



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



DISEÑO DE UN SISTEMA DE SUPERVISIÓN DE LA
CAPACIDAD DE CONTENEDORES DE RESIDUOS SÓLIDOS
PARA OPTIMIZAR EL USO DE RECURSOS DE TRANSPORTE
PARA SU RECOLECCIÓN - DISTRITO DE PUNO AÑO 2022

TESIS

PRESENTADA POR:

LUCERO NALDY FLORES CASTRO

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2023



Reporte de similitud

NOMBRE DEL TRABAJO

DISEÑO DE UN SISTEMA DE SUPERVISIÓN DE LA CAPACIDAD DE CONTENEDORES DE RESIDUOS SÓLIDOS PARA OPTIMIZAR EL USO DE RECURSOS DE TRANSPORTE PARA SU RECOLECCIÓN - DISTRITO DE PUNO AÑO 2022

AUTOR

LUCERO NALDY FLORES CASTRO

RECUENTO DE PALABRAS

17611 Words

RECUENTO DE CARACTERES

101194 Characters

RECUENTO DE PÁGINAS

92 Pages

TAMAÑO DEL ARCHIVO

3.9MB

FECHA DE ENTREGA

Sep 25, 2023 12:41 PM GMT-5

FECHA DEL INFORME

Sep 25, 2023 12:42 PM GMT-5

● 15% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos:

- 12% Base de datos de Internet
- Base de datos de Crossref
- 10% Base de datos de trabajos entregados
- 4% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado

Christian Augusto Romero Goyzueta
INGENIERO ELECTRÓNICO
CIP: 133009

Karlos Alexander Ccantuta Chirapo
ING. ELECTRÓNICO
CIP: 94113

Sub Director de Investigación (E)
E.P.I.E.

Resumen



DEDICATORIA

A la dualidad de Dios que con sus bendiciones y pruebas me preparo en mi camino profesional, a mi familia y a mi madre que con su amor y ejemplo me enseñó a caminar y ser perseverante sin importar las dificultades, a mis hermanos Yissel y Jesus por su compañía en todos los malos y buenos momentos, por sus consejos, gracias por creer en mí porque significaron mi principal motivación para culminar esta investigación.

Lucero Naldy.



AGRADECIMIENTOS

Agradezco principalmente a Dios por guiarme y darme fortaleza para seguir adelante en mi vida diaria.

A la Universidad Nacional del Altiplano y a la Carrera Profesional de Ingeniería Electrónica por permitirme mejorar mis capacidades académicas a lo largo de mi carrera profesional.

Quiero expresar mi más profundo agradecimiento a mi asesor de tesis, M.Sc. Christian Augusto Romero Goyzueta, por su inigualable guía, paciencia y dedicación a lo largo de todo este arduo proceso. Su vasto conocimiento y sus valiosos comentarios han sido elementos fundamentales para el desarrollo y la excelencia de este trabajo. Aprecio de corazón su compromiso inquebrantable y su disposición constante para orientarme en cada etapa, proporcionándome las herramientas necesarias para mi crecimiento académico de manera sobresaliente.

A mis Docentes por impartir sus sabias enseñanzas, conocimientos durante los cinco años de mi desarrollo profesional.

Lucero Naldy



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE ACRÓNIMOS

RESUMEN 11

ABSTRACT..... 12

CAPITULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA 14

1.2. HIPÓTESIS 14

1.3. OBJETIVO GENERAL 14

1.4. OBJETIVOS ESPECÍFICOS 14

CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN 16

2.1.1. Antecedentes Locales 16

2.1.2. Antecedentes Nacionales 17

2.1.3. Antecedentes Internacionales 20

2.2. RESIDUOS SÓLIDOS..... 24

2.3. CLASIFICACIÓN DE RESIDUOS..... 25

2.3.1. Por su origen 25

2.3.2. Por su peligrosidad 26

2.3.3. Por su gestión 26



2.4. RESIDUOS SÓLIDOS DOMÉSTICOS.....	27
2.5. MARCO LEGAL	27
2.6. GESTIÓN DE RESIDUOS SÓLIDOS	29
2.7. CELDAS DE CARGA.....	31
2.9. INTERNET DE LAS COSAS	36
2.10. SERVIDOR LINUX	40
2.10.1. Servidores web	41
2.11. RASPBERRY PI.....	43
2.11.1. Raspberry pi como enrutador	44
2.12. ARDUINO MEGA	44
2.12.1. Ide de Arduino	45
2.13. ARDUINO MEGA	46
2.13.1. Ide de Arduino	46
2.14. ETHERNET SHIELD.....	46
2.15. GOOGLE MAPS.....	47

CAPITULO III

MATERIALES Y METODOS

3.1. MATERIALES	48
3.1.1. Hardware	48
3.1.2. Software.....	48
3.2. DISEÑO Y NIVEL DE LA INVESTIGACIÓN	49
3.2.1. Diseño de investigación.....	49
3.2.2. Nivel de la investigación	49
3.3. POBLACION Y MUESTRA DE LA INVESTIGACION.....	49
3.4. UBICACIÓN DE LA INVESTIGACIÓN.....	50



3.5. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCION DE DATOS	51
.....	
3.5.1. Técnicas	51
3.5.2. Instrumentos	51

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. CONTENEDOR DE RESIDUOS SOLIDOS.....	52
4.2. DISEÑO DE LA ESTRUCTURA PARA MEDICIÓN DEL PESO DE LOS RESIDUOS.....	52
4.3. CALIBRACIÓN DE LOS SENSORES.....	55
4.4. MEDICIÓN DEL PESO	59
4.5. CONEXIÓN DE SHIELD ETHERNET AL SERVIDOR	61
4.6. VERIFICACIÓN DE LOS PESOS EN LA NUBE	69
4.7. RUTAS POSIBLES A SEGUIR SEGÚN LA UBICACIÓN DE LOS CONTENEDORES DE LA CIUDAD DE PUNO	75
V. CONCLUSIONES	84
VI. RECOMENDACIONES.....	85
VII. REFERENCIAS BIBLIOGRÁFICAS.....	86
ANEXOS.....	90

Área: Telecomunicaciones y Telemática

Tema: Internet de las Cosas

FECHA DE SUSTENTACIÓN: 25 de octubre del 2023



ÍNDICE DE FIGURAS

Figura 1: Celda de carga y su configuración interna.....	32
Figura 2: Diagrama de conexión.....	32
Figura 3: Amplificador HX711 para la celda de carga.....	35
Figura 4: Modelo de referencia IoT.....	37
Figura 5: Raspberry Pi.....	43
Figura 6: Arduino MEGA	45
Figura 7: Entorno de programación Arduino	46
Figura 8: Ethernet Shield.....	47
Figura 9: Ciudad de Puno	50
Figura 10: Contenedor de residuos orgánicos	52
Figura 11: Base para los sensores.....	53
Figura 12: Ubicación de los sensores HX711	54
Figura 13: Diagrama de conexión de los sensores y el Arduino MEGA	54
Figura 14: Cobertura de los sensores de peso	55
Figura 15: librería y variables simbólicas	56
Figura 16: Creación del objeto HX711 y configuración en el void setup()	57
Figura 17: Configuración en el void loop().....	58
Figura 18: Definición de variables simbólicas	59
Figura 19: Creación de 4 objetos de HX711	59
Figura 20: Variables flotantes	60
Figura 21: Se definen cuatro objetos de la clase HX711.....	60
Figura 22: Cálculo del peso en lb y Kg	61
Figura 23: Librerías de conexión Ethernet	61
Figura 24: Definimos direcciones IP, mac, Gateway, dns, subnet	63



Figura 25: Se establece la conexión TCP	65
Figura 26: Conexión con el servidor	67
Figura 27: Se mantiene la conexión con el Servidor	68
Figura 28: Diagrama de conexión del prototipo	70
Figura 29: Prototipo integrado totalmente.....	71
Figura 30: Contenedor sin residuos	72
Figura 31: Resultados del peso sin residuos en el contenedor	72
Figura 32: Contenedor con residuos.....	73
Figura 33: Resultados con 15% de contenido	73
Figura 34: Prueba con contenedor lleno.....	74
Figura 35: Pesos con el contenedor lleno	74
Figura 36: Contenedor AV. Simón Bolívar.....	75
Figura 37: Contenedor Avenida el ejercito	76
Figura 38: Contenedor próximo al Parque del niño	76
Figura 39: Contenedor próximo a Colegio Villa de Lago.....	77
Figura 40: Contenedor próximo a EMSA PUNO.....	77
Figura 41: Contenedores próximo al mercado Huajsapata.....	78
Figura 42: Contenedor próximo al mercado Bellavista.....	78
Figura 43: Contenedor próximo al tobogán.....	79
Figura 44: Ubicación de contenedores en la ciudad de Puno	80
Figura 45: Ruta generada de acuerdo a la capacidad de los contenedores	81



ÍNDICE DE ACRÓNIMOS

IoT	: Internet de las cosas
OS	: Sistema Operativo
INEI	: Instituto Nacional de Estadística e Informática
OSI	: Modelo de interconexión de sistemas abiertos
HTTP	: Protocolo de transferencia de hipertexto
IP	: Dirección del Protocolo de Internet
MAC	: Control de acceso a medios
CSI	: Interfaz Serie para Cámaras
SD	: Secure Digital
API	: Interfaz de programación de aplicaciones
DNS	: Sistema de nombres de dominio



RESUMEN

En la actualidad, la supervisión adecuada de la capacidad de los contenedores es crucial para prever la contaminación ambiental y los problemas de salud pública. Por tal razón, esta investigación se orienta en el diseño de un sistema de supervisión de la capacidad de los contenedores de residuos sólidos, con el objetivo de optimizar el uso de los recursos de transporte para su recolección y adicionalmente reducir el desbordamiento. En este proyecto, se ha elaborado un sistema que mide el peso de los contenedores para determinar si están en su máxima capacidad, en media capacidad o vacíos. Los datos obtenidos por los sensores de peso se envían a un servidor Linux a través del Internet de las cosas (IoT), permitiendo a cualquier persona visualizar el estado de los contenedores. Una vez identificados los contenedores que están en su máxima y media capacidad, se establece una ruta para el camión recolector que cubra los lugares donde se encuentran dichos contenedores. Como resultado, se ha logrado un ahorro de combustible en el transporte de los camiones recolectores y se ha mejorado de manera eficiente la gestión de los desechos sólidos en la ciudad de Puno. Finalmente, el término sostenibilidad se aplica en este proyecto teniendo en cuenta que se refiere a la calidad de atender las demandas actuales de la sociedad sin generar un impacto negativo en las futuras generaciones.

Palabras claves: Contenedor de residuos sólidos, servidor, Internet de las cosas, sostenibilidad.



ABSTRACT

Currently, proper supervision of container capacity is crucial to prevent environmental contamination and public health problems. For this reason, this research is oriented towards the design of a system for monitoring the capacity of solid waste containers, with the aim of optimizing the use of transportation resources for their collection and also reducing overflow. In this project, a system has been elaborated that measures the weight of the containers to determine if they are at their maximum capacity, at medium capacity or empty. The data obtained by the weight sensors is sent to a Linux server through the Internet of Things (IoT), allowing anyone to view the status of the containers. Once the containers that are at their maximum and average capacity have been identified, a route is established for the collection truck that covers the places where said containers are located. As a result, fuel savings have been achieved in the transportation of collection trucks and solid waste management has been efficiently improved in the city of Puno. Finally, the term sustainability is applied in this project taking into account that it refers to the quality of meeting the current demands of society without generating a negative impact on future generations.

Keywords: Solid waste container, server, Internet of things, sustainability.



CAPITULO I

INTRODUCCIÓN

El Internet de las cosas o red de objetos conectados (Internet of Things, IoT) es aquella red de dispositivos conectados, como sensores, dispositivos de control y otros objetos cotidianos, que están conectados a internet y se comunican entre sí. Estos dispositivos pueden recopilar y compartir datos, lo que permite una mayor eficiencia y automatización en una variedad de aplicaciones, como la industria, el hogar inteligente, la salud y la movilidad.

La idea del Internet de las cosas se remonta a la década de 1980, cuando el ingeniero de sistemas Kevin Ashton escribió un artículo titulado "¿Qué es el Internet de las cosas?" para el Auto-ID Center del centro universitario de Cambridge. En el artículo, Ashton propuso la idea de que todos los objetos podrían tener identificadores únicos y estar conectados a Internet para recopilar y compartir datos.

Sin embargo, fue en la década de 2000 cuando el concepto comenzó a tomar forma con el desarrollo de dispositivos conectados y la popularidad de la conectividad inalámbrica y el aumento del uso de sensores y dispositivos móviles. A medida que los costos de estos componentes han disminuido, hubo un aumento en el número de dispositivos conectados, incluyendo electrodomésticos, automóviles, dispositivos de seguridad y dispositivos de salud. Cada vez que estos dispositivos se vuelven más comunes, se prevé que el Internet de las Cosas siga expandiéndose y generando cambios significativos en nuestra forma de vida y en nuestro entorno laboral.

El Internet de las cosas (IoT) ha transformado nuestra vida de varias maneras, algunas de las cuales son: mayor comodidad, mejora de la eficiencia, mayor seguridad, mejora de la salud y mayor eficiencia en las industrias.



En este proyecto de investigación se diseña un sistema de supervisión de la capacidad de los contenedores de residuos sólidos, para lo cual se usará sensores para determinar el volumen y la masa, estas magnitudes nos permitirán optimizar los recursos de transporte para la recolección de desechos sólidos del núcleo urbano.

Finalmente se implementa un prototipo del sistema que genera la mejor ruta posible considerando las magnitudes de volumen y masa del contenedor.

1.1. PLANTEAMIENTO DEL PROBLEMA

¿Cómo realizar el diseño de un sistema de supervisión de la capacidad de contenedores de residuos sólidos para optimizar el uso de recursos de transporte para su recolección en el distrito de Puno?

1.2. HIPÓTESIS

El diseño de un sistema de supervisión de la capacidad de contenedores de residuos sólidos permitirá optimizar el uso de recursos de transporte para su recolección en el distrito de Puno en el año 2022

1.3. OBJETIVO GENERAL

Diseñar un sistema de supervisión de la capacidad de contenedores de residuos sólidos para optimizar el uso de recursos de transporte para su recolección en el distrito de Puno en el año 2022

1.4. OBJETIVOS ESPECÍFICOS

- Diseñar un sistema de supervisión de la capacidad en volumen y masa de contenedores de residuos sólidos urbanos.
- Diseñar un sistema de administración de contenedores de residuos sólidos urbanos a fin de optimizar el uso de recursos de transporte para su recolección en la ciudad de Puno.



- Diseñar e implementar un prototipo del sistema que genere una mejor opción y ruta para la recolección de residuos sólidos urbanos desde los contenedores.



CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

En el marco de la presente investigación, se consideraron los siguientes antecedentes:

2.1.1. Antecedentes Locales

Cornejo empleó el método de ajuste lineal múltiple a fin de realizar la conexión real entre la recolección de desechos sólidos llevada a cabo por la Entidad municipal provincial de Puno (MPP) y el pago por los servicios realizados por la población puneña. Analizó la cantidad de desechos sólidos, la carga de transporte, y el pago de atenciones de saneamiento y movilidad. Según la MPP, la generación de residuos sólidos urbanos al día es de 78.68 Toneladas en un día, esto significa que la cantidad recolectada de residuos al mes es de 2 393.18 Toneladas, teniendo como capacidad recolectora sólo de 2 390 Toneladas, por lo que existe un déficit de aproximadamente de 49 toneladas. Así mismo el informe emitido por la Sub Gerencia de Gestión Ambiental y Salud Pública de la Municipalidad Provincial de Puno indica que existe un déficit en los pagos pues actualmente el costo anual asciende a 5'546,456.04 y sólo se recauda 1'452,897.46, por lo que existe un déficit de aproximadamente 4'093,558.58 (Cornejo, 2018).

De la misma forma Quilla realiza la valoración respecto al tratamiento de desechos sólidos urbanos en la población de Huancané. El investigador empleó diferentes enfoques para llevar a cabo su estudio, incluyendo el sistema de estimación condicional, el sistema de tarifa de desplazamiento y el patrón Logit Probit para la valoración económica. Menciona que este proceso también está fuertemente influenciado por otros factores ambientales como la contaminación, factores sociales: educación, tamaño del hogar,



edad. Género; y factores económicos tales como: nivel de ingreso, gestión municipal (Quilla, 2017).

Por su parte, Flores y Huanca proponen el empleo de estrategias comunicativas informativas en la ciudad de Azángaro con el objetivo de influir en el comportamiento, conducta y actitudes de los ciudadanos relacionado al uso de desechos sólidos en la gestión medioambiental. Finalmente, llega a la conclusión de que existe una falta de eficiencia en la coordinación, organización, realización y elaboración de materiales en los métodos de comunicación relacionadas con las tareas de gestión de desechos sólidos (Flores & Huanca, 2018).

Por otro lado, Rodrigo llevó a cabo una investigación en la entidad municipal provincial de Puno a fin de medir el impacto del sistema de administración ambiental en el manejo de desechos sólidos. Utilizó un enfoque no experimental de tipo correlacional causal, con una persecución explicativa y descriptiva, empleando un método analítico y sistémico. En sus conclusiones, señala que el régimen de restos sólidos en la ciudad de Puno, muestra un nivel de eficiencia que varía entre bueno y regular, debido a la escasez de un adecuado sistema de administración ambiental y por la ausencia de directrices para la planificación estratégica (Rodrigo, 2022).

2.1.2. Antecedentes Nacionales

El Decreto Legislativo N° 1278 determina los derechos, deberes, facultades y responsabilidades en toda la sociedad, con el propósito de fomentar una mejora constante de la eficacia en el uso de materiales y garantizar una administración y manejo de los desechos sólidos que sea poco costosa, sanitaria y ambientalmente adecuada. Todo esto se desarrolla de acuerdo con los deberes, principios y directrices determinadas en este Decreto Legislativo.



En el Artículo 8, se establecen medidas para maximizar la efectividad de las cosas y los ciclos de vida. Esto implica la elaboración de productos y suministro de servicios en todas las áreas del país, utilizando materias primas de manera eficiente y llevando a cabo investigaciones constantes para mejorar la productividad en el uso de materiales y desechos. Esto conlleva a mejorar los procesos de producción, fomentar la innovación y utilizar directamente los materiales negociables en actividades económicas.

En el Artículo 9, se aborda el tratamiento de actividades de producción. Los insumos directos de los materiales deben ser utilizados de acuerdo con la misma actividad o a través de investigaciones, desarrollos u otras actividades similares. Además, se establece la evaluación de residuos, con el objetivo de transformarlos según los métodos establecidos por las reglas de residuos sólidos. Los materiales peligrosos deben ser considerados como tales y se deben aplicar normas especiales para su embalaje, manipulación y transporte. Los materiales abandonados se consideran residuos sólidos y deben ser procesados adecuadamente para su distribución final en rellenos sanitarios u otras infraestructuras similares, sin perder su utilidad (Ministerio de Justicia, 2017).

De acuerdo con la investigación realizada por Seminario y Tineo sobre la utilización de desechos sólidos en un supermercado de la localidad, el propósito fundamental de esta tesis es estudiar el proyecto de tratamiento de desechos sólidos utilizado por la megatienda conocida como Tottus, localizado en el centro comercial Open Plaza de la ciudad de Piura, y de acuerdo a ello proponer mejoras que puedan ayudar al sistema actualmente implementado. El trabajo se llevó a cabo en cuatro pasos. En primer lugar, se compiló información general sobre la administración de residuos a través de fuentes bibliográficas tanto físicas como en línea. En segundo lugar, se compilaron informaciones vinculadas con el plan de administración de desechos implementado por la megatienda Tottus - Open Plaza Piura, a través de visitas de inspección al lugar



realizadas por el inspector del mercado y entrevistas al administrador de Emaús, la empresa encargada del transporte y tratamiento de desechos sólidos, con el fin de obtener información sobre el proceso de tratamiento de los desechos. En tercer lugar, se examinó la información recopilada con el fin de cuantificar, calificar y describir el plan de administración aplicada por la megatienda mencionada. Hoy en día, el hipermercado origina un aproximado de 397 264 kg de desechos sólidos al año, entre ellos el 34.8% pertenecen a papel cartón y material sintético. Se detectaron las principales insuficiencias en el procedimiento de manejo de desechos sólidos implementado por la megatienda Tottus - Open Plaza Piura: la ausencia de reutilización de residuos internos como plástico y cartón, la omisión de evaluación de los desechos orgánicos, la inconsciencia y la responsabilidad del consumidor hacia el plan de gestión y la preservación del medio ambiente, la escasa divulgación externa del plan de manejo de residuos del hipermercado, la carencia de separación de basura en contenedores, el inadecuado tratamiento de cajas de cartón y plástico, la emanación de olores en los contenedores de residuos orgánicos, la falta de definición y clasificación de indicadores clave de gestión para supervisar los procesos, y la insuficiente frecuencia de recolección de materiales peligrosos. Basándose en los resultados obtenidos de la investigación, se han dado a conocer una variedad de mejoras que se dividen en cinco grupos según los objetivos buscados: la proposición de reutilización de desechos sólidos, la propuesta de aumentar la implicación de los clientes en el procedimiento de administración, la sugerencia de incrementar la participación de los colaboradores en el curso de administración, la proposición de optimizar el almacenamiento de los residuos sólidos y la sugerencia de mejorar la política de supervisión y la gestión de los desechos sólidos. Se han planificado y documentado diversas acciones y propuestas como entrevistas, actividades de sensibilización, una inspección interna de documentos de Tottus - Open Plaza Piura y la elección de cajas de



cartón para empacar los productos. Se detallan las actividades específicas, los responsables y los requisitos para facilitar su implementación (Seminario & Tineo, 2018).

2.1.3. Antecedentes Internacionales

Según Durán (2019), se examinaron las principales iniciativas en sistemas de administración de redes para Internet de las Cosas (IoT) y se identificaron los requisitos esenciales para gestionar dispositivos en cualquier implementación. Estos dispositivos abarcan el aprovisionamiento e identificación, estructuración y control, monitoreo y diagnóstico, y actualización y sostenimiento de software. Además, se analizaron tendencias en métodos de ejecución como: “Simple Network Management Protocol, Telecommunications Management Network, Web-Based Enterprise Management y Policy-Based Management”. También se desarrollaron las especificaciones y diseños de la infraestructura del sistema de administración de red para IoT, que incluye dispositivos IoT, gateways IoT y nubes, siguiendo un modelo de cómputo. Este diseño facilita la administración de dispositivos en una red IoT, centrándose en la gestión de datos de los sensores de los dispositivos IoT mediante una infraestructura basada en objetos de información. Finalmente, se plantean posibles desarrollos futuros para el sistema de administración de red elaborado, como la implementación de SNMP v3 en la comunicación entre dispositivos IoT y gateways IoT. En cuanto a la comunicación entre el gateway IoT y la aplicación de gestión, se seleccionó el protocolo SOAP, pero también se podrían incorporar protocolos adicionales como REST, brindando al usuario la opción de elegir según sus necesidades específicas de rendimiento (Duran, 2019).

Gahona y Gavilema han realizado un estudio sobre el desarrollo de una red de Internet de las Cosas (IoT) para un edificio empresarial, donde señalan que actualmente se están implementando procesos de mejoras de las redes IoT con el fin de ofrecer soluciones oportunas a la población, empleando tecnología inalámbrica e integración con



software de notificación en caso de acceso no permitido, hurto y alarmas de evacuación en casos de catástrofes naturales. En su actividad, se describe el diseño y la prueba de funcionamiento de una red IoT para proteger los departamentos en el edificio de la empresa Consel. Esta red IoT, se empleó la tecnología Aruba HPE, junto con host terminales IoT Wi-Fi como dispositivos de videovigilancia, detectores de presencia, detectores de llama e intercomunicadores de videoporteros. Se menciona que se consideró la marca Dahua para garantizar mayor protección y cumplir con las necesidades de los residentes, administradores y colaboradores a través del seguimiento y inspección centralizada de los dispositivos de seguridad IoT Wi-Fi. Mediante la prueba del diseño de la red IoT Wi-Fi, se visualizaron los factores de tráfico de información en dos eventos planteados, con el objetivo de asegurar que el incremento de hosts IoT no afectara el desempeño en términos de latencia, desempeño, carga y drops de paquetes. Se realizó una evaluación de los gastos relacionados con la puesta en marcha, funcionamiento y aspectos técnicos mediante el empleo de análisis que consideraron los costos por unidad, y se calcula que la restauración de la inversión se logrará aproximadamente en un año, lo cual se suma al costo total de cada departamento en el edificio. La escalabilidad del edificio fue considerada en el diseño de la terminal IoT 1844, proporcionando datos y alarmas a los usuarios de los dispositivos IoT, y los dispositivos sugeridos pueden obtenerse en cualquier lugar elegido por el equipo de Albu para cumplir con los requisitos de Cono Corporativo. La simulación del retraso de envío a través de la red se realizó para evaluar el impacto de siete dispositivos IoT distribuidos en el departamento y en ciertas áreas del edificio, logrando reducir el retraso en un 14%. En términos de rendimiento de la red IoT, el escenario propuesto permitió una carga del 11.9%. La calidad del servicio, basada en WFQ (Weighted Fair Queuing) como una forma de priorizar el tráfico de red, mejoró en un 57.01%. El monto total de la inversión de 404,006.02 USD se recuperó en



aproximadamente 13 meses, y el margen de beneficio fue del 4.909% (Gaona & Gavilema, 2020).

Bocanegra, Gamarra y Tipian, en su informe acerca de la administración de residuos sólidos en el País durante el período del COVID-19, se destaca que al pasar cinco días desde la confirmación del primer caso positivo de COVID-19 en el país (6 de marzo de 2020), el Ministerio de Salud (MINSA) anunció una Situación de Urgencia en el ámbito de la salud a nivel nacional, válida por un período de 90 días contados desde el calendario, que abarcaba hasta el 10 de junio de 2020, mediante el Decreto Supremo N° 008-2020-SA2. En ese contexto, se pusieron en marcha acciones para prevenir y gestionar la situación. con el objetivo de contener la propagación del COVID-19. Posteriormente, a través del Decreto Supremo N° 020-2020-SA3, se extendió dicha Emergencia Sanitaria por 90 días adicionales, es decir, hasta el 09 de septiembre de 2020, con el propósito de continuar realizando acciones preventivas, de control y atención sanitaria con el fin salvaguardar la salud de la población en todo el territorio nacional. Los desechos sólidos municipales incorporan los desechos domésticos y los residuos generados por la limpieza de espacios públicos, como playas, el comercio y otras actividades urbanas no domésticas, donde los servicios de limpieza pública se encargan de su recolección en toda su jurisdicción. En el Perú, se produce más de 7 millones de toneladas de residuos sólidos urbanos al año, aproximadamente 20,000 toneladas por día y casi 1,000 toneladas por hora. Del total de estos desechos, el 70% proviene de los hogares. Durante la pandemia, fue crucial mantener el aislamiento social y promover la reactivación económica. Además de los residuos sólidos habituales, también se deben tener en cuenta los residuos generados como resultado del sistema en sí y las consecuencias específicas de la situación actual. En este sentido, los servicios de limpieza y recolección de residuos sólidos son considerados servicios esenciales donde, el Estado debe proteger durante un estado de



emergencia nacional. De lo contrario, la interrupción de dichos servicios podría provocar una acumulación temporal de residuos en calles, plazas y espacios públicos, lo que representaría riesgos para nuestro medio ambiente y la salud pública, ya que ocasionaría aparición de vectores como roedores, cucarachas, moscas y mosquitos, capaces de transferir enfermedades (Bocanegra, Gamarra, & Tipian, 2020).

Gomez-Diaz en su artículo del año 2020 indica que: "Si estamos promoviendo la sostenibilidad de la inducción y estamos buscando diferentes miembros de su familia, comprenda que la energía y el consumo de agua se gestionan y administran el consumo de energía y agua pueden ayudar al medio ambiente puede ser más cauteloso. La mejora industrial y del entorno de consumo, la mejora del medio ambiente y el agua y la energía serán tema de desarrollo y resultado de esta investigación, así como la utilización y el entendimiento de las herramientas electrónicas y técnicas para monitorear la gestión y gasto apropiados. El desperdicio en un área residencial puede generar consumo excesivo por ejemplo de agua, la cual puede ser utilizada en el consumo humano. Es posible entender la "sostenibilidad de la inducción" al promover la tecnología en todos los miembros de la sociedad, para mejorar la gestión y la preocupación por la energía y el consumo de agua, lo que a nivel macro pueden ayudar a varias ciudades, las que empiecen a ser más sostenibles, y a la vez más cuidadosas con el medio ambiente. Indican además que, si se inicia la enseñanza sostenible desde los hogares, esto impulsará cada vez más a la consciencia de las personas, ayudando a las empresas e industrias y a toda la comunidad a ocuparse del medio ambiente, y el uso responsable de la energía y el agua. Por ejemplo, la utilización de dispositivos electrónicos que, cada vez que el usuario utilice y mantenga un el grifo del fregadero abierto, o cada vez que se enciende la luz sin permanecer en el lugar realice un aviso para corregir estos descuidos, al mismo tiempo que comience a



crear una conciencia en el hogar y así recortar costos innecesarios que perjudiquen su economía y de manera global dañen el planeta (Gomez-Diaz, 2020).

García y Socorro en su artículo realizan el estudio de caso acerca de la situación de la utilización de los desechos sólidos en la ciudad de Machala, Ecuador. Tomaron como propósito dos localizaciones: un mercado y una urbanización privada, realizando el estudio situacional para ambos sectores. A partir de los análisis realizados se dio un diagnóstico situacional a partir de encuestas y entrevistas personales. Luego se propuso un plan de administración y manejo integral de los desechos, lo que permitirá el cumplimiento de la ley respecto al medio ambiente, contribuyendo a un ambiente sano y saludable (García, Socorro, & Vanessa, 2019).

2.2. RESIDUOS SÓLIDOS

Se tienen varios conceptos de lo que son residuos sólidos. Por ejemplo (Portocarrero, 2018) menciona que un residuo es cualquier objeto, elemento o sustancia en estado sólido que es abandonado, desechado o descartado. Asimismo, se hace referencia a que también puede ser cualquier material que carezca de valor o utilidad para su propietario. Según el Instituto Nacional de Estadística e Informática (INEI), los desechos se refieren a "sustancias, productos o subproductos en forma sólida o semisólida que una entidad debe eliminar o está obligada a eliminar de acuerdo con las leyes y regulaciones aplicables", con el objetivo de prevenir los riesgos que puedan ocasionar a la salud y al medio ambiente" (INEI, 2021). Además, el Ministerio del Ambiente refiere que los residuos son todas los elementos, productos o subproductos en estado semisólido o sólido, que el motriz debe desechar o está forzado a desechar de acuerdo con la normativa vigente o en consideración de la salud pública (Avalos, 2020).



2.3. CLASIFICACIÓN DE RESIDUOS

Existen varios criterios para su clasificación, pero tomaremos los indicados por (Avalos, 2020), el cual los clasifica en tres tipos: por su origen, su peligrosidad y por último por su gestión.

2.3.1. Por su origen

Se toma como criterio la actividad que lo origina, por lo que se le puede clasificar como:

- Residuos domésticos: se refiere a los residuos generados en hogares o lugares similares. Normalmente son los de mayor volumen.
- Residuos comerciales: son los generados por actividades comerciales como mercados, hoteles, restaurantes y actividades relacionadas.
- Residuos de establecimientos de salud pública: también conocidos como residuos de salud pública, varían según el tipo y nivel de establecimiento de salud.
- Residuos institucionales: son los generados por instituciones educativas, entidades gubernamentales, militares, religiosas, terminales aéreas y terrestres, y otras oficinas.
- Desechos de limpieza de lugares públicos: son aquellos que se originan durante el proceso de limpieza de espacios comunes utilizados por la población, como parques, calles, jardines, etc.
- Residuos de actividades de construcción: incluyen residuos de demolición y reparación, como ladrillos, madera, cerámica, entre otros.
- Residuos industriales: se clasifican según el proceso de producción, la materia prima utilizada, los productos intermedios generados y los envases y embalajes utilizados. También se incluyen los desechos industriales equiparable a urbanos,



ocasionados por pequeñas industrias que generalmente se gestionan junto con los residuos urbanos debido a su volumen reducido.

- Residuos del sector agropecuario: son principalmente residuos de cultivos o de la limpieza de campos de cultivo. Esto incluye, entre otros, residuos de poda, envases de fertilizantes, restos de plástico de invernaderos, mangueras, estructuras de madera y metal, tuberías, restos de productos fitosanitarios, etc.
- Residuos de instalaciones o actividades especiales: aquí se incluyen, por ejemplo, los residuos mineros, que requieren un manejo especial debido a sus características.

2.3.2. Por su peligrosidad

- Residuos peligrosos: se refieren a aquellos que, debido a las propiedades inherentes de sus materiales o a su manejo, transporte o almacenamiento, patentizan un riesgo para la vida humana.
- Desechos no peligrosos: son los que se generan como parte de la actividad diaria o habitual, ya que no manifiesta un peligro en la vida humana ni para el medio ambiente.

2.3.3. Por su gestión

- Residuos municipales: se refieren a los residuos generados en hogares, locales comerciales y áreas públicas. En esta categoría se incluyen comúnmente alimentos, papel, botellas, envases, productos de higiene personal, maleza, residuos de limpieza de parques, calles y avenidas, así como otros productos similares.
- Residuos no municipales: son aquellos cuyas características o manejo representan un riesgo para la vida humana, como los generados en centros de salud,



instalaciones industriales, labores de construcción, sector agrícola y en montajes especiales, como la actividad minera o metalúrgica.

2.4. RESIDUOS SÓLIDOS DOMÉSTICOS

(Secretaría de Medio Ambiente y Recursos Naturales, 2016) define que los desechos sólidos son productos desechados en los hogares, así como los residuos urbanos generados en espacios públicos que no pertenecen a otra categoría, como los residuos de la construcción. Esta investigación se centró principalmente en el estudio de este tipo de residuos.

2.5. MARCO LEGAL

Si bien el Perú ha progresado en cuanto a la legislación, aún existen vacíos legales en relación al tema ambiental y al uso adecuado de los residuos sólidos. Dentro de la legislación vigente encontramos:

- La Constitución Política del Perú, en sus artículos 2 (inciso 22) y 67, determina que todas las personas poseen derecho a un ambiente equilibrado, y el Estado fomenta el uso sostenible de los recursos naturales.
- La Ley de Gestión Integral de desechos Sólidos, aprobada por medio del Decreto Legislativo el 24 de abril del 2017, busca garantizar una administración y uso adecuados de los residuos sólidos, con enfoque en la disminución, advertencia de riesgos ambientales, y cuidado de la salud y el bienestar de las personas. En el Artículo 7 (inciso 33) se establecen los materiales para un manejo eficiente de los residuos sólidos, y el Artículo 24 (inciso 24.1) responsabiliza a las municipalidades distritales en el uso y regulación de los desechos sólidos.
- La Ley General del Ambiente (Ley 28611), aprobada el 15 de octubre de 2005, determina los principios y normas básicas a fin de garantizar el derecho a un ambiente saludable y equilibrado. En el Artículo 67 destaca la prioridad del



saneamiento básico por parte de los tres niveles de gobierno: nacional, regional y local.

- La Ley del Sistema Nacional de Impacto Ambiental (Ley 27446), junto con la Normativa sobre el Registro Nacional de Consultoras Ambientales y el Sistema de Evaluación de Impacto Ambiental (SEIA), forman parte de las Normas Ambientales Generales. El propósito es alcanzar un desempeño ambiental integral y coordinado, con un enfoque preventivo para mitigar y modificar los efectos adversos derivados de la actividad humana.
- La Ley Orgánica de Municipalidades (Ley 27972) determina la rivalidad de los órganos de gobierno local. En su Artículo 80, se menciona la regulación y control del transcurso de preparación final de los residuos sólidos, líquidos e industriales como una función específica.
- La Ley de Gobiernos Regionales (Ley 27867) norma la estructura, organización, rivalidad y labor de los gobiernos regionales. Entre sus funciones, el Artículo 50 destaca la expresión, coordinación y vigilancia de estrategias para el control del medio ambiente y la salud en las ciudades.
- La Ley General de Salud (Ley 26842) decreta que el cuidado de la salud es de interés público, siendo el compromiso del Estado regularla, vigilarla y fomentarla. El Artículo 104 prohíbe la descarga de desechos o sustancias contaminantes sin considerar las normas del cuidado sanitaria y ambiental.
- La Ley de Seguridad y Salud en el Trabajo (Ley 29783) decreta las bases a fin de implementar un Sistema de Gestión de la Seguridad y Salud en el trabajo, con el propósito de evitar accidentes laborales. El Artículo 49 establece la obligación del empleador de decretar las bases con el fin de emplear una adecuada



implementación del Sistema de Gestión de la Seguridad y alcanzar la prevención de accidentes

2.6. GESTIÓN DE RESIDUOS SÓLIDOS

Según el Plan Nacional de Gestión Integral de Desechos Sólidos 2016 – 2024 (Ministerio del Ambiente, 2016) la gestión de residuos sólidos se describe como el conjunto de actividades técnicas y administrativas destinadas a planificar, organizar, concebir, llevar a cabo y evaluar políticas, tácticas, proyectos y programas orientados hacia el manejo adecuado de los desechos sólidos, tanto en el ámbito municipal como en el no municipal, a nivel nacional, regional y local. Este documento también menciona tres ejes estratégicos: fortalecimiento de la facultad de las personas o entidades que participan en el manejo de los residuos sólidos, realizado de manera institucional en las entidades encargadas de la administración ambiental en el Perú, y fomento de inversiones públicas y privadas para cubrir las necesidades operativas y presupuestarias existentes. A partir de estos ejes se derivan los siguientes principios rectores: educación dirigida principalmente a los generadores de residuos sólidos, prevención y precaución para disminuir los efectos en las personas y el medio ambiente, integración de planes a nivel nacional, regional y local, control de las fuentes de desechos para reducir su cantidad, eficiencia en el uso de recursos asignados, reciclaje y valorización de los recursos, sostenibilidad ambiental, económica y social en la gestión, autosuficiencia financiera de los generadores de residuos, responsabilidad proporcional al volumen y tipo de residuo generado, responsabilidad extendida del productor en los residuos de tratamiento especial, promoción de una economía sostenible y responsable con el medio ambiente, participación ciudadana y comunicación para incluir a los generadores en la administración de residuos, y aplicación de ciencia y tecnología para mejorar el manejo de residuos. En esta tesis, se enfoca principalmente en el uso de la ciencia y tecnología



como implemento para colaborar en la administración de residuos sólidos y mejorar su manejo. A partir de estos principios rectores se dan los siguientes lineamientos de política generales: educación ambiental enfocado a la población para el correcto uso de desechos sólidos y disminución de su producción, descentralización de la determinación respecto al uso de los residuos, multisectorialidad en la participación de todos los sectores implicados así como que este manejo sea tratado por diferentes profesionales teniendo así un carácter multidisciplinario, información y vigilancia relacionada a la protección ambiental y de vigilancia sanitaria, participación ciudadana no sólo para la utilización y la reducción de desechos sólidos sino también para su participación en las políticas, lineamientos y programas para el uso adecuado de los residuos y por último cooperación técnica para la transferencia tecnológica de las posibles soluciones a los problemas ocasionados por la generación de residuos sólidos.

Concordante con esta Política Nacional de Gestión Integral de Residuos Sólidos (Callirgos & Mendez, 2015) indica las siguientes fases para su manejo:

- Generación y acumulación de los residuos en contenedores generalmente inadecuados en los domicilios y centros de recolección de las municipalidades.
- Recolección de los desechos generalmente en camiones o carros recolectores de basura. Es en este punto donde se enfoca la posible solución presentada en esta investigación
- Movilidad de los residuos a los vertederos.
- Depósito de los residuos sólido recolectados a los vertederos, que generalmente reciben el nombre de incontrolados, pues no se tiene un tratamiento final de procesamiento



- En pocos casos se realiza la recuperación de materiales reciclados o re aprovechables, generalmente esta labor la realizan personas de pocos recursos económicos en condiciones antihumanas.
- Quema o combustión de los residuos restantes.

Los mismos autores mencionan que otros sectores, generalmente en sectores de mejor nivel socio económico se adicionan las siguientes etapas:

Los materiales aprovechables: papel, cartón plástico son seleccionados, almacenados y vendidos en función a la oferta del mercado

Se realiza la incineración de residuos para calentamiento de agua, generalmente esto sucede en residuos de zonas rurales o residuos de jardinería en zonas urbanas

En menor escala almacenamiento de residuos comestible para la alimentación de animales, también sucede generalmente en zonas rurales, o las zonas de periferia de las ciudades en condiciones de insalubridad.

2.7. CELDAS DE CARGA

Para medir el contenido de los contenedores se utilizará el peso de los desechos sólidos y para determinar esta variable se usarán las celdas de carga, los cuales son definidas como un transductor que traslada valores de fuerza mecánica como: compresión, tensión, carga o peso en una señal eléctrica (Altamirano-González, Manotoa-Balseca, Rodríguez-Escobar, & Hidalgo-Calero, 2022). Su mecanismo de funcionamiento se basa en las galgas extensiométricas que convierte la fuerza que deforma el material en señales eléctricas mediante un puente de Wheatstone, que se evidencia en la figura 1.

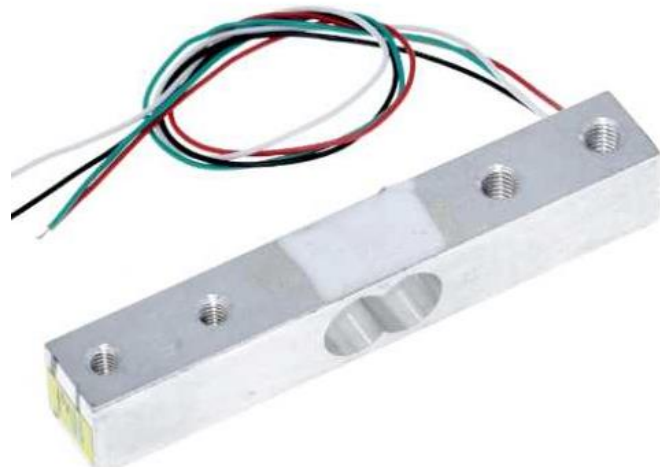


Figura 1: Celda de carga y su configuración interna
Fuente: (Cardona-Castro, 2021)

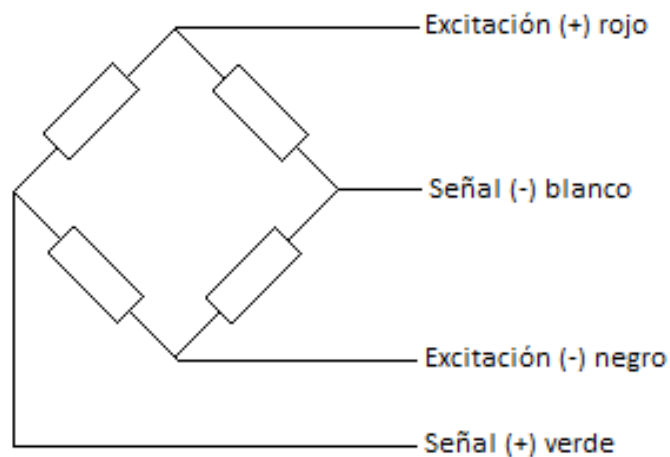


Figura 2: Diagrama de conexión
Fuente: (Cardona-Castro, 2021)

En el mercado existen muchas variantes de celdas de carga, pero mencionaremos las siguientes:

- Miniatura: tienen una alta capacidad en un espacio mínimo disponible, utilizadas para bancos de prueba y prototipos, son de construcción robusta y alta precisión.
- Deformación: Son utilizados para mediciones precisas, construidas en base a un alambre o una lámina fina que varía su resistencia en función a la presión aplicada. Tiene una aislante que aísla el ruido eléctrico al mismo tiempo que disipa el calor



generado, así como utiliza el puente de Wheatstone para dar los resultados en forma de señales eléctricas.

- Viga: Son de baja capacidad, con el elemento dilatante de aluminio, con rangos de medida de 1 a 500 kg. Se utilizan generalmente en ambientes industriales para pesaje estático y dinámico y mediciones de fuerza.
- Plataforma: Selladas herméticamente para aplicaciones en ambientes hostiles y de alta precisión. Utilizan galgas extensiométricas para tener altas precisiones: 0.02%.
- Tipo S: Utilizan también la compresión para para brindar el mejor rendimiento en un encapsulado compacto y versátil.
- Estilo recipiente: Para medir fuerzas de compresión axial, acondicionados también para ambientes hostiles.
- Compensación hidrostática: Para trabajos marinos, incluso a grandes profundidades, toda su construcción es de acero inoxidable.
- Tensión compresión: Para medir fuerzas de dos tipos: tensión y compresión realizadas con acero inoxidable soldado.

Dentro de los parámetros a tomar en cuenta para la elección de la celda de carga tenemos:

- Tiempo de duración de la medición
- Velocidad de la medición
- Dirección de la carga. Compresión o tensión
- Precisión medida generalmente en porcentaje
- Temperatura de funcionamiento y condiciones ambientales.
- Como va a estar montado en la estructura de medición
- Si son necesarias las certificaciones como ASTM E74



- Costo de las operaciones
 - Requisitos de salida: por ejemplo, protocolo RS232, RS485, 0 a 5 V, 4 a 20mA
- Se muestra las características técnicas de la celda de carga con una capacidad y precisión de 50kg (Tostatronic, 2020)

Especificaciones.

- Tensión de entrada: $\leq 10V$
- Peso máximo permitido: 50 Kg (110 Libras)
- Sensibilidad: 1.0mV/V (Tolerancia ± 0.1 mV/V)
- Material utilizado: Aluminio
- Rango de temperatura de funcionamiento: 0°C a 50°C

Datos técnicos:

- Descripción: UM (Unidad de Medida) - Valor
- Capacidad: KG (Kilogramos) - 40-50
- Tolerancia de error: mv/v (milivoltios por voltio) - 0.05
- Sensibilidad de salida: mv/v - 1.0 ± 0.1
- No linealidad: %FS - 0.03
- Repetibilidad: %FS - 0.03
- Histéresis: %FS - 0.03
- Arrastrarse (3 minutos): %FS - 0.03
- Cruce por cero (1 minuto): %FS - 0.03
- Efecto de temperatura en el punto cero: %FS/10°C (porcentaje de la escala completa por cada 10 grados Celsius) - 1
- Efecto de temperatura en la salida: %FS/10°C - 0.05
- Salida en el punto cero: mV/V (milivoltios por voltio) - ± 0.1
- Resistencia de entrada: Ω (Ohmios) - 1000 ± 20

- Resistencia de salida: Ω - 1000 ± 20
- Resistencia de aislamiento: $M\Omega$ (Megaohmios) - ≥ 5000
- Voltaje de excitación: V (Voltios) - ≤ 10
- Rango de temperatura de trabajo: $^{\circ}C$ (grados Celsius) - 0 a +50
- Capacidad de sobrecarga: %FS - 150

Respecto a los requisitos de salida para esta investigación se utilizó el módulo HX711 para cumplir con este requisito.

2.8. AMPLIFICADOR HX711

Como la salida del puente es una señal en milivoltios, se tiene que utilizar un módulo conversor analógico digital. Este es el HX711 que es un amplificador junto con un conversor analógico digital de 24 bits que soporta aplicaciones a pequeña escala y también procesos industriales. Para su comunicación con los dispositivos externos utiliza el protocolo I2C por una interfaz de dos pines, como se muestra en la figura x2 (Cardona-Castro, 2021).

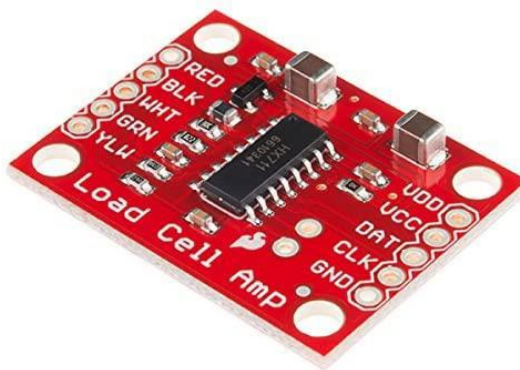


Figura 3: Amplificador HX711 para la celda de carga
Fuente: (Cardona-Castro, 2021)

Los rasgos distintivos técnicas de este módulo son las siguientes (EIRL, 2017):

- Dos canales de entrada diferencial que se pueden seleccionar
- Amplificador de ganancia programable de bajo fragor en el chip, con opciones de ingreso de 32, 64 y 128



- Administrador de fuente de alimentación integrado en el chip con el fin de la alimentación analógica de la celda de carga y el convertidor analógico-digital (ADC)
- Oscilador incorporado en el circuito integrado
- Función de reinicio incorporada en el circuito integrado
- dispositivo digital simple e interfaz en serie: control mediante pines, no es necesario programar
- Velocidad de salida de datos seleccionable: 10SPS (muestras por segundo) o 80SPS
- Utilización de corriente, incorporado el regulador de fuente de alimentación analógica en el chip: menos de 1.5mA en funcionamiento normal, menos de 1uA en modo de apagado
- Voltaje de alimentación operativo admitido: 2.6 ~ 5.5V
- Temperatura de operación admitida: -40 ~ +85 °C
- Dimensiones: 35 mm x 20 mm

2.9. INTERNET DE LAS COSAS

Existen muchas definiciones o acepciones del término Internet de las cosas o IoT, tenemos por ejemplo la proporcionada por (Pisano, 2018) que indica que Internet de las cosas es un concepto acerca de la conexión de cualquier dispositivo a Internet utilizando protocolos o normas estandarizadas, para ser monitorizadas o controladas a distancia, a veces sin intervención humana. Complementando esta definición (Nguyen & De Cremer, 2016) agrega que se puede captar los datos en volúmenes considerables para su posterior análisis y así convertirlos en información y posteriormente en conocimiento. También menciona que esta tecnología es heterogénea por lo que varios dispositivos, plataformas, fabricantes y diseñadores se pueden comunicar sin ningún problema.

2. Modelo de referencia de Internet de las cosas.

Debido a que el modelo OSI de ISO referencia todo el proceso de comunicaciones en redes de datos, se utilizó el modelo de referencia proporcionado por la sugerencia ITU-T Y.2060 (Union Internacional de Telecomunicaciones, 2012) que indica que este conjunto de procesos se puede dividir en cuatro capas: aplicación, soporte a servicios y aplicaciones, de red y por último capa de aparatos, como se evidencia en la figura 4

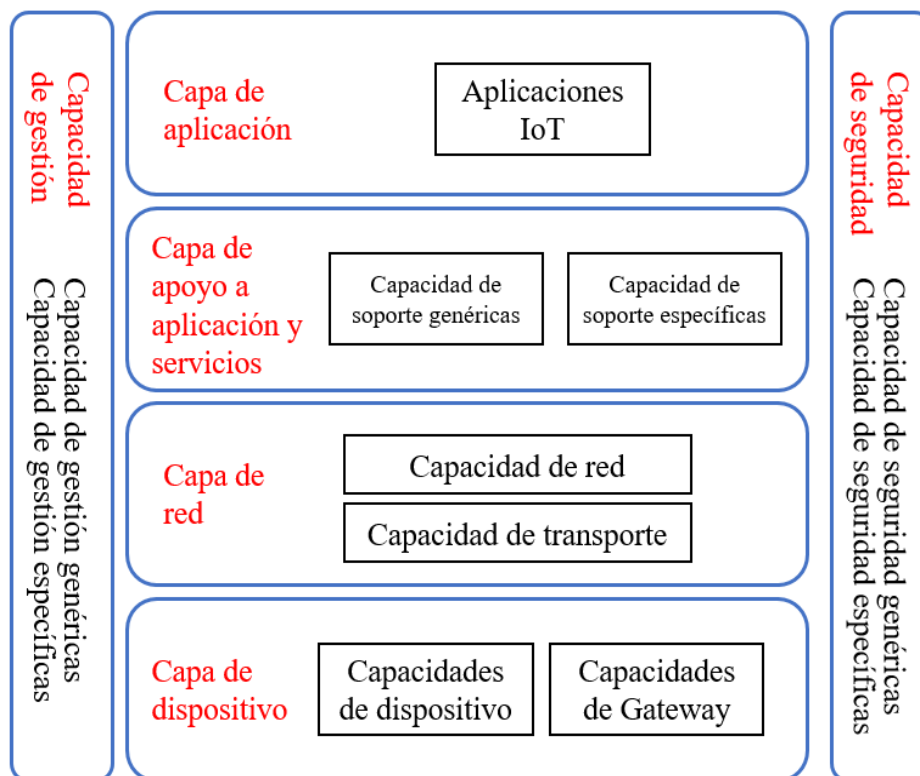


Figura 4: Modelo de referencia IoT.

Fuente: (Union Internacional de Telecomunicaciones, 2012)

- Capa de aplicación: También denominada de integración está formada por la aplicación, plataforma e integración que van a proporcionar el valor comercial al modelo y a la infraestructura IoT.
- Capa de soporte a las aplicaciones y los servicios. Esta capa del modelo de referencia tiene dos conjuntos de funciones:
 - Funciones generales de soporte: Son funciones generales que se pueden utilizar en diversos aplicativos IoT como procesamiento y el acopio de



información. E A partir de estas se pueden generar las capacidades específicas que se describen a continuación.

- Funciones específicas de apoyo: Son funciones a fin de satisfacer los requerimientos específicos de diversos aplicativos. En la práctica, pueden estar en diferentes conjuntos de funciones exactas que brindan diferentes capacidades para admitir diferentes aplicaciones de IoT.
- Capa de red: También esta capa cumple dos funciones:
 - Capacidad de red: Tiene el equivalente a las capas 3 y 4 del modelo OSI, es decir control del acceso y conectividad y algunas funciones de autorización y autenticación.
 - Capacidad de transporte: Para el transporte y la conectividad de la información a través de las aplicaciones y los servicios.
- Capa de aparatos: Como se evidencia en la Figura x3 también cumple dos funciones:
 - Capacidad de dispositivo: Aquí se realizan los procesos de recopilación de información, envío de los mismos y recepción de los datos de la red. Si es conveniente se puede definir aquí las redes adhoc y el control de consumo de potencia de los dispositivos.
 - Capacidad de Gateway: Para permitir la comunicación con distintos tipos de dispositivos de diferentes fabricantes sean alámbricos o inalámbricos o con diferentes tecnologías de comunicación.

De la figura 4 también se describe dos capas que son transversales a las mencionadas anteriormente: Gestión y seguridad. La capa de gestión está relacionada con la gestión en todas sus dimensiones: configuración, rendimiento, registro de actividades cotidianas y de fallas, cambios en la configuración. También para la detección y registro



de actividades no cotidianas como cambios en las configuraciones y situaciones internas y externas que tienen impacto en la red como el tráfico propio de la red y del proporcionado por los proveedores de servicio de internet.

La capa de seguridad no se relaciona directamente con cada capa, si no la controla a partir de los principios de autorización, autenticación y registro de los sucesos relacionados con la seguridad.

3. Modelo de comunicación entre las cosas

Para la comunicación entre las distintas “cosas” o dispositivos de la infraestructura de internet de las cosas el Organismo de Arquitectura Interna (IAB) en su RFC 7452 dicta cuatro modelos para la interconexión de las cosas: equipo a equipo, equipos a la nube, dispositivo a Gateway y modelo de intercambio de información por medio de back-end.

Equipo a equipo: no se necesita ningún otro dispositivo o plataforma para su comunicación. Aquí se ubican protocolos como: Bluetooth y ZigBee. Actualmente se utilizan en dispositivos con bajas tasas de transmisión de datos. La seguridad es baja y generalmente no son fácilmente compatibles con otros dispositivos de otros fabricantes.

- **Dispositivo a la nube:** Para la transferencia de datos se utiliza una plataforma o servicio proporcionado por su proveedor. Se basa generalmente en el protocolo IP y generalmente ofrece interfaces Web para la comunicación. Tiene la ventaja que, al ser un servicio estandarizado, puede funcionar casi con cualquier dispositivo de cualquier fabricante.
- **Dispositivo a Gateway:** Muy semejante al anterior, pero con la diferencia de que existe un dispositivo intermedio: Gateway que permite la conexión a la nube de algunos dispositivos que no tiene esta capacidad, como los denominados wearables. Así mismo permite interconexión entre distintos tipos de protocolos de comunicación como Bluetooth, ZigBee y LoRaWAN.



- **Intercambio de datos a través de back-end:** Para tener capacidades de portabilidad de los datos, proveedores, de servicios de comunicación y administración de IoT este modelo proporciona las capacidades de interoperabilidad para la mejora de estos procesos.

4. Comunicación y transporte

Como existen muchos protocolos y tecnologías que permiten el funcionamiento de IoT mencionaré algunos que son agrupados según (Rose, Eldridge, & Chapin, 2015) en tecnologías de acceso y protocolos de datos.

- **Tecnologías de acceso:** Aquí se encuentran los protocolos relacionados al transporte de la información obtenida por los sensores y actuadores: Bluetooth, Zwave, ZigBee para cortas distancias y bajas tasas de transmisión de datos. Así también WiFi con sus variantes, Lower Power Wide Area (LPWAN), Sigfox, LoraWan, LTE-M, NB-IoT, 4 y 5G.
- **Protocolos de datos:** Los encargados de tratar con la capa de aplicación. Tenemos a HTTP y HTTPS con RS y JSON para acceder a recursos de los servidores; MQTT y MQTT-SN para transmitir datos que requieren poco ancho de banda; CoAp similar a http pero utiliza UDP, por lo que es más simple; XMPP para mensajería instantánea; AMQP para publicar y suscribir con capacidades de seguridad; DDS, semejante al anterior pero que tienen capacidad de funcionar en tiempo real..

2.10. SERVIDOR LINUX

Es una variante del sistema operativo Linux diseñada específicamente a fin manejar las necesidades operativas y de almacenamiento más intensas de organizaciones más grandes y su software. Debido a su estabilidad, seguridad y flexibilidad, los servidores Linux se utilizan ampliamente en la actualidad y se consideran una de las



opciones más populares para servidores, superando a los servidores estándar de Windows (Urrego Morera, 2022).

La importancia de usar Linux en lugar de software de código cerrado como Windows es que Linux es de código abierto. Esto ayuda a mantener los costos de configuración y mantenimiento bajos. Muchas de las variantes patentadas en el sistema operativo Linux estándar, como Debian, CentOS, Ubuntu y Red Hat, ofrecen a los usuarios una flexibilidad significativa en términos de configuración, operación y mantenimiento de sus servidores.

Además, los servidores Linux son generalmente más ligeros que los sistemas operativos que requieren una interfaz gráfica, lo que los hace más adecuados para ejecutarse en servidores físicos y en la nube. Al no requerir una interfaz gráfica, los servidores Linux pueden ejecutarse con menos recursos, lo que puede ser beneficioso para organizaciones con presupuestos limitados o para aquellas que buscan maximizar la productividad y la eficacia de sus servidores.

2.10.1. Servidores web

El término "servidor web" puede hacer referencia al hardware y al software, o a la combinación de los dos trabajando en conjunto (Culqui Briceño, 2022).

En lo que respecta al componente físico, se puede definir un servidor web como una computadora especializada que alberga tanto el software del servidor web como los recursos que componen un sitio web, como páginas HTML, imágenes, hojas de estilo CSS y archivos JavaScript. Asimismo, este servidor está conectado a Internet y facilita la transferencia de información con otros dispositivos conectados a la red.

En cuanto al aspecto del software, un servidor web se compone de múltiples componentes encargados de supervisar cómo los usuarios acceden a los diversos archivos que alberga. Como parte esencial, el servidor web incluye un servidor HTTP, que es un



tipo de programa capaz de interpretar las URL (las direcciones web) y los registros HTTP (que los usuarios emplean para ver páginas web). A través de los nombres de dominio de los sitios web alojados, los usuarios pueden acceder al servidor HTTP, el cual envía el contenido de dichos sitios web al dispositivo del usuario final. Tenemos algunos ejemplos de servidores WEB:

- **Apache:** Es un servidor de internet o web de software libre y de código con acceso al público que posibilita a los usuarios alojar sus páginas web en la red de Internet. Además, es uno de los servidores web más longevos y fiables, mantenido por la Fundación Apache Software, cuya primera versión fue lanzada en 1995. La amplia aceptación de Apache lo convierte en la elección preferida tanto para los desarrolladores web como para las empresas que desean un servidor web confiable y escalable. Con su gran capacidad de personalización, Apache se adapta a los requerimientos específicos de cada sitio web, gracias a la gran cantidad de módulos y complementos disponibles. Además, su estabilidad, seguridad y flexibilidad son factores clave que contribuyen a la confiabilidad de Apache como servidor web. En pocas palabras, Apache es una opción altamente recomendable para aquellos que buscan un servidor web versátil y eficiente para alojar sus sitios web.
- **Nginx:** Es el servidor web de código abierto, ha ganado reconocimiento tanto su capacidad como servidor web, así como por su versatilidad en proxy inverso, caché HTTP e igualador de carga. Empresas destacadas como Autodesk, Atlassian, Intuit, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, Salesforce, VMWare, Xerox, LinkedIn, Cisco, Facebook, Target, Citrix Systems, Twitter, Apple e Intel, entre otras, confían en Nginx para sus necesidades de servidor web. Gracias a su enfoque en el perfeccionamiento del rendimiento a

grandes escalas, Nginx sobresale en comparación con otros servidores web populares, específicamente en situaciones que involucran contenidos estáticos y/o altas cargas de peticiones simultáneas. Esto explica la razón por la que Kinsta ha elegido utilizar Nginx como motor principal para su servicio de alojamiento.

2.11. RASPBERRY PI

Es una computadora pequeña que se puede sostener en la mano sin problemas. Esta computadora viene con un procesador ARM Cortex-A53 de cuatro núcleos de 1.4 GHz de 64 bits y 1 GB de RAM, con una conexión inalámbrica LAN de doble banda 2.4/5 GHz 802.11ac, Ethernet de 10/100/1000 Mbps, Bluetooth 4.2, 4 puertos USB, un puerto HDMI integrado, un puerto de audio compuesto de 3.5 mm y un puerto de video compuesto, además posee 40 pines GPIO, una abertura para tarjeta de memoria Micro SD, una unidad de procesamiento gráfico 3D con VideoCore IV, una interfaz de cámara (CSI) y una interfaz de pantalla (DSI) (Monk, 2021).

Los modelos de Raspberry Pi tienen una interesante convención de nomenclatura que los diferencia por el nombre del modelo y la generación. Entre los modelos se incluyen A, A+, B, B+, Zero y Compute Module. Cada modelo se diferencia por los conectores disponibles y el tamaño de la placa principal. Hasta el momento, se han construido varias generaciones.

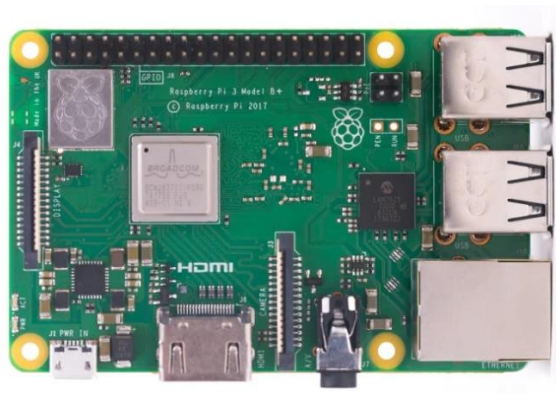


Figura 5: Raspberry Pi
Fuente: (Monk, 2021)

2.11.1. Raspberry pi como enrutador

Utilizar un Raspberry Pi como enrutador implica hacer uso de las capacidades del dispositivo para gestionar el tráfico de red entre dispositivos dentro de una red local. Un enrutador se ocupa de mandar y recibir paquetes de datos entre distintas redes, garantizando que la información llegue correctamente a su destino (Stone, 2019).

Al emplear un Raspberry Pi como enrutador, se obtiene un mayor control sobre la configuración de la red local, permitiendo llevar a cabo tareas como enrutamiento, firewall, asignación de señales IP y resolución de nombres de dominio. Además, se pueden aprovechar las capacidades de programación y personalización del Raspberry Pi para adaptar el enrutador a las necesidades específicas.

Sin embargo, también es importante tener en cuenta las limitaciones de rendimiento y capacidad de procesamiento del Raspberry Pi. Aunque puede funcionar bien como un enrutador básico en una red doméstica o pequeña, puede tener dificultades para manejar un tráfico de red intensivo o aplicaciones que requieran un alto rendimiento. Utilizar un Raspberry Pi como enrutador puede ser una opción interesante para proyectos pequeños o educativos, pero si se necesita un enrutador de alto rendimiento para una red empresarial o de gran tamaño, es recomendable considerar enrutadores dedicados y de mayor capacidad.

2.12. ARDUINO MEGA

Esta placa Arduino Mega 2560 se enfoca en el controlador integrado ATmega2560 y presenta una variedad de características. Incluye 54 pines digitales (15 de ellos se pueden configurar como salidas PWM), 16 entradas analógicas, 4 puertos seriales de dispositivos (UART), un oscilador de cristal de 16 MHz, conectividad USB, un conector para la fuente de alimentación, un encabezado ICSP y un botón de reinicio. Todos los componentes necesarios para respaldar el funcionamiento del microcontrolador

se encuentran en la placa, y esta puede conectarse a una computadora mediante un cable USB o recibir energía a través de un adaptador de corriente alterna a corriente continua o una batería. (Kurniawan, 2019).

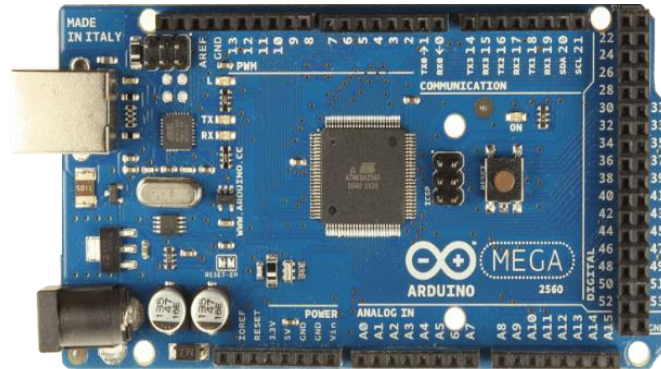


Figura 6: Arduino MEGA
Fuente: (Kurniawan, 2019)

2.12.1. Ide de Arduino

El desarrollo de la electrónica se ha vuelto más sencillo gracias a la combinación del software Arduino (IDE) y las placas Arduino (hardware). Juntos, permiten construir dispositivos digitales e interactivos utilizando otros componentes.

La aplicación de programación Arduino (IDE) es una herramienta de código abierto utilizada para programar las placas Arduino. Fue creada por arduino.cc y ofrece una plataforma completa para escribir y cargar código en estas placas. Además, cuenta con una amplia selección de bibliotecas y ejemplos de proyectos pequeños.

El software Arduino (IDE) es compatible con varios sistemas operativos, como Windows, Linux y Mac OS X, y es compatible con lenguajes de programación como C/C++. Es altamente accesible tanto para principiantes como para usuarios avanzados, ya que facilita el aprendizaje de la programación electrónica

2.13. ARDUINO MEGA

2.13.1. Ide de Arduino

y la robótica, y permite la creación de prototipos interactivos (Natheem S, 2022).

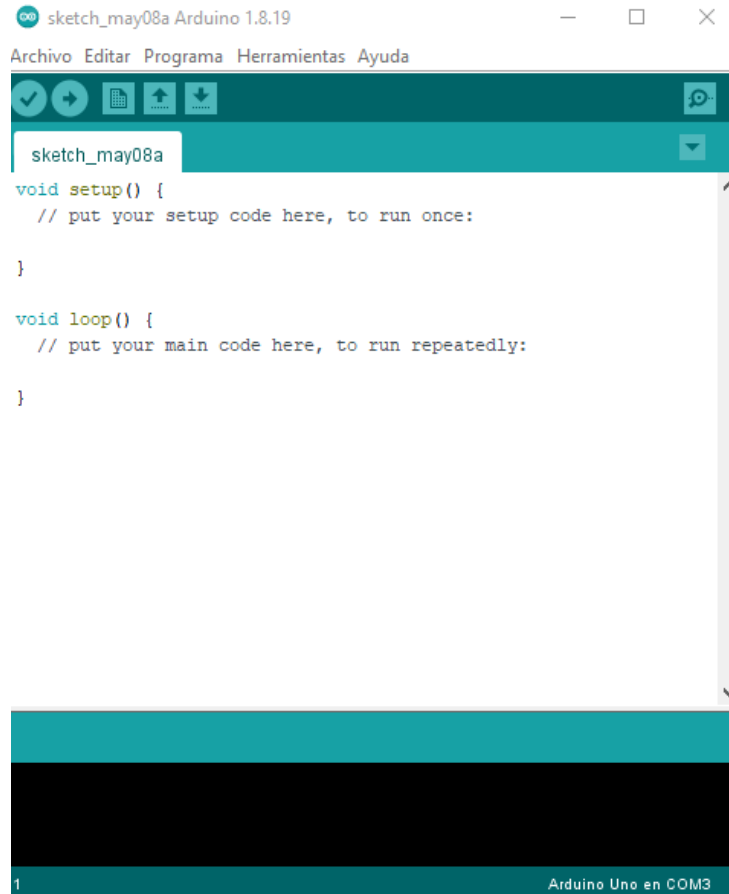


Figura 7: Entorno de programación Arduino
Elaboración propia

2.14. ETHERNET SHIELD

Es una herramienta el cual permite la conectividad a Internet a través de la biblioteca Ethernet de la placa Arduino. Esta biblioteca puede ser utilizada para escribir bocetos (programas de Arduino escritos en el IDE) que nos ayudarán a configurar este escudo para conectarse a la red. Este escudo es conciliable con casi la mayoría de las versiones de placas prototipados y permite que la placa reciba y envíe datos a todo el mundo, brindando una conexión a Internet estable (Pizarro Peláez, 2019).

Además, la opción de tarjeta SD también está disponible y podemos leer y escribir en esta tarjeta utilizando la biblioteca SD. Esto abre una cantidad infinita de posibilidades simplemente permitiendo que su proyecto se conecte a Internet.

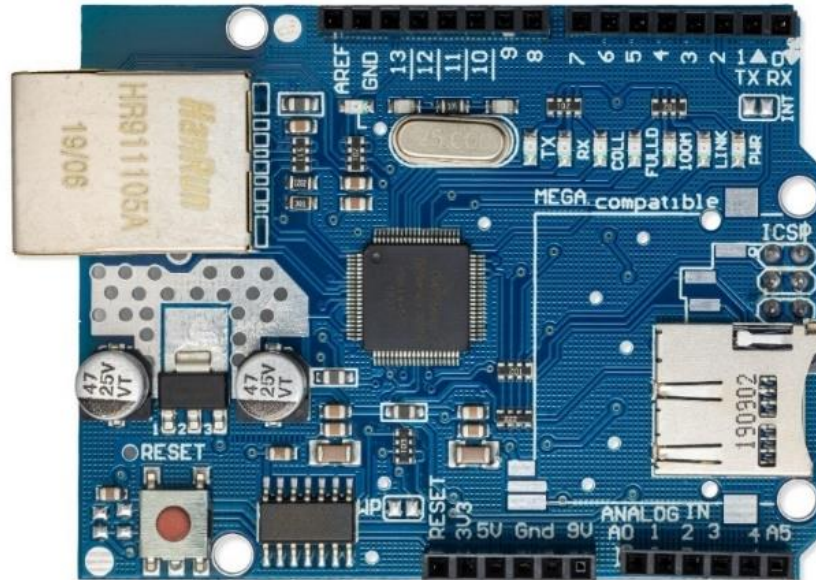


Figura 8: Ethernet Shield
Fuente: (Pizarro Peláez, 2019)

2.15. GOOGLE MAPS

Es un aplicativo de cartografía en línea que pertenece a Alphabet Inc. Esta herramienta ofrece mapas interactivos, fotografías satelitales de todo el mundo, indicaciones para llegar de un lugar a otro, fotos que se encuentran al nivel de las calles mediante Google Street View, información en lapso real sobre el tráfico (Google Traffic) y diversas opciones en navegación, como a través de las caminatas, en autos, en bicicleta (versión beta) y transportes públicos. Además, cuenta con un navegador GPS denominado Google Maps Go. También hay una versión para escritorio y un aplicativo denominado Google Earth, las cuales también son de acceso gratuito (Google, 2023).



CAPITULO III

MATERIALES Y METODOS

3.1. MATERIALES

3.1.1. Hardware

Computador:

- Modelo: HP
- Procesador: AMD A10-5745M APU with Radeon(tm) HD Graphics, 2100 MHz
- Memoria instalada (RAM): 8.00GB de RAM
- Sistema: Sistema Operativo de 64 bits con Windows Home

Arduino:

- Microcontrolador: ATmega328
- Voltaje Operativo: 5v
- Voltaje de Entrada (Recomendado): 7 – 12 v
- Pines de Entradas/Salidas Digital: 14 (De las cuales 6 son salidas PWM)
- Pines de Entradas Análogas: 6
- Memoria Flash: 32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocidad del Reloj: 16 MHZ.

3.1.2. Software

- Arduino IDE (Windows)
- Sistema Operativo de 64 bits con Windows Home
- Microsoft Word 2016 profesional

3.2. DISEÑO Y NIVEL DE LA INVESTIGACIÓN

3.2.1. Diseño de investigación

Esta investigación se basa en un enfoque experimental, el cual utiliza los principios científicos y los implementa en un prototipo. La esencia de este tipo de experimento implica la manipulación intencional de una acción para analizar los posibles resultados. Estos experimentos pueden llevarse a cabo tanto en un entorno de laboratorio como en situaciones reales. Los experimentos resultan especialmente eficaces para la investigación aplicada y explicativa, ya que permiten manipular las condiciones en las que las personas se encuentran, lo cual puede estar limitado a ciertos temas específicos (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2010).

3.2.2. Nivel de la investigación

El nivel de investigación es de carácter exploratorio, lo cual implica el grado de profundidad del conocimiento que se busca obtener a través de la investigación. Las investigaciones de naturaleza exploratoria se centran en abrir nuevas perspectivas en el desarrollo del conocimiento humano.

3.3. POBLACION Y MUESTRA DE LA INVESTIGACION

La población del presente estudio es finita, su composición está dada por el número de contenedores de desechos en la ciudad. El universo acotado o población está conformada por la cantidad de contenedores, que en este caso es 25. Para detectar una equivocación del 5% con un nivel de confiabilidad del 90%, se busca obtener muestras con una fórmula que permita medir la extensión adecuado de la población, estimando que la probabilidad de éxito es del 90%, mientras que la probabilidad de falla es del 10%. se calcula de la siguiente manera:

$$n = \frac{N \cdot Z_a^2 \cdot p \cdot q}{d^2 \cdot (N - 1) + Z_a^2 \cdot p \cdot q}$$

Donde:

N = tamaño de la población

Z_{α} = nivel de confiabilidad

p = probabilidad de logro proporción esperada

q = probabilidad de fallar

d = precisión (Error máximo admisible en términos de proporción)

Si tenemos los valores $N= 25$, $Z_{\alpha} = 1.645$ (Nivel de confianza 90%), $p = 0.9$,
 $q=0.1$ y $d=0.05$.

El resultado que muestra la fórmula es 20.05, lo que indica que para una población de 25 contenedores se tomará una muestra de 20 contenedores para este proyecto, este valor de muestra permite al investigador obtener los resultados deseados.

3.4. UBICACIÓN DE LA INVESTIGACIÓN

El trabajo de investigación se realizó en el distrito Puno en la provincia de Puno en la región de Puno, el cual posee las siguientes coordenadas geográficas: $15^{\circ}50'36''S$ $70^{\circ}01'25''O$ a 3818msnm. Para la investigación se tomó como referencia los contenedores públicos de desechos sólidos.



Figura 9: Ciudad de Puno
Elaboración propia



3.5. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCION DE DATOS

3.5.1. Técnicas

Existen varias técnicas de recopilación, pero en esta situación se usó la observación, la técnica de observación es un método de investigación en el cual se observa y registra el comportamiento de personas, animales o cosas, sin intervenir en el sistema observado. Puede ser realizada de forma estructurada, mediante una lista de comportamientos a observar, o de forma no estructurada, permitiendo la observación libre y el registro de todo lo que se vea relevante. Los datos tienen que ser verificados en el proceso de la investigación, en base a ello se hará el sistema de supervisión de la capacidad de contenedores de desechos sólidos.

3.5.2. Instrumentos

Un investigador puede utilizar varios instrumentos para recolectar datos mediante la técnica de observación, algunos ejemplos son:

- Registro de observación: Es una herramienta utilizada para registrar las observaciones en un formato estandarizado, ya sea mediante una lista de verificación o un formato abierto.
- Cámaras de video y audio: Es una herramienta para registrar la observación en un formato visual y/o auditivo.

En términos simples las herramientas que se utilizan para la recolección de datos son hardware y software que permita el almacenamiento de los datos como también la manipulación de estos, adicionalmente se verifican datos de eventos ocurridos en el procedimiento.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. CONTENEDOR DE RESIDUOS SOLIDOS

En la ciudad de Puno hay contenedores con gran capacidad de almacenamiento, sin embargo, para esta investigación hemos optado por utilizar un contenedor de residuos de tamaño estándar. De esta manera, podremos medir con mayor facilidad el peso de los residuos depositados en el mismo. En la imagen adjunta se muestra el contenedor utilizado para este fin.



Figura 10: Contenedor de residuos orgánicos
Elaboración propia

4.2. DISEÑO DE LA ESTRUCTURA PARA MEDICIÓN DEL PESO DE LOS RESIDUOS

Se tuvo en cuenta el funcionamiento del sensor de peso para diseñar la estructura. Cuando se aplica una fuerza sobre el sensor, las galgas extensiométricas se deforman ligeramente, ocasionando un cambio en su resistencia eléctrica. Este cambio de

resistencia eléctrica es medido por el puente de Wheatstone, el cual lo convierte en un cambio de voltaje que se amplifica y se lee mediante el Arduino. La lectura obtenida por el Arduino es posteriormente convertida en una medida de peso utilizando una fórmula de calibración.

Para desarrollar esta implementación, se parte de una base sólida donde se colocan los sensores de peso. Estos sensores cuentan con orificios para colocar los tornillos, y además se conectan amplificadores para la conexión con el Arduino. El proceso de armado se puede observar a continuación.

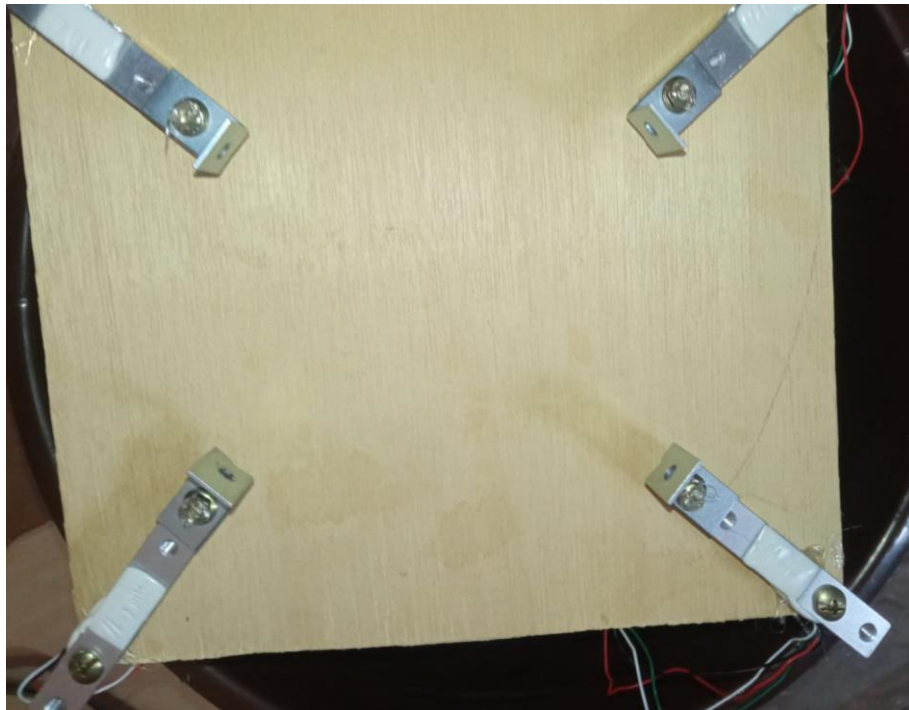


Figura 11: Base para los sensores
Elaboración propia

En la figura 12 se evidencia cuatro sensores colocados convenientemente. Para mejorar y no perjudicar a la deformación de la resistencia interna de los sensores se adiciona unos soportes conocidos como “ángulos de 90 grados”.

En medio de los sensores y el Arduino existe un componente conocido como “HX711”, que nos permite amplificar la señal adquirida de los sensores, ya que la señal de deformación en la resistencia del sensor es muy pequeña.

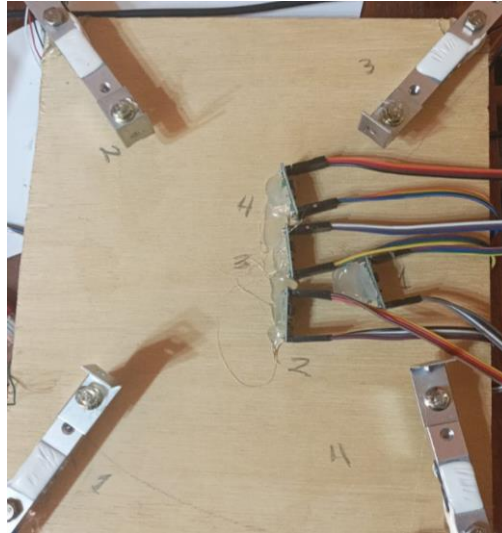


Figura 12: Ubicación de los sensores HX711
Elaboración propia

La figura 13 muestra 4 amplificadores ubicados de manera estratégica en la base de madera.

Mostramos la conexión de los sensores, los amplificadores y el Arduino MEGA

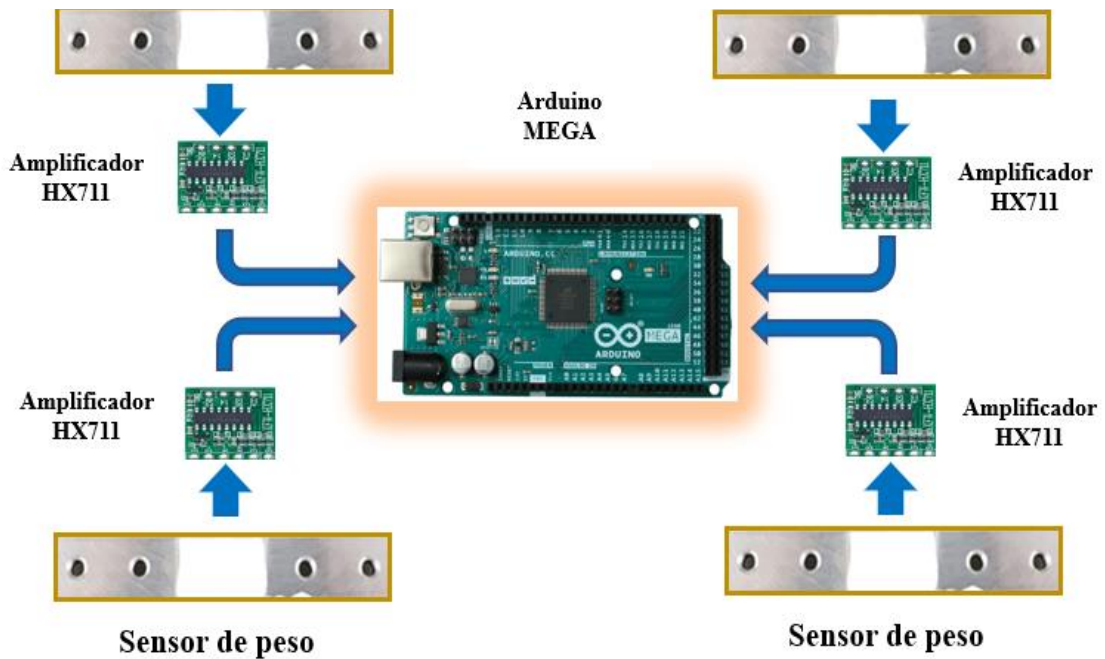


Figura 13: Diagrama de conexión de los sensores y el Arduino MEGA
Elaboración propia

Se hizo una cobertura final, el cual es el soporte y el que logra la deformación de los sensores, es en esta tapa donde se coloca los residuos de basura.



Figura 14: Cobertura de los sensores de peso
Elaboración propia

4.3. CALIBRACIÓN DE LOS SENSORES

El factor de calibración en Arduino para un sensor de peso es un valor numérico utilizado para ajustar la salida del sensor y obtener mediciones precisas de la carga que se está midiendo. Aunque cada sensor de peso tiene una salida analógica que se relaciona con la carga medida, el enlace exacto de la salida del sensor y la carga medida puede variar debido a factores ambientales como la temperatura y la variabilidad en la fabricación del sensor. Por lo tanto, es necesario calibrar el sensor de peso antes de utilizarlo en una aplicación para asegurarse de que la salida del sensor corresponda con la carga real que se está midiendo.

Para lograr la calibración hacemos la programación en el IDE de Arduino, entonces usamos la librería “hx711.h”, considerando que las librerías son colecciones de código previamente escritas que simplifican la tarea de programar y hacen que las tareas específicas sean más fáciles de realizar. Estas librerías se utilizan para extender las

capacidades de Arduino y brindar a los programadores acceso a herramientas de programación y funciones avanzadas para sus proyectos.

Adicionalmente usamos la función `#define` en Arduino es una macro de preprocesador que se utiliza para definir constantes y valores simbólicos en el código. Con `define`, se pueden establecer valores que no modifican en el momento de la ejecución del programa, lo que puede ser útil para definir números fijos, pines de entrada/salida, mensajes de texto, entre otros. Las constantes simbólicas definidas en nuestro código son DOUT Y CLK.

```
#include "HX711.h"  
#define calibration_factor -7050.0  
#define DOUT 3  
#define CLK 2
```

Figura 15: librería y variables simbólicas
Elaboración propia

Un objeto en Arduino es una instancia de una clase que representa un componente o entidad dentro del programa. Cada objeto tiene sus propias propiedades y métodos, que permiten interactuar con él y realizar acciones específicas. En el contexto de la programación de microcontroladores, los objetos son particularmente útiles para controlar sensores conectados a la placa y realizar mediciones precisas de variables físicas. De esta manera, los objetos permiten simplificar y modularizar el código, facilitando su lectura y mantenimiento.

Para nuestro código tenemos la instrucción `"HX711 scale(DOUT, CLK);"` instancia un objeto llamado "scale" de la clase "HX711", utilizando las constantes "DOUT" y "CLK" definidas previamente mediante la función "define". Este objeto está diseñado para representar el sensor de peso y se puede utilizar para llevar a cabo mediciones mediante los métodos proporcionados por la clase.

La función `void setup()` es una función en Arduino que se aplica una única vez al iniciar del programa. En esta función se define la rapidez de transmisión de información, la rapidez de transmisión se establece mediante la función `"Serial.begin()"`, que se encuentra en el `void setup` del programa. La velocidad predeterminada en Arduino es de 9600 baudios, pero se puede cambiar a valores más altos o más bajos de acuerdo a su utilidad para la aplicación, pero para este proyecto de investigación se usa la configuración por defecto.

La línea de código `"Serial.println("HX711 scale ");"` utiliza la función `"println"` de la clase `"Serial"` para imprimir la cadena de caracteres `"HX711 scale"` en la consola del monitor serie, seguida de un salto de línea. Esto permite visualizar en el monitor serie el estado del objeto `"scale"` de la clase `HX711`.

`scale.set_scale(calibration_factor)` en esta línea, se configura el factor de calibración del sensor de peso `"scale"` mediante la variable `"calibration_factor"`. De esta manera, se puede ajustar la salida del sensor para obtener mediciones precisas de la carga que se está midiendo.

La función de tara en los sensores de peso de Arduino permite reiniciar la medición a cero antes de realizar una lectura. En Arduino, se utiliza el método `"tare()"` de la clase del sensor de peso para establecer la tara en cero.

Y para finalizar el código en la función `void setup()`, imprimimos un mensaje en el monitor serial indicándole que imprima “Readings” o lecturas en español.

```
HX711 scale(DOUT, CLK);  
void setup() {  
  Serial.begin(9600);  
  Serial.println("HX711 scale ");  
  scale.set_scale(calibration_factor);  
  scale.tare();  
  Serial.println("Readings:");  
}
```

Figura 16: Creación del objeto HX711 y configuración en el `void setup()`
Elaboración propia

La función `loop()` en Arduino es esencial para el funcionamiento del programa ya que se ejecuta continuamente después de que se ha completado el cargo `setup()`. Dentro de la función `loop()` se escribe el código que se desea que se ejecute de forma repetida en un bucle infinito mientras la placa Arduino esté encendida. De esta manera, se asegura que el programa siga ejecutando las tareas necesarias mientras la placa esté en funcionamiento.

La línea de código `Serial.print(scale.get_units(), 1);`, se utiliza para imprimir la medición de peso obtenida a través del objeto "scale" utilizando el método `get_units()` en la consola del monitor serie. El segundo parámetro, "1", indica que solo se imprimirá un decimal de la medición, con el propósito de mejorar la precisión en la visualización de los datos.

Para imprimir las unidades del proyecto usamos `Serial.print(" kg");` en Arduino se utiliza para imprimir en la consola del monitor serie la unidad de medida del peso obtenido. En este caso, se imprime "kg", indicando que la medición de peso se realiza en kilogramos.

```
void loop() {  
    Serial.print("Reading: ");  
    Serial.print(scale.get_units(), 1);  
    Serial.print(" kg");  
    Serial.println();  
}
```

Figura 17: Configuración en el void loop()
Elaboración propia

El número "-564000" obtenido al ejecutar este código en Arduino representa el factor de calibración del sensor de peso, que se utiliza para obtener mediciones precisas de los residuos sólidos.

4.4. MEDICIÓN DEL PESO

Incluimos la librería hx711, y definimos como variables simbólicas a los pines 14, 15, 16, 17, 18, 19, 20, 21, estos se utilizan para leer el valor analógico de la deformación de la resistencia. Además, se define el valor obtenido en el código anterior.

```
#include "HX711.h"

#define calibration_factor -564000

#define LOADCELL_DOUT_PIN_1 14
#define LOADCELL_SCK_PIN_1 15
#define LOADCELL_DOUT_PIN_2 16
#define LOADCELL_SCK_PIN_2 17
#define LOADCELL_DOUT_PIN_3 18
#define LOADCELL_SCK_PIN_3 19
#define LOADCELL_DOUT_PIN_4 20
#define LOADCELL_SCK_PIN_4 21
```

Figura 18: Definición de variables simbólicas
Elaboración propia

Las líneas siguientes líneas de código en Arduino crean cuatro objetos de la clase HX711, llamados "scale1", "scale2", "scale3" y "scale4". Cada objeto representa un sensor de peso diferente que se conectará a la placa Arduino. Los objetos se utilizarán para realizar mediciones a través de métodos proporcionados por la clase HX711.

```
HX711 scale1;
HX711 scale2;
HX711 scale3;
HX711 scale4;
```

Figura 19: Creación de 4 objetos de HX711
Elaboración propia

Se definen tres variables de tipo float llamadas "total_lb", "total_kg" y "total_perc". Estas variables se utilizan para almacenar los cálculos realizados en el programa.

```
float total_lb;  
float total_kg;  
float total_perc;
```

Figura 20: Variables flotantes
Elaboración propia

El siguiente código de Arduino crea cuatro objetos de la clase HX711 para conectar cuatro sensores de peso diferentes. En primer lugar, se utiliza la función "begin()" para definir los pines de entrada y salida de datos correspondientes para cada sensor. Después, se establece el factor de calibración para cada objeto utilizando el método "set_scale()", lo que permite obtener mediciones precisas del peso. Finalmente, se utiliza la función "tare()" para establecer el peso actual en cero y permitir que las mediciones futuras reflejen únicamente la diferencia de peso desde ese punto.

```
void setup() {  
  
    scale1.begin(LOADCELL_DOUT_PIN_1, LOADCELL_SCK_PIN_1);  
    scale1.set_scale(calibration_factor);  
    scale1.tare();  
    scale2.begin(LOADCELL_DOUT_PIN_2, LOADCELL_SCK_PIN_2);  
    scale2.set_scale(calibration_factor);  
    scale2.tare();  
    scale3.begin(LOADCELL_DOUT_PIN_3, LOADCELL_SCK_PIN_3);  
    scale3.set_scale(calibration_factor);  
    scale3.tare();  
    scale4.begin(LOADCELL_DOUT_PIN_4, LOADCELL_SCK_PIN_4);  
    scale4.set_scale(calibration_factor);  
    scale4.tare();  
  
}
```

Figura 21: Se definen cuatro objetos de la clase HX711
Elaboración propia

El siguiente fragmento de código en Arduino realiza el cálculo del peso total a partir de las mediciones obtenidas por los cuatro sensores de peso, representados por los objetos "scale1", "scale2", "scale3" y "scale4" de la clase HX711. Los valores de cada sensor se suman y se almacenan en la variable "total_lb" como la suma total en libras.

Para obtener la medida en kilogramos, se divide "total_lb" por 2.2046 y se guarda en la variable "total_kg". Además, se calcula el porcentaje del peso máximo soportado por los sensores, que está establecido en 2 kg, dividiendo "total_kg" entre 2 y multiplicando por 100, el resultado se almacena en la variable "total_perc". Con esta información, se puede monitorear la carga de los sensores y prevenir posibles sobrecargas.

```
void loop() {  
  
total_lb = scale1.get_units() + scale2.get_units() + scale3.get_units() + scale4.get_units();  
total_kg = total_lb / 2.2046;  
total_perc = total_kg * 100 / 2;  
  
}
```

Figura 22: Cálculo del peso en lb y Kg
Elaboración propia

4.5. CONEXIÓN DE SHIELD ETHERNET AL SERVIDOR

Para transferir los datos obtenidos por nuestros sensores de peso a Internet, utilizamos una placa llamada Shield Ethernet. Esta placa se puede agregar a una placa de Arduino y permite la conexión a una red Ethernet.

Se utiliza la librería SPI.h para establecer comunicación con dispositivos externos mediante el protocolo SPI (Interfaz Serial Periférica). En este caso, se utiliza para establecer comunicación con la shield Ethernet y así conectar la placa de Arduino a una red Ethernet.

Por otro lado, la librería Ethernet.h es esencial para conectar la placa de Arduino a una red Ethernet a través de la shield Ethernet. Esta librería proporciona una API de red que permite al Arduino enviar y recibir datos a través de la red de manera eficiente y segura.

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

Figura 23: Librerías de conexión Ethernet
Elaboración propia



Definimos la dirección “mac” del servidor de destino, esta parte de código en Arduino define una variable de tipo byte llamada "mac" que contiene una dirección MAC. La dirección MAC es un identificador único asignado a la interfaz de red de un dispositivo y se utiliza para identificarlo en una red. En este caso, la dirección MAC es {0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05} y se utiliza para identificar la interfaz de red de la placa de Arduino conectada a la shield Ethernet.

Para el envío de información al servidor necesitamos necesariamente usar direcciones IP, tanto para el módulo de sensores de peso y el servidor. En el entorno de programación de Arduino, la línea de código "IPAddress server (108,62,123,70);" se utiliza para crear un objeto "IPAddress" llamado "server" y asignarle la dirección IP 108.62.123.70, también se define la dirección IP del módulo de sensores de peso.

Sabiendo que el gateway es una dirección IP que se utiliza en una red para enrutar los datos hacia destinos fuera de la red local. En otras palabras, actúa como una especie de "puerta" hacia otra red. En una red local, el gateway suele ser el enrutador que se conecta a Internet. Y para nuestro proyecto lo necesitamos definir en el programa.

La línea de código "byte subnet[] = { 255, 255, 255, 0 };" declara un array de bytes que se utiliza para almacenar los valores de una máscara de subred. Una máscara de subred es un número de 32 bits que se utiliza para determinar la porción de red y la porción de host de una dirección IP. En este caso, la máscara de subred es "255.255.255.0", lo que significa que los primeros tres octetos de la dirección IP se utilizan para identificar la red, mientras que el último octeto se utiliza para identificar los dispositivos de la red.

La línea de código "byte myDns[] = { 8, 8, 8, 8 };" es una declaración de un array de bytes en el lenguaje de programación de Arduino. Este array de bytes se utiliza para almacenar la dirección IP de un servidor DNS que se utilizará para traducir los nombres de dominio en direcciones IP. En esta investigación, la dirección IP almacenada en el

array es "8.8.8.8", que corresponde a uno de los servidores DNS públicos de Google. Cuando un dispositivo en la red necesita traducir un nombre de dominio en una dirección IP, envía una consulta al servidor DNS especificado en el código, en este caso, el servidor DNS de Google.

```
byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };  
  
IPAddress server(108,62,123,70);  
  
IPAddress ip(172,16,100,133);  
  
byte gateway[] = { 172, 16, 100, 1 };  
  
byte subnet[] = { 255, 255, 255, 0 };  
  
byte myDns[] = { 8, 8, 8, 8 };
```

Figura 24: Definimos direcciones IP, mac, Gateway, dns, subnet
Elaboración propia

La función "EthernetClient client;" declara una instancia de la clase EthernetClient. Esta clase se utiliza para establecer una conexión de red a través del protocolo TCP utilizando el módulo Ethernet del Arduino. La clase EthernetClient se utiliza para crear y administrar conexiones de red TCP entre el Arduino y otros dispositivos en la red. Una vez que se ha establecido una conexión TCP, se pueden enviar y recibir datos de forma bidireccional entre los dispositivos. Para utilizar la clase EthernetClient, primero se debe establecer una conexión Ethernet utilizando el módulo Ethernet y luego crear una instancia de la clase EthernetClient. Luego, se puede utilizar la instancia de la clase para conectarse a un servidor remoto y enviar y recibir datos a través de la conexión TCP.

"bool printWebData = true;" declara una variable booleana llamada "printWebData" y le asigna el valor "true". Una variable booleana es un tipo de variable que solo puede tener dos valores posibles: "true" o "false". En este caso, la variable "printWebData" se utiliza para indicar si se debe imprimir o no los datos recibidos de un



servidor web. Al asignar el valor "true" a la variable, se indica que se debe imprimir los datos recibidos del servidor web. Si la variable se hubiera establecido en "false", entonces los datos no se imprimirían.

"char sensor_id[] = "MOVIL-01";" declara una variable llamada "sensor_id" del tipo de dato cadena de caracteres (string) y le asigna el valor "MOVIL-01". Esta variable se utiliza para almacenar un identificador único para un sensor o dispositivo en un programa de Arduino. El tipo de dato "char" se utiliza para representar caracteres individuales dentro de una cadena de caracteres. Por lo tanto, "char sensor_id[]" define un arreglo de caracteres que puede contener múltiples caracteres. En resumen, la línea de código hace una declaración de la variable "sensor_id" que utiliza una cadena de caracteres para almacenar un identificador único para un sensor o dispositivo en un programa de Arduino. El identificador se almacena como una secuencia de caracteres dentro de un arreglo de caracteres.

La función Ethernet.begin() se utiliza para establecer la conexión de red y se le pasan varios parámetros:

- mac: Es la dirección MAC única asignada a la interfaz Ethernet.
- ip: Es la dirección IP asignada a la placa Arduino.
- myDns: Es la dirección IP del servidor DNS.
- gateway: Es la dirección IP de la puerta de enlace o router que conecta la red local a Internet.
- subnet: Es la máscara de subred que se utiliza para dividir la dirección IP en dos partes, una que identifica la red y otra que identifica el host.

```
EthernetClient client;

bool printWebData = true;

char sensor_id[] = "MOVIL-01";

void setup() {

    Ethernet.begin(mac, ip, myDns, gateway, subnet);
}
```

Figura 25: Se establece la conexión TCP
Elaboración propia

La sentencia condicional `if` utiliza el método `connect()` de la clase `EthernetClient`. Esta línea de código se utiliza para establecer una conexión con un servidor web en un puerto específico.

La función `connect()` se utiliza para establecer una conexión a un servidor web utilizando el protocolo HTTP y se le pasan dos parámetros:

- `server`: Es un objeto de tipo `IPAddress` que contiene la dirección IP del servidor web al que se desea conectarse.
- `port`: Es el número de puerto que utiliza el servidor web para escuchar las solicitudes de los clientes. En este caso, se está utilizando el puerto 80, que es el puerto estándar para las solicitudes HTTP.

La sentencia condicional `if` se utiliza para evaluar si la conexión se ha establecido correctamente. Si la conexión es exitosa, la función `connect()` devuelve un valor verdadero (`true`), lo que significa que la sentencia `if` se ejecutará. Si la conexión falla, la función `connect()` devuelve un valor falso (`false`), lo que significa que la sentencia `if` no se ejecutará y se procederá a la siguiente instrucción después de la llave cerrada.

En resumen, la sentencia `if (client.connect(server, 80))` se utiliza para verificar si la conexión con el servidor web en la dirección IP y el puerto especificados ha sido establecida correctamente y si es así, se ejecuta el bloque de código dentro de las llaves `{ }`. Si la conexión no se establece correctamente, la sentencia `if` no se ejecutará.



Luego de haber establecido la conexión con el servidor llamamos a métodos del objeto client, que es una instancia de la clase EthernetClient en Arduino. Estas llamadas se utilizan para enviar una solicitud HTTP a un servidor web en una dirección IP específica. El código construye una URL con algunos parámetros que se envían al servidor web. La URL se construye mediante varias llamadas a los métodos print() del objeto client, que agregan texto y variables a la solicitud HTTP.

Hacemos una explicación detallada de cada línea de código:

- **client.print("GET /LOADCELL/formaction_loadcell.php?total_kg=");:** Esta línea de código agrega una cadena de texto que indica el tipo de solicitud HTTP que se está realizando (en este caso, una solicitud GET). Luego, se agrega la ruta relativa al archivo PHP que procesará los datos del sensor de peso, seguida del primer parámetro que es "total_kg".
- **client.print(total_kg);:** Esta línea de código agrega el valor de la variable total_kg a la solicitud HTTP. total_kg es una variable que contiene el peso total leído por el sensor de peso.
- **client.print("&total_perc=");:** Esta línea de código agrega una cadena de texto que separa los parámetros de la URL y agrega el segundo parámetro "total_perc".
- **client.print(total_perc);:** Esta línea de código agrega el valor de la variable total_perc a la solicitud HTTP. total_perc es una variable que contiene el porcentaje de peso total leído por el sensor de peso.
- **client.print("&sensor_id=");:** Esta línea de código agrega una cadena de texto que separa los parámetros de la URL y agrega el tercer parámetro "sensor_id".
- **client.print(sensor_id);:** Esta línea de código agrega el valor de la variable sensor_id a la solicitud HTTP. sensor_id es una variable que contiene el ID único del sensor de peso.

- **client.println(" HTTP/1.1");**: Esta línea de código agrega una cadena de texto que indica la versión del protocolo HTTP que se está utilizando.
- **client.print("Host: ");**: Esta línea de código agrega una cadena de texto que indica el nombre de host del servidor web.
- **client.println(server);**: Esta línea de código agrega la dirección IP del servidor web que se especificó anteriormente en el código.
- **client.println("Connection: close");**: Esta línea de código agrega una cadena de texto que indica que la conexión se cerrará después de que se haya enviado la solicitud HTTP.
- **client.println();**: Esta línea de código agrega una línea en blanco que indica el final de la solicitud HTTP.

En pocas palabras, este fragmento de código se utiliza para construir y enviar una solicitud HTTP GET al servidor web con algunos parámetros y datos que se leyeron del sensor de peso. El servidor web procesará esta solicitud y responderá con una respuesta HTTP correspondiente. Adicionalmente si no se logra la conexión con el servidor simplemente le decimos que no realice ninguna acción con la sentencia else.

```
if (client.connect(server, 80)) {  
  
    client.print("GET /LOADCELL/formation_loadcell.php?total_kg=");  
    client.print(total_kg);  
    client.print("&total_perc=");  
    client.print(total_perc);  
    client.print("&sensor_id=");  
    client.print(sensor_id);  
    client.println(" HTTP/1.1");  
    client.print("Host: ");  
    client.println(server);  
    client.println("Connection: close");  
    client.println();  
  
} else {  
  
}
```

Figura 26: Conexión con el servidor
Elaboración propia

El siguiente código es una estructura de control de bucle que se utiliza para mantener la conexión del cliente con el servidor activa mientras el servidor web todavía está disponible. En resumen, el código ejecuta lo siguiente:

- El bucle `while (client.connected())` se ejecuta siempre que la conexión del cliente con el servidor web esté activa.
- La variable `len` se inicializa con el número de bytes disponibles para leer en el búfer de recepción de la conexión.
- Si hay datos disponibles para leer (`len > 0`), se lee un máximo de 80 bytes del búfer utilizando la función `client.read(buffer, len)`.
- Si la variable `printWebData` es verdadera, se procesa y muestra la respuesta del servidor web, que se encuentra en el búfer de recepción.
- Si la conexión se interrumpe (`!client.connected()`), el cliente detiene la conexión con `client.stop()`, espera 28358 milisegundos y luego continúa con la ejecución del programa.

En resumen, el código se utiliza para leer la respuesta del servidor web mientras se mantiene la conexión activa y se detiene la conexión cuando se completa la lectura.

```
while (client.connected()) {  
  int len = client.available();  
  if (len > 0) {  
    byte buffer[80];  
    if (len > 80) len = 80;  
    client.read(buffer, len);  
    if (printWebData) {  
  
    }  
  }  
}  
  
if (!client.connected()) {  
  client.stop();  
  delay(28358);  
}
```

Figura 27: Se mantiene la conexión con el Servidor
Elaboración propia



4.6. VERIFICACIÓN DE LOS PESOS EN LA NUBE

Considerando que el término "nube" se refiere a la capacidad de almacenar y recuperar datos a través de Internet, podemos concluir que la nube desempeña un rol esencial en la gestión y conservación de información en la era digital. La nube proporciona una solución práctica y conveniente para el resguardo de datos, ya que permite su acceso desde cualquier lugar y en cualquier momento, siempre que haya una conexión a Internet disponible. Esto elimina la necesidad de depender de dispositivos físicos, como discos duros o unidades de almacenamiento, y disminuye el riesgo de perder datos debido a problemas técnicos o accidentes.

La seguridad de los datos en la nube también es una preocupación clave. Los proveedores de servicios en la nube implementan medidas de seguridad robustas, como encriptación de datos, autenticación de usuario y copias de seguridad redundantes, para proteger la información almacenada. Sin embargo, es importante tener en cuenta las políticas de privacidad y seguridad del proveedor y tomar precauciones adicionales, como utilizar contraseñas seguras y mantener actualizados los sistemas de protección contra malware.

Para la verificación de los pesos en la nube es necesario aclarar que el servidor que se usa en esta investigación es de pago y el servicio solo nos permitirá verificar los datos en la nube mas no podremos hacer modificaciones en ello.

Los servidores Linux de pago son servicios proporcionados por proveedores especializados en alojamiento web, en los cuales los usuarios pagan una tarifa para acceder a un servidor virtual o dedicado que utiliza el sistema operativo Linux. Además, los servidores Linux de pago suelen ofrecer un rendimiento y estabilidad superiores en comparación con el alojamiento compartido, dado que los recursos del servidor se asignan exclusivamente al usuario o grupo de usuarios que contratan el servicio.

El siguiente diagrama resume el prototipo desarrollado, donde se muestra que los datos del contenedor de residuos sólidos son procesados por el Arduino Mega. Estos datos se envían al "shield Ethernet", que permite la conexión con el router. El router, a su vez, redirecciona los datos a internet, lo que nos posibilita acceder a ellos desde cualquier parte del mundo. En nuestro caso, los operarios de los camiones recolectores utilizarán estos datos, ya que estarán almacenados en un servidor accesible para ellos.

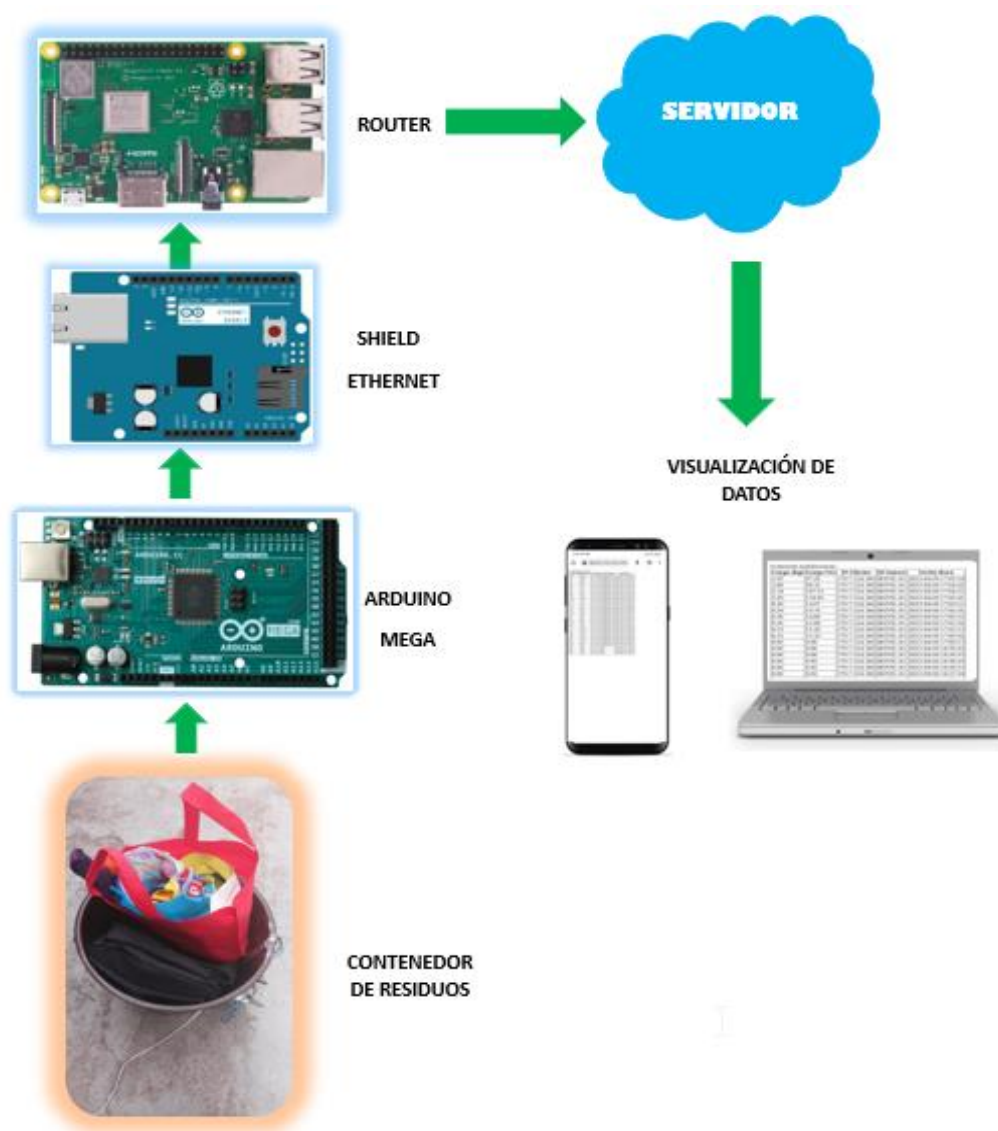


Figura 28: Diagrama de conexión del prototipo
Elaboración propia

Un prototipo completamente integrado, refiere que se han unido y combinado todas las partes esenciales, tales como hardware, software, sensores, actuadores y otros

componentes pertinentes, con el fin de lograr el funcionamiento conjunto del producto o sistema. Esto implica que todas las funcionalidades y características previstas están presentes y operativas en el prototipo.

La integración total del prototipo permite evaluar y poner a prueba su rendimiento en condiciones similares a las reales, ya que se simulan o reproducen las interacciones entre los componentes y se verifica la viabilidad y eficacia de la solución propuesta. Además, facilita la detección temprana de posibles problemas de compatibilidad, interconexión o funcionamiento antes de avanzar hacia la etapa de producción.



Figura 29: Prototipo integrado totalmente
Elaboración propia

La primera prueba la hacemos sin residuos en el contenedor, veamos.



Figura 30: Contenedor sin residuos
Elaboración propia

En el servidor podemos visualizar el resultado del peso, que este caso nos muestra un valor de 0 Kg.

Conexión Satisfactoria!

Carga (Kg)	Carga (%)	IP Cliente	ID Sensor	Fecha Hora
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:43
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:44
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:14
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:56:44
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:56:14
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:55:44
0.00	0.03	179.7.224.186	MOVIL-01	2023-04-06 16:55:15
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:54:45
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:54:15
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:53:45
0.00	0.03	179.7.224.186	MOVIL-01	2023-04-06 16:53:16
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:52:46

Figura 31: Resultados del peso sin residuos en el contenedor
Elaboración propia

La segunda prueba con residuos sólidos, específicamente con una carga de 15% del total.



Figura 32: Contenedor con residuos
Elaboración propia

Los resultados en el servidor muestran la carga en la imagen.

Conexión Satisfactoria!

Carga (Kg)	Carga (%)	IP Cliente	ID Sensor	Fecha Hora
0.31	15.43	179.7.224.186	MOVIL-01	2023-04-06 17:01:12
0.31	15.33	179.7.224.186	MOVIL-01	2023-04-06 17:00:42
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 17:00:13
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:43
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:43
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:44
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:14
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:56:44
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:56:14
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:55:44
0.00	0.03	179.7.224.186	MOVIL-01	2023-04-06 16:55:15
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:54:45
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:54:15
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:53:45

Figura 33: Resultados con 15% de contenido
Elaboración propia

Se hace la tercera prueba final donde se llena el contenedor.



Figura 34: Prueba con contenedor lleno
Elaboración propia

El peso del contenedor la vemos en el servidor.

Conexión Satisfactoria!

Carga (Kg)	Carga (%)	IP Cliente	ID Sensor	Fecha Hora
1.95	97.29	179.7.224.186	MOVIL-01	2023-04-06 17:05:10
1.88	94.12	179.7.224.186	MOVIL-01	2023-04-06 17:04:41
2.14	107.21	179.7.224.186	MOVIL-01	2023-04-06 17:04:11
2.20	110.05	179.7.224.186	MOVIL-01	2023-04-06 17:03:41
0.30	14.97	179.7.224.186	MOVIL-01	2023-04-06 17:03:11
0.29	14.75	179.7.224.186	MOVIL-01	2023-04-06 17:02:41
0.30	14.89	179.7.224.186	MOVIL-01	2023-04-06 17:02:12
0.30	15.08	179.7.224.186	MOVIL-01	2023-04-06 17:01:42
0.31	15.43	179.7.224.186	MOVIL-01	2023-04-06 17:01:12
0.31	15.33	179.7.224.186	MOVIL-01	2023-04-06 17:00:42
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 17:00:13
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:43
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:43
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:44

Figura 35: Pesos con el contenedor lleno
Elaboración propia

El servidor muestra una página web donde se visualiza el peso en Kg, el porcentaje de carga, IP del cliente, ID del sensor, la fecha y la hora. Estos datos obtenidos se muestran periódicamente, esto está definido en el programa de Arduino desarrollado.

Recibir datos periódicamente en el servidor brinda beneficios como el monitoreo en tiempo real, la toma de decisiones basada en datos, el mantenimiento predictivo, la eficiencia operativa, la mejora de la experiencia del usuario.

4.7. RUTAS POSIBLES A SEGUIR SEGÚN LA UBICACIÓN DE LOS CONTENEDORES DE LA CIUDAD DE PUNO

El uso de fotografías de los contenedores de la ciudad de Puno brinda la posibilidad de determinar su nivel de llenado y estado. Una vez que se obtiene esta información, se puede generar un mapa o una visualización que muestre el estado de cada contenedor en tiempo real. Esto brinda una visión general de la distribución de los contenedores llenos y vacíos en la ciudad de Puno, lo que facilita la planificación eficiente de la recolección de residuos.

Se ha fotografiado contenedores de la ciudad de Puno:

Avenida Bolívar:



Figura 36: Contenedor AV. Simón Bolívar
Elaboración propia

Avenida el Ejercito:



Figura 37: Contenedor Avenida el ejercito
Elaboración propia

Jr. Beltrán Rivera:



Figura 38: Contenedor próximo al Parque del niño
Elaboración propia

Leoncio Prado:



Figura 39: Contenedor próximo a Colegio Villa de Lago
Elaboración propia

Contenedores Avenida circunvalación:



Figura 40: Contenedor próximo a EMSA PUNO
Elaboración propia



Figura 41: Contenedores próximo al mercado Huajsapata
Elaboración propia

Avenida El Sol:



Figura 42: Contenedor próximo al mercado Bellavista
Elaboración propia

Avenida Floral:



Figura 43: Contenedor próximo al tobogán
Elaboración propia

Google Earth brinda la posibilidad de explorar y visualizar el planeta Tierra en un formato tridimensional, ofreciendo una representación detallada y realista de nuestro mundo. Esta aplicación obtiene imágenes de satélite y fotografías aéreas provenientes de diversas fuentes, como satélites comerciales, organizaciones gubernamentales y colaboraciones con instituciones. Estas imágenes son capturadas desde distintos ángulos y altitudes, lo que permite obtener una vista completa y detallada del planeta.

La ubicación de los contenedores de la ciudad se muestra en el siguiente mapa, los contenedores están clasificados en colores, sabiendo que:

- **Color rojo:** El contenedor está por encima de su capacidad.
- **Color amarillo:** El contenedor está en media capacidad.
- **Color verde:** El contenedor está vacío.



Figura 44: Ubicación de contenedores en la ciudad de Puno
Elaboración propia

Con el objetivo de optimizar el proceso de recolección de residuos, se ha implementado una nueva estrategia para los camiones recolectores. A partir de ahora, estos vehículos únicamente realizarán el recorrido por los contenedores que se encuentren a medio llenar y aquellos que estén completamente llenos. Esta medida busca maximizar la eficiencia en la recolección de residuos, evitando desplazamientos innecesarios y optimizando el uso de recursos de combustible. Al enfocarse en los contenedores a medio llenar, se evita que sobrepase su capacidad, lo que permite un uso más eficiente del tiempo y los recursos de recolección. Asimismo, la recolección de los contenedores completamente llenos garantiza que no se produzcan desbordamientos ni acumulación de residuos en las áreas de disposición. Con esta nueva estrategia, se espera reducir los costos de operación y minimizar el impacto ambiental asociado con la recolección de residuos. Además, se fomenta una mayor conciencia sobre la necesidad de utilizar los contenedores de manera adecuada, evitando su sobrecarga y promoviendo una gestión responsable de los desechos. Esta iniciativa es un paso importante hacia la mejora continua en la gestión

de residuos, y se espera que tenga un impacto positivo en la eficiencia de la recolección, la reducción de costos y el cuidado del medio ambiente.

Gracias al desarrollo y aplicación del prototipo, ahora contamos con la capacidad de generar rutas de recolección de residuos basadas en la cantidad de basura presente en los contenedores. Esta innovadora solución utiliza una nomenclatura de colores para indicar los niveles de capacidad de los contenedores, siendo el amarillo para aquellos que se encuentran a media capacidad y el rojo para aquellos que están desbordados. La implementación de esta metodología de colores permite a los camiones recolectores priorizar la recolección de residuos sólidos de los contenedores amarillos y rojos, asegurando una recolección eficiente y oportuna en función de las necesidades de cada punto de recolección.

La ruta debe seguir el camión recolector es por que aquellos contenedores representados de color amarillo y de color rojo.



Figura 45: Ruta generada de acuerdo a la capacidad de los contenedores
Elaboración propia



En la investigación de Durán en el 2019, se examinaron las principales iniciativas en sistemas de administración de redes para Internet de las Cosas y se identificaron los requisitos esenciales para gestionar dispositivos en cualquier implementación. Estos dispositivos abarcan el aprovisionamiento e identificación, estructuración y control, monitoreo y diagnóstico, y actualización y sostenimiento de software (Duran, 2019). La presente investigación, a diferencia de la de Durán, se usaron sistemas de administración de redes para Internet de las Cosas y se realiza el monitoreo o supervisión y diagnóstico del estado actual de la aplicación IoT.

Gahona y Gavilema en el 2020 han realizado un estudio sobre el desarrollo de una red de Internet de las Cosas (IoT) para un edificio empresarial, donde señalan que actualmente se están implementando procesos de mejoras de las redes a redes IoT con el fin de ofrecer soluciones oportunas a la población, empleando tecnología inalámbrica e integración con aplicaciones de notificación en caso de acceso no permitido, hurto y alarmas de evacuación en casos de catástrofes naturales (Gaona & Gavilema, 2020). La presente investigación, a diferencia de la de Gahona y Gavilema, han realizado un estudio sobre el desarrollo de una red IoT para una ciudad o distrito, supervisando los elementos IoT con tecnología inalámbrica pero más orientada a la gestión de residuos sólidos.

Bocanegra, Gamarra y Tipian en el 2020, en su informe acerca de la administración de desechos sólidos en el Perú durante el período del COVID-19, indican que, en el Perú, se produce más de 7 millones de toneladas de residuos sólidos urbanos al año, aproximadamente 20,000 toneladas por día y casi 1,000 toneladas por hora. Del total de estos desechos, el 70% proviene de los hogares. Durante la pandemia, fue crucial mantener el aislamiento social y promover la reactivación económica (Bocanegra, Gamarra, & Tipian, 2020). La presente investigación, en complemento con la de Bocanegra, Gamarra y Tipian, se gestiona los residuos sólidos con la tecnología IoT,



señalando la importancia que tiene su gestión para actividades económicas, sociales, para la salud y otros.

García y Socorro en su artículo del año 2019, realizan el estudio de caso acerca de la situación de la utilización de los desechos sólidos en la ciudad de Machala, Ecuador. Tomaron como propósito dos localizaciones: un mercado y una urbanización privada, realizando el estudio situacional para ambos sectores. A partir de los análisis realizados se dio un diagnóstico situacional a partir de encuestas y entrevistas personales. Luego se propuso un plan de administración y manejo integral de los desechos, lo que permitirá el cumplimiento de la ley respecto al medio ambiente, contribuyendo a un ambiente sano y saludable (García, Socorro, & Vanessa, 2019). La presente investigación, en complemento con la de García y Socorro, se administra los residuos sólidos con la tecnología IoT, señalando la importancia que tiene su administración para cumplir leyes y permitir la sostenibilidad de la localidad.



V. CONCLUSIONES

El diseño de un sistema de supervisión de la capacidad de los contenedores de residuos sólidos urbanos tiene un impacto significativo, debido a que se obtiene una visión precisa del nivel de llenado de contenedores. Por lo tanto, este sistema de supervisión desempeña un papel fundamental que optimizará la recolección de residuos y minimizará el impacto ambiental, lo que a su vez contribuye a la creación de una ciudad más sostenible.

Se diseñó un sistema de gestión de contenedores de residuos sólidos urbanos, lo que permitió planificar las rutas de manera eficiente, sin la necesidad de realizar viajes innecesarios y de esta manera poder ahorrar recursos como el combustible y el tiempo. La solución es compacta económico, de alto rendimiento energético y la conectividad a Internet está verificada y se puede lograr de muchas formas.

De los resultados obtenidos se observó cómo el prototipo desarrollado con Arduino genera la mejor opción de ruta para la recolección de residuos sólidos urbanos. Además, la información del estado de los contenedores se puede verificar en un servidor Linux que está nube al que se puede acceder desde diversos dispositivos como celulares o computadoras.



VI. RECOMENDACIONES

Capacitar a los operarios con el propósito de garantizar un adecuado funcionamiento del sistema y maximizar sus beneficios. Mediante la capacitación, se asegura que los operarios estén plenamente informados, preparados y comprometidos en el uso efectivo del sistema, lo que a su vez promoverá una gestión más eficiente de los residuos y el logro de los objetivos planteados.

Establecer alianzas con instituciones, organizaciones gubernamentales, municipios y entidades relacionadas con el medio ambiente y la gestión de residuos. Por consiguiente, estas alianzas permitirán ampliar el alcance del proyecto, compartir conocimientos y recursos, y generar un mayor impacto en la implementación del prototipo a nivel regional y nacional.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Altamirano-González, M., Manotoa-Balseca, F., Rodríguez-Escobar, P., & Hidalgo-Calero, A. (2022). Desarrollo de un sistema de control para la automatización de una máquina de ensayo universal. *Dominio de las Ciencias*, 8(2), 190-205.
doi:<http://dx.doi.org/10.23857/dc.v8i1.2640>
- Avalos, L. (2020). *Diseño de un plan de gestión de residuos sólidos domésticos para el balneario de Puerto Malabrigo, distrito de Razuri 2019*.
- Bocanegra, K., Gamarra, F., & Tipian, P. (Julio de 2020). *Defensoría del Pueblo*.
Obtenido de Gestión de los Residuos Sólidos en el Perú en tiempos de Covid-19:
<https://www.defensoria.gob.pe/wp-content/uploads/2020/07/Informe-Especial-N%C2%B0-24-2020-DP.pdf>
- Callirgos, L., & Mendez, D. (2015). *Gestión integral para el tratamiento de residuos sólidos en el distrito de Trujillo, provincia de Trujillo, la Libertad*. Universidad Privada Antenor Orrego.
- Cardona-Castro, E. (2021). *Diseño e implementación de un módulo de adquisición de datos para un secador convectivo*.
- Cornejo, R. (2018). *Recolección y transporte de residuos sólidos urbanos y su relación con el pago del servicio en la ciudad de Puno*. Repositorio institucional de la Universidad Nacional del Altiplano.
<http://repositorio.unap.edu.pe/handle/UNAP/8815>.
- Culqui Briceño, W. A. (2022). *Evaluación de rendimiento carga y stress de servidores web Linux, bajo la norma ISO/IEC 25000*. Pimentel - Perú: Universidad Señor de Sipan .



- Duran, M. (2019). *Sistema de Gestión de Red Para Internet de las Cosas*. Repositorio institucional de la Pontificia Universidad Javeriana.
- EIRL, T. (2017). *TESLAELECTRONIC*. Obtenido de Sensor de carga HX711:
<https://www.teslaelectronic.com.pe/producto/sensor-de-carga-hx711/>
- Flores, Y., & Huanca, A. (2018). *Estrategias comunicacionales y manejo de los residuos sólidos en la gestión medio ambiental de la ciudad de Azángaro 2018*. Repositorio institucional de la Universidad Nacional del Altiplano.
<http://repositorio.unap.edu.pe/handle/UNAP/10716>.
- Gaona, R., & Gavilema, A. (2020). *Diseño de la red de Internet de las Cosas (IoT) para el edificio de la empresa Consel*. Universidad Politécnica Salesiana Sede Quito.
- García, R., Socorro, A., & Vanessa, A. (2019). Control and environmental management of solid waste, case study. *Revista Universidad y Sociedad*, 11(1), 265-271.
- Gomez-Diaz, L. (2020). Herramienta para la Supervisión y Gestión del Consumo de Energía y Agua en el Hogar, Aplicando la Sostenibilidad Inductiva. *MODUM*, 2(1), 77-88.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2010). *Metodología de la Investigación*. Mexico D.F.: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
- INEI. (Enero de 2021). *Perú Anuario de estadísticas Ambientales*. Obtenido de Publicaciones Digitales:
https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1760/libro.pdf



Ministerio de Justicia. (24 de Abril de 2017). *Sistema Peruano de Información Jurídica*.

Obtenido de <https://www.minam.gob.pe/wp-content/uploads/2018/06/Decreto-Legislativo-N%C2%B0-1278.pdf>

Ministerio del Ambiente. (1 de julio de 2016). *gob.pe*. Obtenido de Ministerio del

Ambiente: <https://www.gob.pe/institucion/minam/informes-publicaciones/2634-plan-nacional-de-gestion-integral-de-residuos-solidos-2016-2024>

Nguyen, B., & De Cremer, D. (20 de Enero de 2016). *The European Business Review*.

Obtenido de The Fairness Challenge of the Internet of Things:

<https://www.europeanbusinessreview.com/the-fairness-challenge-of-the-internet-of-things/>

Pisano, A. (2018). *Internet de las Cosas*.

Portocarrero, S. (2018). *Análisis de manejo de residuos sólidos en el distrito de Yanque, provincia de Caylloma Arequipa 2018*. Universidad Nacional de San Agustín.

Quilla, C. (2017). *Valoración económica del tratamiento y gestión del manejo de los residuos sólidos urbanos en la ciudad de Huancané*. Repositorio institucional de la Universidad Nacional del Altiplano.

<http://repositorio.unap.edu.pe/handle/UNAP/5090>.

Rodrigo, J. (2022). *Sistema de gestión ambiental y su influencia con el manejo de los residuos sólidos de las municipalidades de la región Puno*. Repositorio institucional de la Universidad Nacional del Altiplano.

<http://repositorio.unap.edu.pe/handle/UNAP/18316>.

Rose, K., Eldridge, S., & Chapin, L. (2015). *The internet of things: An overview*.

Geneva, Switzerland: The Internet Society.



Secretaria de Medio Ambiente y Recursos Naturales. (2016). *Informe de la situación del medio ambiente en Mexico*. Senarmat.

Seminario, R., & Tineo, A. (2018). *Gestión de residuos sólidos en un hipermercado local*. Universidad Nacional de Piura
https://pirhua.udep.edu.pe/bitstream/handle/11042/3819/ING_611.pdf.

Tostatronic. (2020). *Tostatronic venta de componentes electrónicos e informáticos*.
Obtenido de Celdas de carga de 50 kg de precisión SEN-10245:
<https://tostatronic.com/store/sensores/2051-galga-extensometrica-de-50kg-de-precision-0683489667997.html>

Union Internacional de Telecomunicaciones. (2012). *Serie Y: Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la Próxima generación Y.2060*. UIT-T.

Urrego Morera, A. (2022). Seguridad en Sistemas Operativos Linux. *Universidad Piloto de Colombia*, 11.



ANEXOS

ANEXO 1

Código de Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include "HX711.h"

#define calibration_factor -564000

#define LOADCELL_DOUT_PIN_1 14
#define LOADCELL_SCK_PIN_1 15
#define LOADCELL_DOUT_PIN_2 16
#define LOADCELL_SCK_PIN_2 17
#define LOADCELL_DOUT_PIN_3 18
#define LOADCELL_SCK_PIN_3 19
#define LOADCELL_DOUT_PIN_4 20
#define LOADCELL_SCK_PIN_4 21

HX711 scale1;
HX711 scale2;
HX711 scale3;
HX711 scale4;

float total_lb;
float total_kg;
float total_perc;

byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };

IPAddress server(108,62,123,70);

IPAddress ip(172,16,100,133);
//IPAddress ip(192,168,1,133);
byte gateway[] = { 172, 16, 100, 1 };
//byte gateway[] = { 192, 168, 1, 1 };
byte subnet[] = { 255, 255, 255, 0 };
//byte myDns[] = { 192, 168, 1, 252 };
//byte myDns[] = { 190, 113, 220, 18 };
byte myDns[] = { 8, 8, 8, 8 };

EthernetClient client;

bool printWebData = true;

char sensor_id[] = "MOVIL-01";

void setup() {
```



```
Ethernet.begin(mac, ip, myDns, gateway, subnet);
}

void loop() {
  delay(1000);

  total_lb = scale1.get_units() + scale2.get_units() + scale3.get_units() +
scale4.get_units();
  total_kg = total_lb / 2.2046;
  total_perc = total_kg * 100 / 2;

  if (client.connect(server, 80)) {

    client.print("GET /LOADCELL/formation_loadcell.php?total_kg=");
    client.print(total_kg);
    client.print("&total_perc=");
    client.print(total_perc);
    client.print("&sensor_id=");
    client.print(sensor_id);
    client.println(" HTTP/1.1");
    client.print("Host: ");
    client.println(server);
    client.println("Connection: close");
    client.println();

  } else {

  }

  while (client.connected()) {
    int len = client.available();
    if (len > 0) {
      byte buffer[80];
      if (len > 80) len = 80;
      client.read(buffer, len);
      if (printWebData) {

      }
    }
  }

  if (!client.connected()) {
    client.stop();
    delay(28358); }}
```



ANEXO 2

Conexión Satisfactoria!

Carga (Kg)	Carga (%)	IP Cliente	ID Sensor	Fecha Hora
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:11:38
0.00	0.09	179.7.224.186	MOVIL-01	2023-04-06 17:11:08
0.00	0.09	179.7.224.186	MOVIL-01	2023-04-06 17:10:38
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:10:08
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:09:38
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:09:09
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:08:39
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:08:09
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:07:39
0.00	0.08	179.7.224.186	MOVIL-01	2023-04-06 17:07:09
1.92	96.17	179.7.224.186	MOVIL-01	2023-04-06 17:06:40
1.94	96.89	179.7.224.186	MOVIL-01	2023-04-06 17:06:10
1.95	97.59	179.7.224.186	MOVIL-01	2023-04-06 17:05:40
1.95	97.29	179.7.224.186	MOVIL-01	2023-04-06 17:05:10
1.88	94.12	179.7.224.186	MOVIL-01	2023-04-06 17:04:41
2.14	107.21	179.7.224.186	MOVIL-01	2023-04-06 17:04:11
2.20	110.05	179.7.224.186	MOVIL-01	2023-04-06 17:03:41
0.30	14.97	179.7.224.186	MOVIL-01	2023-04-06 17:03:11
0.29	14.75	179.7.224.186	MOVIL-01	2023-04-06 17:02:41
0.30	14.89	179.7.224.186	MOVIL-01	2023-04-06 17:02:12
0.30	15.08	179.7.224.186	MOVIL-01	2023-04-06 17:01:42
0.31	15.43	179.7.224.186	MOVIL-01	2023-04-06 17:01:12
0.31	15.33	179.7.224.186	MOVIL-01	2023-04-06 17:00:42
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 17:00:13
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:43
0.00	0.06	179.7.224.186	MOVIL-01	2023-04-06 16:59:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:43
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:58:13
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:44
0.00	0.05	179.7.224.186	MOVIL-01	2023-04-06 16:57:14
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:56:44
0.00	0.04	179.7.224.186	MOVIL-01	2023-04-06 16:56:14



DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo LUCERO NALDY FLORES CASTRO,
identificado con DNI 71711061 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado
INGENIERIA ELECTRONICA

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

“ DISEÑO DE UN SISTEMA DE SUPERVISION DE LA CAPACIDAD DE CONTENEDORES
DE RESIDUOS SOLIDOS PARA OPTIMIZAR EL USO DE RECURSOS DE TRANSPORTE
PARA SU RECOLECCION - DISTRITO DE PUNO AÑO 2022 ”

Es un tema original.


Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

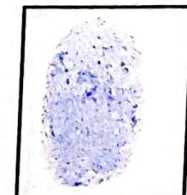
Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 20 de SEPTIEMBRE del 2023



FIRMA (obligatoria)



Huella



AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo LUCERO NALDY FLORES CASTRO
identificado con DNI 71711061 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

INGENIERIA ELECTRONICA

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

" DISEÑO DE UN SISTEMA DE SUPERVISION DE LA CAPACIDAD DE CONTENEDORES DE RESIDUOS SOLIDOS PARA OPTIMIZAR EL USO DE RECURSOS DE TRANSPORTE PARA SU RECOLECCION - DISTRITO DE PUNO AÑO 2022 "

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 20 de SEPTIEMBRE del 2023



FIRMA (obligatoria)



Huella