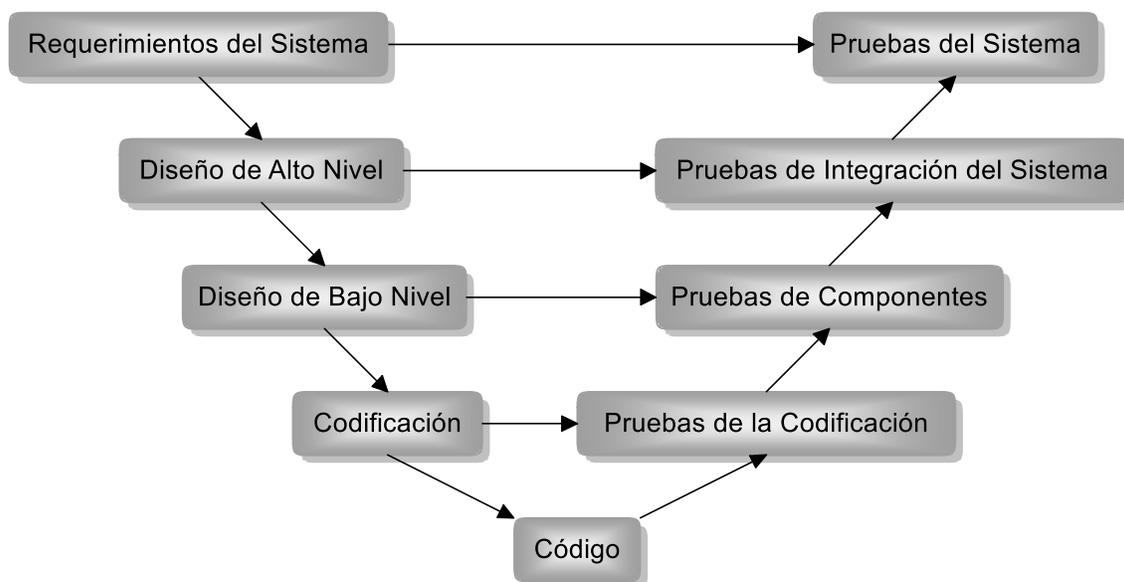


Ilustración 6. Diagrama del modelo en V

Fuente: <http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>

2.7.4.2. *Fases del modelo en V.*

Los requerimientos SRS comienzan el ciclo de vida del modelo al igual que el modelo de cascada. Pero, en este modelo antes de que se inicie el desarrollo, se crea un plan de prueba del sistema. El plan de prueba se centra en cumplir con la funcionalidad especificada en la recopilación de requisitos.

La fase de diseño de alto nivel (HLD) se centra en la arquitectura y el diseño del sistema. Proporciona una visión general de la solución, plataforma, sistema, producto y servicio / proceso. Un plan de prueba de integración se crea en esta fase, así como para probar las piezas de la capacidad de los sistemas de software para trabajar juntos.

La fase de diseño de bajo nivel (LLD) es donde se diseñan los componentes de software reales. Define la lógica real para cada componente del sistema. Diagrama de clase con todos los métodos y la relación entre las clases viene bajo LLD. También se crean pruebas de componentes en esta fase.

La fase de implementación es, de nuevo, donde tiene lugar toda la codificación. Una vez completada la codificación, la ruta de ejecución continúa por el lado derecho de la V, donde ahora se ponen a prueba los planes de prueba desarrollados anteriormente.

Codificación: Se encuentra en la parte inferior del modelo en V. El diseño del módulo es convertido en código por los desarrolladores. Las pruebas unitarias son realizadas por los desarrolladores en el código escrito por ellos.

2.7.4.3. *Ventajas del modelo en V.*

- Simple y fácil de usar.
- Las actividades de prueba como la planificación, el diseño de pruebas ocurren mucho antes de la codificación. Esto ahorra mucho tiempo. Por lo tanto una mayor probabilidad de éxito sobre el modelo de cascada.
- El seguimiento proactivo de defectos, es decir, los defectos se encuentran en las primeras etapas.
- Evita el flujo descendente de los defectos.
- Funciona bien para pequeños proyectos donde los requisitos son fáciles de entender.

2.7.4.4. *Desventajas del modelo en V.*

- Muy rígido y menos flexible.
- El software se desarrolla durante la fase de implementación, por lo que no se producen prototipos tempranos del software.
- Si se producen cambios a mitad de camino, los documentos de prueba junto con los documentos de requisitos deben actualizarse.

2.7.4.5. *Cuando utilizar el modelo en V.*

- El modelo en V se debe utilizar para pequeñas y medianos proyectos donde los requisitos están claramente definidos y establecidos.
- El modelo V se debe ser elegido cuando se dispone de amplios recursos técnicos con los conocimientos técnicos necesarios.

Capítulo 3. Diseño del Sistema.

Para el desarrollo de este capítulo se utilizó el modelo en cascada como metodología, debido a que es un modelo lineal-secuencial utilizado en proyectos cortos, el mismo que permitió realizar el diseño del sistema de una manera adecuada y organizada. Para iniciar con el desarrollo de este capítulo se llevó a cabo el análisis de la situación actual del parqueadero de la UTN, análisis de requisitos o requerimientos, elección de los componentes que integran el sistema y el diseño del sistema.

3.1. Análisis de la situación actual.

El análisis de la situación actual es una base fundamental de todo proyecto ya que permite conocer el estado del problema y las condiciones en que se encuentra actualmente, lo cual ayudó a establecer los requerimientos necesarios a ser considerados en el diseño de este proyecto de titulación. Para obtener información de la situación actual de los parqueaderos de la UTN se utilizó técnicas de investigación que faciliten la recopilación de información y el análisis de los resultados.

3.1.1. Técnicas de recolección de información.

Las técnicas son los procedimientos e instrumentos que se utiliza para la adquisición de información de un tema en particular. Encuestas, entrevistas, observaciones y todo lo que se deriva de ellas. En este caso se aplicó encuestas a las personas que hacen uso de los parqueaderos y una entrevista al jefe de seguridad de la UTN.

3.1.1.1. Encuesta.

Las encuestas son instrumentos de investigación descriptiva que permiten obtener información sistemáticamente del personal directamente implicado a través de un cuestionario, ya sea utilizando medios físicos o digitales, con el propósito de aclarar un asunto de interés para el encuestador.

La encuesta fue aplicada a una muestra de usuarios del parqueadero, tanto administrativos, docentes y estudiantes de las distintas facultades de la Universidad Técnica del Norte, lo que se busca conocer principalmente es si utilizan el parqueadero de automóviles o motocicletas, el horario que existe mayor afluencia en estos parqueaderos, el tiempo promedio que tardan en estacionarse desde que ingresan, si consideran importante disminuir el tiempo que les toma en estacionarse y si creen necesario que la UTN cuente con un sistema que permita conocer la disponibilidad de estacionamientos.

Las encuestas fueron realizadas de manera digital mediante el uso de las tecnologías actuales como son las redes sociales, ya que estos medios permiten encuestar de manera rápida, una recepción y manipulación de los resultados inmediatos, además que ayuda a reducir los costos del proceso de encuestado. Las preguntas planteadas en la encuesta son de tipo cerrado, teniendo preguntas de respuesta única y de respuesta múltiple, que permiten obtener respuestas rápidas y precisas, las cuales permiten facilitar el proceso de cuantificación, tabulación y análisis de resultados. El formato de la encuesta aplicada se muestra en el *ANEXO 1* de este proyecto de titulación.

3.1.1.2. Entrevista.

Las entrevistas son una forma mucho más personal de la investigación que las encuestas. En la entrevista personal, el entrevistador trabaja directamente con el entrevistado. A diferencia de las encuestas por internet, el entrevistador tiene la oportunidad de sondear o hacer preguntas de seguimiento. Y, las entrevistas son generalmente más fáciles para el entrevistado, sobre todo si lo que se busca es opiniones o impresiones.

La entrevista fue aplicada al jefe del departamento de seguridad de la Universidad Técnica de Norte, se desea conocer los horarios que se llenan los parqueaderos y sus principales problemas que se presentan cuando esto sucede, los horarios de apertura y cierre de los accesos vehiculares, la cantidad de estacionamientos reservados y la cantidad de dispositivos electrónicos que se encuentran en funcionamiento para el ingreso vehicular.

Las preguntas para esta entrevista fueron de tipo abiertas, ya que al existir una interrelación o diálogo con el entrevistado, este expresa personalmente sus respuestas y opiniones, lo cual permitió obtener información más detallada. El cuestionario utilizado en la entrevista realizada al jefe del departamento de seguridad de la Universidad Técnica del Norte se encuentra en el *ANEXO 2*.

3.1.2. Identificación de la población.

La población o también llamado universo, es el conjunto de todos los miembros que tienen una característica común, esta población puede ser finita o infinita. Se considera población infinita cuando no se conoce el número de elementos que la integran y población finita cuando está

delimitada y se conoce el número sujetos que la integran, en este caso se tiene una población finita ya que el número de usuarios que poseen el dispositivo electrónico para ingresar con su vehículo y hacer uso de los parqueaderos de la UTN son 700, los cuales se encuentran distribuidos de la siguiente manera.

Tabla 10. Distribución de los dispositivos electrónicos.

Tipo de Usuario	Numero de Dispositivos Electrónicos	Porcentaje
Administrativo	200	28.57%
Docente	330	47.14%
Estudiante	170	24.28%
TOTAL	700	100%

Fuente: Jefe del departamento de seguridad de la UTN

3.1.2.1. *Calculo del tamaño muestral.*

Un muestreo apropiado permite que se obtenga una parte de la población con una medida confiable de todo el conjunto. Como se está trabajando con una población finita (Fisher y Navarro, 1997), se aplica la fórmula:

$$n = \frac{NK^2p(1-p)}{(e^2(N-1)) + K^2p(1-p)}$$

Donde:

n = Tamaño de la muestra

N = Tamaño de la población

K = Nivel de confianza. Es un valor constante que, cuando no se tiene su valor, se lo toma en relación al 95% de confianza equivale a 1,96 (valor más usual) o en relación al 99% de confianza equivale 2,58, este dato queda a criterio del investigador.

p = Probabilidad de éxito

q = (1 - p) = Probabilidad de fracaso

e = Error muestral.

Para este caso de estudio se toman los siguientes datos:

N = 700

K = 1,96 con un nivel de confianza del 95%

p = Probabilidad de éxito del 50% (0.5)

q = Probabilidad de fracaso del 50% (0.5)

e = Error muestral del 10% (0,1)

$$n = \frac{(700)(1,96)^2(0,5)(0,5)}{(0,1^2(700 - 1)) + 1,96^2(0,5)(0,5)} = 84$$

De acuerdo con el cálculo se obtuvo que el tamaño muestral es de 84, es decir que la encuesta se aplicó a 84 usuarios de los parqueaderos de la UTN, para lo cual se subdividió el número total de encuestas entre los diferentes tipos de usuarios de acuerdo al porcentaje que tiene cada grupo.

Tabla 11. Número de usuarios a los que se les aplico la encuesta.

Tipo de Usuario	Porcentaje	Usuarios a los que se les aplico la encuesta
Administrativo	28.57%	24
Docente	47.14%	40
Estudiante	24.28%	20
TOTAL	100%	84

Fuente: Autoría

3.1.3. Universidad Técnica del Norte.

La Universidad Técnica del Norte (UTN) es una entidad de Educación Superior que realiza una labor académica e investigativa con el fin de formar profesionales críticos, capaces de contribuir al desarrollo nacional y regional. Se encuentra ubicada en Ecuador, provincia de Imbabura, su matriz está situada en la ciudad de Ibarra, barrio El Olivo, Av. 17 de Julio 5 -21 y Gral. José María Córdova.



Ilustración 7. Vista superior de la UTN.

Fuente: Google Earth.

3.1.5. Parquederos de la UTN.

La UTN actualmente cuenta con nueve parquederos para automóviles y dos parquederos para motocicletas, los cuales por el momento no tienen definido un nombre por parte de la Unidad de seguridad y Salud ocupacional y ambiente (USSOA) de la UTN, por tal motivo para este caso de estudio se los identifico de la siguiente manera: Parquedero seguido de la letra A en caso de automóviles y la letra M para motos y el número del parquedero (Parquedero A1, Parquedero M1), esto permitió referirse a un parquedero en específico, adicionalmente para identificar la ubicación de cada parquedero más fácilmente, se los asoció con el nombre de la edificación que se encuentra más cercana. A continuación en la *Ilustración 9* se puede apreciar la ubicación de cada uno de los parquederos.

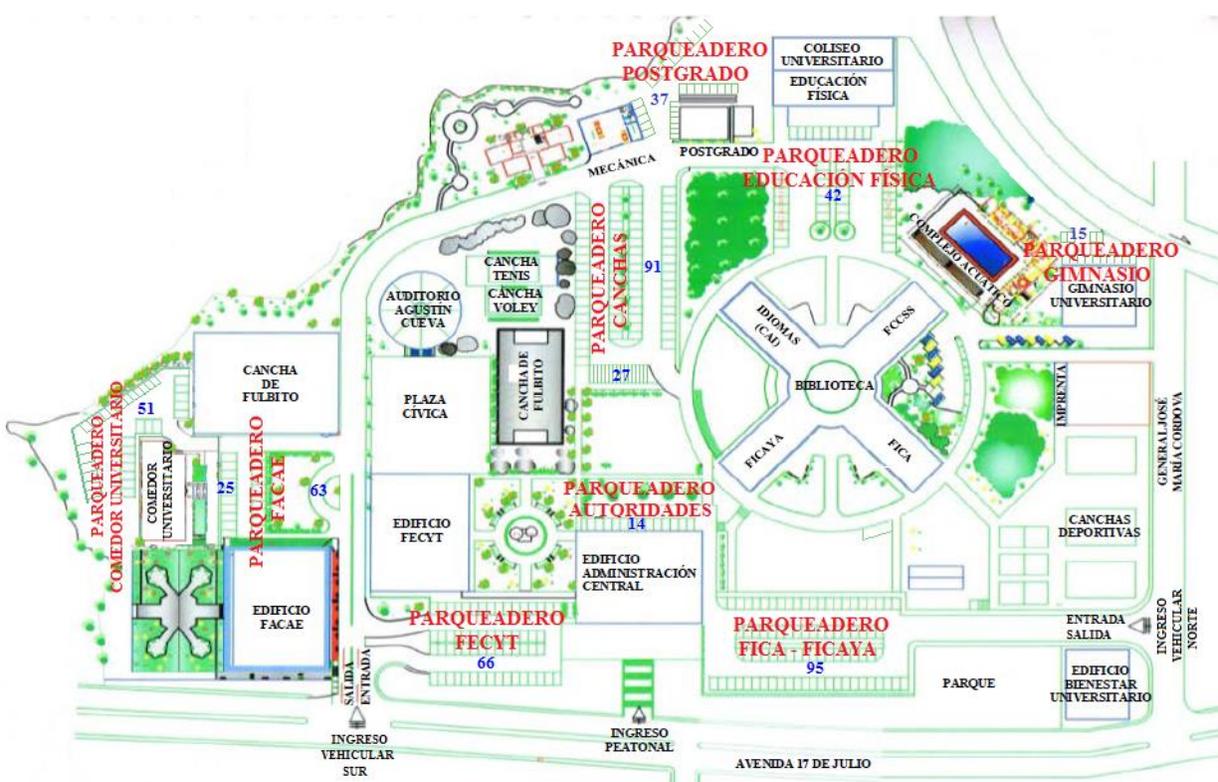


Ilustración 9. Ubicación de los parquederos en el campus de la UTN.

Fuente: Adaptado de UTN

A continuación en las siguientes tablas se muestra la cantidad de estacionamientos de cada parqueadero y la capacidad total de plazas de aparcamiento que tiene el campus de la Universidad Técnica del Norte.

Tabla 12. Capacidad de los parqueaderos de automóviles de la UTN.

Parqueaderos de Automóviles	Ubicación	Nombre dado al Parqueadero	Cantidad de Estacionamientos
Parqueadero A1	FECYT	Parqueadero de la FECYT	66
Parqueadero A2	FACAE	Parqueadero de la FACAE	25
Parqueadero A3	Comedor Universitario	Parqueadero del Comedor Universitario	51
Parqueadero A4	Postgrado	Parqueadero de Postgrado	37
Parqueadero A5	Gimnasio	Parqueadero del Gimnasio	15
Parqueadero A6	Educación Física	Parqueadero de Educación Física	42
Parqueadero A7	Canchas	Parqueadero de las Canchas	91
Parqueadero A8	Edificio Central	Parqueadero de Autoridades	14
Parqueadero A9	FICA - FICAYA	Parqueadero de la FICA y FICAYA	95
TOTAL			436

Fuente: Autoría

Tabla 13. Capacidad de los parqueaderos de motocicletas de la UTN.

Parqueaderos de Motos	Ubicación	Nombre del Parqueadero	Cantidad de Estacionamientos
Parqueadero M1	FACAE	Parqueadero de la FACAE	63
Parqueadero M2	Canchas	Parqueadero de las Canchas	27
TOTAL			90

Fuente: Autoría

3.1.5.1. *Tamaño de los parqueaderos.*

En este apartado se realiza la medición de los parqueaderos, con la ayuda de la herramienta google earth, con el objetivo de conocer el tamaño de cada parqueadero, ya que estos espacios son el área de trabajo para la cual se realizó el diseño del sistema.

Tabla 14. Dimensiones de los parqueaderos.

Parqueadero	Nombre del Parqueadero	Lado 1	Lado 2	Área
Parqueadero A1	Parqueadero de la FECYT	47 m	32 m	1504 m ²
Parqueadero A2	Parqueadero de la FACAE	38 m	17 m	646 m ²
Parqueadero A3	Parqueadero del Comedor Universitario	65 m	38 m	2470 m ²
Parqueadero A4	Parqueadero de Postgrado	50 m	55 m	2250 m ²
Parqueadero A5	Parqueadero del Gimnasio	27 m	13 m	351 m ²
Parqueadero A6	Parqueadero de Educación Física	42 m	40 m	1680 m ²
Parqueadero A7	Parqueadero de las Canchas	75 m	32 m	2400 m ²
Parqueadero A8	Parqueadero de Autoridades	40 m	10 m	400 m ²
Parqueadero A9	Parqueadero de la FICA y FICAYA	77 m	37 m	2849 m ²

Fuente: Autoría

Al conocer el tamaño de los parqueaderos se obtuvo que el parqueadero de mayor extensión es el A9, por lo cual en base a esas dimensiones se puede decir que el alcance de los transceptores inalámbricos de la WSN deben ser de al menos 100 metros.

3.1.5.2. Estacionamientos reservados.

Existen varias plazas de estacionamiento reservadas, las mismas que son de uso exclusivo para autoridades, personas en situación de discapacidad y vehículos de la UTN.

- **Reservado para autoridades y visitas.**

En el parqueadero A8 se encuentran designados 10 estacionamientos para el uso de los decanos y sub-decanos de las diferentes facultades (FICA, FICAYA, FCCSS, FECYT, FACA), un estacionamiento para el rector de la UTN y los 3 estacionamientos restantes para visitas. Además se dispone de 2 estacionamientos para los vicerrectores en el parqueadero A9.



Ilustración 10. Parqueadero A8 reservado para autoridades y visitas.

Fuente: Autoría.



Ilustración 11. Estacionamientos reservados para vicerrectores en el parqueadero A9.
Fuente: Autoría.

- **Reservado para personas con discapacidad**

Existe un estacionamiento designado para el uso de personas con discapacidad y se encuentran ubicado en el Parqueadero A9.



Ilustración 12. Estacionamiento para personas con discapacidad.
Fuente: Autoría.

- **Reservado para vehículos de la UTN**

Los espacios reservados para los vehículos oficiales de la institución son 32 y se encuentran en el parqueadero A9 debido a que es un lugar cercano al edificio de la administración central y visible donde se los puede brindar mayor seguridad.



Ilustración 13. Estacionamientos para vehículos de la UTN.

Fuente: Autoría.

- **Lugar de aparcamiento de los buses de la UTN**

Actualmente la UTN no cuenta con un parqueadero adecuado para el aparcamiento de los autobuses y de la maquinaria pesada, por tal motivo se hace uso de un espacio verde para estacionar estos vehículos, el mismo que no se encuentra considerado como parqueadero por parte de la USSOA. Este lugar está ubicado entre el edificio del CAI y la FICAYA.



Ilustración 14. Lugar de aparcamientos de los buses de la UTN.

Fuente: Autoría.

Para poder distinguir con mayor facilidad la ubicación de los estacionamientos reservados, en la *Ilustración 15* se los marco de color morado los designados para autoridades, de color amarillo los reservados para vehículos de la institución, de color azul para personas con discapacidad y de color rosado el área designada para estacionar los buses.

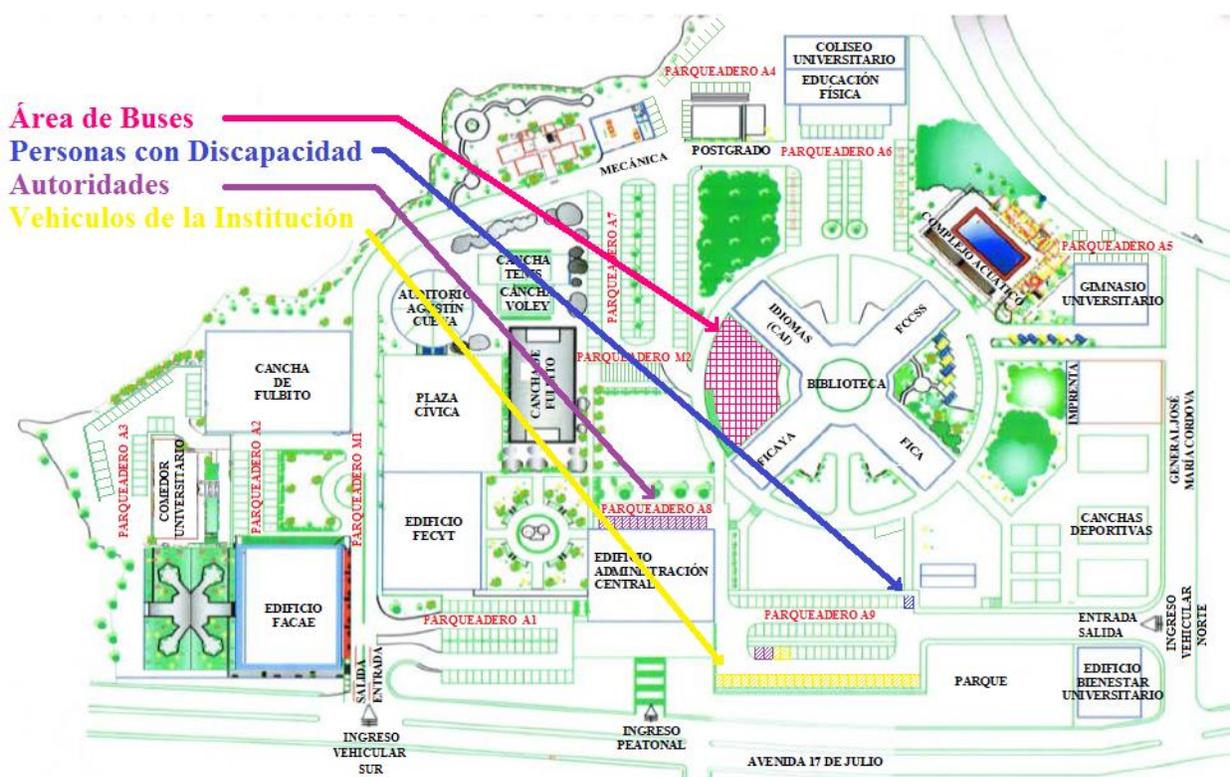


Ilustración 15. Ubicación de los estacionamientos reservados.

Fuente: Adaptado de UTN.

3.1.5.3. Estacionamientos de uso libre.

Teniendo en cuenta el total de estacionamientos que tienen los parqueaderos y la cantidad de estacionamientos que se encuentran reservados, se obtuvo el número de lugares de aparcamiento de los cuales pueden hacer uso los docentes, estudiantes y trabajadores de la UTN.

Tabla 15. Cantidad de estacionamientos disponibles para automóviles.

Parqueaderos	Nombre del Parqueadero	Estacionamientos	Reservados	De uso libre
Parqueadero A1	Parqueadero de la FECYT	66	0	66
Parqueadero A2	Parqueadero de la FACAE	25	0	25
Parqueadero A3	Parqueadero del Comedor Universitario	51	0	51
Parqueadero A4	Parqueadero de Postgrado	37	0	37
Parqueadero A5	Parqueadero del Gimnasio	15	0	15
Parqueadero A6	Parqueadero de Educación Física	42	0	42
Parqueadero A7	Parqueadero de las Canchas	91	0	91
Parqueadero A8	Parqueadero de Autoridades	14	14	0
Parqueadero A9	Parqueadero de la FICA y FICAYA	95	35	60
TOTAL		436	49	387

Fuente: Autoría

En el caso de los parqueaderos M1 y M2 el total presentado en la *Tabla 13* es la cantidad de estacionamientos que se puede hacer uso, ya que en este caso no hay estacionamientos reservados.

3.1.5.4. Control de los parqueaderos de la UTN.

La Universidad Técnica del Norte mediante su Unidad de Seguridad, Salud Ocupacional y Ambiente (USSOA) ha establecido un instructivo para el correcto uso de parqueaderos y circulación interna por parte de toda clase de vehículos y peatones, el cual se aplica para todo el personal del campus universitario, los mismos que deben respetar y acatar las disposiciones de estacionamiento y circulación interior de la Universidad y entre las cuales se tiene:

- Estacionar los vehículos en reversa
- Respetar la señalización existente
- Velocidad máxima de circulación 20km/h
- No usar en exceso el claxon o bocina del vehículo
- Usar la radio del vehículo a volumen bajo
- Respetar y dar preferencia al peatón y personas con discapacidad
- Personas con discapacidad harán uso de parqueos exclusivos
- No estacionar en redondeles, frente a tomas de incendio, pasos peatonales, espacios destinados para personas con discapacidad, parqueos reservados para autoridades y en zonas que dificulte la libre circulación de vehículos y peatones.

Para facilitar el cumplimiento de cada una de las disposiciones antes mencionadas, existe la señalética necesaria en los parqueaderos y las vías de acceso, tal como se puede apreciar en las siguientes ilustraciones.



(a)

(b)



(c)

Ilustración 16. (a) Señal de estacionarse en reversa. (b) Límite de velocidad de circulación interna. (c) Señalización de los lugares prohibidos para estacionarse.

Fuente: Autoría.

3.1.6. Puertas de acceso vehicular.

El campus de la UTN cuenta con dos accesos vehiculares, la puerta norte se encuentra en la calle General José María Córdova (*Ilustración 17 (a)*) y la puerta sur en la Av. 17 de Julio (*Ilustración 17 (b)*).



(a) (b)
 Ilustración 17. Accesos vehiculares. (a) Puerta norte. (Puerta sur).
 Fuente: Autoría.

El ingreso sur suele abrirse para la comunidad universitaria a partir de las 6H30 hasta las 21h45 en horario de lunes a sábado, los domingos y días feriados las puertas permanecen cerradas, y el horario de apertura del ingreso norte en la mañana es partir de 6h45 hasta las 8h15 y en la tarde existen dos horarios, desde 12h40 hasta las 14h15 y de 16h45 hasta las 17h15 de lunes a viernes y los días sábados, domingos permaneces cerradas. El motivo por el cual la puerta norte no permanezca abierta continuamente se debe a la falta de personal del departamento de seguridad ya que consideran que es necesario que siempre permanezca un guardia en la puerta mientras esta se encuentra abierta.

Los estudiantes, docentes, personal administrativo y de apoyo que deseen ingresar con sus vehículos a las instalaciones de la UTN deben portar su dispositivo electrónico para poder acceder, el mismo que debe ser adquirido previamente en el departamento de recaudación, debido a que actualmente los ingresos vehiculares cuentan con un sistema de control de accesos para evitar el ingreso de personas ajenas a la institución. El número de dispositivos electrónicos entregados a la comunidad universitaria es de 700 y los estacionamientos de uso libre son 387, es por este motivo que en horas de la mañana de 8h00 - 8h30 a 12h30 y en la tarde de 14h30 – 15h00 a 17h15 existe

un aglomeramiento vehicular en los parqueaderos. (Datos proporcionados por el jefe del departamento de seguridad de la UTN).

3.1.7. Análisis de resultados.

Luego de haber realizado la encuesta a 84 usuarios de los parqueaderos de la UTN, se obtuvo resultados que permiten conocer sus opiniones e inconvenientes, lo cual permitirá fundamentar de una manera adecuada el desarrollo de este sistema. Las preguntas permitieron corroborar que los parqueaderos más usados son los de los automóviles, y que el parqueadero se llena durante los horarios de ingreso del personal que trabaja en la institución.

La tabulación de cada una de las preguntas aplicadas en la encuesta se encuentra en el *ANEXO 3* de este documento. Para conocer sus experiencias se preguntó el tiempo que les lleva estacionarse, teniendo en cuenta que el tiempo que se demoren es un factor muy importante que el sistema debe ayudar a reducir, ya que se obtuvo que para el 91% de los encuestados es necesario reducir el tiempo que les toma encontrar un espacio libre y estacionarse.

De esta encuesta se puede concluir que las personas encuestadas coinciden en dar sus respuestas teniendo en cuenta que cada persona hace uso del parqueadero en distintos horarios, por lo tanto el desarrollo de este sistema ayudará a reducir el tiempo que le toma estacionar su vehículo desde que ingresa al campus universitario y por ende mejorar la experiencia al utilizar los parqueaderos.

3.2. Introducción al desarrollo del sistema.

En esta sección se describe aspectos relevantes del sistema, los cuales permitieron tener claro cuál es el propósito, ámbito y funciones del sistema, lo cual ayudó a realizar el diseño de este sistema de una manera adecuada.

- **Propósito del sistema.**

El propósito de este sistema es proporcionar a los usuarios información acerca de la cantidad de estacionamientos libres que existen en cada uno de los diferentes parqueaderos de la UTN, de esta manera los usuarios al conocer que parqueadero cuenta con estacionamientos disponibles se dirigirán directamente a ese parqueadero disminuyendo así el tiempo que les toma estacionar su vehículo.

- **Ámbito del sistema.**

El sistema de monitoreo de la disponibilidad de estacionamientos en los parqueaderos del campus El Olivo de la UTN, es un sistema que al tratarse únicamente del diseño, las pruebas de funcionamiento se lo realizara a través de un prototipo, estas pruebas servirán para la comprobación de su utilidad y posibles implementaciones futuras. Para comprender de una mejor manera que es un prototipo, se lo define a continuación.

Un prototipo es una simulación del producto final. Es como una maqueta interactiva cuyo objetivo principal es probar si el funcionamiento es el correcto o si hace falta corregirlo. Los prototipos dan vida a cualquier diseño y proporcionan una gran cantidad de información ya que

permiten poner a prueba la viabilidad y la utilidad de nuestros diseños y también ayudan a descubrir mejoras e innovaciones inesperadas que pueden hacer nuestro proyecto aún mejor.

3.2.1. Descripción general del sistema.

El sistema busca emplear una red inalámbrica de sensores destinada a la detección del estado de los estacionamientos (libre u ocupado), con lo cual se puede determinar el número de estacionamientos libres en los parqueaderos, todos los datos recolectados serán enviados a una plataforma de almacenamiento, los mismos que podrán ser observados en la interfaz de visualización de dicha plataforma.

3.2.1.1. *Funciones del proyecto.*

Las principales funciones de este sistema son:

- Detectar la presencia del vehículo en el estacionamiento y cambiar de estado en el que se encuentre el estacionamiento apenas se detecte la presencia o ausencia del mismo con el objetivo de contabilizar la cantidad de estacionamientos libres que tiene cada parqueadero.
- Los datos recolectados serán enviados a una plataforma de almacenamiento en la nube para poder llevar un registro.
- La información podrá ser visualizada a través de la interfaz de la plataforma de almacenamiento, inicialmente estos datos solo podrán ser visualizados por el administrador del sistema.

3.2.1.2. Restricciones.

- Las pruebas del sistema se realizarán únicamente en 5 estacionamientos de un parqueadero.
- Deben existir puntos de conexión de energía eléctrica para energizar a todos los componentes del sistema.
- El sistema funcionará con la red eléctrica de la UTN, por tal motivo si esta se pierde el sistema quedará fuera de servicio, ya que el sistema también hará uso de la red de datos de la UTN y al no contar con energía eléctrica esta red igualmente queda fuera de servicio.
- Los nodos soportarán variaciones de clima leves, es decir no estarán bajo ninguna norma que certifique su grado de protección IP.
- El sistema podrá ser manipulado únicamente por el administrador del sistema.
- El sistema será aplicable para los parqueaderos de automóviles y no de motocicletas.

3.3. Requerimientos del sistema.

En este proyecto los requerimientos del sistema se los estableció basándose en el estándar ISO / IEC / IEEE 29148: 2011, debido a que es un estándar que brinda directrices para la aplicación de los requisitos y procesos relacionados con los requisitos que se van a implementar. La ingeniería de requisitos es una función de gran importancia que sirve para establecer y mantener los requisitos que debe cumplir el sistema, software o servicio. Así como también se ocupa de obtener, analizar, desarrollar, determinar métodos de verificación, validación, comunicación, documentación y gestión de los requisitos.

El análisis de requerimientos del sistema se lo realizó teniendo en cuenta las consideraciones del estándar antes mencionado, por tal motivo los requerimientos que se analizan a continuación son de stakeholders, funcionales y de arquitectura, para lo cual se utilizó algunos términos abreviados y acrónimos que se describen en la *Tabla 16*.

Tabla 16. Términos abreviados y acrónimos.

Términos abreviados y acrónimos usados durante el análisis de requerimientos.	
Acrónimos y abreviados	Descripción
StSR	Requerimientos de Stakeholders del sistema.
SySR	Requerimientos Funcionales del sistema.
SRSR	Requerimientos de Arquitectura del sistema.

Fuente: Autoría

3.3.1. Requerimientos de stakeholders.

Stakeholders se refiere al conjunto de las partes interesadas del proyecto, las cuales se las describe en la *Tabla 17*. Además los requerimientos de stakeholders pueden incluir otros dos grupos de requisitos como son los operacionales y de usuarios, los mismos que se detallan en la *Tabla 18*.

Tabla 17. Lista de Stakeholders del proyecto.

Stakeholders
1. Universidad Técnica del Norte
2. Unidad de seguridad, salud ocupacional y ambiente (USSOA)
3. Comunidad universitaria
4. Ing. Jaime Michilena

5. Ing. Daniel Jaramillo

6. Ing. Luis Suarez

7. Sr. Wilmer Bedoya

Fuente: Autoría

Tabla 18. Requerimientos de Stakeholders del sistema.

<i>StSR</i>						
REQUERIMIENTOS DE STAKEHOLDERS						
#	Requerimiento	Prioridad			Relación	Verificación
		Alta	Media	Baja		
StSR 1	El vehículo debe estacionarse correctamente en los lugares marcados.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
StSR 2	El sensor debe ser ubicado en un lugar donde puede detectar la presencia del vehículo con facilidad.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS OPERACIONALES						
StSR 3	Es necesario un nodo sensor para cada estacionamiento.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
StSR 4	Existencia de puntos de energía eléctrica para cada dispositivo del sistema.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
StSR 5	Existencia de conexión a internet en cada nodo central para enviar los datos a una plataforma de almacenamiento en la nube.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
StSR 6	Disponer de dispositivos para la visualización de datos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

REQUERIMIENTOS DE USUARIO

StSR 7	Los datos mostrados deben ser actualizados.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
StSR 8	Los datos deben ser mostrados en un formato adecuado para el usuario.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
StSR 9	Los datos deben ser mostrados en un lugar visible para el usuario.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fuente: Autoría

3.3.2. Requerimientos funcionales.

Estos requerimientos hacen referencia a todas las funcionalidades que debe tener el sistema, los cuales comprenden los requisitos de uso, de las interfaces, los modos, estados y físicos, los cuales se encuentran presentados en la *Tabla 19*, teniendo en cuenta que de este modo se puede identificar claramente lo que se necesita para el sistema.

Tabla 19. Requerimientos funcionales del sistema.

SySR					
REQUERIMIENTOS DE FUNCIONALES					
#	Requerimiento	Prioridad			Relación Verificación
		Alta	Media	Baja	
SySR 1	El sistema deberá realizar el monitoreo de la disponibilidad de estacionamientos en un ambiente externo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

REQUERIMIENTOS DE USO

SySR 2	El sistema debe estar energizado.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SySR 3	El sistema debe estar encendido y debe tener conexión a internet.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
REQUERIMIENTO DE INTERFACES				
SySR 4	Es necesario ingresar a la plataforma de almacenamiento en la web para visualizar los datos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SySR 5	La presentación de los datos debe ser de manera clara.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
REQUERIMIENTOS DE MODOS/ESTADO				
SySR 6	Todo el sistema debe estar en funcionamiento durante los horarios que se encuentre habilitado el parqueadero.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
REQUERIMIENTOS FÍSICOS				
SySR 7	Los nodos deben estar ubicados fijamente en un lugar que no obstaculicen los estacionamientos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SySR 8	El nodo central debe estar ubicado en un lugar donde tenga línea de vista con los nodos sensores.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SySR 9	El dispositivo de visualización, debe ser colocado en un lugar visible por el usuario y que no interfiera en el tránsito vehicular.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fuente: Autoría

3.3.3. Requerimientos de arquitectura.

Para describir estos requerimientos se presenta la *Tabla 20*, en la que se presenta los requisitos lógicos, de diseño, de software, de hardware y eléctricos, lo cual será de gran ayuda para la elección de los componentes de software y hardware que serán parte del sistema.

Tabla 20. Requerimientos de arquitectura del sistema.

<i>SRSH</i>					
REQUERIMIENTOS DE ARQUITECTURA.					
#	Requerimiento	Prioridad			Relación Verificación
		Alta	Media	Baja	
REQUERIMIENTOS LÓGICOS					
SRSH 1	Las placas o módulos deben poseer entradas y salidas digitales.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 2	La placa de procesamiento debe ser compatible con el módulo inalámbrico.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
REQUERIMIENTOS DE DISEÑO					
SRSH 3	Implementación de un sistema de bajo costo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 4	Implementar Hardware y Software que sea de distribución libre.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
REQUERIMIENTOS SOFTWARE					
SRSH 5	El software de programación debe ser compatible con la placa de desarrollo a elegir.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

SRSH 6	Contar con disponibilidad de librerías entre los módulos y la placa de procesamiento.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 7	El software de visualización debe ser compatible con los dispositivos donde se visualizara.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 8	La plataforma de almacenamiento y visualización debe ser capaz de actualizarse constantemente.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
REQUERIMIENTOS DE HARDWARE				
SRSH 9	Sensores capaces de detectar la presencia de un vehículo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 10	Sensores utilizados en ambientes externos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 11	Capacidad de procesar rápidamente los datos recolectados.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 12	Disponibilidad de pines para la conexión de varios periféricos y compatibilidad para integrar un Shield.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 13	La placa de procesamiento debe contar con un puerto Ethernet para la conexión a internet o compatibilidad para integrar un shield.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 14	Retardo en la transmisión de datos bajo (baja latencia).	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 15	El alcance para la comunicación inalámbrica	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	debe ser de hasta 100 metros.			
SRSH 16	Debe soportar hasta 100 nodos por master.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 17	Debe existir disponibilidad de los dispositivos en el mercado local a un precio asequible.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 18	El tamaño de los elementos que conforman los nodos debe ser de tamaño reducido.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 19	El dispositivo de visualización debe tener la capacidad de conectarse a internet.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 20	El dispositivo de visualización debe soportar interfaz web.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>REQUERIMIENTOS ELÉCTRICOS</i>				
SRSH 21	Los dispositivos deben tener un bajo consumo de energía.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fuente: Autoría

3.4. Elección de hardware y software del sistema.

Habiendo ejecutado el análisis de requerimientos se procede a realizar la selección del hardware y software que formaran parte del sistema, lo cual se lo llevara a cabo mediante una valoración de cada una de las opciones de acuerdo a las tablas de los requerimientos, es decir la opción que cumpla con todos o con la mayoría de los requerimientos será la que se elija. Las opciones presentadas en esta valoración han sido escogidas ya que son las que más se adaptan al sistema.

3.4.1. Elección de Hardware.

Una vez realizada la elección de cada uno de los elementos que formaran parte del sistema se realizara una breve descripción de sus principales características.

3.4.1.1. Sensor.

Este dispositivo será el encargado de detectar la presencia del automóvil en el estacionamiento. A continuación en la *Tabla 21* se tiene la valoración de las diferentes opciones de sensores que pueden ser utilizados para la detección de la presencia de vehículos.

Tabla 21. Elección del sensor.

<i>Elección del sensor para la detección de vehículos.</i>					
<i>Hardware</i>	<i>Requerimientos</i>				<i>Valoración Total</i>
	SRSH 9	SRSH 10	SRSH 17	SRSH 18	
Ultrasónico HC-SR04	1	0	1	1	3
Infrarrojo Sharp 2Y0A21	1	1	1	1	4
Magnético HMC5883L	0	1	1	1	3
1 – Cumple 0 – No Cumple					
Elección: Sensor Infrarrojo Sharp 2Y0A21					

Fuente: Autoría

- **Sensor infrarrojo Sharp 2Y0A21.**

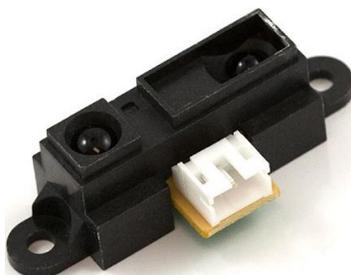


Ilustración 18. Sensor infrarrojo Sharp 2Y0A21.

Fuente: https://http2.mlstatic.com/sensor-sharp-D_NQ_NP_914311-MPE20540722842_012016-F.webp

GP2Y0A21YK0F es una unidad de sensor de medición de distancia, compuesta de una combinación integrada de PSD (detector de posición sensible), IRED (diodo emisor de infrarrojos) y circuito de procesamiento de señal. La variedad de la reflectividad del objeto, la temperatura ambiental y la duración operativa no se ven fácilmente influenciadas por la detección de distancia debido a su método de triangulación.

Este dispositivo emite el voltaje correspondiente a la distancia de detección. Entonces, este sensor también se puede usar como un sensor de proximidad. El componente Sharp 2Y0A21 se comunica con Arduino a través de un pin analógico, además no es necesario contar con ninguna librería lo cual lo hace muy sencillo de utilizar.

Este sensor consta de tres pines: alimentación, tierra y salida. La alimentación del sensor debe ser entre 4,5 y 5,5 voltios, y lo más estable posible para evitar que arroje datos erróneos, la tensión del pin de salida variará entre 0,3 y 3,1 voltios en función de la distancia medida. En la *Tabla 22* se muestran sus principales características.

Tabla 22. Especificaciones técnicas del sensor.

Especificaciones	Valores
Voltaje de Operación	4.5 – 5.5 V
Consumo de corriente	30 – 40 mA
Distancia de medición	10 – 80 cm
Tipo de salida	Análoga
Temperatura de operación	-10 a 70 °C
Tamaño	29.5x13x13.5 mm
Data rate	160 Hz

Fuente: Adaptado de (Socle Technology Corp, s.f.)

3.4.1.2. Placa de procesamiento de datos.

Teniendo en cuenta que la recopilación de datos es a través de sensores es necesario contar con un dispositivo que se encargue de procesar los mismos, para ello en la *Tabla 23* se realiza la elección de la placa de procesamiento que se utilizó.

Tabla 23. Elección de la placa de procesamiento de datos.

<i>Elección de la placa de procesamiento de datos.</i>						
<i>Hardware</i>	<i>Requerimientos</i>					<i>Valoración Total</i>
	SRSH 11	SRSH 12	SRSH 13	SRSH 17	SRSH 18	
Arduino Nano	1	0	0	1	1	3
Arduino Uno	1	1	1	1	1	5
Arduino Mega2560	1	1	1	1	0	4

Arduino Yun	1	1	1	0	1	4
Arduino Ethernet	0	1	1	1	1	4

1 – Cumple

0 – No Cumple

Elección para el nodo sensor: Arduino Uno

Elección para el nodo central: Arduino Mega y Arduino Ethernet

Fuente: Autoría

Se realizó la elección de 3 placas debido a que la placa Arduino Uno fue para el nodo sensor y las placas Arduino Mega y Arduino Ethernet fueron para el nodo central, no se optó por la placa Arduino Yun para el nodo central a pesar de tener la misma valoración, debido a que tiene un costo más elevado que el Arduino Mega y el Arduino Ethernet. A continuación, se describe cada una de las placas.

- **Arduino Uno.**



Ilustración 19. Arduino UNO.

Fuente: <https://store.arduino.cc/usa/arduino-uno-rev3>

El Arduino Uno es una placa micro controladora basada en la ATmega328P. Tiene 14 pines digitales de entrada / salida (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, una toma de alimentación, una cabecera ICSP y un botón de reinicio. Contiene todo lo necesario para soportar el micro controlador; Simplemente se conecta a un ordenador con un cable USB o con un adaptador AC-DC o batería para alimentarlo. (Arduino, 2017) Sus principales especificaciones técnicas se detallan a continuación en la *Tabla 24*.

Tabla 24. Especificaciones técnicas del Arduino UNO.

Especificaciones	Valores
Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Tensión de entrada (recomendado)	7-12V
Tensión de entrada (límite)	6-20V
Pines digitales de E / S	14 (de los cuales 6 proporcionan salida PWM)
Pines digitales PWM E / S	6
Pines de Entradas Análogas	6
Corriente CC por pin de E / S	20 mA
Corriente de CC para el Pin de 3.3V	50 mA
Memoria flash	32 KB (ATmega328P) de los cuales 0,5 KB utilizado por bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz

LED_BUILTIN	13
Longitud	68,6 mm
Anchura	53,4 mm
Peso	25 g

Fuente: Adaptado de (Arduino, 2017)

- **Arduino Mega2560.**



Ilustración 20. Arduino Mega2560.

Fuente: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

El Arduino Mega 2560 es una placa micro controladora basada en el ATmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, Y un botón de reinicio. La placa Mega 2560 es compatible con la mayoría de los shield diseñados para el Uno. (Arduino, 2017) Sus principales especificaciones técnicas se detallan a continuación en la *Tabla 25*.

Tabla 25. Especificaciones técnicas del Arduino Mega2560.

Especificaciones	Valores
Microcontrolador	ATmega2560
Tensión de funcionamiento	5V
Tensión de entrada (recomendado)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales de E / S	54 (de los cuales 15 proporcionan PWM salida)
Pines de Entradas Análogas	16
Corriente CC por pin de E / S	20 mA
Corriente de CC para el Pin de 3.3V	50 mA
Memoria flash	256 KB de los cuales 8 KB utilizados por bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101,52 mm
Anchura	53,3 mm
Peso	37 g

Fuente: Adaptado de (Arduino, 2017)

- **Arduino Ethernet Shield.**

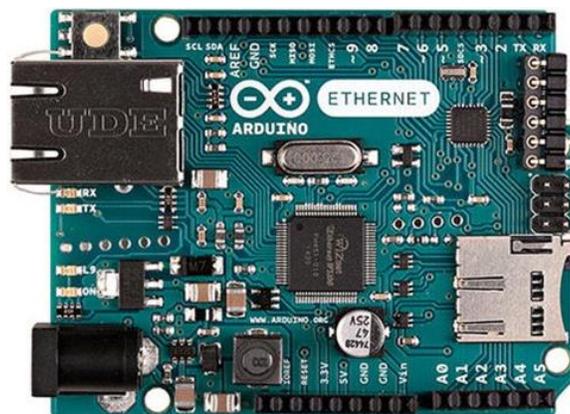


Ilustración 21. Arduino shield Ethernet.

Fuente: <https://store.arduino.cc/usa/arduino-ethernet-rev3-without-poe>

El Ethernet de Arduino es un tablero del microcontrolador basado en el ATmega328. Tiene 14 pines digitales de entrada / salida, 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión RJ45, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Hay que tener en cuenta que los pines 10, 11, 12 y 13 están reservados para la interfaz con el módulo Ethernet y no deben utilizarse de otra manera. Esto reduce el número de pines disponibles a 9, con 4 como salidas PWM. Esta placa difiere de las otras en que no tiene un chip de controlador USB a serie integrado, pero tiene una interfaz Ethernet Wiznet. Esta es la misma interfaz que se encuentra en el blindaje Ethernet. (Arduino, 2017)

Se puede acceder a un lector de tarjetas microSD integrado, que se puede utilizar para almacenar archivos para servir a través de la red, mediante la Biblioteca SD. El pin 10 está reservado para la interfaz Wiznet, SS para la tarjeta SD está en el Pin 4. La versión 3 de la placa introduce el pinout estándar 1.0, que consiste en:

- Se agregaron pines SDA y SCL que están cerca del pin de AREF y dos otros pines nuevos colocados cerca del pin RESET, esto será la oportunidad de proteger que usan componentes I2C o TWI para ser compatibles con todas las placas Arduino.
- El IOREF que permite a los shields adaptarse a la tensión proporcionada desde la placa. Los shields que utilizan el pin IOREF serán compatibles tanto con la placa que utilizan el AVR, que funcionan con 5V y con el Arduino Due que funcionan con 3.3V. Junto al pin IOREF hay un pin no conectado, que está reservado para futuros propósitos.

Tabla 26. Especificaciones técnicas del Arduino Ethernet.

Especificaciones	Valores
Microcontrolador	ATmega328
Tensión de funcionamiento	5V
Tensión de entrada enchufe (recomendado)	7-12V
Voltaje de entrada Plug (límites)	6-20V
Tensión de entrada PoE (límites)	36-57V
Pines digitales de E / S	14 (de los cuales 4 proporcionan salida PWM)
Pines reservados:	10 a 13 utilizados para SPI 4 utilizado para la tarjeta SD 2 W5100 interrupción (cuando se puentea)
Pines de Entradas Análogas	6
Corriente CC por pin de E / S	40 mA
Corriente de CC para el Pin de 3.3V	50 mA
Memoria flash	32 KB (ATmega328) de los cuales 0,5 KB utilizados por el gestor de arranque
SRAM	2 KB (ATmega328)

EEPROM	1 KB (ATmega328)
Velocidad de reloj	16 MHz
W5100 Controlador de Ethernet integrado TCP / IP	
Enchufe magnético de Power Over Ethernet	
Tarjeta Micro SD, con traductores de tensión activa	
Longitud	68,6 mm
Anchura	53,3 mm
Peso	28 gr

Fuente: Adaptado de (Arduino, 2017)

3.4.1.3. *Transceptor inalámbrico.*

Debido al lugar donde se encontraran ubicados cada nodo sensor se recomienda contar con una conexión inalámbrica para transmitir los datos de cada nodo sensor al nodo central o Gateway, es por ello que en la *Tabla 27* se elige el dispositivo que se encargara de la comunicación inalámbrica.

Tabla 27. Elección del módulo de comunicación inalámbrica.

<i>Elección del módulo de comunicación inalámbrica.</i>						
<i>Hardware</i>	<i>Requerimientos</i>					<i>Valoración Total</i>
	SRSH 14	SRSH 15	SRSH 16	SRSH 17	SRSH 21	
WiFi ESP8266	1	1	0	1	0	3
Bluetooth HC-05	0	0	0	1	0	1

XBee S2C	1	1	1	1	1	5
-----------------	---	---	---	---	---	---

1 - Cumple

0 – No Cumple

Elección: XBee S2C.

Fuente: Autoría

- **Módulo XBee S2C.**



Ilustración 22. XBee S2C.

Fuente: <http://www.alselectro.com/xbee-series-2-s2c-with-spi.html>

XBee S2C cuenta con una interfaz SPI, mejora la salida de potencia y el protocolo de datos de la serie 2. Los módulos de la serie S2C le permiten crear redes de malla complejas basadas en el firmware de la malla XBee ZB ZigBee. Este módulo permite una comunicación muy fiable y simple entre microcontroladores, computadoras, sistemas, realmente cualquier cosa con puerto serie. Se admiten redes Punto a Punto y multipunto. El XBee ZB (S2C) introduce la interfaz serie SPI en la radio ZigBee de Digi. (Digi International, 2017) En la *Tabla 28* se muestran sus principales especificaciones técnicas, y para más detalle revisar *ANEXO 7*.

Tabla 28. Especificaciones técnicas XBee ZB S2C.

Especificaciones	Valores
Interiores y urbanos	Hasta 200 pies (60 m)
Rango de visión de RF al aire libre	Hasta 1200 m (4000 pies)
Salida de potencia de transmisión	6.3mW (+ 8dBm) Modo Boost 3.1mW (+ 5dBm) Modo normal
Frecuencia de datos RF	250 Kbps
Sensibilidad del receptor	-102dBm (1% PER) Modo Boost -100dBm (1% PER) Modo Normal
Tensión de alimentación	2.1 - 3.6V
Corriente de operación (Transmisión)	45 mA (+ 8dBm) Modo Boost 33 mA (+ 5dBm) Modo Normal
Corriente de operación (Recepción)	31 mA (+ 8dBm) Modo Boost 28 mA (+ 5dBm) Modo Normal
Corriente de desconexión	<1uA
Frecuencia	ISM 2,4 GHz
Dimensiones	Pasante: 0.960 "x 1.087" SMT: 0.866 "x 1.33" x 0.120 "
Temperatura de funcionamiento	-40 a 85°C
Opciones de Antena	Agujero pasante: PCB, látigo integrado, U.FL, RPSMA SMT: cojín del RF, PCB, o U.FL
Topologías de red	Punto a punto, Estrella, Malla
número de canales	16 Canales de Secuencia Directa
Opciones de filtración	PAN ID, canal y origen / destino

Fuente: Adaptado de (Digi International, 2017)

3.4.1.4. *Dispositivo para la visualización de la información.*

Una vez que se ha recolectado la información y se la ha procesado es necesario contar con un dispositivo donde se los pueda visualizar, es por ello que en la *Tabla 29* se realiza la elección del dispositivo a utilizarse.

Tabla 29. Elección del dispositivo de visualización.

<i>Elección del dispositivo de visualización.</i>				
<i>Hardware</i>	<i>Requerimientos</i>			<i>Valoración Total</i>
	SRSH 17	SRSH 19	SRSH 20	
Dispositivo móvil (Celular, Tablet)	1	1	1	3
Rótulos electrónicos	1	0	0	1
Pantallas led	0	1	1	2
1 - Cumple				
0 – No Cumple				
Elección: Pantallas led.				

Fuente: Autoría

A pesar de tener mayor valoración el dispositivo móvil, no fue elegido debido a que no es recomendable que un conductor utilice su dispositivo móvil (Celular o Tablet) mientras conduce, ya que según las estadísticas de la Agencia Nacional de Tránsito, conducir con el teléfono a la mano es una de las razones por las cuales hay más accidentes de tránsito. De enero a septiembre del 2016, según los datos de la Agencia Nacional de Tránsito, conducir desatento al tránsito por usar teléfono celular ha causado 4,880 accidentes de tránsito. Eso significa el 21,60% de los accidentes en el país. (ANT, 2017)

- **Pantalla led MOD PC-MINI.**



Ilustración 23. Pantalla led MOD PC-mini.
Fuente: <http://www.rotuloselectronicos.net/>

Por lo antes mencionado se eligió el uso de pantallas, a pesar de que estas no tienen un precio asequible y su disponibilidad en el mercado nacional es escaso, se recomienda el uso de pantallas MOD PC-mini, debido a su alta luminosidad, su gran ángulo y distancia de visualización permitirá que los usuarios del parqueadero observen de manera clara la información mostrada. En la *Tabla 30* se muestran sus principales especificaciones técnicas, y para más detalle revisar *ANEXO 8*.

Tabla 30. Especificaciones técnicas de la pantalla led.

Especificaciones	Valores
Dimensiones de pantalla con marcos	200x2135x30 cm.
Dimensiones visibles pantalla LED	180x90 cm.
Superficie de pantalla con marcos	2,7 m ²
Superficie visible pantalla LED	1'62 m ²
Anchura de marco y perfil decorativo	10 cm. (por cada lado)
Pixel Pitch Visual/Real	7 mm/14 mm

Densidad de pixel	5.102 px/m ² .
Composición de pixel	2R1V1A
Definición gráfica visual	256x128 px
Brillo	7.000 NITS
Ángulo de visualización	160° H - 90° V
Distancia visualización	10 a 75 m.
Voltaje de entrada	AC 110V/220V/60Hz
Consumo eléctrico Max:	1.800 W - Medio: 700 W
Vida media de led mínimo	10 años
Grado de protección	IP65
Temperatura de funcionamiento	-40°C~60°C
Condiciones de humedad	10%~95%
Tipo de comunicación internet	ADSL, WiFi, 3G o Router 3G

Fuente: Autoría

3.4.1.5. Otros elementos.

Además fue necesario la utilización del Shield XBee, el cual sirve para conectar la placa Arduino con el módulo XBee S2C.

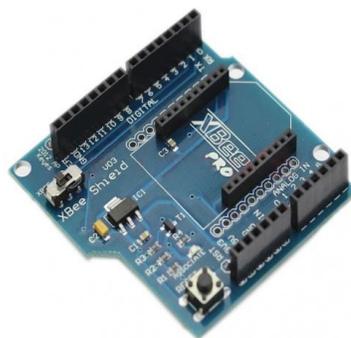


Ilustración 24. XBee Shield

Fuente: <https://store.arduino.cc/usa/arduino-wirelss-sd-shield>

El Shield XBee permite que una placa Arduino se comunice de forma inalámbrica con un módulo inalámbrico. Está basado en los módulos XBee de Digi, pero se puede usar con cualquier módulo con la misma huella. Se puede usar como un reemplazo del serial / USB o puede ponerlo en un modo de comando y usarlo para configurar el módulo XBee. El Shield divide cada uno de los pines de XBee en una almohadilla de soldadura de agujeros pasantes.

El Shield XBee es una placa en la que se monta el módulo XBee y a su vez se conecta sobre el Arduino, evitando la necesidad de cableado, además de que no obstruye o bloquea los pines sobrantes para que estos puedan ser utilizados.

3.4.2. Elección de Software.

A continuación se realiza la elección y descripción de cada uno de los software que se utilizó en este sistema de monitoreo.

3.4.2.1. IDE Arduino.

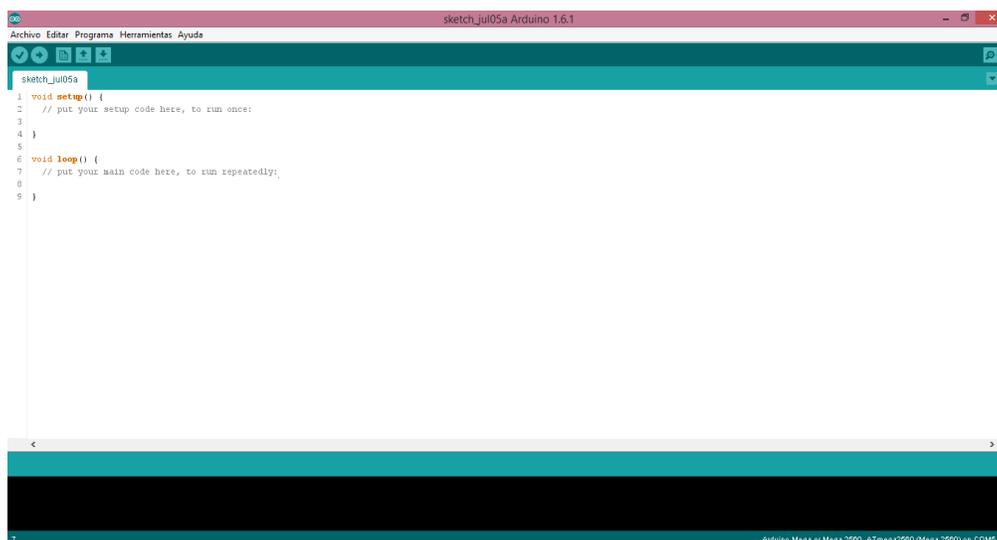


Ilustración 25. Interfaz del IDE de Arduino

Fuente: (Arduino, 2017)

En este caso se ha eligió directamente el IDE de Arduino ya que no es necesario realizar ninguna elección debido a que si las placas de procesamiento son Arduino se debe utilizar el software propietario de las mismas.

3.4.2.2. Almacenamiento en la nube.

Existen varias plataformas que permiten almacenar datos y visualizar los mismos de una manera comprensible, cada plataforma posee su propia interfaz y características diferentes que permiten observar los datos almacenados, como es en este caso lo datos recolectados por los nodos sensores. En la *Tabla 31* se realizó la valoración de cada una de las plataformas de almacenamiento en la nube.

Tabla 31. Elección de la nube de almacenamiento.

<i>Elección de la nube de almacenamiento.</i>					
<i>Hardware</i>	<i>Requerimientos</i>				<i>Valoración Total</i>
	StSR 8	SRSH 4	SRSH 7	SRSH 8	
Xively	1	0	1	1	3
Exosite	1	0	1	1	3
Ubidots	1	1	1	1	4
1 – Cumple					
0 – No Cumple					
Elección: Ubidots					

Fuente: Autoría

- **Ubidots.**

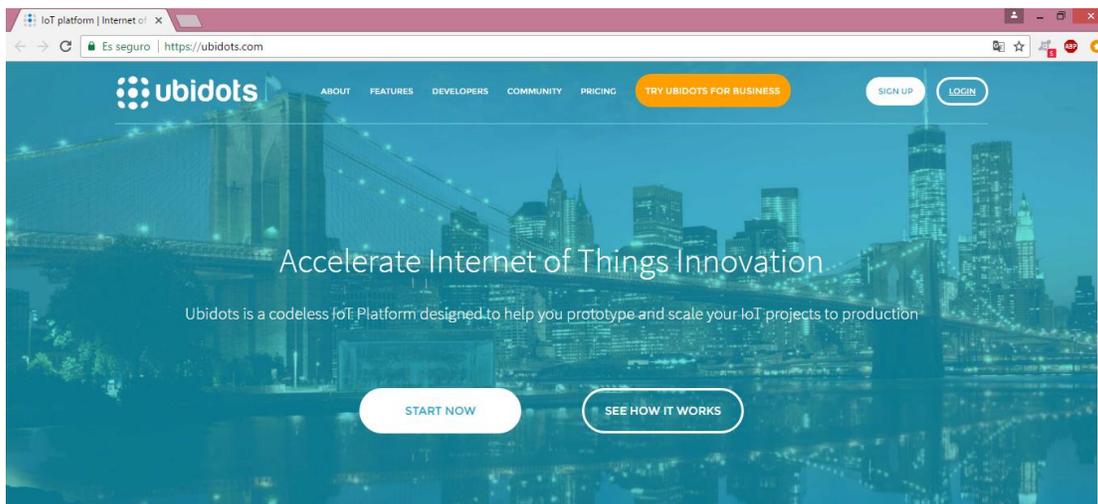


Ilustración 26. Interfaz web de Ubidots
Fuente: <https://ubidots.com/>

Ubidots es una plataforma que permite crear líneas de código y asociar dispositivos con gran facilidad, permite observar los datos recolectados en tiempo real y llevar un registro de ello, posee varias formas de mostrar los datos y generar alertas, no se necesita tener aplicaciones extras para poder ver los datos, ya que solo basta con ingresar a la página web de Ubidots e iniciar sesión para visualizar los datos recolectados por los dispositivos que tengamos asociados, lo cual facilita la monitorización de los mismos.

3.5. Diseño del sistema.

El área de estudio para el diseño de este sistema son los parqueaderos del campus universitario El Olivo, por lo cual se tomó en consideración toda la información y aspectos recabados anteriormente para establecer de una manera adecuada las siguientes fases de desarrollo del sistema: el diagrama de bloques del proyecto, la arquitectura del sistema, el diseño de la WSN, el diagrama de flujo, el almacenamiento y visualización de la información en la nube.

3.5.1. Diagrama de bloques.

Para tener claro cuáles son las principales fases de diseño del sistema se procedió a realizar el diagrama de bloques que se muestra en la *Ilustración 27*, de este modo se puede apreciar de manera rápida los bloques que se tomaron en cuenta al momento de realizar el diseño del sistema.

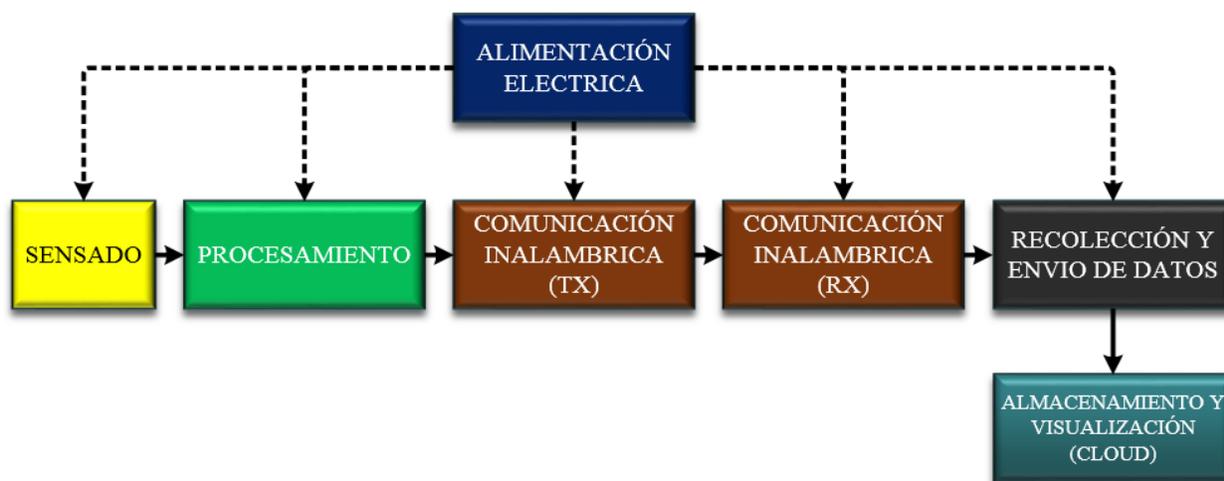


Ilustración 27. Diagrama de bloques del sistema.
Fuente: Autoría

El diagrama de bloques del sistema se subdivide en varias etapas:

- Bloque de sensado, comprende el sensor del sistema los cuales se encarga de detectar si el estacionamiento se encuentra o no disponible.
- Bloque de procesamiento se encarga de recibir la señal del sensor e interpretarla para poder transmitirla.
- Bloque de comunicación inalámbrica (TX), este se encarga de transmitir de manera inalámbrica los datos proporcionados por la placa de procesamiento a un receptor central.

- Bloque de comunicación inalámbrica (RX), este receptor central se encarga de recibir todos los datos enviados por los transmisores y pasarlos al bloque de recolección.
- Bloque de recolección y envío de datos, se encarga de recolectar los datos que anteriormente fueron transmitidos inalámbricamente y enviarlos a una plataforma de almacenamiento a través de internet.
- Bloque de energía, es el encargado de proporcionar energía eléctrica a cada uno de los componentes del sistema.
- Bloque de almacenamiento y visualización, todos los datos recolectados son enviados a la plataforma Ubidots para su almacenamiento y visualización mediante su interfaz WEB.

3.5.2. Arquitectura del sistema.

La arquitectura del sistema está formada por dos partes principales que son: la WSN y la plataforma PaaS, donde la WSN está constituida por una serie de sensores y un único nodo central en cada parqueadero, y la plataforma PaaS se encarga de almacenar los datos proporcionados por todos los nodos centrales y mostrarlos mediante su propia interfaz web.

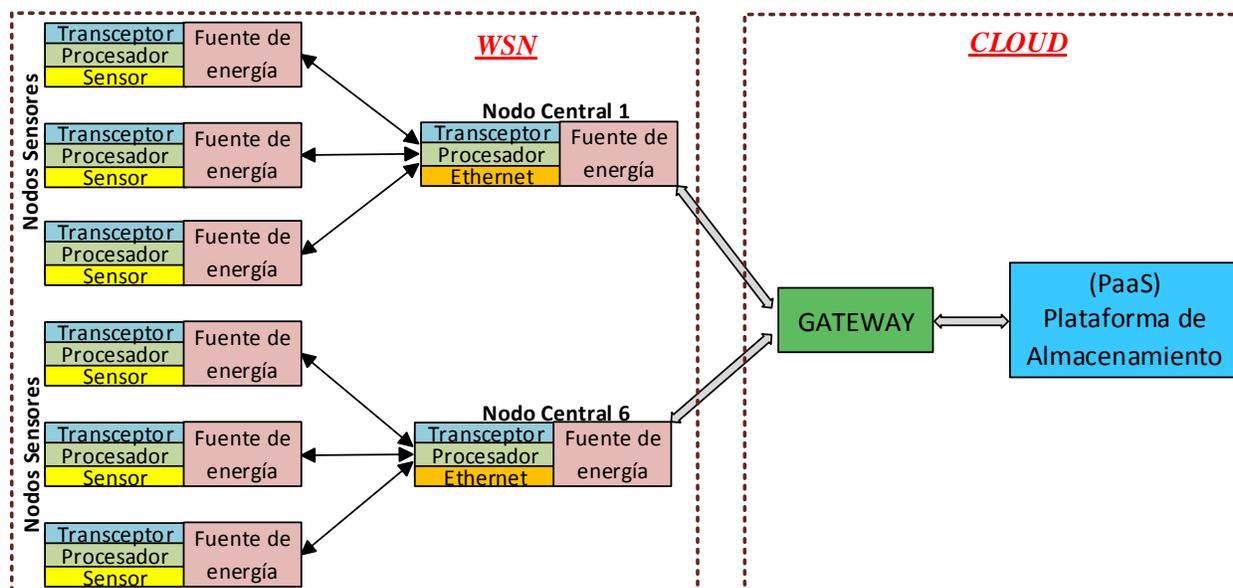


Ilustración 28. Arquitectura del sistema.

Fuente: Autoría

3.5.3. Diseño de la WSN.

En este apartado se describe todo lo relacionado con el diseño de la red inalámbrica de sensores, como son: la topología de la red, la ubicación de los componentes de la WSN y el desarrollo de los nodos, teniendo en cuenta los parámetros que fueron considerados al momento de elegir los componentes del sistema.

El diseño de la red se lo desarrolló para los nueve parqueaderos de automóviles. El mismo que está constituido por un nodo central por parqueadero y un nodo sensor por estacionamiento, como se lo puede apreciar en la *Tabla 32*.

Tabla 32. Cantidad de nodos centrales y sensores.

Parqueadero	Nombre del Parqueadero	Nodo Central	Nodos Sensores
Parqueadero A1	Parqueadero de la FECYT	1	66
Parqueadero A2	Parqueadero de la FACAE	1	25
Parqueadero A3	Parqueadero del Comedor Universitario	1	51
Parqueadero A4	Parqueadero de Postgrado	1	37
Parqueadero A5	Parqueadero del Gimnasio	1	15
Parqueadero A6	Parqueadero de Educación Física	1	42
Parqueadero A7	Parqueadero de las Canchas	1	91
Parqueadero A8	Parqueadero de Autoridades	1	14
Parqueadero A9	Parqueadero de la FICA y FICAYA	1	95
TOTAL		9	436

Fuente: Autoría

3.5.3.1. Topología de la WSN.

Los principales aspectos que tomaron en cuenta para la elección de la topología de la WSN son: el consumo de energía y el ambiente de trabajo; de tal manera que al tratarse de un ambiente externo no existe mayor cantidad de obstáculos para la comunicación inalámbrica, y además, es necesario que los nodos tengan cierta autonomía energética, por lo cual se requiere un bajo consumo de energía.

Por tal motivo la topología que se utilizó es tipo estrella, ya que al utilizar esta configuración es posible poner a los nodos sensores en modo sleep cuando no tengan ninguna actividad, lo cual permite tener un gran ahorro de energía, cosa que no se puede hacer en una

configuración tipo árbol o malla, además el alcance de los XBee S2C en exteriores es de 1200m en teoría, lo cual es suficiente para poder tener una comunicación directa entre los nodos sensores y el nodo central sin necesidad de tener nodos router.

3.5.3.2. Ubicación de los nodos.

En este apartado se definen los lugares en los que deben ir ubicados los nodos sensores y centrales, para lo cual se toma en cuenta los parámetros previamente establecidos en el análisis de requerimientos, y de esta manera lograr que el sistema funcione adecuadamente.

- **Ubicación de los nodos sensores.**

Los lugares donde serán ubicados cada uno de estos nodos son los estacionamientos, los mismos que se encuentran señalizados en todos los parqueaderos del campus universitario, entonces al tener ya un área delimitada el sensor va a ser colocado en un sitio donde detecte con gran facilidad la presencia del automóvil y además no obstaculice de ninguna manera al automóvil que se estacione.

Teniendo en cuenta que se trata de un ambiente externo, se considera que el sensor debe ser colocado sobre el suelo (entre los topes de estacionamiento) ya que de esta manera el sensor será cubierto completamente por el automóvil, lo cual facilita la detección de este último por parte del sensor y además no es necesario implementar ningún tipo de infraestructura adicional para la colocación de los mismos. Por tales motivos se ha establecido que el sensor sea colocado tal como se muestra en la *Ilustración 29*.

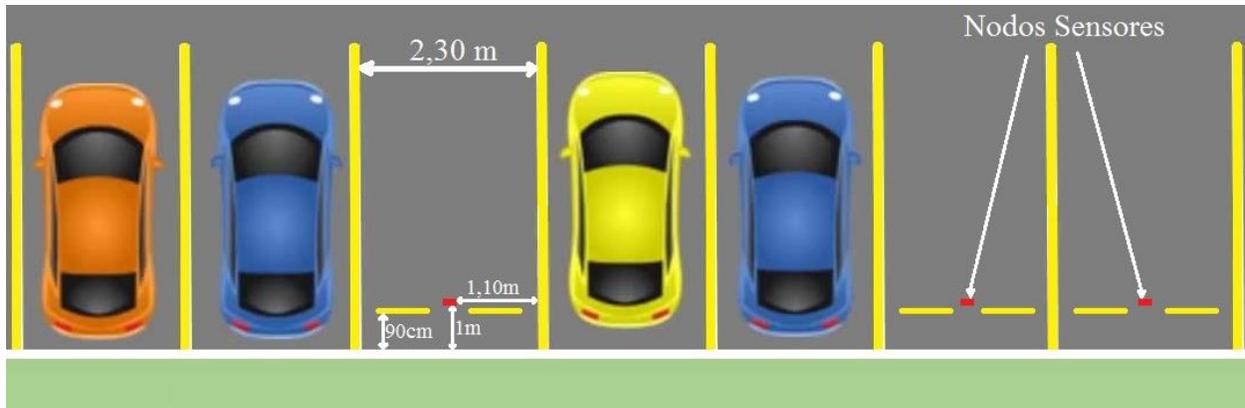


Ilustración 29. Ubicación de los nodos sensores.
Fuente: Autoría

- **Ubicación de los nodos centrales.**

Los parámetros más importantes a tomar en cuenta para la ubicación de los nodos centrales, es que estos nodos requieren tener acceso cableado a internet, conexión eléctrica y deben estar cerca a sus nodos sensores, por lo cual se consideró ubicarlos en los edificios que se encuentran más cercanos a los parqueaderos y en los cuales se encuentran colocados AP externos, ya que es un buen lugar para tener acceso a la red y a la energía eléctrica aprovechando la infraestructura que ya se encuentra instalada. Por lo tanto en la *Ilustración 30* se puede apreciar la ubicación de cada uno de los nodos centrales.

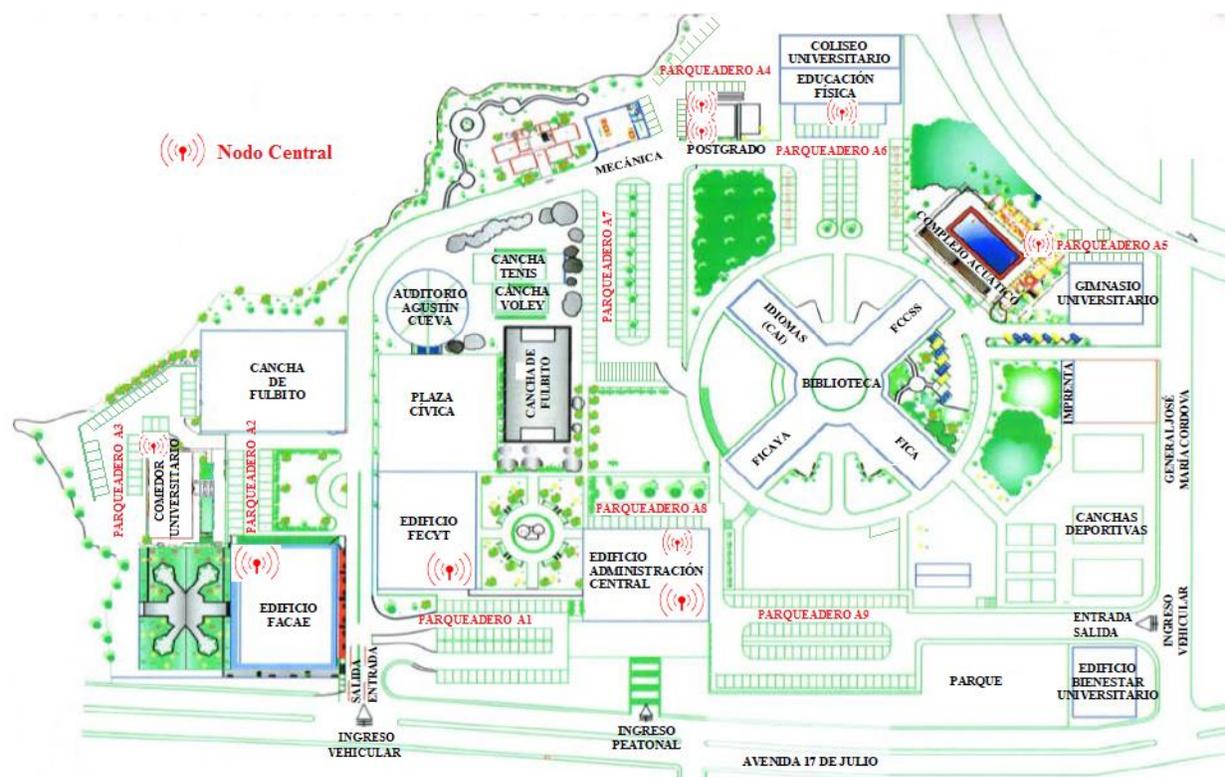


Ilustración 30. Ubicación de los nodos centrales.

Fuente: Adaptado de: UTN.

Luego de establecer dónde estarán ubicados los nodos centrales, se midió distancia entre el sitio establecido para los nodos centrales y el estacionamiento más lejano del parqueadero, siendo 95 metros la distancia más larga, pero que se encuentra dentro del alcance de los XBee.

Tabla 33. Distancia máxima entre el nodo central y el nodo sensor.

Ubicación del nodo Sensor	Ubicación del Nodo Central	Distancia máxima entre nodos
Parqueadero de la FECYT	FECYT	58m
Parqueadero de la FACAE	FACAE	53m
Parqueadero del Comedor Universitario	Comedor Universitario	40m
Parqueadero de Postgrado	Postgrado	42m
Parqueadero del Gimnasio	Complejo Acuático	30m

Parqueadero de Educación Física	Educación Física	55m
Parqueadero de las Canchas	Postgrado	95m
Parqueadero de Autoridades	Administración Central	20m
Parqueadero FICA - FICAYA	Administración Central	89m

Fuente: Autoría

3.5.3.3. *Nodo sensor.*

Las partes por las cuales está conformado cada uno de los nodos sensores son: un sensor infrarrojo Sharp 2Y0A21 incluyendo una adaptación de Shield el cual fue diseñado, una placa de procesamiento Arduino Uno, un transceptor inalámbrico XBee S2C, un Shield XBee y una fuente de energía eléctrica.

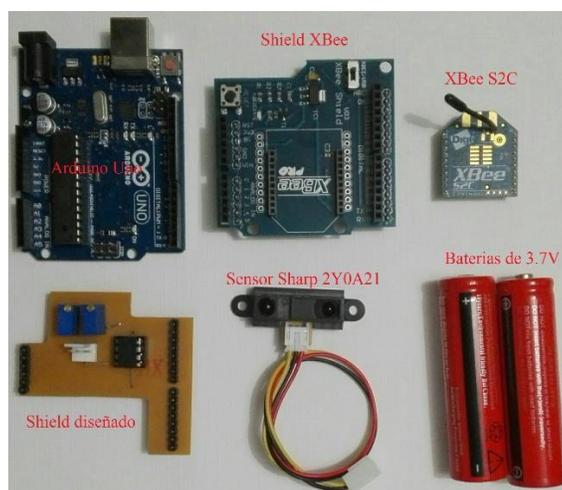


Ilustración 31. Componentes de los nodos sensor.

Fuente: Autoría.

El sensor infrarrojo Sharp 2Y0A21 es el encargado de detectar la presencia del automóvil en el estacionamiento, el dato que arroja el sensor es analógico por lo cual se realiza una adaptación de Shield para convertir el dato analógico a digital y enviar ese dato al Arduino Uno mediante las

interrupciones 0 y 1, el Arduino uno es el encargado de procesar los datos recibidos y enviarlos mediante los pines de la comunicación serial al XBee S2C, el Shield XBee permite realizar la conexión entre el Arduino Uno y el XBee S2C, posee un pequeño circuito regulador de voltaje para suministrar el voltaje al que trabaja el módulo XBee, el XBee S2C se encarga de comunicarse inalámbricamente con el nodo central mediante la tecnología zigbee.

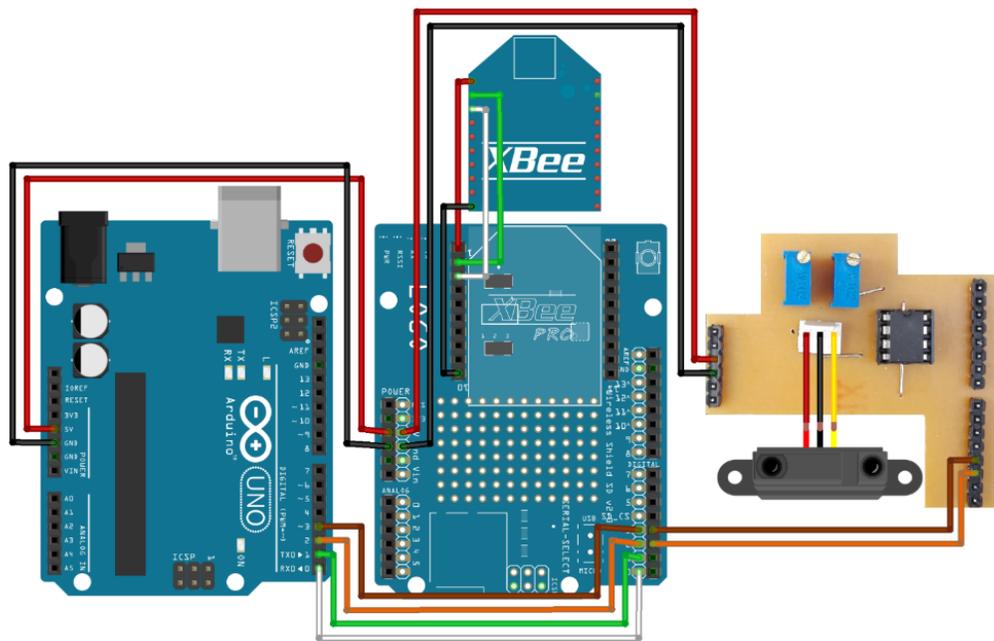


Ilustración 32. Diagrama de conexión del nodo sensor.
Fuente: Autoría – Elaborado en Fritzing.

La adaptación de shield para el sensor infrarrojo fue realizada en el software Eagle, el cual es un programa de diseño de diagramas y de placas de circuito impreso (PCB). Fue diseñada en base a un Arduino uno, por lo cual no existe ningún problema de compatibilidad para montarlo sobre él. Los componentes que contiene esta placa son dos potenciómetros de precisión de 1 kilo ohmio, un integrado LM358 y un molex macho de tres pines.

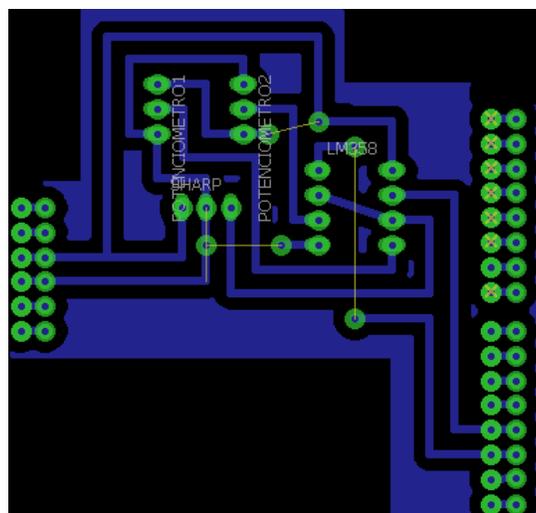


Ilustración 33. Diseño de la PCB - Shield para el sensor infrarrojo.
Fuente: Autoría.

Además de cumplir con todos los requerimientos de hardware, estos dispositivos tienen gran facilidad para acoplarse entre ellos, solo basta con montar una placa sobre de otra, esto es de gran importancia ya que se los puede remover fácilmente en caso de que se necesite realizar algún replazo o mantenimiento de alguno de los componentes.

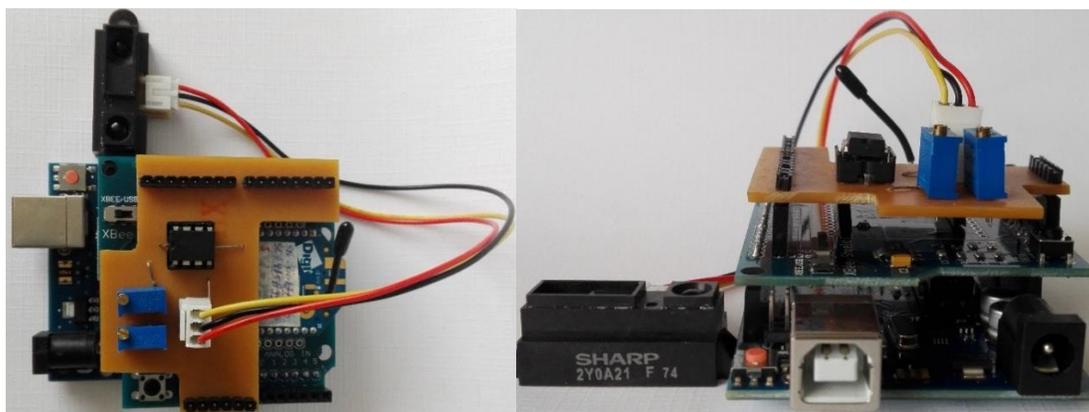


Ilustración 34. Montaje de las placas del nodo sensor. (a) Vista superior. (b) Vista frontal.
Fuente: Autoría.

Teniendo en cuenta que los nodos sensores van a estar instalados al aire libre, estos dispositivos necesitan estar protegidos de factores externos, es por ello que se utilizó una caja de

protección de 10x6.8x5 centímetros para colocar el nodo dentro de ella, dicha caja se muestra en la *Ilustración 35*.



Ilustración 35. Caja de protección para el nodo sensor.
Fuente: Autoría.

Posteriormente se realizó la adecuación de la caja para que todos los componentes del nodo sensor estén ubicados dentro de la caja de una manera organizada, lo primero fue colocar las baterías en la tapa de la caja de manera que se puedan extraer de manera fácil y posteriormente se hizo una perforación en la tapa donde estará ubicado el sensor.



Ilustración 36. Vista interna del nodo sensor.
Fuente: Autoría.

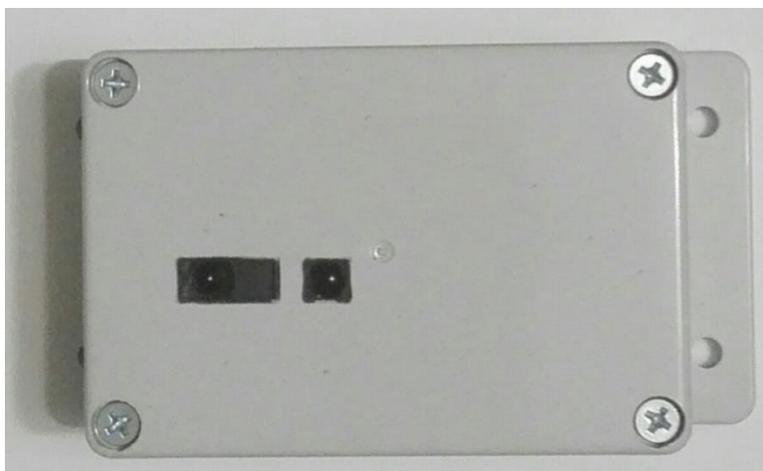


Ilustración 37. Nodo sensor finalizado.

Fuente: Autoría.

- **Alimentación eléctrica.**

Todo dispositivo electrónico requiere energía eléctrica para su funcionamiento, en este caso la tensión de alimentación recomendada para el Arduino Uno es de (7-12 V), y a través del cual serán energizados los demás dispositivos que conforman el nodo. A continuación se realiza el cálculo de consumo de corriente del nodo.

Tabla 34. Consumo de corriente de los elementos del nodo sensor.

Dispositivo	Consumo de corriente - Modo Normal	Consumo de corriente – Modo Dormido
Sharp 2Y0A21	28mA	28mA
Arduino Uno	19,5mA	0.36mA
Shield XBee	2mA	2mA
XBee S2C	30mA	1uA
Shield diseñado	8mA	8mA
TOTAL	87.5mA	39.36mA

Fuente: Autoría

- T_{cn} = Tiempo Consumo Normal
- T_{cd} = Tiempo Consumo Dormido
- I_{cn} = Intensidad de Corriente Consumo Normal
- I_{cd} = Intensidad Corriente Consumo Dormido

Se aplica la siguiente ecuación para calcular el consumo de corriente que tiene el nodo.

$$Consumo = \frac{T_{cn} * I_{cn} + T_{cd} * I_{cd}}{T_{cn} + T_{cd}}$$

En este caso los valores que se toman en cuenta son:

$$T_{cn} = 0.08 \text{ horas} \quad T_{cd} = 0.92 \text{ horas} \quad I_{cn} = 87.5 \text{ mA} \quad I_{cd} = 39.36 \text{ mA}$$

Ingresando los datos detallados anteriormente a la ecuación se tiene:

$$Consumo = \frac{0.08h * 87.5mA + 0.92h * 39.36mA}{0.08h + 0.92h}$$

$$Consumo = 43.21mA$$

Las baterías que se escogió para alimentación de los nodos sensores son de Ion Litio de 3,7 V a 3000 mAh, por lo cual se instalaran dos baterías en serie para obtener 7,4 V. Conociendo la capacidad de las baterías y el consumo de los nodos sensores se procede a calcular el tiempo de descarga.

$$Tiempo \ de \ descarga = \frac{Capacidad \ de \ la \ Bateria}{Consumo}$$

$$\text{Tiempo de descarga} = \frac{3000 \text{ mAh}}{43.21 \text{ mA}}$$

$$\text{Tiempo de descarga} = 69.42 \text{ horas}$$

$$\text{Tiempo de descarga} = 2 \text{ días } 21 \text{ horas } 25.7 \text{ minutos}$$



Ilustración 38. Baterías Ultrafire.

Fuente: <http://www.ultrafire.com/Batteries-c854>

3.5.3.4. *Nodo central.*

Cada uno de los nodos centrales están conformados por: una placa de procesamiento Arduino mega2560, un shield Ethernet, un shield XBee, un XBee S2C y una fuente de energía eléctrica.

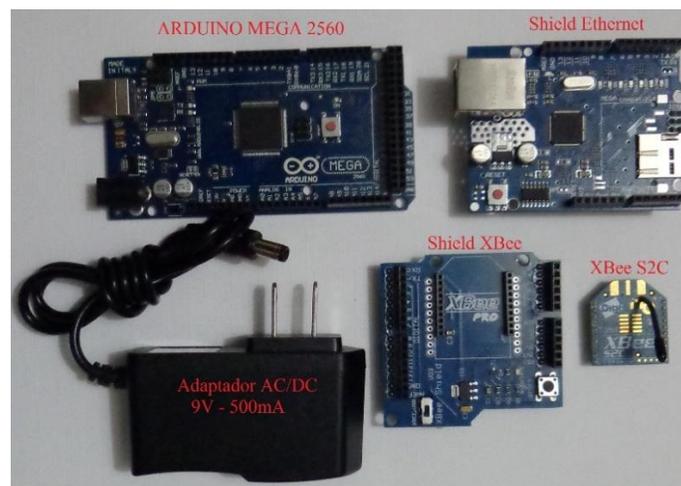


Ilustración 39. Componentes del nodo central.

Fuente: Autoría.

El módulo XBee S2C es el encargado de comunicarse inalámbricamente con los nodos sensores mediante la tecnología Zigbee, el Shield XBee se ocupa de la comunicación entre el XBee S2C y la placa de procesamiento mediante los pines de la comunicación serial y de proporcionar el voltaje adecuado al XBee S2C, el Arduino mega2560 se encarga de procesar todos los datos recibidos por el XBee S2C y enviarlos mediante la interfaz serial periférica (SPI) al Shield Ethernet, el cual se ocupa de la conexión hacia la red TCP/IP para realizar el envío de datos hacia la nube.

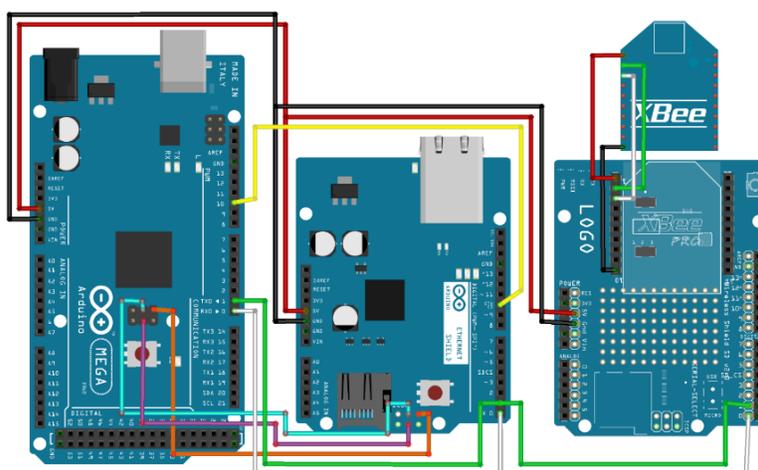
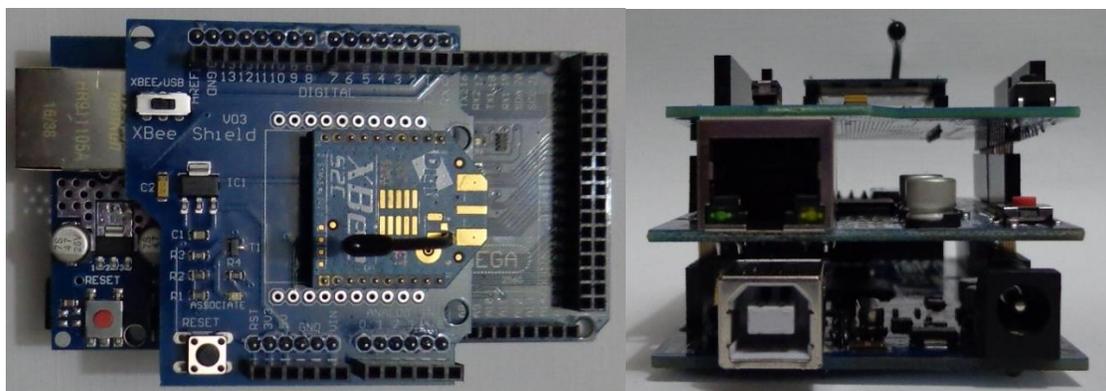


Ilustración 40. Diagrama de conexión del nodo central.

Fuente: Autoría – Elaborado en Fritzing.

Al igual que los componentes de hardware del nodo sensor, estos dispositivos también pueden acoplarse fácilmente entre ellos, colocando una placa sobre otra, pero en este caso el Shield Ethernet es el que primero debe ir sobre el Arduino mega, ya que necesita conectarse con los pines ICSP del Arduino para su funcionamiento.



(a)

(b)

Ilustración 41. Montaje de las placas del nodo central. (a) Vista superior. (b) Vista frontal.

Fuente: Autoría.

De igual manera que los nodos sensores, los componentes del nodo central también deben estar protegidos de alguna manera ante el polvo, agua o la manipulación de los dispositivos del nodo por parte de terceros, es por ello que en este caso se utilizó una caja de derivación de 18x14x8 centímetros para colocar el nodo, la misma que se puede apreciar en la *Ilustración 42*.



Ilustración 42. Caja de protección para el nodo central.

Fuente: Autoría.

Para concluir con el armado del nodo central se realizó la adecuación de la caja, lo cual consistió en hacer dos perforaciones para colocar un Jack RJ45 hembra y un conector hembra invertido, los mismos que fueron conectados al puerto Ethernet y al conector de alimentación

eléctrica del Arduino respectivamente, para que no sea necesario abrir la caja para realizar la conexión del cable de red y del adaptador de 9V.



Ilustración 43. Vista interna del nodo central.
Fuente: Autoría.



Ilustración 44. Nodo central finalizado.
Fuente: Autoría.

- **Alimentación eléctrica.**

Al igual que los nodos sensores, los nodos centrales también requieren de una fuente de alimentación eléctrica, la tensión de alimentación recomendada para el Arduino mega es de (7-12 V), a través de la cual se energizaran los demás dispositivos que conforman el nodo, a continuación en la *Tabla 35* se presenta el consumo de corriente que tiene cada dispositivo.

Tabla 35. Consumo de corriente de los elementos del nodo central.

Dispositivo	Consumo de corriente
XBee S2C	30mA
Shield XBee	2mA
Shield Ethernet	188mA
Arduino Mega 2560	32mA
TOTAL	252mA

Fuente: Autoría

El consumo de corriente total del nodo central es de 252mA, entonces teniendo en cuenta que es un consumo de corriente es alto y que estos nodos no son netamente inalámbricos ya que tendrán un cable de red para su acceso a internet se optó por alimentarlo mediante un adaptador de corriente AC/DC de 9 voltios y 500mA.



Ilustración 45. Adaptador AC/DC para Arduino.

Fuente: <http://cdtecnologia.net/arduino/16-fuente-de-energia-para-arduino.html>

3.5.4. Diagramas de flujo.

Los diagramas de flujo que se presentan a continuación son una representación gráfica de los procesos que realiza cada parte del sistema, y tuvieron como objetivo servir de guía al momento de la realización de los programas de las placas Arduino que se encuentran en los nodos, dichos programas se los puede observar en los ANEXOS 9 y 10.

3.5.4.1. Diagrama de flujo del nodo sensor.

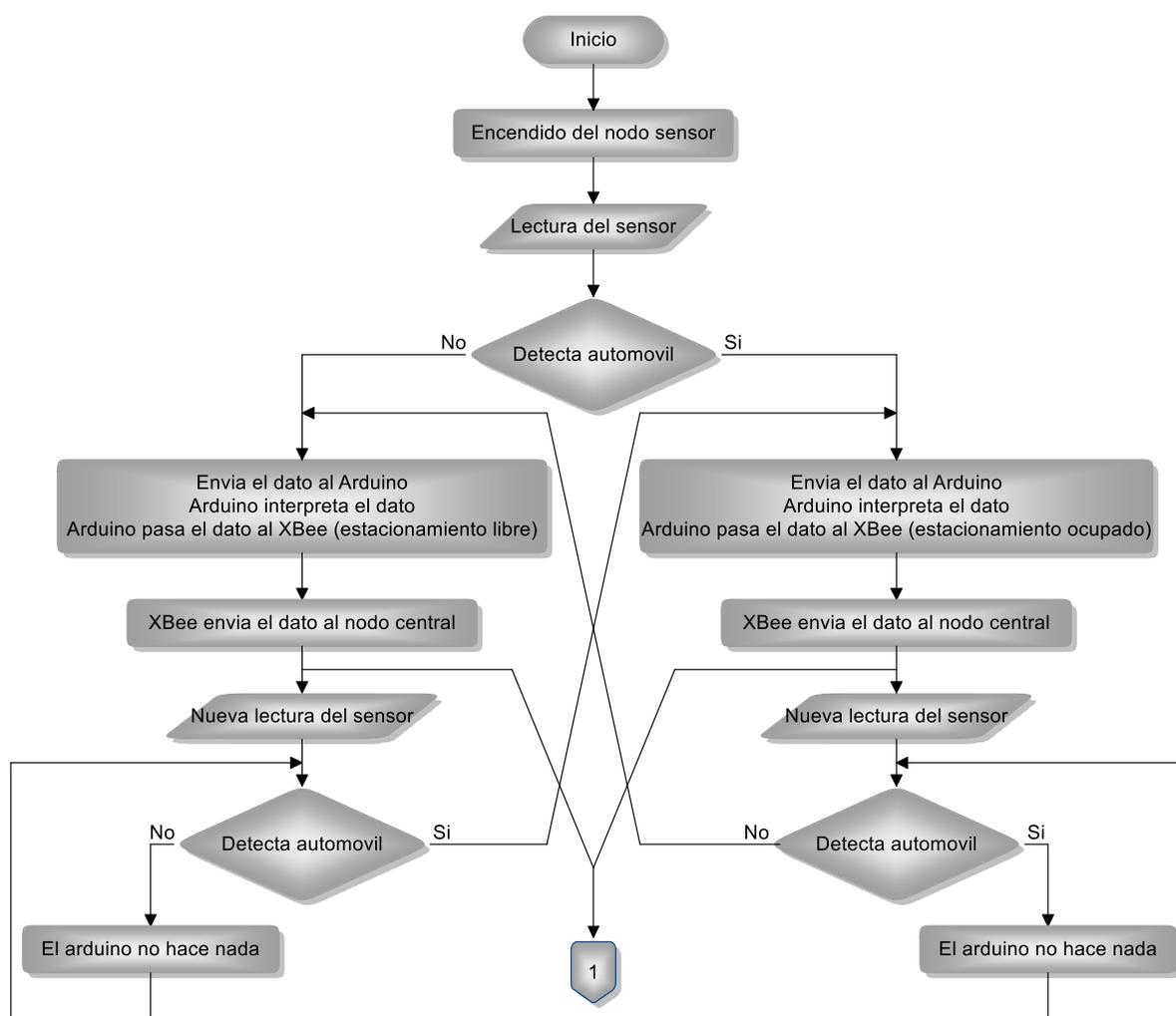


Ilustración 46. Diagrama de flujo del nodo central.

Fuente: Autoría.

3.5.4.2. Diagrama de flujo del nodo central.

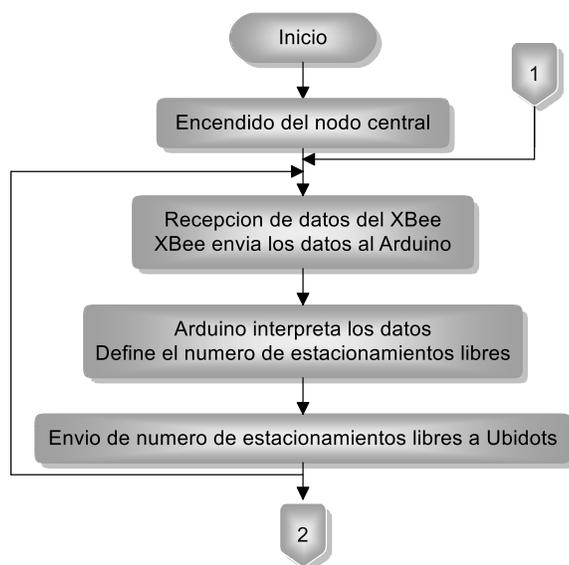


Ilustración 47. Diagrama de flujo del nodo central.
Fuente: Autoría.

3.5.4.3. Diagrama de flujo de la plataforma Ubidots.

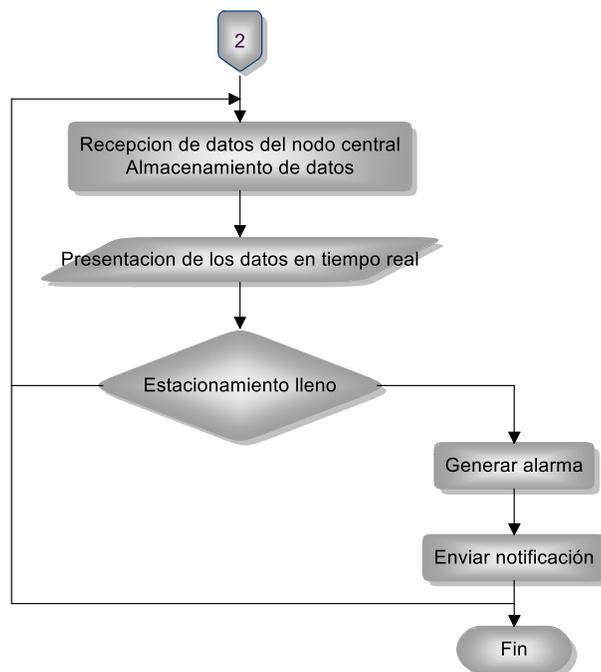


Ilustración 48. Diagrama de flujo de la plataforma de Ubidots.
Fuente: Autoría.

3.5.5. Almacenamiento y visualización de la información en la nube.

El monitoreo de la disponibilidad de los estacionamientos se lo realizó a través de la plataforma Ubidots, por lo cual en este apartado se describen los pasos a seguir para poder realizar el almacenamiento y visualización de los datos. Primeramente se debe ingresar a su página web oficial <https://ubidots.com/>, dirigirse a la opción de registro y registrarse ingresando un nombre de usuario, el correo electrónico y una contraseña, una vez creada la cuenta, Ubidots envía un mensaje de bienvenida al correo electrónico con el cual se realizó el registro.

Ahora se deben realizar algunas configuraciones, una vez que se ha ingresado en la cuenta de Ubidots, hay que dirigirse a Dispositivos y posteriormente en Añadir dispositivo, configurar el nombre del dispositivo creado e ingresar en él.

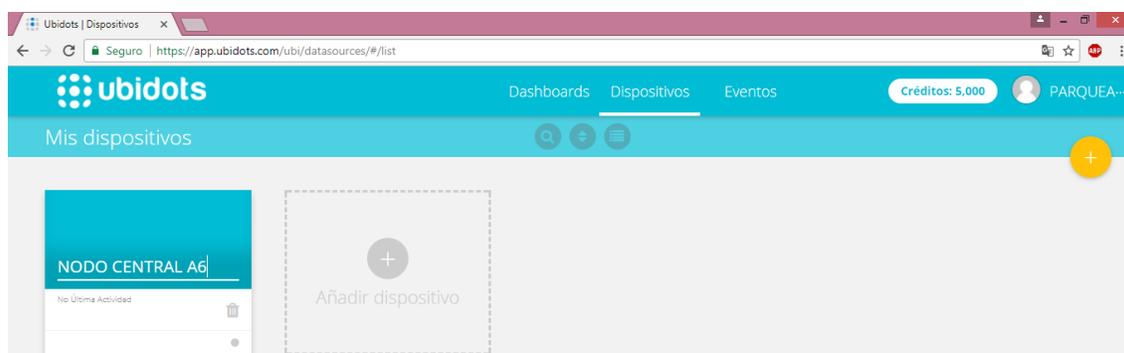


Ilustración 49. Creación de dispositivos en Ubidots.

Fuente: (Ubidots, 2017)

Al ingresar al dispositivo se debe agregar una variable y de igual manera hay que configurar el nombre que se le quiere dar, en esta caso la variable ha sido llamada Parqueadero A9, la cual almacena la cantidad de estacionamientos vacíos. Se debe tener muy en cuenta la ID de la Variable que se creó, ya que esta ID permite crear la conexión con Arduino.

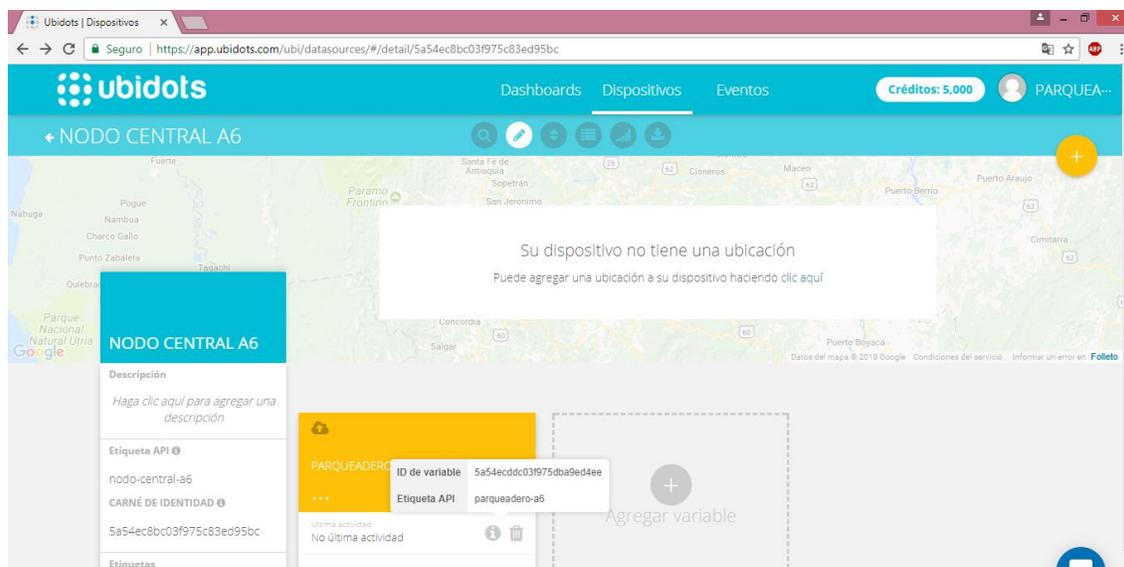


Ilustración 50. Creación de la variable en Ubidots.
Fuente: (Ubidots, 2017)

Ahora se debe dirigir a la parte superior derecha de la pantalla e ingresar en Credenciales API

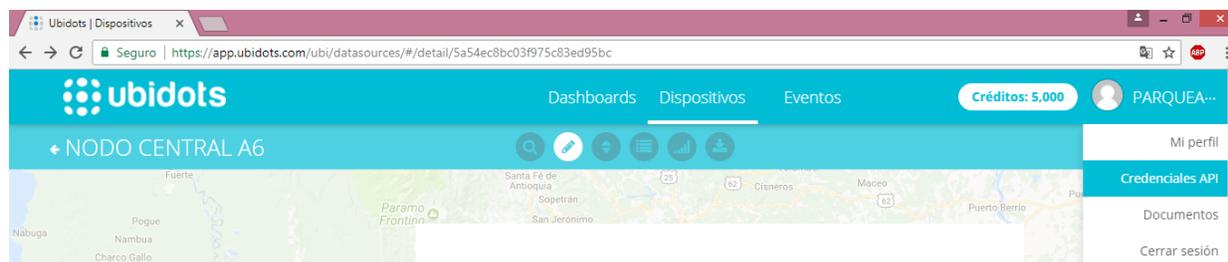


Ilustración 51. Ingreso a las credenciales API de Ubidots.
Fuente: (Ubidots, 2017)

En esta nueva ventana se debe crear un Token, esto es código de autenticación temporal, el cual permite realizar la comunicación entre Arduino y la plataforma Ubidots.

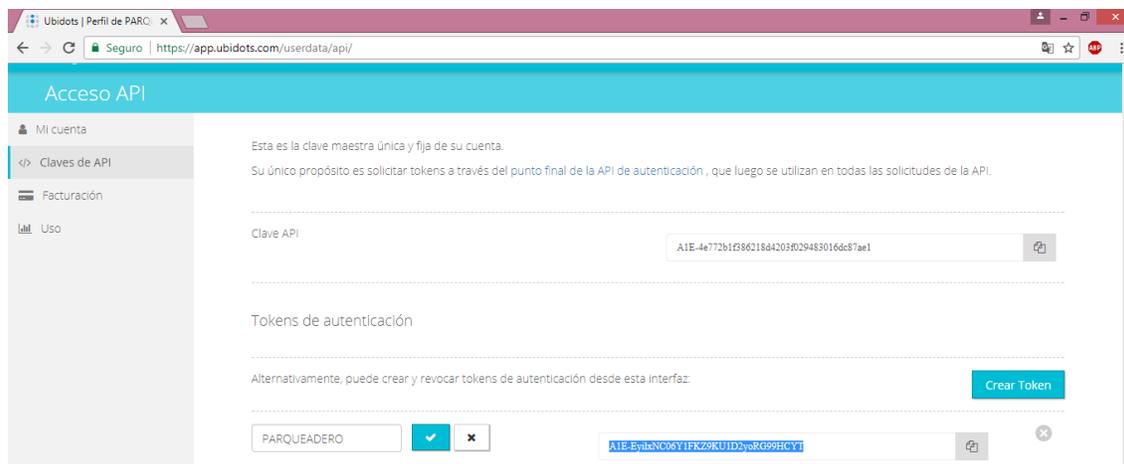


Ilustración 52. Creación del Token en Ubidots.
Fuente: (Ubidots, 2017)

Una vez finalizados todos estos pasos ya se está listo para ingresar el ID de la variable y el Token en el código de Arduino tal como se muestra en la *Ilustración 53*, y se iniciará la comunicación entre el Shield Ethernet de Arduino y Ubidots, teniendo en cuenta que la placa de Arduino debe estar conectada a Internet para poder enviar los datos. Para más detalle del código ver *ANEXO 10*.

```
#include <Ethernet.h>
#include <SPI.h>
#include <UbidotsEthernet.h>
#define ID "Your_variable_ID_here" // Put here your Ubidots variable ID
#define TOKEN "Your_token_here" // Put here your Ubidots TOKEN
```

Ilustración 53. Ingreso del Token Ubidots y su ID de variable.
Fuente: (Ubidots, 2017)

3.5.5.1. Almacenamiento de datos.

Para poder observar los datos almacenados es necesario ingresar a la ventana de Dispositivos, acceder al dispositivo y posteriormente a la variable, cada variable lleva un registro

de los datos almacenados y se los puede ver de acuerdo a nuestra elección, pueden ser de la última hora, día, semana, mes o elegir cualquier fecha.

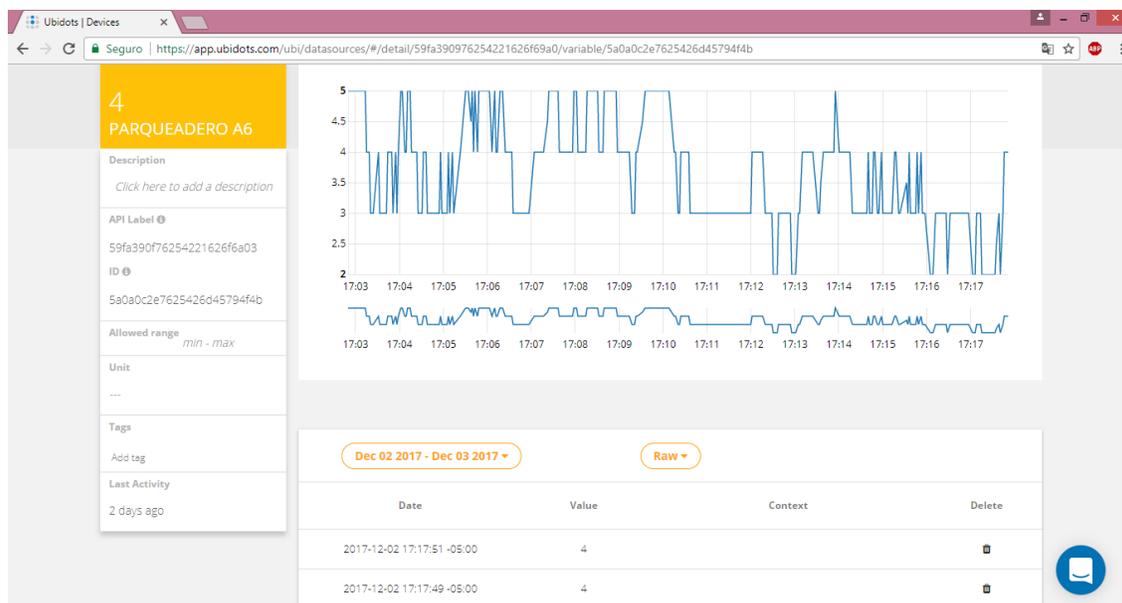


Ilustración 54. Almacenamiento de datos en Ubidots.
Fuente: (Ubidots, 2017)

3.5.5.2. Visualización de la información.

Para poder visualizar los datos en tiempo real es necesario agregar Widgets, esto se lo realiza en la página principal de Ubidots (Dashboard).

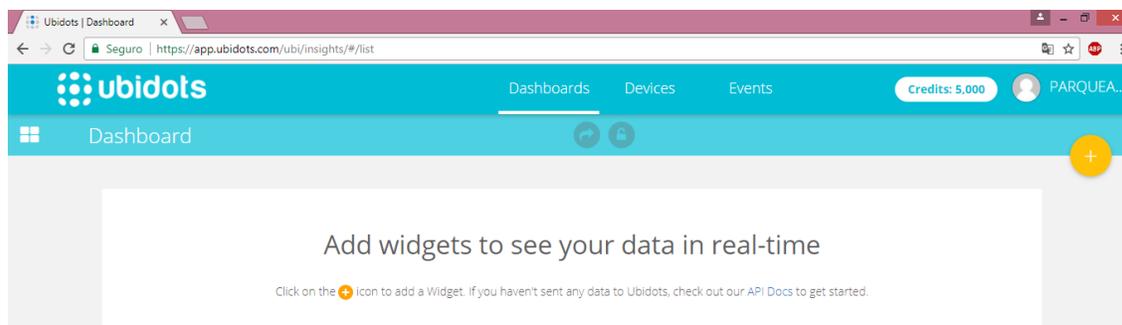


Ilustración 55. Página principal de Ubidots.
Fuente: (Ubidots, 2017)

Esta plataforma tiene la capacidad de presentar los datos de varias maneras, en este caso se escogió Metric, ya que se necesita mostrar un valor numérico.

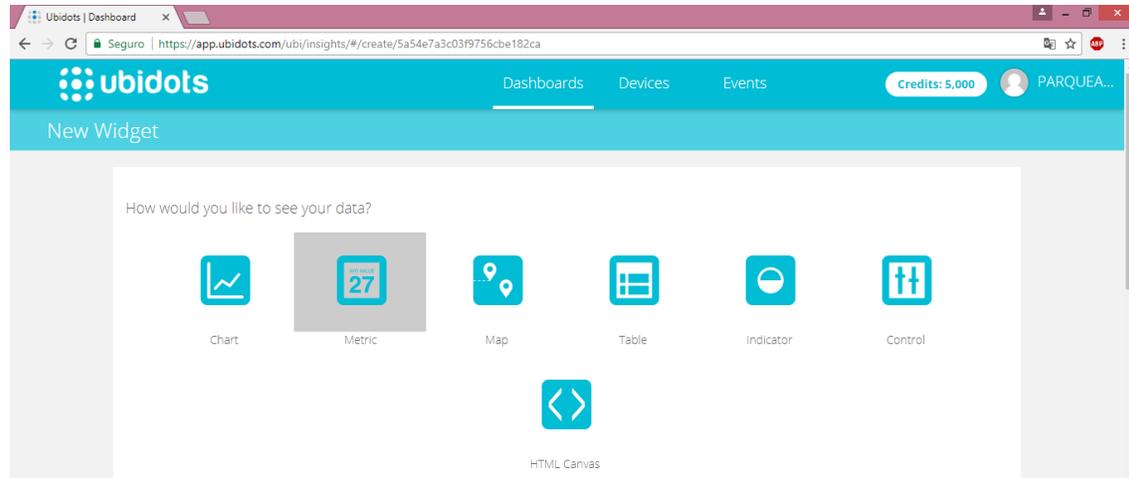


Ilustración 56. Elección de la presentación de datos en Ubidots.
Fuente: (Ubidots, 2017)

Luego se debe seleccionar qué dato mostrar y escoger el dispositivo y la variable que estará asociada a este Widget.

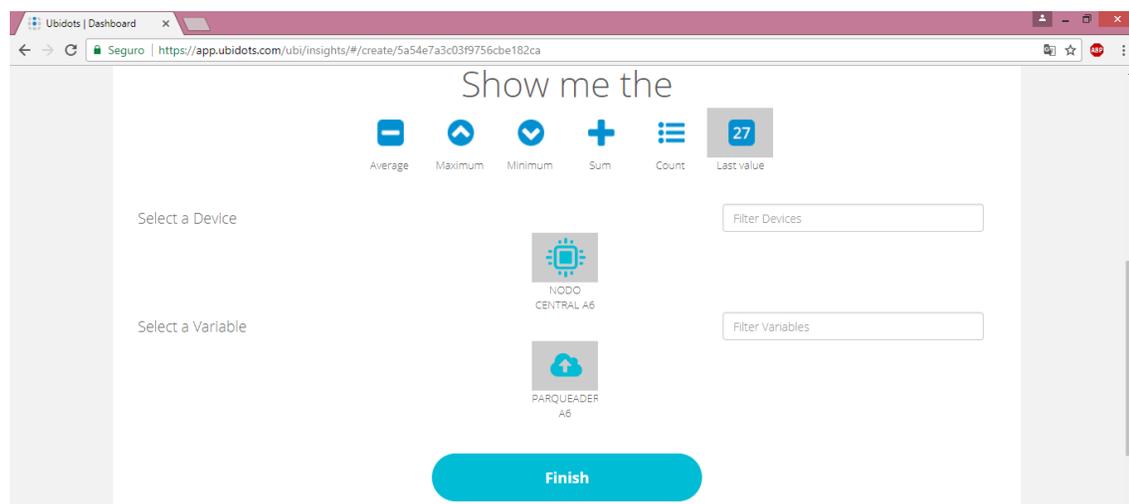


Ilustración 57. Elección del dato que va a mostrar Ubidots.
Fuente: (Ubidots, 2017)

Y finalmente los datos pueden ser visualizados de una manera adecuada en la interfaz web de Ubidots, para lo cual solo basta con tener un dispositivo con navegador web y que tenga acceso a internet para abrir la interfaz antes mencionada, la misma que se puede apreciar en la *Ilustración 58*.



Ilustración 58. Interfaz de monitoreo en Ubidots.
Fuente: (Ubidots, 2017)

3.5.5.3. Creación de alertas.

Las alertas son algo que Ubidots no deja de lado, ya que permite crear eventos y generar alertas si llega a ocurrir dicho evento. Para lo cual se ingresa en la ventana de eventos, se añade un nuevo evento, se selecciona el dispositivo que se va asociar y se selecciona la variable.

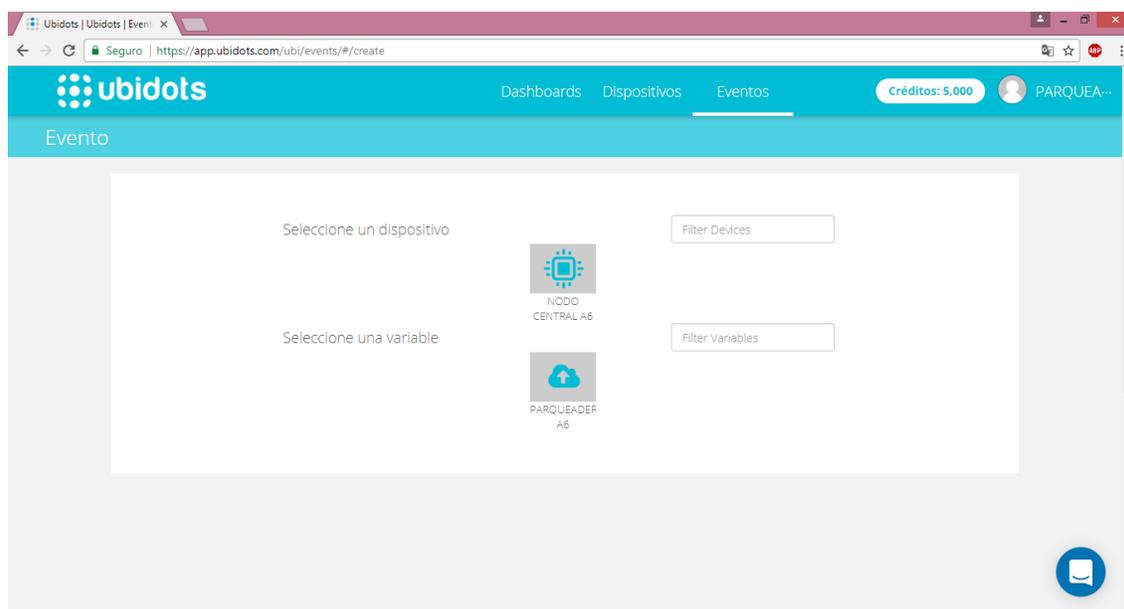


Ilustración 59. Creación de eventos en Ubidots.

Fuente: (Ubidots, 2017)

Posteriormente aparecen más opciones, entre las cuales están valor, tiempo de Inactividad o ubicación, en este caso se elige la opción valor para poder generar una alerta cuando un parqueadero se encuentre lleno.

Si

PARQUEADERO A6

Menos mayor menos o igual mayor o igual igual

que

Continuar

Ilustración 60. Condición para que se genere la alerta en Ubidots.

Fuente: (Ubidots, 2017)

Y se elige el medio por el cual se quiere recibir la alerta y el mensaje que se enviara.

Enviar correo electrónico Enviar SMS Enviar Telegram WebHook Establecer una variable

Agregar correo electrónico

Email

Tema

Mensaje

Terminar

Ilustración 61. Elección del medio por el cual Ubidots genera la alerta.

Fuente: (Ubidots, 2017)

Capítulo 4. Pruebas de Funcionamiento.

Una vez concluida la fase de diseño se procede a realizar la implementación del prototipo del sistema, el mismo que permitirá verificar el funcionamiento del diseño propuesto, para lo cual se lo aplicara en un área limitada donde se realizaran las pruebas respectivas.

4.1. Implementación del prototipo.

En esta sección se lleva a cabo la implementación del prototipo del sistema, lo cual incluye la colocación de los nodos sensores y del nodo central en el parqueadero A9 y la utilización de dispositivos electrónicos para realizar la visualización de los datos mediante la interfaz web de Ubidots.

4.1.1. Colocación del nodo sensor.

Los cinco nodos sensores fueron colocados en cinco estacionamientos, es decir un nodo por estacionamiento y se ubicaron sobre el suelo en los lugares establecidos en el capítulo 3.



Ilustración 62. Colocación del nodo sensor.

Fuente: Autoría.



Ilustración 63. Nodos sensores instalados.
Fuente: Autoría.

4.1.2. Colocación del nodo central.

El nodo central se ubicó en el laboratorio #2 de la facultad de ingeniería en ciencias aplicadas, ya que existen puntos de acceso a internet y de energía eléctrica cerca y es un lugar que tiene línea de vista al parqueadero.



Ilustración 64. Instalación del nodo central
Fuente: Autoría.

4.1.3. Dispositivos para la visualización de la información.

La visualización de la cantidad de estacionamientos que se encuentran disponibles se lo puede realizar en cualquier dispositivo que tenga un navegador web y conexión a internet, es por ello que en este caso se utilizó un computador de escritorio. Solo basta con ingresar en el enlace que se muestra a continuación.

<https://app.ubidots.com/ubi/public/getdashboard/page/2EvM2IJ7T6N1KG0FZZzh-DOr214>

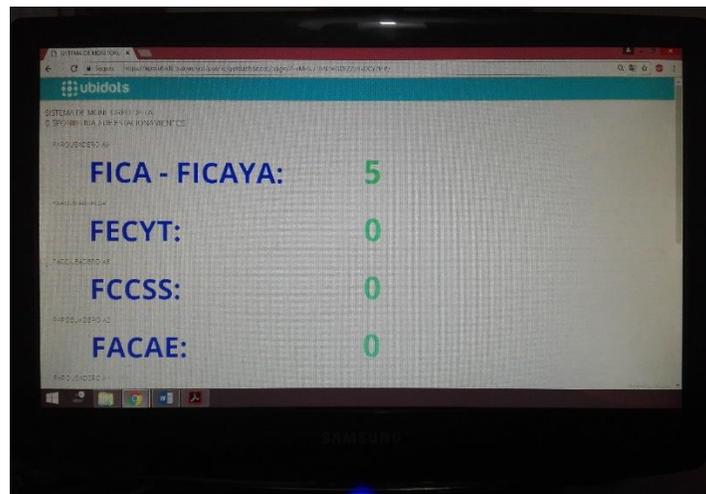
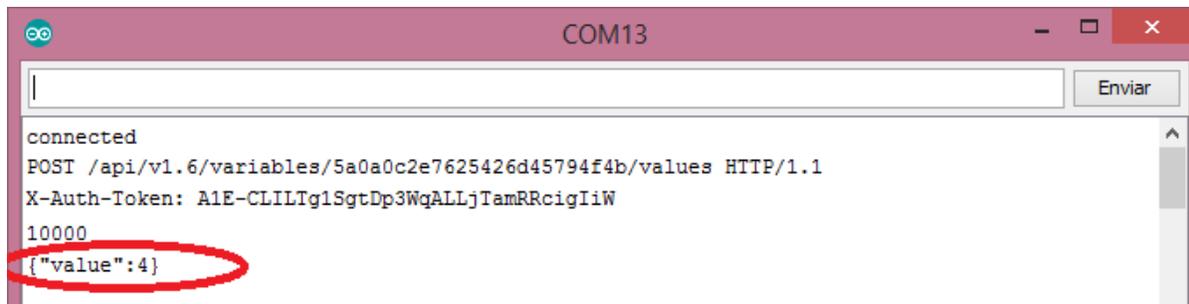


Ilustración 65. Interfaz de visualización en dispositivos móviles.
Fuente: Autoría.

4.2. Pruebas de funcionamiento del Prototipo.

En esta etapa se procede a realizar las pruebas del prototipo, las mismas que permitirán validar el funcionamiento del sistema. Los nodos sensores estarán funcionando al mismo tiempo con el objetivo de detectar la presencia o ausencia del vehículo en el estacionamiento, para luego enviar ese dato de manera inalámbrica al nodo central, el mismo que se encargara de subirlo a la nube, por lo cual se espera visualizar en tiempo real la cantidad de estacionamientos que se

error en la transmisión inalámbrica, pero si el nodo sensor no detecta el automóvil entonces no cambiara de estado y por lo tanto no realizara ninguna transmisión al nodo central.



```

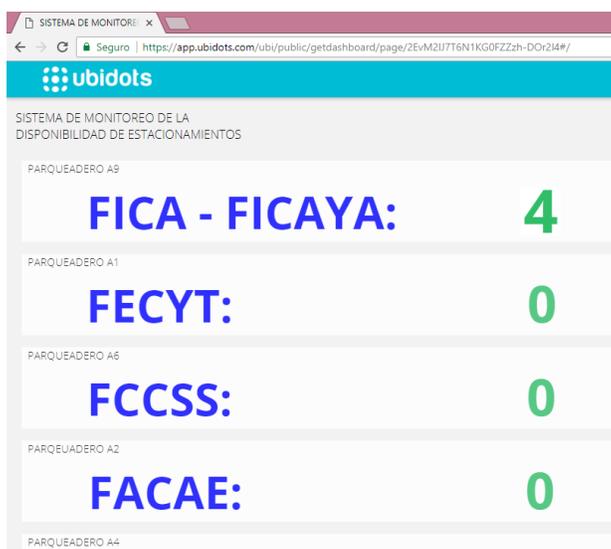
connected
POST /api/v1.6/variables/5a0a0c2e7625426d45794f4b/values HTTP/1.1
X-Auth-Token: A1E-CLILTg1SgtDp3WqALLjTamRRcigIiW
10000
{"value":4}

```

Ilustración 67. Valor que envía el nodo central a Ubidots.
Fuente: Autoría.

4.2.3. Pruebas de la plataforma Ubidots.

Y por último se verifico que los datos del nodo central sean enviados hacia la plataforma en la nube. Para ello se comprobó que Ubidots reconozca al nodo central como conectado y que el valor del nodo central sea el mismo que se muestra en Ubidots. Tal como se observa en la *Ilustración 68*.



PARQUEADERO	Nombre	Valor
PARQUEADERO A9	FICA - FICAYA:	4
PARQUEADERO A1	FECYT:	0
PARQUEADERO A6	FCCSS:	0
PARQUEADERO A2	FACAE:	0
PARQUEADERO A4		

Ilustración 68. Numero de estacionamientos disponibles en Ubidots.
Fuente: Autoría.

A continuación se presenta las pruebas realizadas conforme se las describió anteriormente, con el objetivo de comprobar que las diferentes partes por las cuales está conformado el sistema funcionen correctamente.

4.2.4. Pruebas desarrolladas en el día.

Las pruebas fueron realizadas tal como se lo describe anteriormente, dichas pruebas fueron realizadas durante el día con el objetivo de verificar si la luz solar afecta a los sensores en el proceso de detección de los vehículos. Se realizaron quince pruebas por cada nodo sensor tal como se puede apreciar en la *Tabla 36*.

Tabla 36. Pruebas del prototipo del sistema realizadas en el día.

Nodo Sensor	Pruebas realizadas	Detecciones exitosas	Transmisiones exitosas	Datos subidos en tiempo real
Nodo #1	15	15	15	15
Nodo #2	15	14	13	13
Nodo #3	15	15	14	14
Nodo #4	15	14	13	13
Nodo #5	15	14	14	14
TOTAL	75	72	69	69

Fuente: Autoría

4.2.5. Pruebas desarrolladas en la noche.

De igual manera se realizaron pruebas durante la noche para comprobar si la ausencia de luz afecta la detección de los vehículos. El número de pruebas realizadas fue igual a las realizadas durante el día y los datos obtenidos se lo puede observar en la *Tabla 37*.

Tabla 37. Pruebas del prototipo del sistema realizadas en la noche.

Nodo Sensor	Pruebas realizadas	Detecciones exitosas	Transmisiones exitosas	Datos subidos en tiempo real
Nodo #1	15	15	15	15
Nodo #2	15	14	14	14
Nodo #3	15	15	13	13
Nodo #4	15	15	14	14
Nodo #5	15	14	14	14
TOTAL	75	73	70	70

Fuente: Autoría

4.2.6. Resultados de las pruebas.

El haber realizado pruebas en el día y en la noche, permitió comprobar que la luz solar y la ausencia de luz no interfieren de ninguna manera en el funcionamiento del sistema. En total se realizaron 150 pruebas con los cinco sensores, de los cuales se obtuvo un error del 3,33% en la detección de los vehículos, un error del 7,34% en la transmisión de datos de manera inalámbrica y un error del 7,34% en la visualización de los datos en tiempo real. Con cual se obtiene que el sistema tiene un porcentaje de éxito promedio de 93,99%.

Tabla 38. Resultados de las pruebas.

Pruebas	Número de Pruebas	% Detecciones exitosas	% Transmisiones exitosas	% Datos subidos en tiempo real
Prueba 1				
Realizada en el día	75	96%	92%	92%
Prueba 2	75	97.33%	93.33	93.33

Realizada en la noche				
TOTAL	150	96.66%	92.66%	92.66%

Fuente: Autoría

4.3. Costos del sistema.

A continuación se presenta el costo de cada uno de los factores involucrados en el desarrollo del sistema de monitoreo de la disponibilidad de estacionamientos.

El costo se refiere a la inversión económica que se debe realizar para implementar el sistema, se toman en cuenta todos los factores que intervienen tales como: hardware (nodos sensores - nodos centrales - pantallas led), software, infraestructura y trabajo de ingeniería.

4.3.1. Costos de hardware del nodo sensor.

Se refiere al costo de los componentes que conforman el nodo sensor, el total de la *Tabla 39*, es el costo de un nodo sensor.

Tabla 39. Costos de los elementos utilizados en el nodo sensor.

Descripción	Cantidad	Precio Unitario	Precio Total
Arduino Uno	1	\$16.90	\$16.90
Shield XBee	1	\$8.00	\$8.00
XBee S2C	1	\$45.00	\$45.00
Sensor Sharp 2Y0A21	1	\$10.50	\$10.50
Elaboración del Shield	1	\$5.00	\$5.00
Adaptador AC/DC	1	\$7.50	\$7.50

Caja de protección del nodo sensor	1	\$4.00	\$4.00
Otros elementos	1	\$3.00	\$3.00
TOTAL			\$99.90

Fuente: Autoría

4.3.2. Costos de hardware del nodo central.

Se trata del costo de los componentes que forman parte del nodo central, y el total de la *Tabla 40* representa el costo de un nodo central.

Tabla 40. Costos de los elementos utilizados en el nodo central.

Descripción	Cantidad	Precio Unitario	Precio Total
Arduino Mega 2560	1	\$19.00	\$19.00
Shield XBee	1	\$8.00	\$8.00
XBee S2C	1	\$45.00	\$45.00
Shield Ethernet	1	\$16.00	\$16.00
Adaptador AC/DC	1	\$7.50	\$7.50
Caja de protección del nodo central	1	\$10.95	\$10.95
Otros elementos	1	\$5.00	\$5.00
TOTAL			\$111.45

Fuente: Autoría

4.3.3. Costos de la pantalla led.

En este caso el total que se muestra en la *Tabla 41* representa el costo de una de las pantallas led que se utilizaran para presentar la información.

Tabla 41. Costo de la pantalla led.

Descripción	Cantidad	Precio Unitario	Precio Total
Pantalla led Publicity PC-MINI	1	\$4,000.0	\$4,000.0
TOTAL			\$4,000.0

Fuente: Autoría

4.3.4. Costos de infraestructura.

Entre los costos de infraestructura se encuentran la instalación de los nodos sensores, nodos centrales y pantallas led; Los cables de red utilizados para los nodos centrales; Y el consumo de energía eléctrica y servicio de internet, mismos que se los excluye debido a que al estar conectados a la infraestructura de la Universidad Técnica del Norte son financiados por dicha institución.

Tabla 42. Costos de infraestructura.

Descripción	Cantidad	Precio Unitario	Precio Total
Instalación del nodo sensor	436	\$30.00	\$13,080.0
Instalación del nodo central	9	\$30.00	\$270.00
Instalación de la pantalla led	2	\$50.00	\$100.00
Cables y otros materiales	9	\$20.00	\$180.00
Consumo de energía eléctrica	1	\$0.00	\$0.00
Acceso a internet	1	\$0.00	\$0.00
TOTAL			\$13,630.0

Fuente: Autoría

4.3.5. Costos de ingeniería.

Se considera como costo de ingeniería, los honorarios de la persona que se encarga del diseño, estudio de campo y documentación, este valor se da acorde a la dificultad de acceso y condiciones climáticas del sitio en el que se ejecutará el proyecto.

Tabla 43. Costos de ingeniería.

Descripción	Cantidad	Precio Unitario	Precio Total
Estudio de campo, diseño y revisión de la infraestructura.	1	\$700.00	\$700.00
Documentación	1	\$250.00	\$250.00
TOTAL			\$950.00

Fuente: Autoría

4.3.6. Costos de software.

En este caso el costo del software utilizado es \$0.00, ya que estos programas se basan en arquitecturas open source de libre acceso.

Tabla 44. Costos de software.

Descripción	Cantidad	Precio Unitario	Precio Total
IDE de Arduino	1	\$0.00	\$0.00
Ubidots	1	\$0.00	\$0.00
TOTAL			\$0.00

Fuente: Autoría

4.3.7. Costos de implementación del sistema.

A continuación en la *Tabla 45* se presenta un resumen del costo de inversión total del proyecto, el cual se obtiene de la cantidad de dispositivos a implementarse y de la suma de los costos que intervienen en la implementación.

Tabla 45. Costos de la implementación del sistema.

Descripción	Cantidad	Precio Unitario	Precio Total
Nodos sensores	436	\$99.90	\$43,556.4
Nodos centrales	9	\$111.45	\$1,003.05
Pantallas Led	2	\$4,000.0	\$8,000.0
Costo de Infraestructura	1	\$13,630.0	\$13,630.0
Costo de Ingeniería	1	\$950.00	\$950.00
Costo de software	1	\$0.00	\$0.00
		TOTAL	\$67,139.45

Fuente: Autoría

Conclusiones y Recomendaciones.

Conclusiones.

Al finalizar este proyecto de titulación, usando una red de sensores inalámbricos se ha logrado cumplir con los objetivos propuestos y se ha desarrollado un prototipo que se centra en la detección de la presencia o ausencia del vehículo en un estacionamiento, permitiendo informar la cantidad de estacionamientos libres que existen en el parqueadero, mediante una plataforma de almacenamiento en la nube lo cual se logra teniendo acceso a internet.

Se realizó una investigación bibliográfica acerca de temas relacionados con las redes de sensores inalámbricos y los sistemas de parqueo, lo cual permitió desarrollar el diseño de este sistema de una mejor manera.

Se desarrolló un prototipo del sistema para determinar si el diseño propuesto funciona adecuadamente, para ello se realizó una fase de pruebas en el parqueadero de la Universidad Técnica del Norte, comprobando que la detección del vehículo al momento que se estaciona es exitosa ya que los datos si pueden ser vistos en la plataforma Ubidots.

La fase pruebas fue realizada tanto en el día como en la noche, permitiendo así comprobar que la luz solar y la ausencia de luz no interfieren de ninguna manera en el funcionamiento del sistema.

Los sistemas de parqueo vehiculares están teniendo en la actualidad un gran impacto, ya que se encuentran entre los principales sistemas de las Smart cities.

Recomendaciones.

El sistema de monitoreo de los estacionamientos debe ser ubicado en lugares donde el conductor pueda ver con claridad en que parqueadero existen plazas disponibles y no interfiera de ninguna manera al tránsito vehicular o al vehículo al momento de estacionarse.

Establecer los requerimientos de una manera ordenada y clara ayuda a realizar una elección adecuada de los componentes que conformaran el sistema, para lo cual se recomienda utilizar alguna norma como guía para establecer los requerimientos necesarios para el desarrollo del sistema.

Realizar una elección adecuada de cada uno de los elementos que conforman el sistema, hace que las funcionalidades sean las adecuadas y las requeridas por el sistema para que se logre cumplir con los objetivos trazados.

Al momento de elegir los componentes que formaran parte del sistema es importante tomar en cuenta que estos componentes sean compatibles entre sí para evitar problemas al momento de acoplar todo el sistema.

Los nodos sensores al estar ubicados en un ambiente externo deben ser protegidos con algún case que soporte filtraciones de agua para evitar que los componentes electrónicos se mojen y por ende se dañen.

Referencias.

Ahmed, K., & Gregory, M. (12 de Junio de 2012). *Journal of Sensor and Actuator Networks*.

Obtenido de <https://ai2-s2-pdfs.s3.amazonaws.com/d4e1/49a6044ed7aee370e2e02645f37afc191dee.pdf>

ANT. (2017). *Agencia Nacional de Tránsito*. Obtenido de

<http://www.ant.gob.ec/index.php/noticias/>

Arduino. (2017). *ARDUINO*. Obtenido de <https://www.arduino.cc/>

Capella Hernández, J. V. (2011). *Redes inalámbricas de sensores: Una nueva arquitectura eficiente y robusta basada en jerarquía dinámica de grupos*. Valencia: UPV.

Computer Hope. (s.f.). *Computer Hope*. Obtenido de

<http://www.computerhope.com/jargon/w/webpage.htm>

Delgado, P. (06 de Junio de 2014). *El Grupo Informatico*. Obtenido de

<https://www.elgrupoinformatico.com/repaso-las-tecnologias-almacenamiento-actuales-t19307.html>

Digi International. (2017). *DIGI*. Obtenido de <https://www.digi.com/>

Electronilab. (s.f.). *ELECTRONILAB*. Obtenido de Ingeniería y Diseño Electronico:

<https://electronilab.co/>

EsarkFun Electronics. (2015). *SPARKFUN*. Obtenido de <https://www.sparkfun.com/products/13678>

Exosite. (s.f.). *EXOSITE*. Obtenido de <https://exosite.com/>

Fahmy, H. M. (2016). *Wireless Sensor Network*. Singapore: Springer.

Farahani, S. (2008). *ZigBee Wireless Networks and Transceivers*. ELSEVIER.

Farej, Z. K., & Abdul-Hameed, A. M. (Agosto de 2015). *International Journal of Computer Applications*. Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.5839&rep=rep1&type=pdf>

Flores Carbajal, E. E. (30 de Octubre de 2012). *Redes de Sensores Inalámbricas Aplicado a la Medicina*. Obtenido de <https://repositorio.unican.es/xmlui/bitstream/handle/10902/1288/349251.pdf?sequence=1>

Garbarino, J. (2011). Universidad de Buenos Aires.

Green, N. (01 de Mayo de 2015). *KrollOntrack*. Obtenido de <http://blog.ontrackdatarecovery.es/tecnologias-en-sistemas-de-almacenamiento/>

ISTQB Exam Certification. (s.f.). *ISTQB EXAM CERTIFICATION*. Obtenido de <http://istqbexamcertification.com/>

IWEB - INTERNAP. (s.f.). *INTERNAP*. Obtenido de <https://iweb.com/es/almacenamiento-redes>

Libelium. (2016). *Plug & Sense! Smart Parking - Technical Guide*. Libelium Comunicaciones Distribuidas S.L.

Libelium. (s.f.). *LIBELIUM*. Obtenido de <http://www.libelium.com/>

Microsoft Azure. (s.f.). *Microsoft Azure*. Obtenido de <https://azure.microsoft.com/>

Muñoz, D. H. (Julio de 2013). *Universidad Politecnica de Madrid*. Obtenido de http://oa.upm.es/21414/1/PFC_DAVID_HERRADOR_MU%C3%91OZ.pdf

ParkHelp. (s.f.). *ParkHelp The new parking experience*. Obtenido de <http://www.parkhelp.com/why-parkhelp/>

PeopleWeb. (2014). *People WEB*. Obtenido de <http://www.peopleweb.com.ec/index.php/gestor-de-parqueaderos>

Rotulos Electronicos. (2017). *rotuloselectronicos.NET*. Obtenido de <http://www.rotuloselectronicos.net>

Rouse, M. (Junio de 2006). *TechTarget*. Obtenido de <http://searchdatacenter.techtarget.com/definition/sensor-network>

Rouse, M. (Diciembre de 2013). *TechTarget*. Obtenido de <http://whatis.techtarget.com/definition/mobile-app>

Sip. (2015). *SIP*. Obtenido de <http://sip.ec/>

Sistembel. (05 de Marzo de 2015). *SISTEMBEL S.* Obtenido de <https://www.sistembel.com/>

Smart Parking Limited. (s.f.). *Smart Parking.* Obtenido de <http://www.smartparking.com/>

Socle Technology Corp. (s.f.). *Socle Technology Corp.* Obtenido de http://www.socle-tech.com/doc/IC%20Channel%20Product/Sensors/Distance%20Measuring%20Sensor/Analog%20Output/gp2y0a21yk_e.pdf

Techopedia. (s.f.). *techopediaA.* Obtenido de <https://www.techopedia.com/>

Ubidots. (2017). *Ubidots.* Obtenido de <https://ubidots.com/>

Universidad Técnica del Norte. (2008). *UTN.* Obtenido de http://www.utn.edu.ec/web/uniportal/?page_id=2008

Xively. (2007). *xively by LogMeIn.* Obtenido de <https://www.xively.com/>

Glosario de Términos y Acrónimos.

WSN: Wireless Sensor Network (Redes de sensores inalámbricos). Realiza la adquisición y tratamiento de datos con múltiples aplicaciones en distintos campos tales como entornos industriales, domótica, entornos militares, detección ambiental.

SMS: Short Message Service (Servicio de mensajes simples o servicio de mensajes cortos). Es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos, conocidos como mensajes de texto entre teléfonos móviles.

LAN: Local Area Network (Red de área local). Una LAN es una red que conecta los ordenadores en un área relativamente pequeña y predeterminada, como una habitación, un edificio, o un conjunto de edificios.

WLAN: Wireless Local Area Network (Red de área local inalámbrica). Como la denominación lo señala, una WLAN es una red de tipo local cuyos equipos no necesitan estar vinculados a través de cables para conectarse.

IEEE: Institute of Electrical and Electronics Engineers (Instituto de ingeniería eléctrica y electrónica). Es una asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas. Es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías.

Wifi: Wireless Fidelity (Fidelidad inalámbrica). Es una tecnología de comunicación inalámbrica que permite conectar a internet equipos electrónicos, como computadoras, tablets,

smartphones o celulares, etc., mediante el uso de radiofrecuencias para la transmisión de la información.

PCB: Printed Circuit Board (Placa de circuito impreso). Es una superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos.

SPI: Serial Peripheral Interface (Interfaz periférica serial). Es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. Es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica).

IoT: Internet of things (Internet de las cosas). Es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet. Alternativamente, Internet de las cosas es el punto en el tiempo en el que se conectarían a internet más “cosas u objetos” que personas.

PaaS: Platform as a Service (Plataforma como servicio). Es un ambiente de cómputo en la nube al que se obtiene acceso según sea necesario a través de una red de un proveedor de servicios.

UART: Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal). Es el dispositivo que controla los puertos y dispositivos serie, mediante este se puede realizar la comunicación serial entre dispositivos sea PC con alguna placa en particular. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

FFD: Dispositivo de funcionalidad completa, se lo conoce como nodo activo. Puede funcionar en cualquier topología, es capaz de ser un comunicador de red y puede comunicarse con cualquier otro dispositivo.

RFD: Dispositivo de funcionalidad reducida, se lo conoce como nodo pasivo. Se limita a funcionar en la topología estrella, no puede ser un coordinador de red.

TX: En telecomunicaciones se refiere a la transmisión o emisión. Es el proceso de envío y propagación de una señal de información analógica o digital sobre un medio de transmisión físico punto-a-punto o punto-a-multipunto, ya sea por cable, fibra óptica o inalámbricamente.

RX: En telecomunicaciones se refiere a la recepción. Es el proceso de recibir una señal de información analógica o digital sobre un medio de recepción físico punto-a-punto o punto-a-multipunto, ya sea por cable, fibra óptica o inalámbricamente.

Anexos.

ANEXO 1: Formato de encuesta aplicada a los usuarios de los parqueaderos de la UTN.

USO DE LOS PARQUEADEROS DE LA UTN

Encuesta dirigida a las personas que hacen uso del parqueadero de la Universidad Técnica del Norte.

Objetivo: Obtener información de interés directamente de los usuarios del parqueadero, acerca de su uso, el tiempo que les toma estacionar el vehículo y si consideran importante disminuir dicho tiempo y facilitar la búsqueda de estacionamientos mediante el uso de la tecnología.

Indicaciones: Lea detenidamente las preguntas y seleccione la opción que corresponda según su caso.

1. ¿Cuál es el rol que usted desempeña?

Estudiante ()

Docente ()

Administrativo ()

2. ¿Qué tipo de parqueadero utiliza usted?

Autos ()

Motos ()

3. ¿En qué horarios hace uso del parqueadero?

Mañana ()

Tarde ()

Noche ()

4. ¿Según su experiencia en que horarios existe más afluencia vehicular y es difícil encontrar un estacionamiento libre?

Mañana ()

- Tarde ()
 Noche ()
5. ¿Le ha causado inconvenientes estacionar su vehículo en el parqueadero de la UTN?
 Si ()
 No ()
6. ¿Qué hace cuando ingresa al parqueadero y no encuentra un estacionamiento disponible para su vehículo?
 Espera hasta encontrar un estacionamiento libre. ()
 Abandona el campus de la UTN y se estaciona en la calle. ()
 Se estaciona en cualquier lugar, así no sea un estacionamiento marcado. ()
7. ¿Cuál es el tiempo promedio que tarda en estacionar su vehículo desde que entra hasta que se estaciona?
 De 0 a 2,5 minutos ()
 De 2,5 a 5 minutos ()
 De 5 a 10 minutos ()
 Más de 10 minutos ()
8. ¿Es importante para usted disminuir el tiempo que tarda en estacionarse?
 Si ()
 No ()
9. ¿Cree que es necesario que la UTN cuente con un sistema que permita a los usuarios conocer la existencia de estacionamientos libres?
 SI () NO ()
10. ¿Cree que si la UTN cuenta con un sistema que ayude a conocer la existencia de estacionamientos libres, usted podría estacionarse más rápido?
 SI () NO ()
11. ¿Cómo le gustaría conocer la disponibilidad de estacionamientos libres?
 Pantallas informativas en las puertas de acceso vehicular ()
 Página WEB ()
 Aplicación Móvil ()
 Otro ()
 Porque:

ANEXO 2: Preguntas realizadas en la entrevista al jefe de seguridad de la UTN.

USO DE LOS PARQUEADEROS DE LA UTN

Entrevista sobre el uso de los parqueaderos dirigida al jefe de seguridad de la Universidad Técnica del Norte.

Objetivo: Obtener información de interés directamente del jefe del departamento de seguridad de la UTN, acerca del uso, distribución y control que se tiene actualmente en los parqueaderos.

1. ¿Qué horario es considerado hora pico en los parqueaderos?

.....

2. ¿Cómo les identifican a cada una de las puertas de acceso vehicular?

.....

3. ¿Cuál son los horarios de apertura y cierre de los accesos vehiculares?

.....

4. ¿Por qué el acceso vehicular norte suele pasar cerrada?

.....

5. ¿Cuáles son los principales problemas que se tiene en los parqueaderos en las horas pico?

.....

6. ¿Se tiene estacionamientos reservados para las autoridades de la UTN y de ser así, cuantos estacionamientos son?

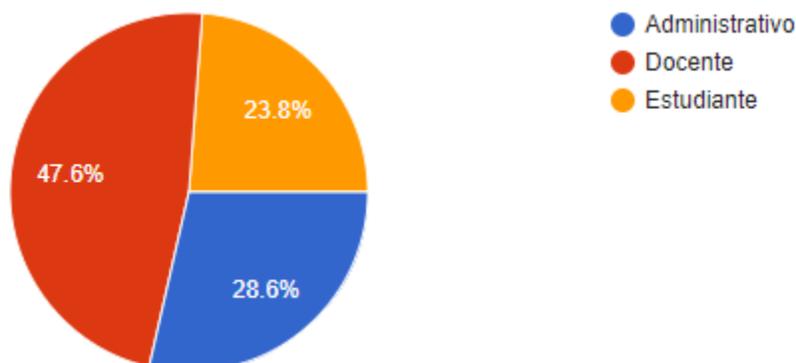
.....

7. ¿Cuántos estacionamientos son reservados para los vehículos oficiales de la UTN?
.....
.....
8. ¿Cuál es la principal razón para que los vehículos de la institución se encuentren en un parqueadero concurrido y no se los estacione donde no haya mucha afluencia vehicular?
.....
.....
9. ¿Por qué los buses se estacionan en un espacio verde y no hacen uso de un parqueadero?
.....
.....
10. ¿Cree que es importante tener estacionamientos reservados para visitas?
SI () NO ()
11. ¿Cuál es el control actual de acceso a los parqueaderos?
.....
.....
12. ¿Cuántos dispositivos electrónicos para el acceso de vehículos se encuentran entregados?
.....
.....

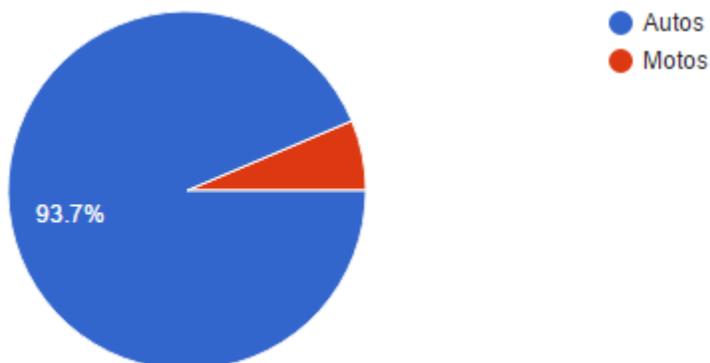
ANEXO 3: Tabulación de las encuestas aplicadas a los usuarios de los parqueaderos de la UTN de los parqueaderos de la UTN.

TABULACIÓN DE LAS ENCUESTAS DEL USO DE LOS PARQUEADEROS DE LA UTN

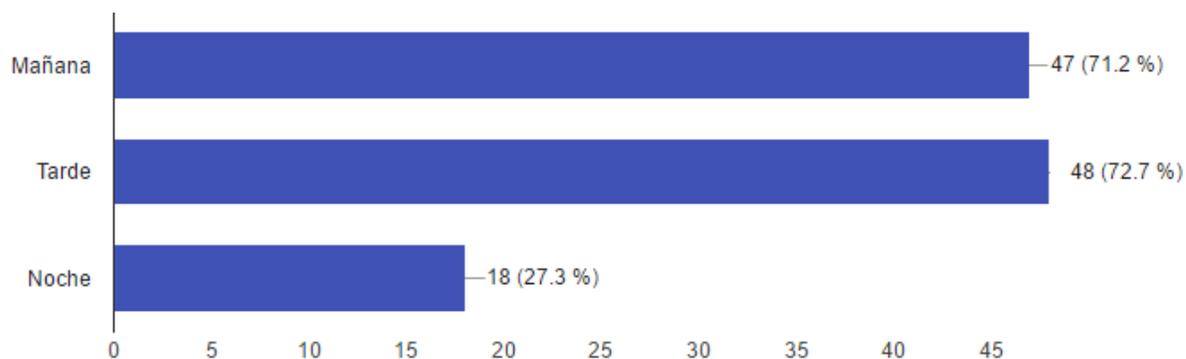
1. ¿Cuál es el rol que usted desempeña?



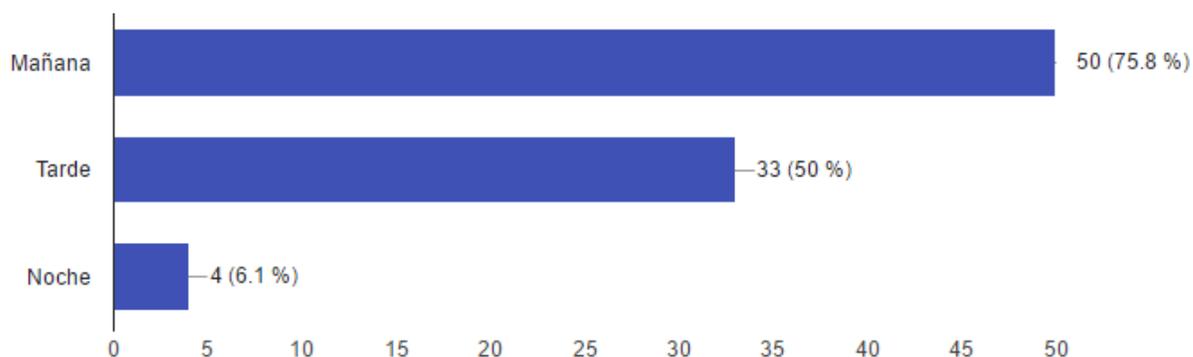
2. ¿Qué tipo de parqueadero utiliza usted?



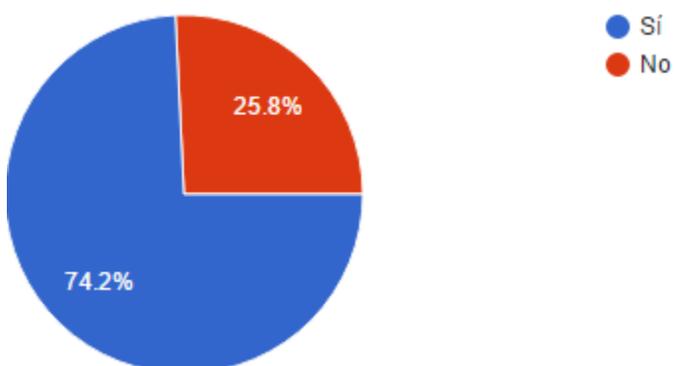
3. ¿En qué horarios hace uso del parqueadero?



4. ¿Según su experiencia en que horarios existe más afluencia vehicular y es difícil encontrar un estacionamiento libre?



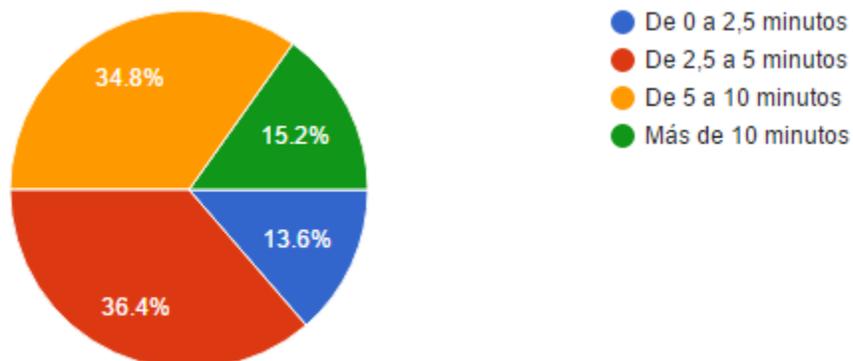
5. ¿Le ha causado inconvenientes estacionar su vehículo en el parqueadero de la UTN?



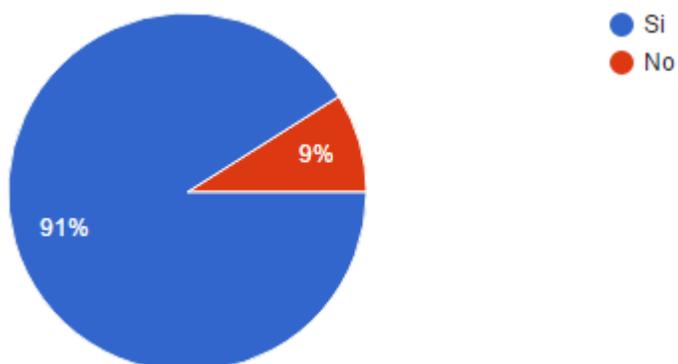
6. ¿Qué hace cuando ingresa al parqueadero y no encuentra un estacionamiento disponible para su vehículo?



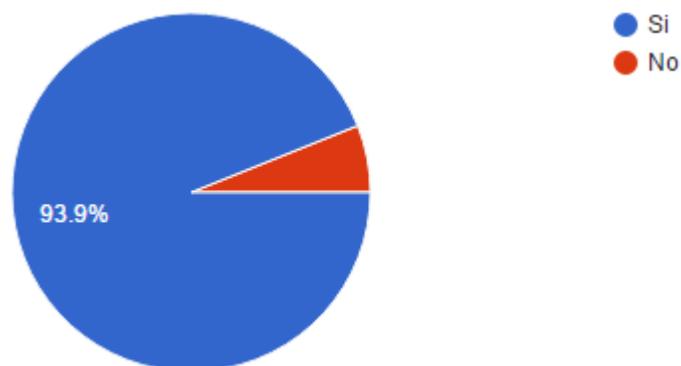
7. ¿Cuál es el tiempo promedio que tarda en estacionar su vehículo desde que entra hasta que se estaciona?



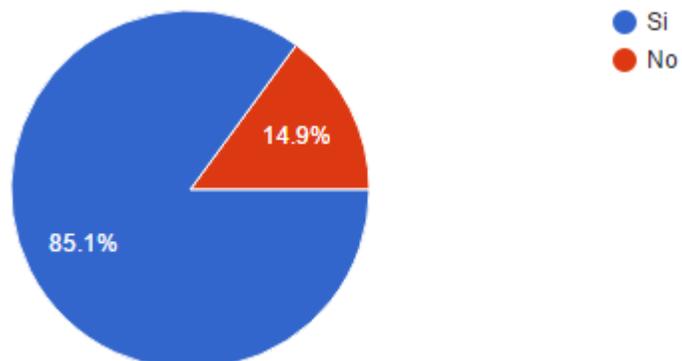
8. ¿Es importante para usted disminuir el tiempo que tarda en estacionarse?



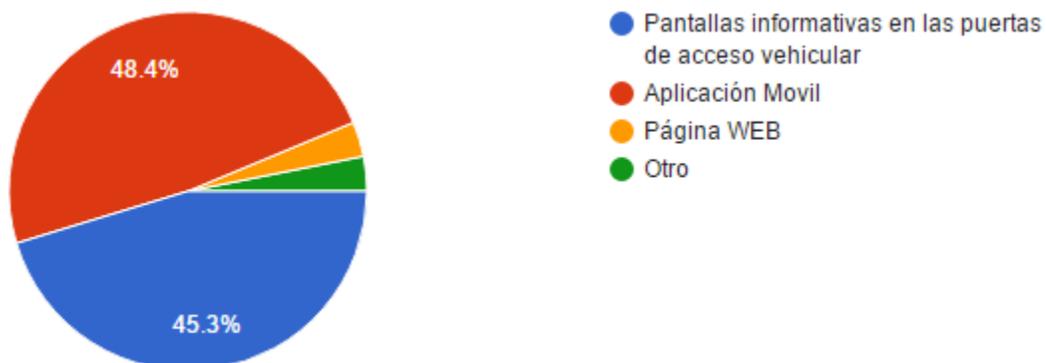
9. ¿Cree que es necesario que la UTN cuente con un sistema que permita a los usuarios conocer la existencia de estacionamientos libres?



10. ¿Cree que si la UTN cuenta con un sistema que ayude a conocer la existencia de estacionamientos libres, usted podría estacionarse más rápido?



11. ¿Cómo le gustaría conocer la disponibilidad de estacionamientos libres?



ANEXO 4: DataSheet Sensor SHARP 2Y0A21.

SHARP

GP2Y0A21YK0F

GP2Y0A21YK0F

Distance Measuring Sensor Unit
 Measuring distance: 10 to 80 cm
 Analog output type

**■Description**

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

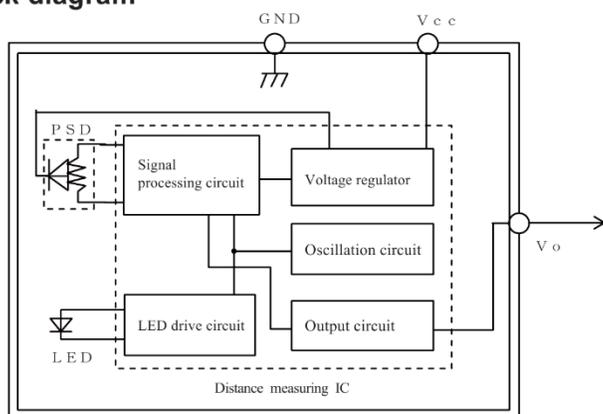
■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

■Applications

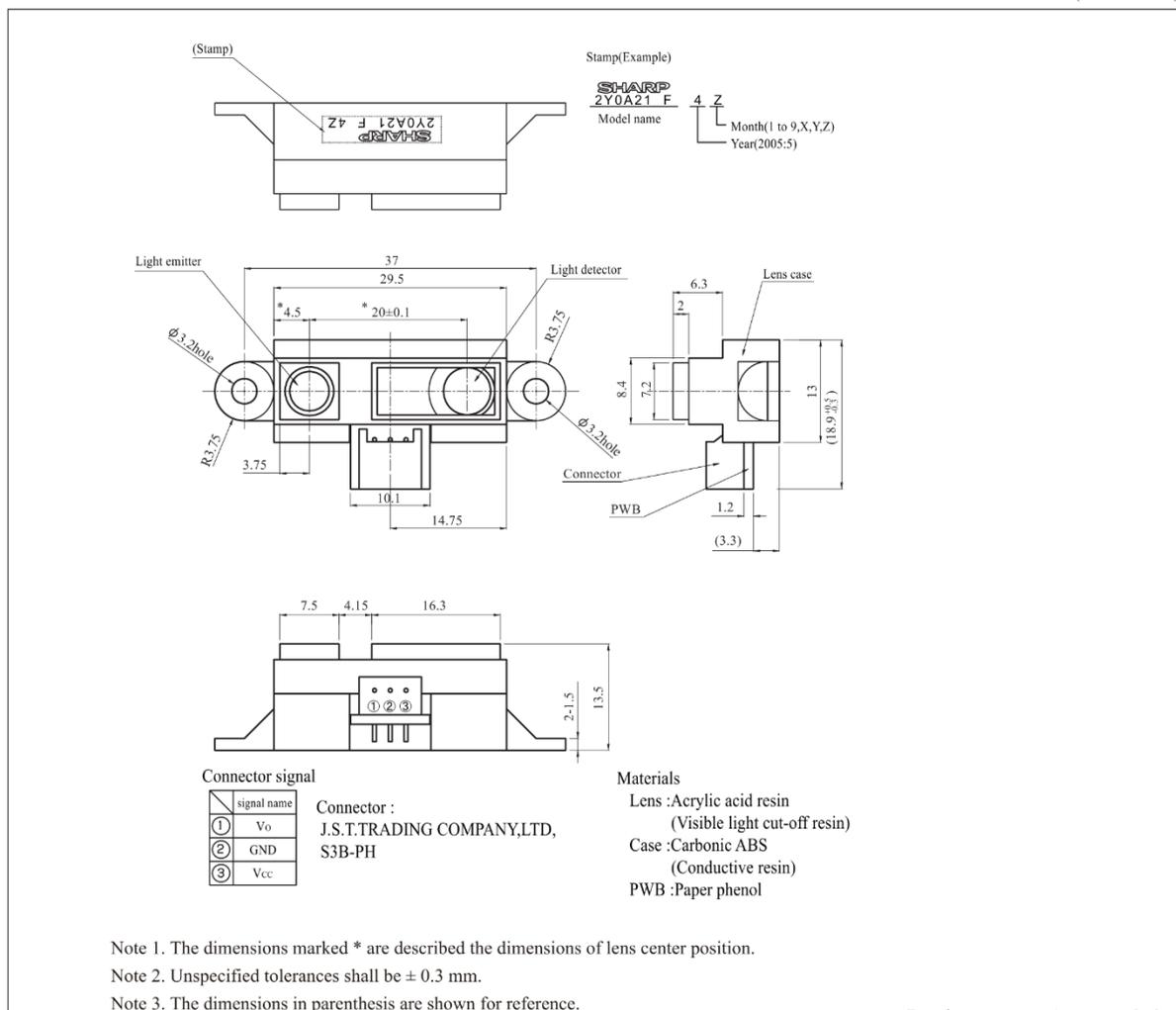
1. Touch-less switch
(Sanitary equipment, Control of illumination, etc.)
2. Robot cleaner
3. Sensor for energy saving
(ATM, Copier, Vending machine)
4. Amusement equipment
(Robot, Arcade game machine)

■ **Block diagram**



■ **Outline Dimensions**

(Unit : mm)



■ Absolute Maximum Ratings (T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	-0.3 to +7	V
Output terminal voltage	V _O	-0.3 to V _{CC} +0.3	V
Operating temperature	T _{opr}	-10 to +60	°C
Storage temperature	T _{stg}	-40 to +70	°C

■ Electro-optical Characteristics (T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I _{CC}	L=80cm (Note 1)	—	30	40	mA
Distance measuring	ΔL	(Note 1)	10	—	80	cm
Output voltage	V _O	L=80cm (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV _O	Output voltage difference between L=10cm and L=80cm (Note 1)	1.65	1.9	2.15	V

* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27•white face, reflectance; 90%)

■ Recommended operating conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	4.5 to 5.5	V

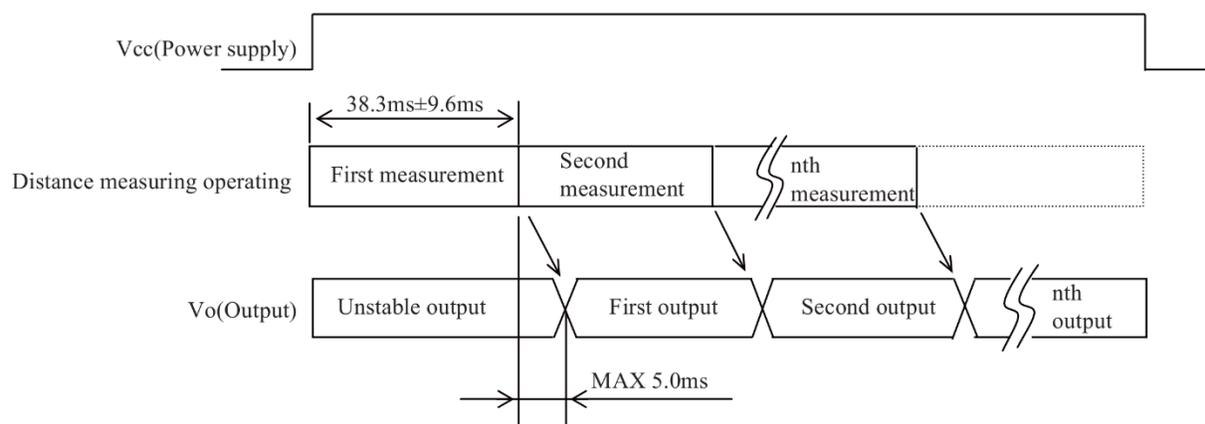
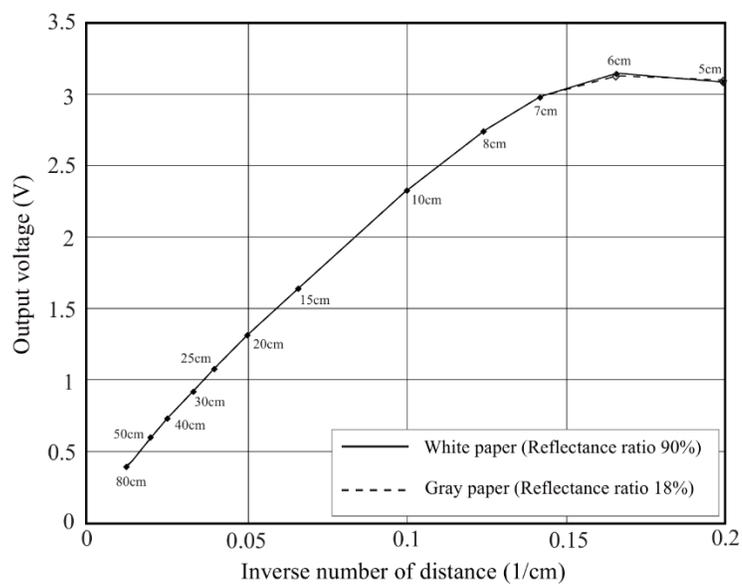
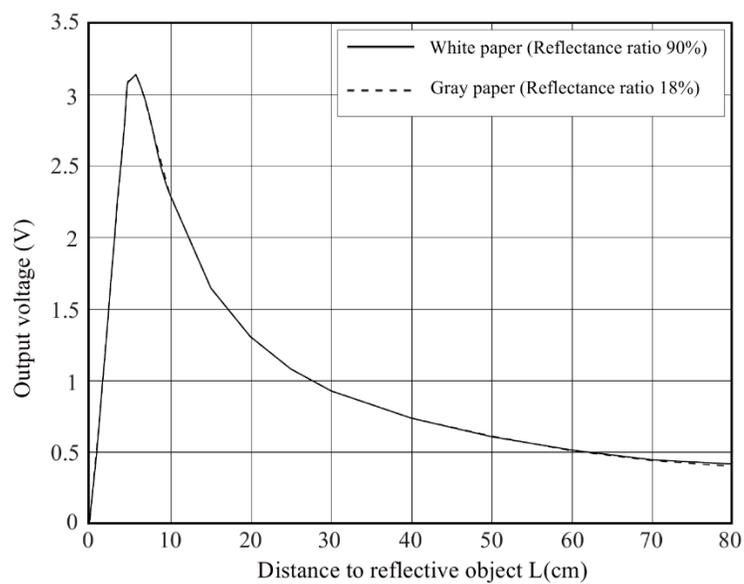
Fig. 1 Timing chart

Fig. 2 Example of distance measuring characteristics(output)



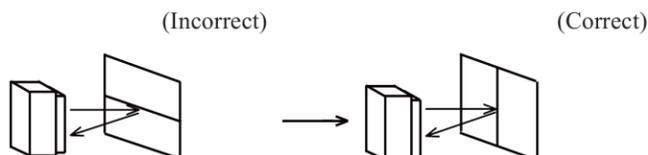
■Notes

●Advice for the optics

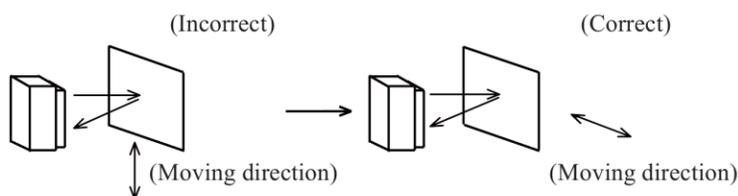
- The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

●Advice for the characteristics

- In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda = 870 \pm 70\text{nm}$), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.
- In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.
- When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.
- Distance to a mirror reflector can not be sometimes measured exactly.
In case of changing the mounting angle of this product, it may measure the distance exactly.
- In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.



- In order to decrease deviation of measuring distance by moving direction of the reflective object, it shall be recommended to set the sensor that the moving direction of the object and the line between emitter center and detector center are vertical.



●Advice for the power supply

- In order to stabilize power supply line, we recommend to insert a by-pass capacitor of $10\mu\text{F}$ or more between Vcc and GND near this product.

●Notes on handling

- There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.

● Presence of ODC etc.

This product shall not contain the following materials.

And they are not used in the production process for this product.

Regulation substances : CFCs, Halon, Carbon tetrachloride, 1.1.1-Trichloroethane (Methylchloroform)

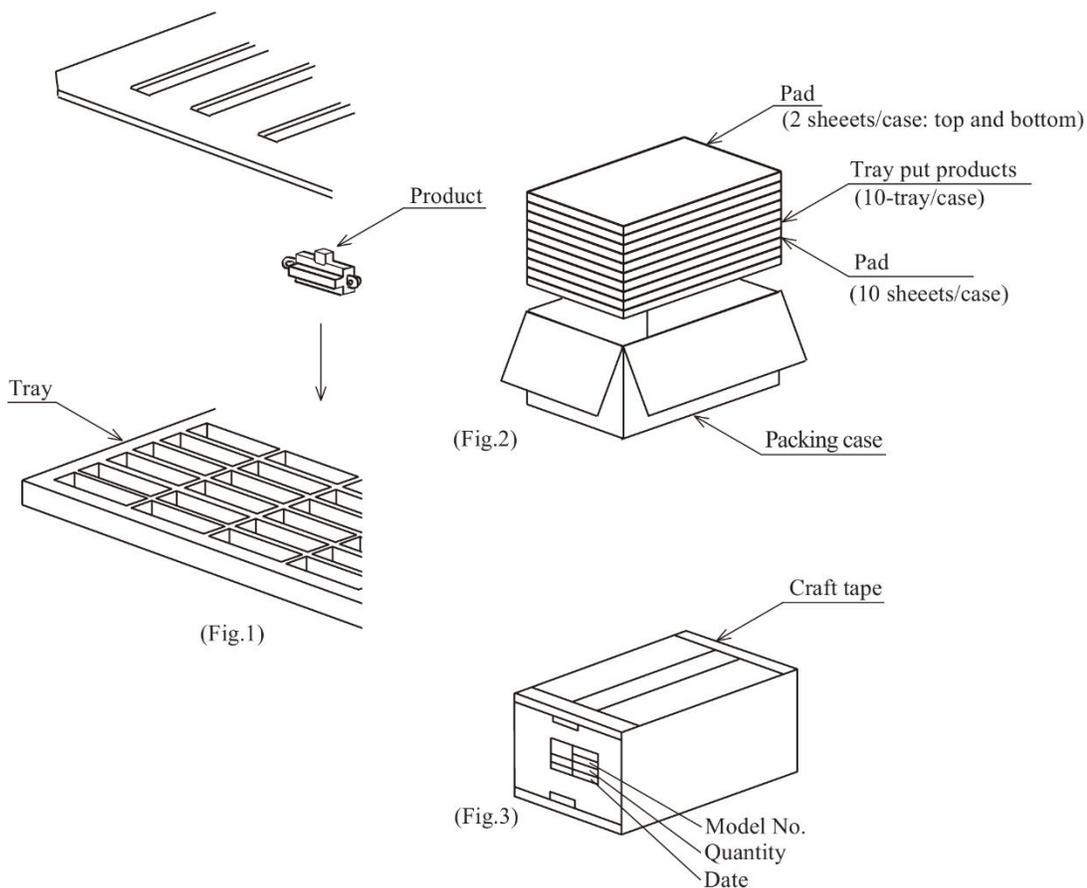
Specific brominated flame retardants such as the PBB and PBDE are not used in this product at all.

This product shall not contain the following materials banned in the RoHS Directive (2002/95/EC).

- Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls (PBB), Polybrominated diphenyl ethers (PBDE).

■ Package specification

Package composition



Packaging method

- 1.Put products of 100pcs. in tray. packing method is showed in the above fig.(Fig.1)
- 2.Put them(10-tray) in the packing box. Put pads on their top and bottom.
And put pads on each trays(Total 10 sheets) (Fig.2).
- 3.Seal the packing box with craft tape.
Print the model No.,quantity,inspection date (1000 pcs./a packing box)(Fig.3).

■ Important Notices

· The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

· Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

· Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:

- Personal computers
- Office automation equipment
- Telecommunication equipment [terminal]
- Test and measurement equipment
- Industrial control
- Audio visual equipment
- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection

with equipment that requires higher reliability such as:

- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- Traffic signals
- Gas leakage sensor breakers
- Alarm equipment
- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

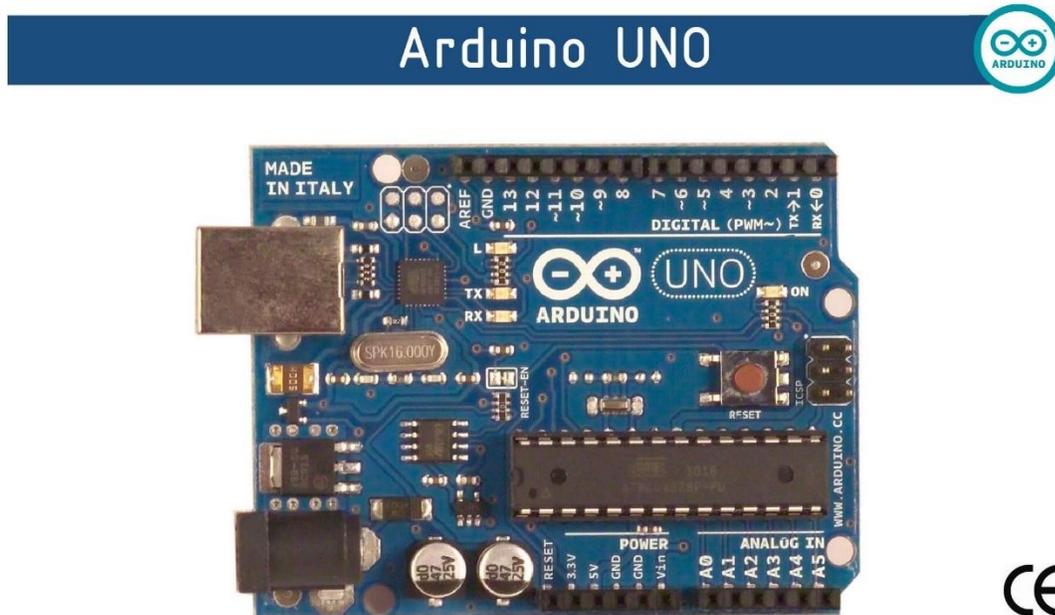
- Space applications
- Telecommunication equipment [trunk lines]
- Nuclear power control equipment
- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

ANEXO 5: DataSheet Arduino UNO.



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7

Technical Specification

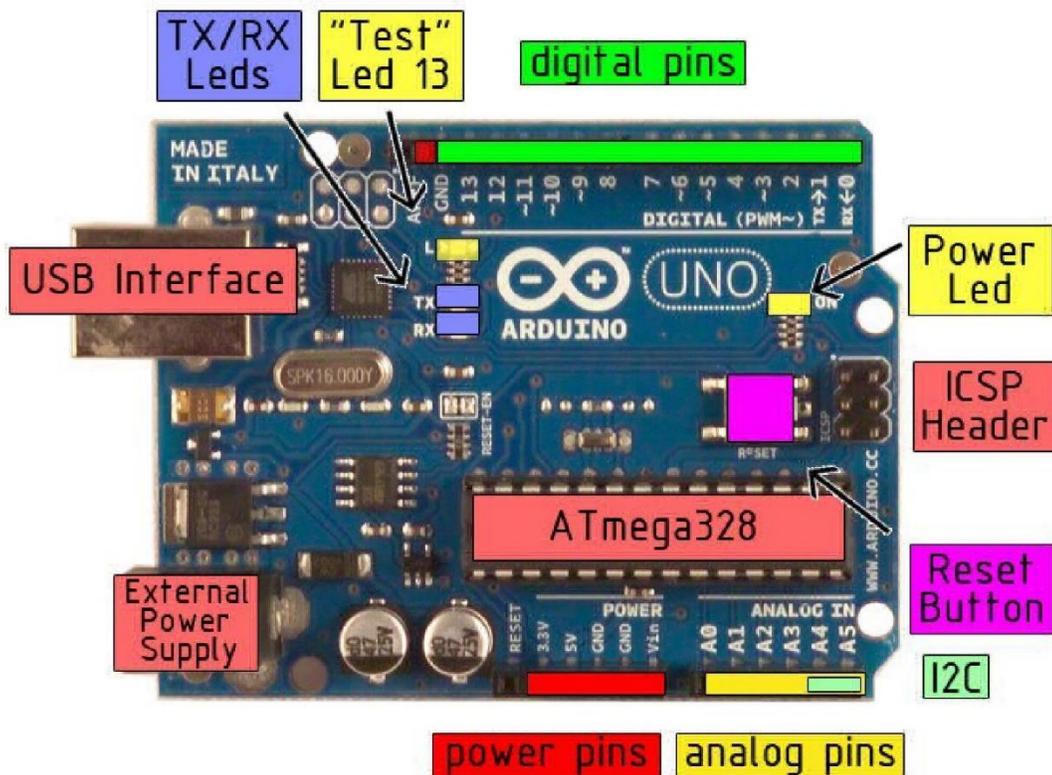


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.

```

Blink | Arduino 0017
File Edit Sketch Tools Help
Blink $
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
  
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

ANEXO 6: DataSheet Arduino Mega2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7

Technical Specification

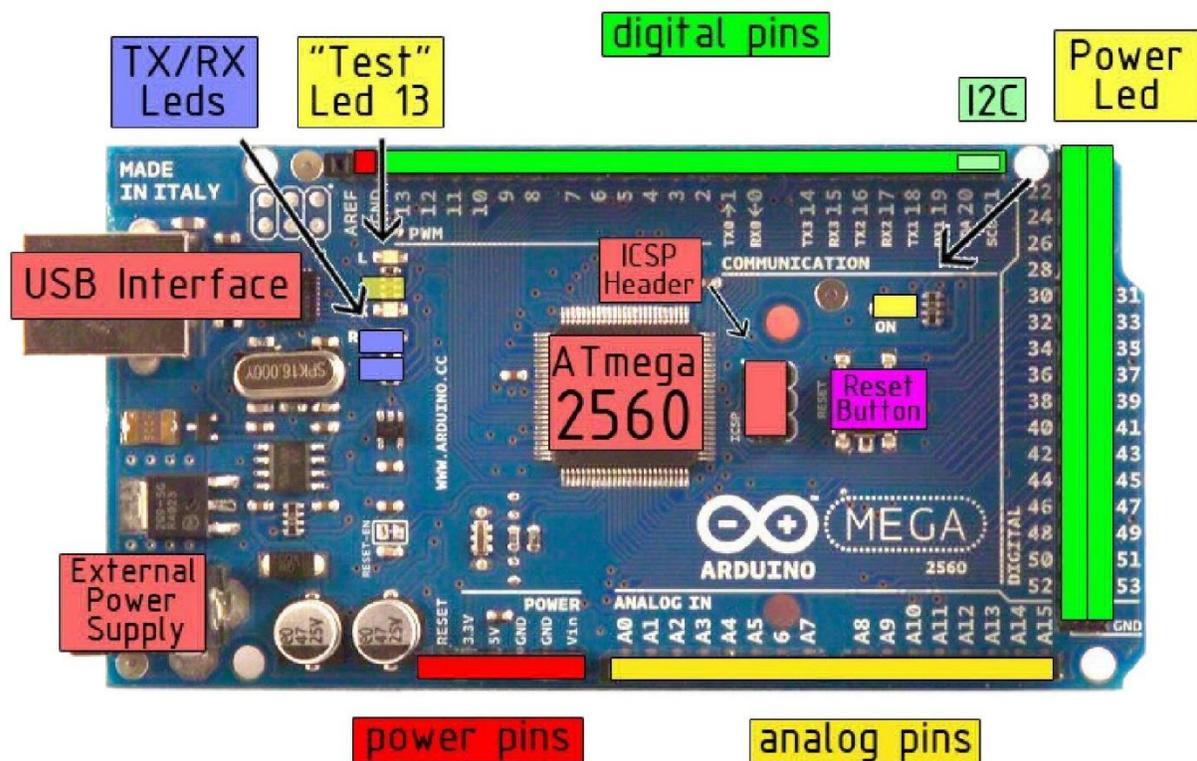


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

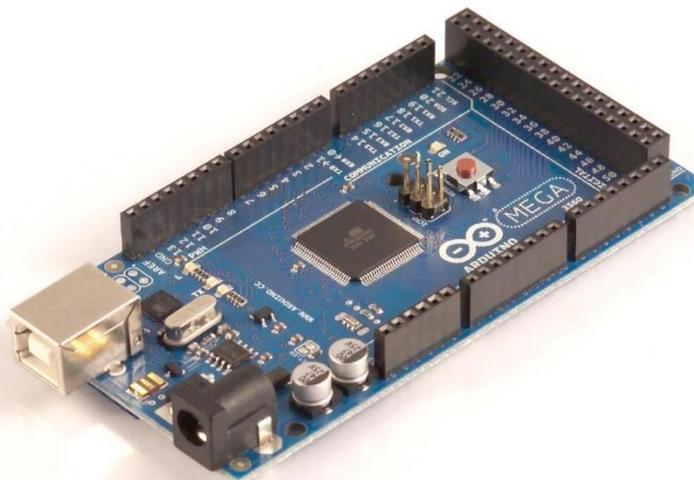
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The Atmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```

Blink | Arduino 0017
File Edit Sketch Tools Help
Blink$
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
  
```



Done compiling.

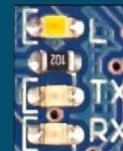
Press Compile button
(to check for errors)



Upload

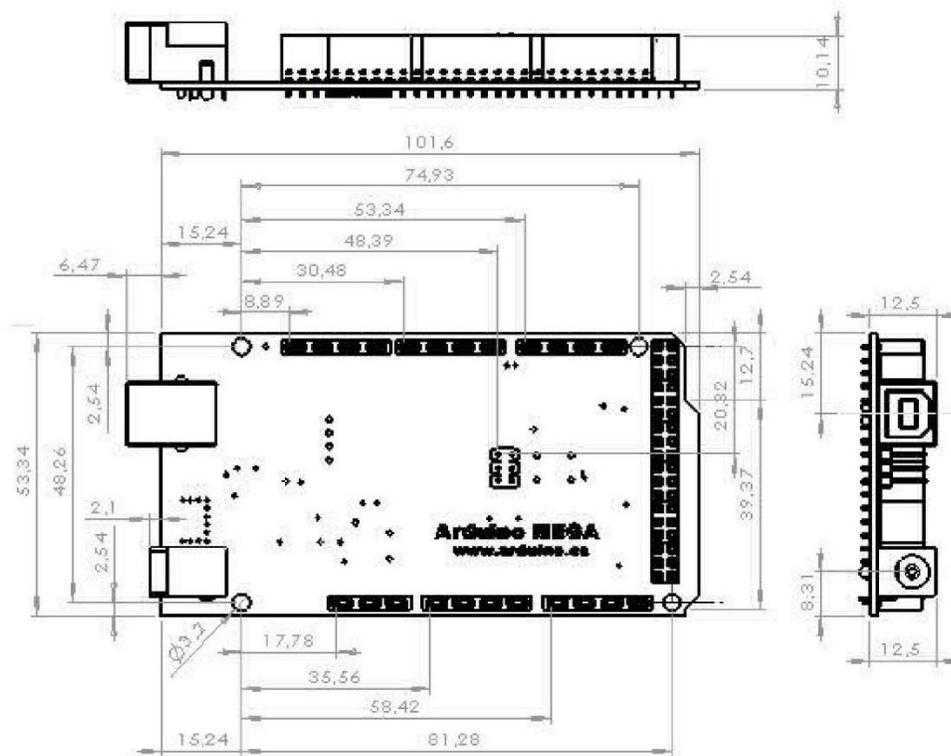
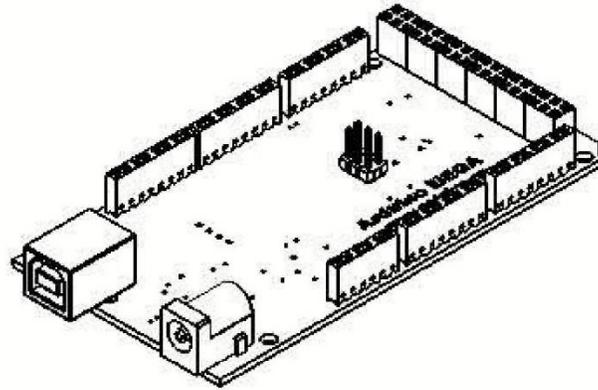


TX RX Flashing



Blinking Led!

Dimensioned Drawing



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

ANEXO 7: DataSheet XBee S2C.

XBee[®]/XBee-PRO S2C ZigBee[®]

RF Module

User Guide

Technical specifications

Performance specifications	16
Power requirements	16
General specifications	16
Networking and security specifications	17
Communication interface specifications	17
Regulatory conformity summary	18
Serial communication specifications	19
GPIO specifications	19
Hardware specifications for the programmable variant	20

Performance specifications

The following table describes the performance specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Indoor/urban range	Up to 60 m (200 ft)	Up to 90 m (300 ft)	Up to 60 m (200 ft)
Outdoor RF line-of-sight range	Up to 1200 m (4000 ft)	Up to 3200 m (2 mi)	Up to 1200 m (4000 ft)
Transmit power output (maximum)	6.3 mW (+8 dBm), boost mode 3.1 mW (+5 dBm), normal mode channel 26 max power is +3 dBm	63 mW (+18 dBm)	6.3 mW (+8 dBm) channel 26 max power is +1 dBm
RF data rate	250,000 b/s		
Receiver sensitivity	-102 dBm, boost mode -100 dBm, normal mode	-101 dBm	-102 dBm, boost mode -100 dBm, normal mode

Power requirements

The following table describes the power requirements for the XBee/XBee-PRO ZigBee RF Module.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Adjustable power	Yes		
Supply voltage	2.1 - 3.6 V 2.2 - 3.6 V for programmable version	2.7 - 3.6 V	2.1 - 3.6 V
Operating current (transmit)	45 mA (+8 dBm, boost mode) 33 mA (+5 dBm, normal mode)	120 mA @ +3.3 V, +18 dBm	45 mA
Operating current (receive)	31 mA (boost mode) 28 mA (normal mode)	31 mA	31 mA
Power-down current	< 1 μ A @ 25°C		< 3 μ A @ 25°C

General specifications

The following table describes the general specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Operating frequency band	ISM 2.4 - 2.5 GHz		
Form factor	through-hole, surface-mount		surface-mount

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Dimensions	through-hole: 2.438 x 2.761 cm (0.960 x 1.087 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	through-hole: 2.438 x 3.294 cm (0.960 x 1.297 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)
Operating temperature	-40 to 85 °C (industrial)		
Antenna options	through-hole: PCB antenna, U.FL connector, RPSMA connector, or integrated wire surface-mount: RF pad, PCB antenna, or U.FL connector		

Networking and security specifications

The following table describes the networking and security specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Supported network topologies	Point-to-point, point-to-multipoint, peer-to-peer, and DigiMesh		
Number of channels	16 Direct sequence channels	15 Direct sequence channels	16 Direct sequence channels
Interface immunity	Direct Sequence Spread Spectrum (DSSS)		
Channels	11 to 26		
Addressing options	PAN ID and addresses, cluster IDs and endpoints (optional)		

Communication interface specifications

The following table provides the device's communication interface specifications.

Interface options	
UART	250 Kb/s maximum
SPI	5 Mb/s maximum (burst)

ANEXO 8: DataSheet Pantalla led MOD PC02.

PANTALLAS FULL COLOR DE LEDS PARA APLICACIONES PUBLICITARIAS COMERCIALES

Su publicidad
Elegante
y llena de **vida**



rotuloselectronicos.NET
"EL MAYOR CENTRO DE DISTRIBUCIÓN DE RÓTULOS ELECTRÓNICOS"



Los medios publicitarios para exterior son el más espectacular y llamativo de forma continua, respectado a los medios estáticos. Su alta luminosidad y el dinamismo de los videos y animaciones publicitarias hacen que se genere una gran atención, con lo que sus clientes salen muy beneficiados. La efectividad de este medio está garantizada. Teniendo en cuenta todo esto, si contamos con ubicaciones de gran paso de personas, tenemos más que asegurados miles de impactos diarios. Esto está haciendo que las tradicionales vallas estáticas queden relegadas a un segundo plano, al tiempo que van siendo sustituidas por este tipo de vallas digitales. El resultado:

Las instalaciones que realizamos habitualmente van dirigidas a la explotación publicitaria de la pantalla, aunque, al mismo tiempo, también puede tener usos informativos.

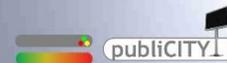
- ✔ Publicidad en zonas de gran paso.
- ✔ Vías de acceso a ciudades.
- ✔ Aparcamientos y centros comerciales.
- ✔ Concesionarios de vehículos.
- ✔ Grandes superficies y supermercados.
- ✔ Ayuntamientos y turismo.
- ✔ Establecimientos junto a vías de acceso.
- ✔ Zonas céntricas de ciudades.

Todos nuestros productos cuentan con diferentes sellos de calidad como reconocimiento al cumplimiento de los estándares establecidos por organismos internacionales.



rotuloselectronicos.NET

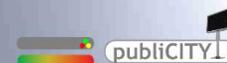
"EL MAYOR CENTRO DE DISTRIBUCIÓN DE RÓTULOS ELECTRÓNICOS"



<p>Pantalla LED: 180x90 cm. Superficie leds: 162 m² Pixel Pitch: 7V/14R Resolución: 256x128 px</p> <p>FULLCOLOR TRICOLOR</p>	<p>Pantalla LED: 180x134 cm. Superficie leds: 243 m² Pixel Pitch: 7V/14R Resolución: 256x192 px</p> <p>FULLCOLOR TRICOLOR</p>	<p>Pantalla LED: 205x154 cm. Superficie leds: 315 m² Pixel Pitch: 8V/16R Resolución: 256x192 px</p> <p>FULLCOLOR TRICOLOR</p>	<p>Pantalla LED: 192x144 cm. Superficie leds: 276 m² Pixel Pitch: 10R Resolución: 192x144 px</p> <p>FULLCOLOR HD</p>	<p>Pantalla LED: 208x160 cm. Superficie leds: 333 m² Pixel Pitch: 10R Resolución: 208x160 px</p> <p>HD FULLCOLOR</p>	<p>Pantalla LED: 307x154 cm. Superficie leds: 473 m² Pixel Pitch: 8V/16R Resolución: 384x192 px</p> <p>FULLCOLOR TRICOLOR</p>	<p>Pantalla LED: 307x230 cm. Superficie leds: 711 m² Pixel Pitch: 8V/16R Resolución: 384x288 px</p> <p>FULLCOLOR TRICOLOR</p>
---	--	--	---	---	--	--

rotuloselectronicos.NET

"EL MAYOR CENTRO DE DISTRIBUCIÓN DE RÓTULOS ELECTRÓNICOS"



200 cms.

150 cms.

Su publicidad Elegante y llena de Vida

Dimensiones de pantalla con marcos
Superficie de pantalla con marcos
Anchura de marco y perfil decorativo
Altura estándar del monoposte
Dimensiones rótulo patrocinio
Altura total
Color de la estructura

Dimensiones visibles pantalla LED	180x90 cms.
Superficie visible pantalla LED	1 62 m ²
Pixel Pitch Visual/Real	7 mm/14mm
Densidad de píxel	5.102 px/m ² . FULLCOLOR
Composición de píxel	
Definición gráfica visual	256x128 px FULLCOLOR
Brillo	7.000 NITS
Angulo de visualización	160° H - 90° V
Distancia visualización	10 a 75 m.
Voltaje de entrada	AC220V/50Hz
Consumo eléctrico pantalla	Max: 1.800 W - Medio: 700 W
Vida media de led	mínimo 10 años
Grado de protección	IP65
Temperatura de funcionamiento	-40°C - 60°C
Condiciones de humedad	10% - 95%
Sistema operativo	Windows 98/me/2000/NT/XP
Tipo de comunicación	internet ADSL, WiFi, 3G pen o Router 3G
Sonda temperatura/humedad/hora	opcional

TRICOLOR	incluido	FULLCOLOR	incluido
	incluido		incluido
	incluido		incluido
	opcional		opcional
	consultar		consultar



ANEXO 9: Código Nodo Sensor – Arduino Uno.

```

#include "LowPower.h"
int contador = 0;
void setup()
{
  Serial.begin(9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
}

void loop()
{
  switch (contador)
  {
    case 0:
      attachInterrupt(0, ServicioBoton, CHANGE);
      attachInterrupt(1, ServicioBoton1, CHANGE);
      break;
    case 1:
      attachInterrupt(0, ServicioBoton, CHANGE);
      break;
    case 2:
      attachInterrupt(1, ServicioBoton1, CHANGE);
      break;
  }
  LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);

  detachInterrupt(0);
  detachInterrupt(1);
  switch (contador)
  {
    case 1:
      Serial.print("A0");
      break;
    case 2:
      Serial.print("A1");
      break;
  }
  Serial.flush();
}

void ServicioBoton()
{
  contador = 2;
}
void ServicioBoton1()
{
  contador = 1;
}

```

ANEXO 10: Código Nodo Central – Arduino Mega2560.

```

#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
byte ip[] = {192, 168, 1, 15};
byte gw [] = {192, 168, 1, 1};
byte dns_[] = {192, 168, 1, 1};
byte subnet_[] = {255, 255, 255, 0};
EthernetClient client;
String idvariable_1 = "5a0a0c2e7625426d45794f4b";
String token = "AlE-CLILTg1SgtDp3WqALLjTamRRcigIiW";

byte i = 0;
byte v[2], v1[2];
boolean stringComplete = 0;
byte parqueo = 0;
byte est[5] = {0, 0, 0, 0, 0};

byte total = 5;
byte disponibles = 0;

void setup()
{
  Serial.begin(115200);
  Serial1.begin(9600);
  while (!Serial) ;

  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("Failed to configure Ethernet using DHCP");

    Serial.println("Configuring manually network parameters");
    Ethernet.begin(mac, ip, dns_, gw, subnet_);
  }
  Serial.println("Sistema listo");
}

void loop()
{
  while (Serial1.available())
  {
    v[i] = (char)Serial1.read();
    i++;
    if (i == 2)
    {
      for (i = 0; i < 2; i++)
        v1[i] = v[i];
      stringComplete = true;
    }
  }
}

```

```

        i = 0;
    }
}

if (stringComplete)
{
    stringComplete = false;
    switch (v1[0])
    {
        case 'A': est[0] = v1[1] - 48;
            break;
        case 'B': est[1] = v1[1] - 48;
            break;
        case 'C': est[2] = v1[1] - 48;
            break;
        case 'D': est[3] = v1[1] - 48;
            break;
        case 'E': est[4] = v1[1] - 48;
            break;
    }
    parqueo = 0;
    for (byte i = 0; i < 5; i++)
    {
        Serial.print(est[i]);
        parqueo += est[i];
    }
    Serial.println("");
}

disponibles = total - parqueo;

ubidots(String(disponibles));
client.flush();
client.stop();
delay(2000);

void ubidots(String valor)
{
    int num_1;
    String var = "{\"value\":\"" + String(valor) + "\"}";
    num_1 = var.length();
    if (client.connect("things.ubidots.com", 80))
    {
        Serial.println("connected");
        client.println("POST /api/v1.6/variables/" + idvariable_1 +
"/values HTTP/1.1\nContent-Type: application/json\nContent-Length: " +
String(num_1) + "\nX-Auth-Token: " + token + "\nHost:
things.ubidots.com\n");
        Serial.println("POST /api/v1.6/variables/" + idvariable_1 +
"/values HTTP/1.1\nContent-Type: application/json\nContent-Length: " +

```

```
String(num_1) + "\nX-Auth-Token: " + token + "\nHost:
things.ubidots.com\n");
    client.print(var + "\n");
    Serial.print(var + "\n");
    client.flush();
}
else
{
    Serial.println("connection terminada");
}
if (!client.connected())
{
    Serial.println();
    Serial.println("Desconectando");
    client.stop();
}

if (client.available())
{
    char c = client.read();
    Serial.print(c);
}
}
```