



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

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

TEMA:

**SISTEMA DE RECOLECCIÓN DE DESECHOS OPTIMIZADO, MEDIANTE IOT EN
EL SECTOR EL OLIVO DE LA CIUDAD DE IBARRA PROVINCIA DE IMBABURA.**

AUTOR: OBANDO SUAREZ JEAN PAUL

DIRECTOR: MSc. CARLOS ALBERTO VASQUEZ AYALA

ASESOR: MSc. LUIS EDILBERTO SUAREZ ZAMBRANO

Ibarra-Ecuador

2022



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA
UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA.

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL CONTACTO	
Cédula de identidad	0604196162
Apellidos y Nombres	Obando Suarez Jean Paúl
Dirección	Quito, Real Audiencia N63-131 y de los Cedros
E-mail	jpobandos@utn.edu.ec
Teléfono móvil	0987540254

DATOS DE LA OBRA	
Título	“Sistema de recolección de desechos optimizado, mediante IoT en el sector El Olivo de la ciudad de Ibarra provincia de Imbabura.”
Autor	Obando Suarez Jean Paul
Fecha	7 de octubre del 2022
Programa	Pregrado
Título	Ingeniería en Electrónica y Redes de Comunicación
Director	Ing. Carlos Alberto Vásquez Ayala, MsC.

Firma: 

Nombre: Obando Suarez Jean Paul



UNIVERSIDAD TECNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 14 días del mes de octubre de 2022

EL AUTOR

A handwritten signature in blue ink, consisting of several loops and a long horizontal stroke, positioned above a solid horizontal line.

Obando Suarez Jean Paul

CI: 0604196162



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

**MAGISTER CARLOS ALBERTO VASQUEZ AYALA, DIRECTOR DEL PRESENTE
TRABAJO DE TITULACIÓN CERTIFICA:**

Que, el presente trabajo de Titulación “Sistema de recolección de desechos optimizado, mediante IoT en el sector El Olivo de la ciudad de Ibarra provincia de Imbabura.” fue desarrollado en su totalidad por el Sr. Obando Suarez Jean Paul, bajo mi supervisión.

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

Ing. Carlos Alberto Vásquez Ayala, MsC.

DIRECTOR

AGRADECIMIENTO

Primeramente, agradezco a Dios por permitirme llegar hasta este momento junto a mi familia quienes han sido un pilar fundamental de apoyo y comprensión.

A la Universidad Técnica del Norte por abrirme las puertas para formarme como profesional.

A los docentes de la carrera, quienes brindaron sus conocimientos a lo largo de mi formación universitaria, de manera especial al MSc, Carlos Vásquez, quien fue el guía de este trabajo de grado.

A todas las personas que de una u otra manera han sido parte de este camino.

Obando Suarez Jean Paul

DEDICATORIA

Dedico este trabajo a mi familia y amigos.

Obando Suarez Jean Paul

RESUMEN

Como proyecto final se presenta un sistema que automatiza y sistematiza la recolección de residuos sólidos urbanos (RSU), con la utilización de sensores ultrasónicos y la tecnología SigFox para IoT, que detecta el nivel de almacenamiento en tiempo real de los contenedores de basura, en el sector “El Olivo”, en la ciudad de Ibarra, este sistema genera rutas de recolección para el personal que realiza esta actividad y presta la información relevante para los moradores del sector.

Los datos obtenidos por el sensor se transmiten a un microcontrolador donde son analizados y procesados para su clasificación en tres niveles “vacío, medio y lleno”, se envían de manera inalámbrica mediante el módulo Wisol RC- Z4 a la red de transmisión Sigfox.

El sistema de transmisión envía un mensaje cada hora del día con la información del nivel de cada contenedor hacia la nube de SigFox, información que sirve de base para la toma de decisiones en el algoritmo de aprendizaje de la ruta óptima, el cual toma como principales parámetros la ubicación predefinida en el sistema junto con el nivel de almacenamiento obtenido en tiempo real de cada uno de los contenedores.

Finalmente, se presenta una aplicación web y una aplicación móvil, las cuales muestran el estado de los contenedores, la posición en la que se encuentran, la ruta óptima y demás datos relevantes.

ABSTRACT

As a final project, the system is presented that automates and systematizes the collection of urban solid waste (RSU), with the use of ultrasonic sensors and SigFox technology for IoT, which detects the storage level in real time of garbage containers, in the “El Olivo” sector, in the city of Ibarra, this system generates collection routes for the personnel who carry out this activity and provides relevant information for the inhabitants of the sector.

The data obtained by the sensor is transmitted to a microcontroller where it is analyzed and processed for its classification into three levels "empty, half and full", they are sent wirelessly through the Wisol RC4 module to the Sigfox transmission network.

The transmission system sends a message every hour of the day with the information on the level of each container to the SigFox cloud, information that serves as the basis for decision making in the optimal route learning algorithm, which takes as main parameters the predefined location in the system along with the storage level obtained in real time from each of the containers.

Finally, a web application and a mobile application are presented, which show the status of the containers, their position, the optimal route and other relevant data.

CONTENIDO

IDENTIFICACIÓN DE LA OBRA.	
CONSTANCIAS	
CERTIFICACIÓN	
AGRADECIMIENTO	10
DEDICATORIA	14
RESUMEN	15
ABSTRACT	16
CONTENIDO	17
CAPÍTULO I: ANTECEDENTES	23
1.1. Introducción	23
1.2. Tema.....	24
1.3. Problema.....	24
1.4. Objetivos	25
1.4.1. Objetivo General.....	25
1.4.2. Objetivos Específicos.....	26
1.5. Alcance.....	26
1.6. Justificación.....	28
2. CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA	32
2.1. Residuos Sólidos Urbanos (RSU)	32
2.1.1. Caracterización de los Residuos Sólidos Urbanos.....	32
2.2. Generación y recolección de residuos sólidos.....	33
2.3. Internet de las cosas IoT.....	37
2.3.1. Arquitectura IoT.....	37
2.4. Sensores y dispositivos.....	39
2.4.1. Sensor ultrasónico	40
2.4.2. Placas de desarrollo o gestión.....	41
2.5. Arduino.....	43
2.5.1. Tipos de placas Arduino	44
2.5.2. Dispositivo Conexión a Sigfox.....	45
2.6. Tecnología LPWAN.....	46
2.7. Tecnología Sigfox	48
2.7.1. Características	48

2.7.2.	Arquitectura de sistema Sigfox	50
2.8.	Modelo de cascada	52
2.8.1.	Etapas / iteraciones	53
2.9.	Algoritmo de programación	55
2.9.1.	Algoritmo de Dijkstra	56
2.10.	IoT aplicado a la recolección de RSU	56
2.11.	Apache Cordova	57
2.11.1.	Arquitectura de Apache Cordova.....	57
3.	CAPÍTULO III: METODOLOGÍA	59
3.1.	Macro Localización.....	59
3.2.	Micro Localización	59
3.3.	Población y muestra	60
3.3.1.	Instrumentos de recolección de información	61
3.3.2.	Resultados de la recolección de información.....	61
3.4.	Diseño de prototipo	61
3.4.1.	Requisitos del sistema.....	62
3.5.	Recursos Humanos.....	65
3.6.	Recursos económicos	66
3.7.	Recurso tecnológico	66
3.8.	Modelación de rutas	70
3.9.	Algoritmo de rutas.....	71
3.9.1.	Modelo de ejecución en Java	71
3.10.	Selección de tecnología LPWAN.....	72
3.11.	Selección dispositivo Kit desarrollo ThinXtra	73
4.	CAPÍTULO IV: DESARROLLO Y PRUEBAS	74
4.1.	Características del Sistema de recolección de desechos	74
4.2.	Implementación del Hardware del sistema	75
4.3.	Implementación del Software del Sistema.....	76
4.3.1.	Capa de Dispositivos.....	77
4.3.2.	Capa de red	80
4.3.3.	Capa de servicios	87
4.3.4.	Capa de aplicación	91

4.4.	Funcionamiento del sistema y pruebas de desarrollo.....	98
4.4.1.	Aplicación Web	98
4.4.2.	Aplicación Móvil	104
5.	CONCLUSIONES Y RECOMENDACIONES	112
5.1.	Conclusiones	112
5.2.	Recomendaciones.....	112
	BIBLIOGRAFÍA	114
	ANEXOS	119
	Anexo 1. Manual de Usuario.....	119
	Anexo 2. Entrevista.....	137
	Anexo 3. Presentación Final del Sistema.....	138
	Anexo 4. Justificación del desarrollo del prototipo en Quito.....	139

Índice de Tablas

Tabla 1.	Características del Sensor Ultrasónico.....	41
Tabla 2.	Tipos de Arduino	44
Tabla 3.	Dispositivos de conexión a Sigfox.....	46
Tabla 4.	Aspectos Técnicos de redes LPWAN	47
Tabla 5.	Especificaciones técnicas red Sigfox.	49
Tabla 6.	Abreviaturas de los requerimientos del sistema.....	62
Tabla 7.	Requerimientos De Stakeholders	63
Tabla 8.	Requerimientos del sistema.	64
Tabla 9.	Requerimientos de Arquitectura	65
Tabla 10.	Recursos Humanos.....	66
Tabla 11.	Recursos Económicos	66
Tabla 12.	Recurso tecnológico.....	68
Tabla 13.	Componentes del sistema.....	69

Índice de Figuras

Figura 1. Capas de la Arquitectura IoT.....	38
Figura 2. Estructura de un sensor ultrasónico.....	40
Figura 3. Elementos más comunes de las tarjetas de desarrollo.....	42
Figura 4. Estructura red Sigfox.....	48
Figura 5. Arquitectura del sistema Sigfox	50
Figura 6. Diagrama de Flujo del proceso general del modelo Cascada.....	52
Figura 7. Modelo de Algoritmo de Dijkstra	56
Figura 8. Arquitectura de Apache Cordova.....	58
Figura 9. Macro Localización.....	59
Figura 10. Macro Localización.....	60
Figura 11. Diagrama de Bloques del sistema de Recolección de Desechos.....	67
Figura 12. Estructura del sistema de recolección de residuos.	69
Figura 13. Ventajas IoT de redes LPWAN.....	72
Figura 14. Diagrama de Conexión Arduino UNO y Sensor Ultrasónico	77
Figura 15. Variables para pines y cálculos	78
Figura 16. Comunicación entre módulo Sigfox y los pines del sensor.....	78
Figura 17. Función principal para monitorear los sensores	79
Figura 18. Conexión con la red Sigfox	80
Figura 19. Módulo Wisol incorporado en microcontrolador.....	80
Figura 20. Registro Región Arduino en Sigfox	81
Figura 21. Datos para registro en Sigfox	81
Figura 22. Datos de Usuario para Registro en Sigfox	82
Figura 23. Pantalla Principal Backend.....	83
Figura 24. Pantalla del Grupo en Backend	84
Figura 25. Backend Sigfox, zona de operación del dispositivo.....	84
Figura 26. Pantalla de ID del Dispositivo.....	85
Figura 27. Backend Sigfox, zona de recepción de mensajes	86
Figura 28. Backend Sigfox, zona de eventos.....	86
Figura 29. Backend Sigfox, zona de estadística	87
Figura 30. Servidor de base de Datos	87

Figura 31. Tabla de Contenedores	88
Figura 32. Tabla de estado.....	88
Figura 33. Tabla de Tipo de Usuario	89
Figura 34. Tabla de Información de Usuarios	89
Figura 35. Conexión con la base de datos	90
Figura 36. Función para insertar datos en la Base de Datos	90
Figura 37. Algoritmo de programación del sistema de Recolección de Desechos.....	92
Figura 38. Algoritmo de Dijkstra.....	93
Figura 39. Archivos para aplicación móvil.....	94
Figura 40. Comunicación con la base de datos.....	94
Figura 41. Información agregada en los contenedores para mostrar en la aplicación móvil.....	95
Figura 42. Información agregada en los usuarios para mostrar en la aplicación móvil.	95
Figura 43. Apache Cordova página oficial	96
Figura 44. Ruta de archivos para la comunicación.....	96
Figura 45. Comando para construir la aplicación móvil.....	97
Figura 46. Ruta de la aplicación móvil.....	97
Figura 47. Carpeta comprimida con la aplicación móvil.....	98
Figura 48. Pantalla de Inicio Aplicación Web.....	98
Figura 49. Error de ingreso de credenciales	99
Figura 50. Menú Principal Aplicación Web	99
Figura 51. Menú de Administrador y Usuario.....	100
Figura 52. Pantalla de Registro de Contenedores	101
Figura 53. Confirmación de Eliminación de Contenedores.....	101
Figura 54. Pantalla de Ubicación de Contenedores	102
Figura 55. Simbología de Mapa de Ubicación de Contenedores.....	102
Figura 56. Tipo de Rutas.....	103
Figura 57. Trazado de Ruta de Recolección	103
Figura 58. Pantalla de registro de Usuarios	104
Figura 59. Aplicación móvil	104
Figura 60. Pantalla de Login.....	105
Figura 61. Barra de navegación	105

Figura 62. Información de Home.....	106
Figura 63. Información de Rutas	107
Figura 64. Cálculo de la ruta más óptima	107
Figura 65. Información del detalle de los contendores	108
Figura 66. Lista de los contendores	109
Figura 67. Perfil de usuario.....	109
Figura 68. Actualización de datos.....	110
Figura 69. Cerrar sesión.....	110
Figura 70. Aceptar cerrar sesión.....	111

CAPÍTULO I: ANTECEDENTES

1.1. Introducción

El crecimiento económico global está asociado con el incremento de la población especialmente en centros urbanos, lo que ha originado el aumento de la generación de residuos sólidos urbanos (RSU) por el consumo social, actividades productivas y componentes ambientales (Pérez et al. 2007). Por ello, las ciudades ven la necesidad de adaptar y adecuar sus sistemas de recolección de residuos en base al crecimiento poblacional y capacidad presupuestaria, que les permita solventar al menos la producción promedio anual de su población (Alcocer et al. 2019).

La recolección de residuos sólidos depende de varios factores como las zonas rurales o urbanas, nivel socioeconómico, actividades económicas de las zonas, densidad demográfica, análisis de días cotidianos y especiales, entre otros factores (Peralta & Vélez, 2011). En Ibarra se ha implementado el sistema de recolección por contenedores en gran parte del territorio, y los vehículos recolectores cumplen una ruta temporizada para cumplir con su trabajo, sin embargo, en muchos casos se ha evidenciado que estos no se encuentran moderadamente llenos, y en otros casos existe la presencia de bolsas de residuos alrededor de los contenedores debido a que estos ya se encuentran ocupados al máximo de su capacidad.

Por lo mencionado, el presente proyecto está orientado a diseñar un sistema automatizado en la detección de información del estado en tiempo real de los contenedores, para establecer así un sistema de recolección de residuos eficiente, mediante la aplicación de sensores remotos y el

diseño específico de un software que permita a los operadores conocer, analizar, comparar y diseñar hojas de ruta en la recolección de residuos sólidos.

1.2. Tema

Sistema de recolección de desechos optimizado, mediante IoT en el sector El Olivo de la ciudad de Ibarra provincia de Imbabura.

1.3. Problema

Los materiales u objetos que han cumplido la función por la cual fueron creados y ya no presentan un valor económico para su usuario son desechados en forma de residuos, estos residuos tienden a acumularse en grandes masas en centros urbanos, por ello se implementa un sistema de recolección de residuos para gestionar, recolectar y eliminar dichos residuos (Sáez & Urdaneta, 2014). El problema en la generación de basura se evidencia en su tratamiento, recolección, almacenamiento y eliminación, debido a que ocasiona elevados costos sociales y económicos que deben asumir los gobiernos, además de originar impactos ambientales para toda la población y el planeta.

Actualmente, el Ecuador presenta una tasa per cápita de generación de residuos sólidos de 0.82 kg/hab-día, el cuál es un valor preocupante en comparación con otros países de la región (Noguera & Oliveros, 2010). Se estima que diariamente en la ciudad de Ibarra se generan 128,83 toneladas de residuos sólidos con una tasa per cápita 0,71 kilogramos por habitante al día, y la ciudad presenta un crecimiento poblacional del 2.3% anual en las regiones urbanas, por lo que es

necesario analizar el sistema de recolección para estimar los recursos necesarios para cumplir con la capacidad de la ciudad (GADP Imbabura, 2012).

En los últimos años ha aumentado la tasa de generación de residuos sólidos en la ciudad de Ibarra por ello, el Municipio optó por aplicar contenedores en varios sectores de la ciudad con el fin de aumentar la eficiencia en el almacenamiento temporal de los residuos. Sin embargo, esto ha originado otros problemas derivados de la recolección de información acerca del estado, nivel de utilización y rutas de recolección de residuos (GADP Imbabura, 2012).

La falta de automatización en la transmisión de información ha generado un problema grave, debido a que se trazan rutas de recolección de contenedores que en muchos casos se encuentran vacíos o ya están completamente llenos con desperdicios y desechos en su alrededor (Alcocer et al. 2019). Estas hojas de ruta se realizan únicamente en función a la cantidad de residuos generados, sin embargo, un sistema automatizado permitiría en tiempo real, detectar el nivel de utilización de los contenedores para de esta manera, generar y adecuar el sistema de recolección de forma óptima y en base a la utilización eficiente de los recursos del municipio.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar un sistema de recolección de desechos mediante el uso de tecnologías IoT para optimizar recursos y evitar problemas sanitarios, estéticos y de salubridad además de mejorar la calidad de vida de las personas en la ciudad de Ibarra, sector El Olivo.

1.4.2. *Objetivos Específicos*

- Elaborar un estado del arte acerca de la tecnología de transmisión para IoT Sigfox, así como los tipos de sensores utilizados para medir distancias, las redes para el intercambio de información, software de aprendizaje de rutas y sus algoritmos
- Determinar requerimientos para Hardware y software que garanticen el funcionamiento del sistema.
- Diseñar un sistema de recolección de desechos utilizando las fases de la metodología en cascada que permita definir la tecnología idónea para transmisión y algoritmo de enrutamiento a utilizarse en el sistema.
- Determinar la ruta más eficiente de recolección posterior al tratamiento de los datos obtenidos de los sensores de cada contenedor mediante el uso de algoritmos.
- Ejecutar las pruebas de operación para validar el correcto funcionamiento del sistema de recolección de desechos.

1.5. Alcance

El presente proyecto está orientado a automatizar y sistematizar la recolección de residuos sólidos urbanos (RSU) específicamente en contenedores de basura, en el sector del Olivo de la ciudad de Ibarra de la provincia de Imbabura, mediante el uso de tecnologías IoT (Internet de las cosas) utilizando equipos electrónicos, software y criterios técnicos que determinen un sistema eficiente en la generación de hojas de ruta de recolección.

Se realizará un prototipo para la implementación de sensores ultrasónicos para determinar el nivel de ocupación de contenedores de basura, basados en la geografía urbana del sector “El

Olivo”, en la ciudad de Ibarra. Se realizará una muestra para determinar el número de contenedores a implementar en el prototipo, además de los dispositivos necesarios para la instalación del sistema de detección de nivel de ocupación.

Se tomará como modelo la metodología en “Cascada para el desarrollo secuencial de Software e IoT”, que es un procedimiento estandarizado que divide los procesos de desarrollo en sucesivas fases del proyecto” (IONOS, 2021). El modelo propone siete fases que se ha de ejecutar en diversas iteraciones: Requisitos de sistema, Requisitos de software, Análisis, Diseño, Implementación, Prueba, Servicio (Royce, 2021). Para cumplir con los objetivos planteados y en función de la primera y segunda fase de la metodología se determinará los requerimientos del sistema, tanto de software como hardware lo cual permitirá obtener una visión más clara de los componentes y limitaciones que forman parte del proyecto.

Se analizará los datos obtenidos para el intercambio de información por medio de softwares de aprendizaje, algoritmos de determinación de rutas y demás componentes. Posterior a la recopilación y análisis de información, y teniendo en cuenta los requerimientos tanto de software como hardware se desarrollará la etapa de diseño que comprende la cuarta fase de la metodología la que iniciará con la toma de datos de cada uno de los contenedores existentes en el prototipo de funcionamiento controlado.

Los datos obtenidos por el sensor serán transmitidos a un microcontrolador donde serán analizados y procesados para su clasificación en tres niveles “vacío, medio y lleno” y enviados de manera inalámbrica mediante un módulo capaz de acoplarse al a red de transmisión Sigfox.

El sistema de transmisión enviará un mensaje 2 veces al día con la información del nivel de cada contenedor hacia la nube, información que servirá de base para la toma de decisiones en nuestro algoritmo de aprendizaje de la ruta óptima el cual toma como principales parámetros la ubicación predefinida en el sistema junto con el nivel de almacenamiento obtenido en tiempo real de cada uno de los contenedores.

Se debe tomar en cuenta que en el sector existen aproximadamente 100 contenedores lo cual conlleva a obtener múltiples posibilidades de rutas que parten desde un mismo punto además de ser un plan piloto extensible y adaptable a sectores donde utilicen contenedores como sistemas de almacenamiento de residuos.

1.6. Justificación

Es necesario implementar un sistema de recolección de RSU automatizado y adaptado a los requerimientos de la ciudad, en función a las tasas de producción per cápita, densidad demográfica, número de vehículos recolectores, distribución de rutas y comportamientos diferenciados de generación de RSU que se pueden dar por feriados, eventos especiales, entre otros. La aplicación de un sistema IoT permitirá a los operadores identificar el comportamiento de generación de RSU en el sector de análisis para establecer una hoja de ruta óptima y eficiente en la zona, además de facilitar y maximizar el uso de recursos del municipio.

En la ciudad de Ibarra no existe ningún sistema que gestione de manera óptima el sistema de recolección partiendo desde la monitorización del contenedor hasta la determinación óptima de rutas de recolección y cada día con el incremento poblacional se incrementa también el

número de contenedores en la ciudad así se evidencia la implementación en el año 2016 de 800 contenedores que fueron distribuidos en tres sectores de la ciudad. Dentro de la Constitución de la República del Ecuador establece en el Capítulo Cuarto de régimen de competencias: Art. 264.- Los gobiernos municipales tendrán las siguientes competencias exclusivas, sin perjuicio de otras que determine la ley: 4. Prestar los servicios públicos de agua potable, alcantarillado, depuración de aguas residuales, manejo de desechos sólidos, actividades de saneamiento ambiental y aquellos que establezca la ley (Asamblea Constituyente, 2008)

En la Sección cuarta de Habitación y vivienda, Art. 375.- El Estado, en todos sus niveles de gobierno, garantizará el derecho al hábitat y a la vivienda digna, para lo cual: 1. Generará la información necesaria para el diseño de estrategias y programas que comprendan las relaciones entre vivienda, servicios, espacio y transporte públicos, equipamiento y gestión del suelo urbano (Asamblea Nacional Constituyente de Ecuador de 2007-2008, 2008). Dentro del Plan Nacional de Desarrollo 2017-2021 en los Objetivos Nacionales de Desarrollo para el Buen Vivir, Eje1: Derechos para todos Durante toda la Vida tiene como Objetivo 1: Garantizar una vida digna con iguales oportunidades para todas las personas.

El proyecto “SISTEMA DE RECOLECCION DE DESECHOS OPTIMIZADO MEDIANTE IoT” tiene como objeto la optimización de recursos, evitar problemas sanitarios, estéticos y de salubridad además de mejorar la calidad de vida de las personas dentro de la ciudad de Ibarra para esto el proyecto se sustenta en los siguientes artículos de la Constitución del Ecuador que promueven el derecho del buen vivir.

El Estado Ecuatoriano dentro de su constitución y derechos del buen vivir en su sección segunda Ambiente sano dice: Art. 14.- Se reconoce el derecho de la población a vivir en un ambiente sano y ecológicamente equilibrado, que garantice la sostenibilidad y el Buen Vivir, Sumak Kawsay. Se declara de interés público la preservación del ambiente, la conservación de los ecosistemas, la biodiversidad y la integridad del patrimonio genético del país, la prevención del daño ambiental y la recuperación de los espacios naturales degradados (Asamblea Nacional Constituyente de Ecuador de 2007-2008, 2008). En la Sección sexta Habitación y vivienda: Art. 30.- Las personas tienen derecho a un hábitat seguro y saludable, y a una vivienda adecuada y digna, con independencia de su situación social y económica (Asamblea Nacional Constituyente de Ecuador de 2007-2008, 2008). Art. 31.- Las personas tienen derecho al disfrute pleno de la ciudad y de sus espacios públicos, bajo los principios de sustentabilidad, justicia social, respeto a las diferentes culturas urbanas y equilibrio entre lo urbano y lo rural. El ejercicio del derecho a la ciudad se basa en la gestión democrática de ésta, en la función social y ambiental de la propiedad y de la ciudad, y en el ejercicio pleno de la ciudadanía (Asamblea Nacional Constituyente de Ecuador de 2007-2008, 2008).

Sección séptima Salud: Art. 32.-La salud es un derecho que garantiza el Estado, cuya realización se vincula al ejercicio de otros derechos, entre ellos el derecho al agua, la alimentación, la educación, la cultura física, el trabajo, la seguridad social, los ambientes sanos y otros que sustentan el buen vivir. El Estado garantizará este derecho mediante políticas económicas, sociales, culturales, educativas y ambientales.

La prestación de los servicios de salud se regirá por los principios de equidad, universalidad, solidaridad, interculturalidad, calidad, eficiencia, eficacia, precaución y bioética,

con enfoque de género y generacional (Asamblea Nacional Constituyente de Ecuador de 2007-2008, 2008). Además de los derechos y garantías que presenta la constitución se debe tomar en cuenta el avance tecnológico que las ciudades han venido presentando, es por esto por lo que el proyecto busca generar soluciones que encajen dentro de estas características aportando hacia el avance de una ciudad inteligente y tecnológica.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

En este capítulo se recoge la fundamentación teórica necesaria para el desarrollo del actual proyecto mediante el análisis de conceptos de forma deductiva partiendo de conceptos generales a detalles específicos relacionados con la investigación. Se definirán los fundamentos necesarios para el desarrollo de la metodología del estudio en base al alcance y cumplimiento de los objetivos planteados.

2.1. Residuos Sólidos Urbanos (RSU)

Los RSU son todos aquellos materiales y objetos cuyo propietario desecha debido a que ya cumplieron su meta y carecen de valor. Estos se generan específicamente en los centros urbanos y sus zonas de influencia (Sáez & Urdaneta, 2014). Los residuos sólidos pueden ser caracterizados en centros de transición y separación para obtener materiales que, con el debido proceso, pueden reintegrarse a la cadena de valor de las actividades económicas. Los residuos que después de una caracterización y valoración no se reintegran a la cadena de valor, toman el nombre de desechos y son enviados a vertederos, rellenos sanitarios o incinerados para su eliminación.

2.1.1. Caracterización de los Residuos Sólidos Urbanos

La caracterización de residuos sólidos es el proceso de determinación, agrupación y separación de clases de residuos en base a su composición y características fisicoquímicas en diferentes fracciones, que permite reintegrar varios grupos a las cadenas productivas. La caracterización de residuos permite identificar principalmente dos tipos de residuos: reciclables y no reciclables (Noguera & Oliveros, 2010). Los RSU no reciclables se enviarán para disposición

final hacia vertederos o rellenos sanitarios, y los RSU reciclables se clasifican por su composición de la siguiente manera:

- Vidrio
- Papel y cartón
- Plástico
- Orgánicos
- Tetra pack
- Metal
- Llantas usadas
- Electrónicos

2.2. Generación y recolección de residuos sólidos

En la ciudad de Ibarra de acuerdo con la ordenanza Municipal de la administración actual, la recolección se realiza diariamente por personal del Municipio, el cual sigue diferentes rutas en transporte especializado, levantando la basura que se encuentra en los contenedores que han sido colocados a lo largo de la ciudad, terminada la recolección se ingresa en el relleno sanitario de la ciudad (Gobierno Autónomo Descentralizado Ibarra, 2019).

La generación de residuos se puede dar por actividades domésticas, comerciales, agrícolas e industriales. Para establecer indicadores de generación es necesario determinar cuál es la producción per cápita producida por habitante al día en una zona determinada (Alcocer et al. 2019). A continuación, en la ecuación 1, se presenta la fórmula de cálculo de producción per cápita (PPC):

$$PPC = \frac{\text{Cantidad total recolectada } \left(\frac{kg}{\text{dia hab}}\right)}{\text{Población atendida } \left(\frac{\text{habitantes}}{\text{dia}}\right)} \quad (\text{Ec. 1})$$

La recolección de residuos debe organizarse con el fin de ofrecer un servicio eficiente y equitativo, evitando la generación de malos olores, polvos, desorden, contaminación ambiental y alteración del paisaje de la ciudad. Para implementar un sistema eficiente es necesario realizar un diagnóstico para la revisión del circuito prediseñado, evaluación de horarios de recolección, el volumen de residuos generados, distancia del punto de recolección hasta el centro de disposición y los tiempos requeridos (Torres et al. 2017).

También se debe considerar cuando el camión se encuentre lleno completamente, evaluando cantidad de residuos generados para de esta manera obtener regularidad horaria. Esto se consigue trazando hojas de rutas (Sáez & Urdaneta, 2014). En cada zona en función de la densidad poblacional y el tamaño físico, se determina el número de viajes necesarios y el tiempo por cada ruta, para establecer los horarios y frecuencias de recolección.

Por esto para obtener buenos rendimientos es necesario fijar los horarios, rutas y días de recolección para cada barrio o sector. Evitando así que los desechos se acumulen en las aceras, vías o lotes baldíos, generando vectores contaminantes. Una vez determinado los factores de la recolección, se divide el proceso de recolección de residuos en cuatro etapas:

2.2.1. *Recolección domiciliaria interna*

Es aquella que se desarrolla en cada hogar, hay que tomar en cuenta que esta recolección es óptima cuando existe una separación diferenciada de los residuos sólidos.

2.2.2. *Recolección Puerta a Puerta*

Es el sistema de recolección en ruta el cual consiste en que el carro recolector sigue una ruta abarcando todos los predios de una zona determinada. Esta ruta está diseñada con el fin de que el sistema de recolección realice una gestión óptima de prestación de servicios y por ende todos los desechos generados en una zona sean recolectados.

2.2.3. *Generación de un sistema de recolección*

Cada zona requiere de un análisis previo y específico para generar medidas de adopción para establecer rutas e itinerarios. Los factores que influyen en el tiempo de recolección son los siguientes:

- Tipo de receptáculo: Recolección en vereda, puerta a puerta o por contenedores de almacenamiento temporal.
- Ubicación del receptáculo: Localización geográfica de los puntos de recolección.
- Número de recolectores: Cantidad de vehículos recolectores y personal.
- Tipo de zona o sector: Se debe determinar si es una zona domiciliaria, comercial, agrícola o industrial.
- Calidad de las vías de circulación: Si las vías permiten la circulación normal del equipo de recolección y de no ser el caso, implementar otras herramientas.
- Tránsito de vehículos: Específicamente en sectores con alto tránsito vehicular en dónde se recomienda realizar la recolección en horarios nocturnos.
- Hora en que se realiza la recolección: Fijar horarios cotidianos en función del cumplimiento de hojas de ruta y factores secundarios de cada zona.

- Climático: Análisis del componente ambiental para efectuar el proceso en horarios y sitios que no representen la generación de riesgos para el trabajador.

2.2.4. *Transporte de residuos*

Se lo realiza mediante el empleo de vehículos adaptados al fin, estas unidades de transporte poseen una gran capacidad de recolección de desechos con el fin de poder brindar el servicio de transporte sin necesidad de realizar varios viajes para la recolección de desechos. Los desechos recolectados son llevados a una estación de transferencia y finalmente a un relleno sanitario.

2.2.5. *Disposición final*

Es el punto en la gestión de residuos sólidos en el cual los residuos que no presentan un valor económico son tratados como desechos y enviados a su eliminación. La legislación ecuatoriana determina que estos desechos deben ser enviados a rellenos sanitarios diseñados de forma técnica para evitar impactos hacia el entorno ambiental (Marmolejo et al., 2009).

Una vez que pasan por las estaciones de transferencia en dónde se clasifica y separa materiales reciclables, los camiones de recolección envían a disposición final empleando maquinaria pesada como tráileres para reincorporar los vehículos recolectores nuevamente y continuar con el proceso (Marmolejo et al., 2009).

La disposición final genera varios problemas socioambientales, debido a las extensas zonas que se requieren para implementar un relleno sanitario y ninguna comunidad quiere uno de estos centros cerca de su entorno, generado dificultades en la ejecución de estos proyectos.

Los rellenos son diseñados para funcionar de 10 a 20 años según la zona, sin embargo, el rápido crecimiento poblacional y el aumento de la tasa de generación de residuos han reducido el tiempo de vida útil de estas estructuras, generando problemas económicos en la ciudad y la necesidad de implementar otros rellenos en distintas áreas, creando pasivos ambientales e inutilizando dichos terrenos (Fazenda & Tavares Russo, 2016).

2.3. Internet de las cosas IoT

Sistema IoT es la agrupación e interconexión de dispositivos electrónicos y objetos mediante una red privada o internet, donde todos ellos pueden ser visibles e interactuar para automatizar un proceso. En respecto a los dispositivos electrónicos pueden ser varios, entre sensores o procesadores hasta dispositivos mecánicos, todos aquellos que puedan conectarse a internet, con el objeto de que exista interacción máquina a máquina sin la necesidad de la intervención humana, a este proceso se lo denomina interacción M2M (IONOS, 2021).

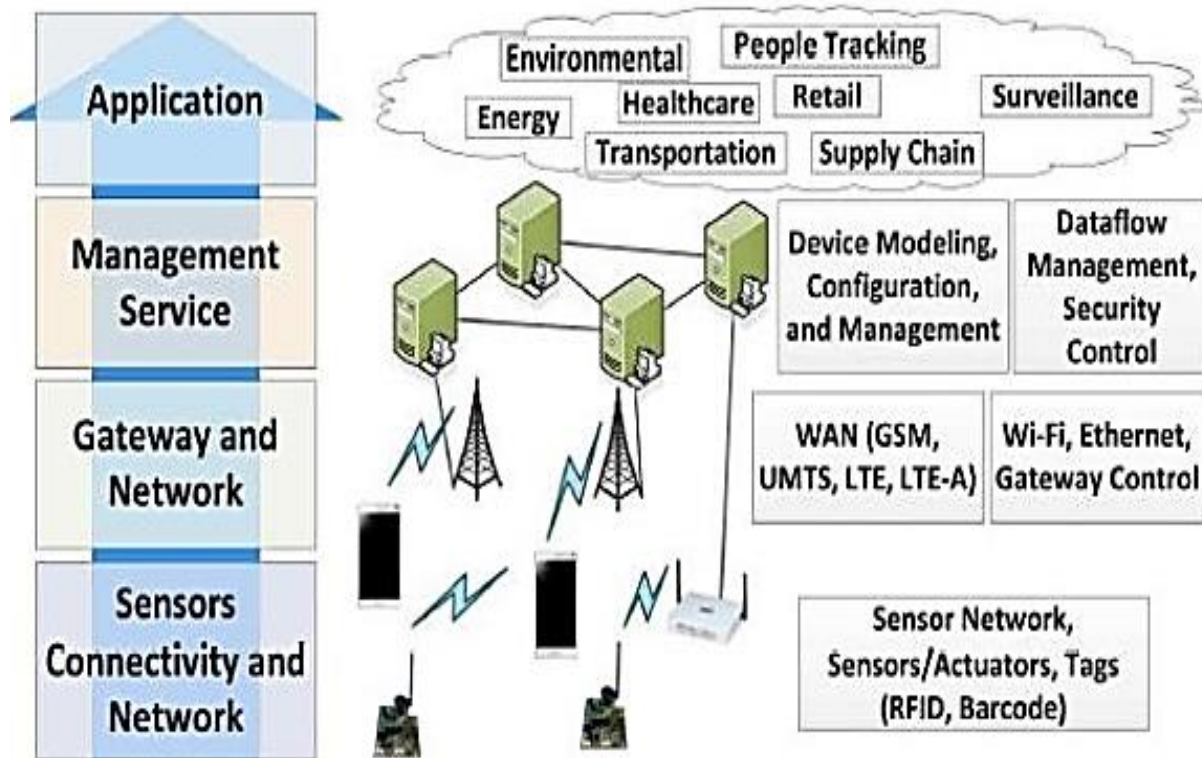
El avance de la ciencia ha generado la evolución y uso masivo de IoT, debido a que puede aplicarse a varios campos y posibilidades. Los IoT emplean la tecnología asociada para recopilar datos por medio de sensores electrónicos y mandarlos a la red a través de internet para su análisis de comportamiento (Alvear et al., 2017).

2.3.1. Arquitectura IoT

Esta sección presenta los componentes de software y hardware que se utilizan para la implementación del sistema IoT. Tomando como referencia la arquitectura base del sistema IoT (Cardozo et al. 2016), se determina las siguientes capas que integran el sistema.

Figura 1.

Capas de la Arquitectura IoT



Fuente: Arduino, (2020)

La figura 1 presenta las interacciones de cada una de las capas de la arquitectura IoT, donde se observa la transferencia de información desde los sensores y microsensores hasta el cliente, correspondiente a la capa de aplicación.

- **Capa Sensores y conectividad Network.** En esta capa se llevan a cabo los procesos de medición, mediante los sensores y microsensores; suministro de energía al sistema y conexión de los datos con la interfaz de comunicaciones a través del programa ya instalado en la tarjeta de desarrollo.

- **Capa de Gateway and Network.** Se encuentran el módulo de conexión, el cual facilita la transferencia de datos a la red, un lenguaje de intercambio, quien permite la transferencia de datos y finalmente un protocolo de comunicaciones. Tiene como objetivo generar la comunicación entre las mediciones realizadas y el servidor Web de la siguiente capa.
- **Capa de Servicio de Administración.** Su objetivo es interconectar los datos recibidos por el servidor back-end con el internet. Está constituido por una plataforma, en la cual se implementa una base de datos que almacena los mensajes recibidos, y un servidor Web inmerso en una plataforma de integración. Esta última se encarga de la elaboración de una API REST que, al relacionarse con una base de datos, permite o no el acceso y uso de los datos recibidos en el lenguaje de intercambio utilizado, además se implementa un servidor, quien será el encargado de atender las solicitudes realizadas por la capa de aplicación.
- **Capa de aplicación:** se genera la comunicación entre los clientes front – end y el servidor de la capa de transporte, exponiendo de una forma fácil a los usuarios los resultados que se obtiene de las solicitudes enviadas por el servidor.

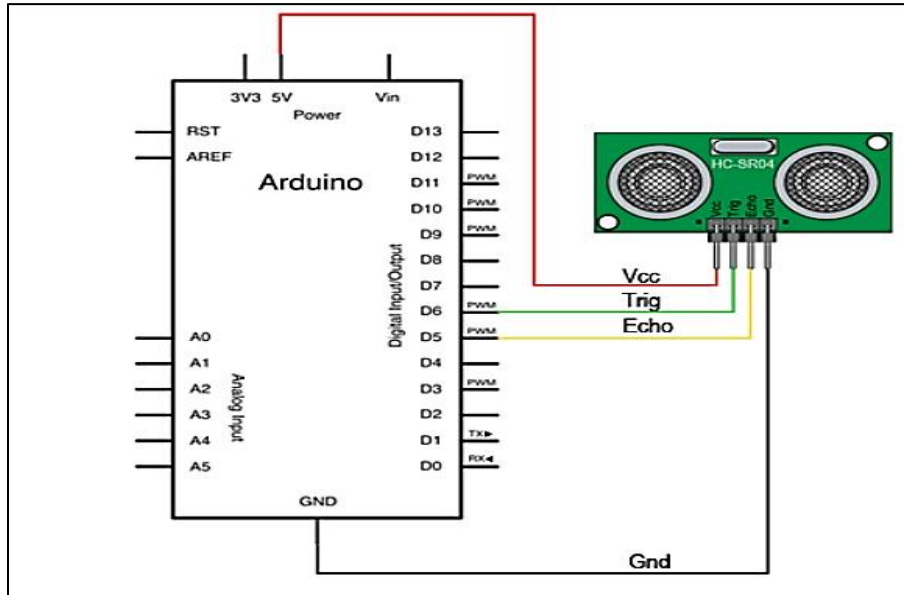
2.4. Sensores y dispositivos

Son dispositivos electrónicos que transforman una cantidad física que puede ser temperatura, nivel, intensidad, opacidad, entre otros, en una cantidad de carga o voltaje eléctrico, el cual se integra en una cadena de señales para determinados procedimientos. La curva de calibración de cada tipo de sensor es la curva que se genera por la evolución de la cantidad de voltaje eléctrico, característica del sensor en base a la cantidad física que recibe (Barrera & Ros, 2016).

2.4.1. Sensor ultrasónico

Figura 2.

Estructura de un sensor ultrasónico



Fuente: Morocho, (2018)

El sensor ultrasónico es un dispositivo que permite calcular las distancias de algún objeto mediante la transmisión de pulsos de altas frecuencias que no pueden ser receptadas por el oído humano. El pulso emitido rebota al momento de chocar en la superficie del objeto en cuestión y retorna al sensor, el cual posteriormente mide el tiempo de retorno de la señal o pulso para calcular la distancia a la que se encuentra dicho objeto (Morocho, 2018).

Las ventajas de estos sensores ultrasónicos son que son fáciles de manejar y de bajo costo, además que permiten medir distancias en rangos de 2cm a 1 metro como se menciona en la tabla 1, con un margen de error de 0.3 cm, el cual es relativamente bajo. El sensor más

utilizado es el HC – SR04, especialmente en proyectos de robótica, detección de obstáculos, distancias y posiciones de objetos (Morocho, 2018).

Tabla 1.

Características del Sensor Ultrasónico

CARACTERÍSTICAS DEL SENSOR ULTRASÓNICO	
Alimentación	5 V
Interfaz sencilla	4 hilos Vcc, Trigg, Echo, GND
Rango de medición	2 cm a 400 cm
Corriente de alimentación	15mA
Frecuencia de pulso	40KHz
Apertura del pulso ultrasónico	15°
Señal de disparo	10 uS
Dimensiones del modulo	45 x 20 x 15 mm

Fuente: Morocho, (2018).

2.4.2. Placas de desarrollo o gestión

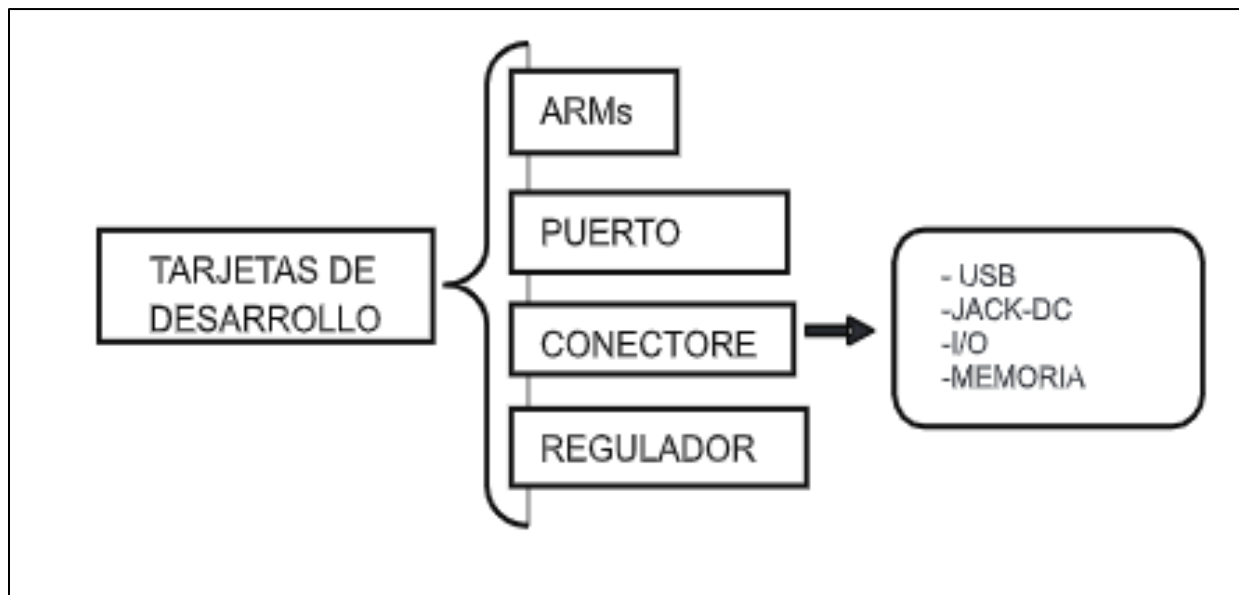
Las placas de desarrollo son dispositivos o circuitos electrónicos que cuentan con un microcontrolador o dispositivo lógico reprogramable, el cual puede ejecutar instrucciones para un fin específico. Generalmente estas placas cuentan con puertos, conectores y reguladores para permitir la comunicación con sensores externos, permitiendo a los usuarios acceder de manera fácil y rápida a los periféricos para la realización de aplicaciones y pruebas, haciendo uso de la electrónica digital (Higuera, 2019).

Según menciona (Higuera, 2019), las tarjetas o placas de desarrollo constan de elementos primordiales para su correcto funcionamiento como son el microcontrolador, puerto JTAG,

conectores de USB, Jack DC, I/O o memoria MicroSD y un regulador. En la figura 3 se puede observar los elementos más comunes de las placas de desarrollo.

Figura 3.

Elementos más comunes de las tarjetas de desarrollo



Fuente: Palma, (2011)

- **Microcontrolador ARM:** Basados en la tecnología RISC, con algunas modificaciones que le permiten usar las memorias y periféricos de una manera más eficiente, reduciendo el consumo de energía y aumentando la capacidad de procesamiento, convirtiéndose en una buena alternativa para el diseño de dispositivos específicos como celulares, consolas de juegos o sistemas embebidos.
- **Puerto JTAG.** (Join Test Action Group) es utilizado para testear los circuitos lógicos y comprobar el funcionamiento de los submódulos de los circuitos integrados. Además, es aplicado como mecanismo para la depuración de aplicaciones embebidas.

- **Reguladores.** Tienen el objetivo de regular el voltaje mediante la reducción de la tensión de alimentación hasta los valores especificados para los componentes de la tarjeta. Una tarjeta de desarrollo estándar TTL puede utilizar un voltaje de 5V, las que operan los ARM usa 3.3 V y existen casos de 2.5 V y 1.8 V para otro tipo de tarjetas lógicas.
- **Conectores.** Los más usados en las tarjetas de desarrollo son: USB, mini USB, memoria MicroSD y Jack – DC.
- **USB y mini USB.** “*Universal Serial Bus*”, el cual, para conectar periféricos, permite transmitir los datos a una velocidad de 12 Mbps. En las tarjetas de desarrollo se utiliza el conector de Mini USB en modo USB – A en host y Device.
- **JACK – DC.** En caso de no requerir la conexión a un computador, este conector se usa para la alimentación externa de las tarjetas.
- **Conectores I/O.** Generalmente están conectados a cada uno de los pines del ARM, lo que posibilita unirlos con otro tipo de circuito. Son conocidos como regletas.
- **Conector para memoria MicroSD.** Las tarjetas al poseer interfaces SPI, pueden soportar tarjetas de memoria SD, MMC o MicroSD. Estas últimas son las más sencillas de utilizar debido a que usan poco espacio, son económicas y facilitan la comunicación entre el ARM. Gracias a sus características son las más usadas y solo se implementan con un conector.

2.5. Arduino.

Es una plataforma de desarrollo de microcontroladores de hardware libre que permiten establecer conexiones con diferentes sensores y actuadores de forma sencilla, utiliza el software Arduino IDE mediante el uso de un lenguaje de programación C++ , para cargarlo en una placa

física con diseños mejorados lo cual permite dar soluciones a diversas problemáticas (Arduino, 2021).

El microcontrolador de Arduino tiene una interfaz de entrada lo cual permite conectar varios tipos de periféricos que trasladan la información al microcontrolador para procesarlos. También cuenta con una interfaz de salida que lleva la información que ha procesado Arduino a otros periféricos (Arduino, 2021).

2.5.1. Tipos de placas Arduino

En el mercado hay gran variedad de microcontroladores con distintas características técnicas, entre las cuales destacan el tipo de procesador, voltaje de uso, velocidad, entradas y salidas analógicas, etc. La tabla 2 muestra distintas placas Arduino disponibles en el mercado y sus características técnicas que diferencia una de otra.

Tabla 2.

Tipos de Arduino

NOMBRE	PROCESADOR	VOLTAJE DE USO/ENTRADA	VELOC. CPU	ENT/SAL ANALÓGICA	E/S DIGITAL/ PWM	SRAM [KB]	USB
101	Intel Curie	3.3V / 7-12V	32MHz	6/0	14/4	24	Regular
Gemma	Attiny85	3.3V / 4-16V	8MHz	1/0	3/2	0.5	Micro
LilyPad USB	Atmega32U4	3.3V / 3.8-5V	8MHz	4/0	9/4	2.5	Micro
Micro	Atmega32U4	5V / 7-12V	16MHz	12/0	20/7	2.5	Micro
MKR1000	SAMD21 Cortex-M0+	3.3V / 5V	48MHz	7/1	8/4	32	Micro
Uno	Atmega328P	5V / 7-12V	16MHz	6/0	14/6	2	Regular
MKRFOX1200	Cortex-M0+	5V	48MHz	7/1	8/4	32	Micro
Ethernet	Atmega328P	5V / 7-12V	16MHz	6/0	14/4	2	Regular

Fuente: Arduino, (2022)

2.5.2. *Dispositivo Conexión a Sigfox*

Sigfox utiliza tres tipos de placas para generar la comunicación de los dispositivos objetos que son:

- **Sigfox Thinxtra RCZ1.** Es una placa que tiene una frecuencia de 868MHz y que ha sido diseñada para Europa y el Medio Oriente bajo un sistema M2COMM. La potencia de salida esta entre 14 dBm hasta 17 dBm y se puede modificar a través del comando AT y así obtener un alto rendimiento, calidad y a un precio económico (Bermeo, 2019).
- **Sigfox Thinxtra RCZ2.** Es una placa tipo transceptor que se diferencia por su frecuencia central que es de 90 MHz y que ha sido diseñada para ser usada en países como México, Estados Unidos y Brasil, mediante un sistema M2COMM. Su potencia de salida es de 22dBm / 100mW con un consumo de corriente de 30Ma Tx, siendo un paquete de alto rendimiento, alta calidad y precio económico (Bermeo, 2019).
- **Sigfox Thinxtra RCZ4. (OPERACION)**Esta placa maneja en un protocolo inalámbrico WSSFM10 y es apta para Sudamérica, operando con una frecuencia de 920MHz. Además, contiene 3 sensores embebidos, uno de ellos siendo el acelerómetro de 3 ejes, mismo que es micro maquinado con una resolución de 14 bits. Su objetivo es medir la fuerza, ya sea dinámica de aceleración o estática, y posee dos sensores digitales, uno de temperatura que opera en 300 – 1100hPa ± 0.12 hPa y otro de presión que están en grados de -40°C $- + 85^{\circ}\pm 1.0$, además posee sensores de luz, de interruptores de láminas, Leds, entre otros (Bermeo, 2019).

En la tabla 3 se muestra una comparación de los diferentes dispositivos de conexión a Sigfox y sus respectivas características técnicas.

Tabla 3.

Dispositivos de conexión a Sigfox

DISPOSITIVO	FRECUENCIA	SISTEMA	POTENCIA	COBERTURA
Sigfox Thinxtra RCZ1	868MHz	M2COMM	14-17 dBm	Europa, Medio Oriente
Sigfox Thinxtra RCZ2	90 MHz	M2COMM.	22dBm / 100mW	México, Estados Unidos y Brasil
Sigfox Thinxtra RCZ4	920MHz	M2COMM	300 – 1100hPa ±0.12hPa	Sudamérica

Fuente: Bermeo, (2019)

2.6.Tecnología LPWAN

La tecnología LPWAN representa un papel importante dentro de la construcción del internet de las cosas. Estas plataformas de redes permiten tener muchos sensores recopilando y enviando datos a largo alcance con una optimización de energía a menores costos que otras opciones de conectividad. Es importante saber que LPWAN es un término general y que hay diferentes tecnologías que permiten esta conectividad, por lo que hay que saber identificar la opción más adecuada dependiendo las necesidades y requerimientos del proyecto a realizar (López Vicario & Vilajosana Guillén, 2019). En la actualidad las redes que permiten esta conectividad son las siguientes:

- SigFox
- LoRa
- LTE-M
- NB-IoT

A continuación, en la tabla 4, se muestran las diferentes características de las redes PLWAN donde se compara cada tecnología con el objetivo de resaltar los puntos técnicos más representativos de cada una y que pueden ser considerados a la hora de desarrollar un proyecto.

Tabla 4.

Aspectos Técnicos de redes LPWAN

	SIGFOX	LoRa WAN	LTE- M	NB-IoT
Modulación	BPSK	CSS	QPSK	QPSK
Frecuencia	Bandas ISM sin licencia 868MHz en Europa 915MHz en Norte América 433MHz en Asia		Bandas de frecuencia LTE con licencia	Bandas de frecuencia LTE con licencia y 2G
Ancho de banda	100 Hz	250 kHz y 125 kHz	1,4 MHz	200 kHz
Velocidad máx. De datos	100 bps	50 kbps	1 Mbps	100 kbps
Bidireccional	Half-dúplex	Half-dúplex	Full-dúplex	Half-duplex
Máximo mensajes / día	140 (UL), 4 (DL)	Ilimitado	Ilimitado	Ilimitado
Longitud máxima de carga útil	12 bytes (UL) 8 bytes (DL)	243 bytes	27.2 (DL), 62.5 (UL)	1600 bytes
Distancia	10 km (urbano) 40 km (rural)	5 km (urbano) 20 km (rural)	1km (urbano) 5km (rural)	1 km (urbano) 10 km (rural)
Duración batería	20 años	15 años	10 años	10 años
Inmunidad a interferencias	Muy alto	Muy alto	Medio	Bajo
Autenticación y encriptación	No soportado	Sí (AES-128)	Sí (LTE)	Sí (LTE)
Tasa de datos adaptativa	No	Si	No	No
Handover	Los dispositivos finales no se unen a una sola estación base			
Comunicación en tiempo real	No	No	Si	No
Localización	Sí (RSSI)	Sí (TDOA)	Sí	No (bajo

Fuente: López Vicario & Vilajosana Guillén, (2019)

2.7. Tecnología Sigfox

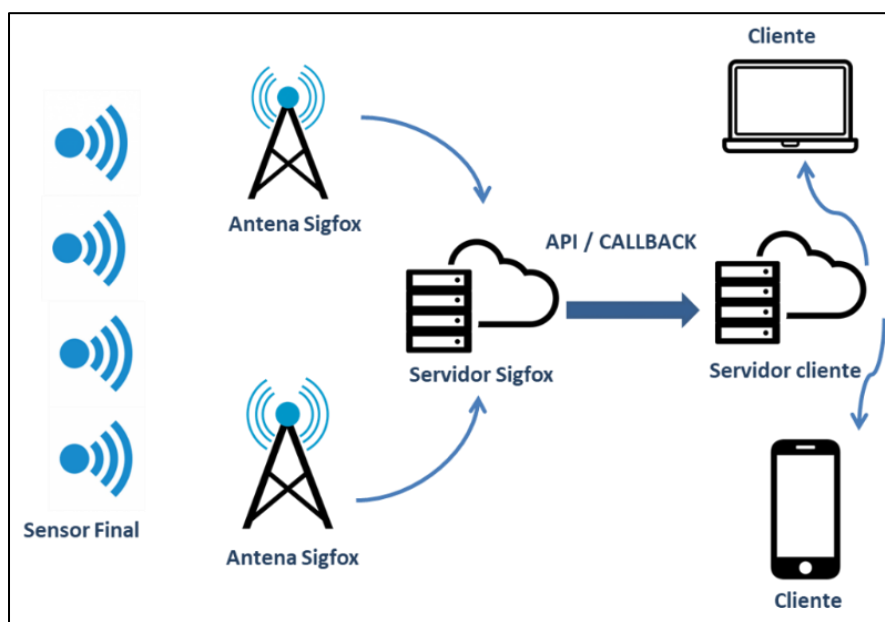
Sigfox trabaja como un operador de red LPWAN, de tal manera que permite dar una solución de conectividad para equipos con tecnología IoT (Pinto, 2020), SigFox está dedicado a complementar los sistemas de conectividad para garantizar el servicio y confiabilidad dirigido a automatizar procesos mediante su relación con el desarrollo del Internet de las cosas (IoT), Es una red diseñada para transmisión de información de baja velocidad, bajos costos y reducido consumo de energía por los dispositivos desconectados.

2.7.1. Características

La infraestructura Sigfox está diseñada con antenas y estaciones bases que son repartidas por el territorio, estos se comunican con la Red Sigfox en dónde se almacena, codifica y se presenta los datos obtenidos (Mora et al., 2016).

Figura 4.

Estructura red Sigfox.



Fuente: DSET Energy, (2021)

Se ha utilizado para la instalación remota de sensores que permiten el monitoreo en escalas, domiciliarias, comunitarias, locales, y de aspectos estratégicos para la automatización, manipulación y conclusión de la información obtenida, dirigido al uso eficiente de recursos (Castillo P. , 2018). A continuación, se presenta la estructura Sigfox:

El servicio Sigfox debe contratarse previamente para proyectos de metering, en sus productos tienen categorías que van desde 2 hasta 140 envíos de información diaria. Posterior se recupera la información de la nube Sigfox a través de Callback, para tratar los datos obtenidos en la interfaz diseñada (Molano et al., 2021).

Además, se puede recibir mensajes desde el sistema hacia los dispositivos sensores instalados, sin embargo, no son sincrónicos y pueden aplicarse hacia el área de domótica, Smart Grid, logística, industria y ciudades inteligentes. Esto debido a que no requiere de grandes cantidades de energía, por lo cual, no debe estar conectado directo al sistema interconectado de energía o poseer baterías de capacidades elevadas (Molano et al., 2021). Las características de la red Sigfox se especifican en la Tabla 5.

Tabla 5.

Especificaciones técnicas red Sigfox.

CARACTERISTICAS	VALOR
Frequency band	EU: 868 MHz - US: 902 MHz
Range (estimated)	3 – 8 km (Urbano) - 30 – 50 km (Rural)
Link Budget	156 dB
Data Throughput	EU: 100 bps - US: 600 bps up to 140 messages per day
Power consumption	Tx: < 50 mA - Rx: 10 – 40 mA- Sleep: < 0.01 mA
Packet size	12 bytes
Devices per access point	1 millon
Security	No network encryption
Status	Being deployed

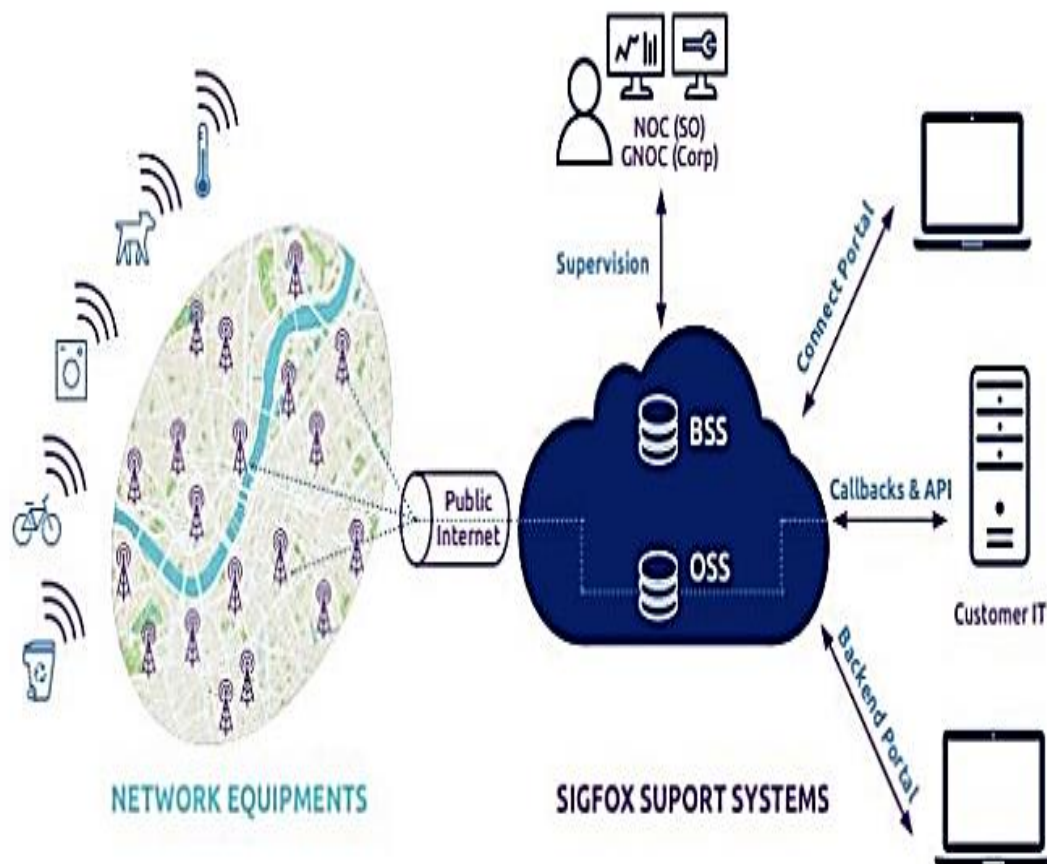
Fuente: Sigfox, (2020)

2.7.2. Arquitectura de sistema Sigfox

Según (Munera, 2018), la red Sigfox tiene principalmente cuatro capas: almacenamiento, comunicación, back-end y presentación, en la figura 5 se puede apreciar la arquitectura del sistema.

Figura 5.

Arquitectura del sistema Sigfox



Fuente: Munera, (2018)

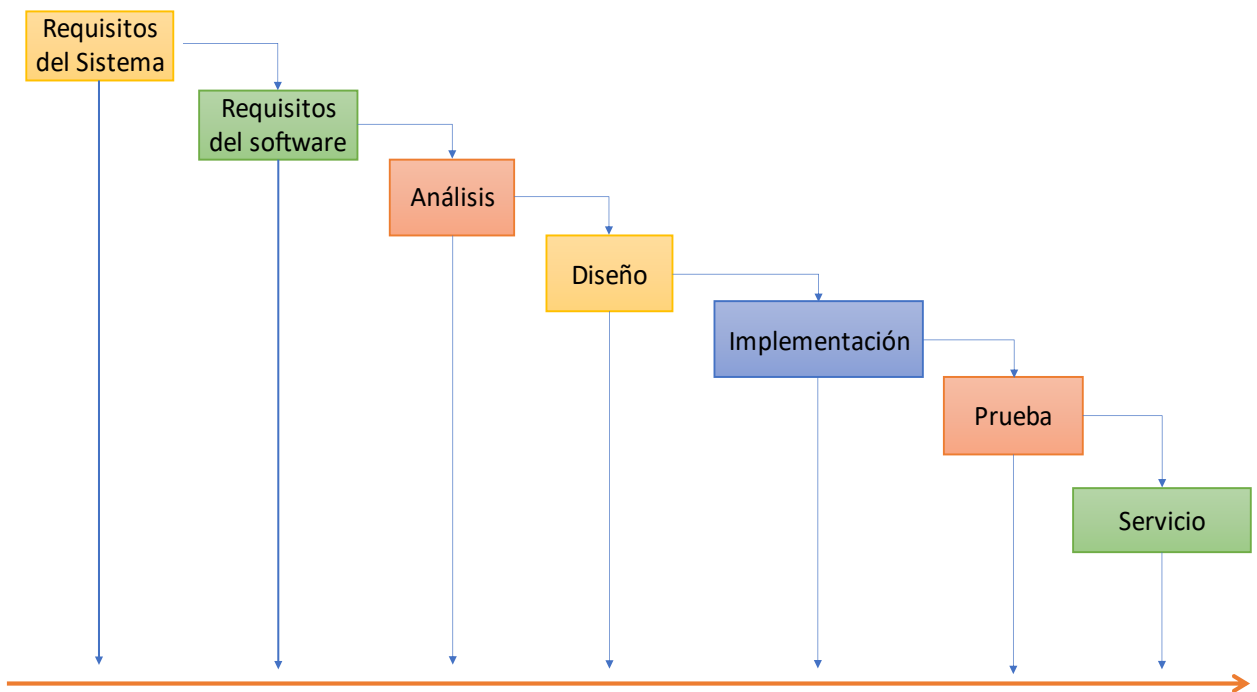
- **Almacenamiento:** El almacenamiento de los datos se realiza a través de dos capas principales.

- **Equipamiento, dispositivos y antenas.** Esta capa consta de todos los dispositivos encargados de enviar los mensajes, junto con las estaciones base que los reciben.
- **Sistema de soporte.** Esta capa es la menos visible, pero engloba una serie de funciones que la convierten en la más compleja. Tiene el objetivo de procesar los mensajes y generar mensajes finales a los clientes. Además, ofrece las funcionalidades a los usuarios (APIs) de generar una acción de respuesta después de la llegada de mensajes y lograr controlar, monitorizar sus dispositivos, herramientas de análisis y demás acciones.
- **Comunicación.** Es toda la información enviada de forma inalámbrica desde los dispositivos hasta las estaciones base, donde esta se encarga de comunicarse con los servidores de Sigfox. Estos procesos lo realizan generalmente con el uso de DLS, 3G y 4G.
- **Back – end.** Es un servidor que se encarga de manejar los mensajes. En los casos que el mismo mensaje se reciba varias veces se debe a la recepción cooperativa, por lo que el servidor debe asegurarse de almacenar una sola vez. El mensaje no es lo único que se guarda ya que se guardan datos adicionales o “*metadata*”, asociada al mismo, lo que permite que el usuario permita recuperarla.
- **Acceso y presentación.** Finalmente, los usuarios pueden acceder a los mensajes mediante la interfaz web y la API, además frente a distintos eventos pueden definir “*callback*” HTTP. Esto es de gran utilidad en el uso de un conjunto de datos propia o de una nube adicional.

2.8. Modelo de cascada

Figura 6.

Diagrama de Flujo del proceso general del modelo Cascada



Fuente: Hardyanto, (2016)

El modelo de cascada es un método de desarrollo de procedimientos de forma lineal, dividiendo los procesos en sucesivas fases para el cumplimiento del proyecto. Este modelo se utiliza convencionalmente para el desarrollo de software y sus fases se ejecutan una sola vez, para originar hipótesis de base para la siguiente fase (García, 2018). Entre las características más importantes de este modelo se puede citar:

- Es el más usado en el desarrollo de softwares
- El desarrollo de software es la sucesión de etapas que genera productos intermedios.
- El cambio de orden de cualquiera de fases generará un producto de menor calidad.

2.8.1. *Etapas / iteraciones*

Según (Hardyanto, 2016), el modelo de cascada se compone de siete fases que deben ser cumplidas secuencialmente en el desarrollo de algún software:

- **Requisitos de sistema:** se determina de forma general los instrumentos, dispositivos, procedimientos y software necesario para el desarrollo del proyecto. También se analiza el objetivo que pretende cumplir el proyecto en conjunto con las técnicas que se van a aplicar. En esta etapa se analiza los requerimientos de los usuarios finales del software, para determinar los objetivos que se deben cubrir. Se debe conocer todo lo que requiere el cliente del sistema y se especifica lo que se debe realizar sin entrar en detalles.
- **Requisitos de software:** en esta etapa se analizan las necesidades del personal directo que va a utilizar el software. Se desarrolla un documento que especifique los requisitos que se requiere del programa, denominado “Memoria SRD” sin entrar en detalles del programa. El ingeniero de software (Analistas) debe comprender la arquitectura del software, así como la función, el rendimiento y las interfaces requeridas.
- **Análisis:** Una vez determinados los requisitos, se analiza la información y conceptos del objeto del sistema, en el caso del proyecto, datos específicos derivados de la recolección de residuos sólidos para determinar qué objetivos debe cubrir. El producto de esta fase es una memoria llamada SRD (Documento de Especificación de Requisitos), la cual contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos. Es importante señalar que en esta etapa se deben verificar todo lo que se requiere en el sistema,

como la interfaz, la función y el rendimiento requerido a través de un esquema de análisis y será aquello lo que seguirá en las siguientes etapas, ya que no se pueden solicitar nuevos requisitos a mitad del proceso de elaboración del software.

- **Diseño:** en esta etapa se determinará como va a funcionar el software de forma general. Se debe especificar las consideraciones relacionadas con la implementación del software, el hardware que se aplicará y la estructura que va a seguir el programa.
- **Implementación:** una vez terminado el diseño se procede a la instalación de los dispositivos electrónicos en campo, y la calibración de las señales generadas hacia el software diseñado para determinar un producto final de análisis.
- **Prueba:** es conocida como la etapa del test, aquí se realiza la comprobación del funcionamiento del software para determinar si cumple o no con los requisitos establecidos previamente. Esta etapa debe ser comprobada por un técnico ajeno al que desarrollo el software. Existen cuatro tipos de pruebas más generales utilizadas; a) pruebas de unidad, la cual se centra en el módulo, usando la descripción del modelo como guía. Prueban todos los caminos importantes con el objetivo de descubrir errores dentro del módulo; b) la prueba de integración, quien tiene como objetivo ejecutar una estructura del programa que esté de acuerdo con el diseño a partir de los módulos testeados en la prueba de unidad. La prueba de integración tiene dos formas: integración no elemental, que combina todos los módulos de manera anticipada probando todo el software en conjunto, y la integración incremental, la cual prueba el software en pequeños

segmentos. En tercer lugar, está la prueba del sistema (c), donde se verifica la funcionalidad y el rendimiento del sistema global; y d) pruebas de regresión, las cuales son una estrategia de prueba donde se vuelve a realizar la prueba ejecutadas anteriormente en una nueva versión modificada, con el objetivo de asegurar la calidad de la nueva funcionalidad.

- **Servicio:** El sistema una vez se encuentra funcionando, debe ser ejecutado y comprobado por el por el cliente, y se debe destinar recursos para su mantenimiento y ajuste en función de las necesidades del cliente.

2.9. Algoritmo de programación

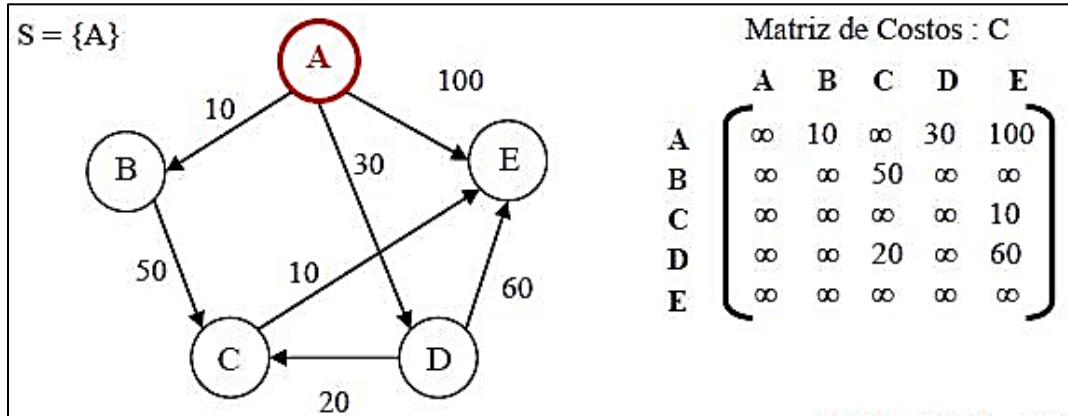
Un algoritmo es un conjunto o serie de instrucciones secuenciales y reglas bien definidas que permiten solucionar un problema mediante el desarrollo de un cómputo, procesamiento de datos y estimación del resultado eficiente basado en la programación. Los algoritmos pueden ser cuantitativos, es decir depende de un cálculo matemático o cualitativo en dónde se desarrollan secuencias lógicas. Es necesario conocer el problema que se quiere resolver con los factores que influyen en este, para desarrollar un algoritmo que en forma real resuelva esta necesidad (Castillo & Espitia, 2020).

También llamado algoritmo de caminos es un algoritmo que permite calcular la distancia más corta desde el nodo inicial a cualquier nodo de una red. Es decir, permite conocer el camino que represente menores costos, desde un vértice hasta otro. El algoritmo utiliza una matriz de costos.

2.9.1. Algoritmo de Dijkstra

Figura 7.

Modelo de Algoritmo de Dijkstra



Fuente: Torrubia & Terrazas, (2012)

2.10. IoT aplicado a la recolección de RSU

De acuerdo a publicaciones de EPA - United States Environmental Protection Agency, Eurostat y The World Bank Group, actualmente cada persona en el mundo genera de 1 a 1.2 kg de residuos por día, y se estima que para el 2025 el per cápita ascenderá a 1.5 kg por persona por día, siendo uno de los mayores problemas ambientales que la población está enfrentando. Paralelamente, la conectividad y el big data está tomando cada día más fuerza, que combinados conforman el internet de las cosas o KIoT – Internet of Things, llevando a cabo la cuarta revolución industrial, por lo que sería lógico utilizar la tecnología en la solución de los problemas que nos aquejan hoy en día.

En este contexto, la empresa Bin – e es una de las pioneras en la fabricación de contenedores inteligentes de residuos llamados Smart Bins, siendo capaces de identificar y clasificar los RSU a través de sensores instalados en su interior. Además, comprimen los

desechos y notifica a la empresa de recolección y gestión sobre los niveles de llenado y el tipo de residuo que almacena. Este sistema también permite el análisis de patrones de comportamiento de las personas con sus desperdicios, siendo información importante para la optimización de la logística y recolección de residuos y al mismo tiempo para mejorar los hábitos de la población en el consumo y desperdicio de alimentos y otros materiales de uso diario (EPA.gov, 2021).

2.11. Apache Cordova

Es un esquema de código abierto, para desarrollar aplicaciones móviles, que simplifica la programación, debido a que estructura el código ahorrando tiempo y esfuerzos, este framework utiliza las tecnologías web HTML5, JavaScript y CSS3. La ejecución de aplicaciones se realiza dentro de contenedores que son dirigidos a plataformas las cuales se encuentran basadas en enlaces de API, de tal manera que permiten cumplir con estándares que accederán a las capacidades de sensores, dispositivos, etc. (Apache Cordova, 2022).

2.11.1. Arquitectura de Apache Cordova

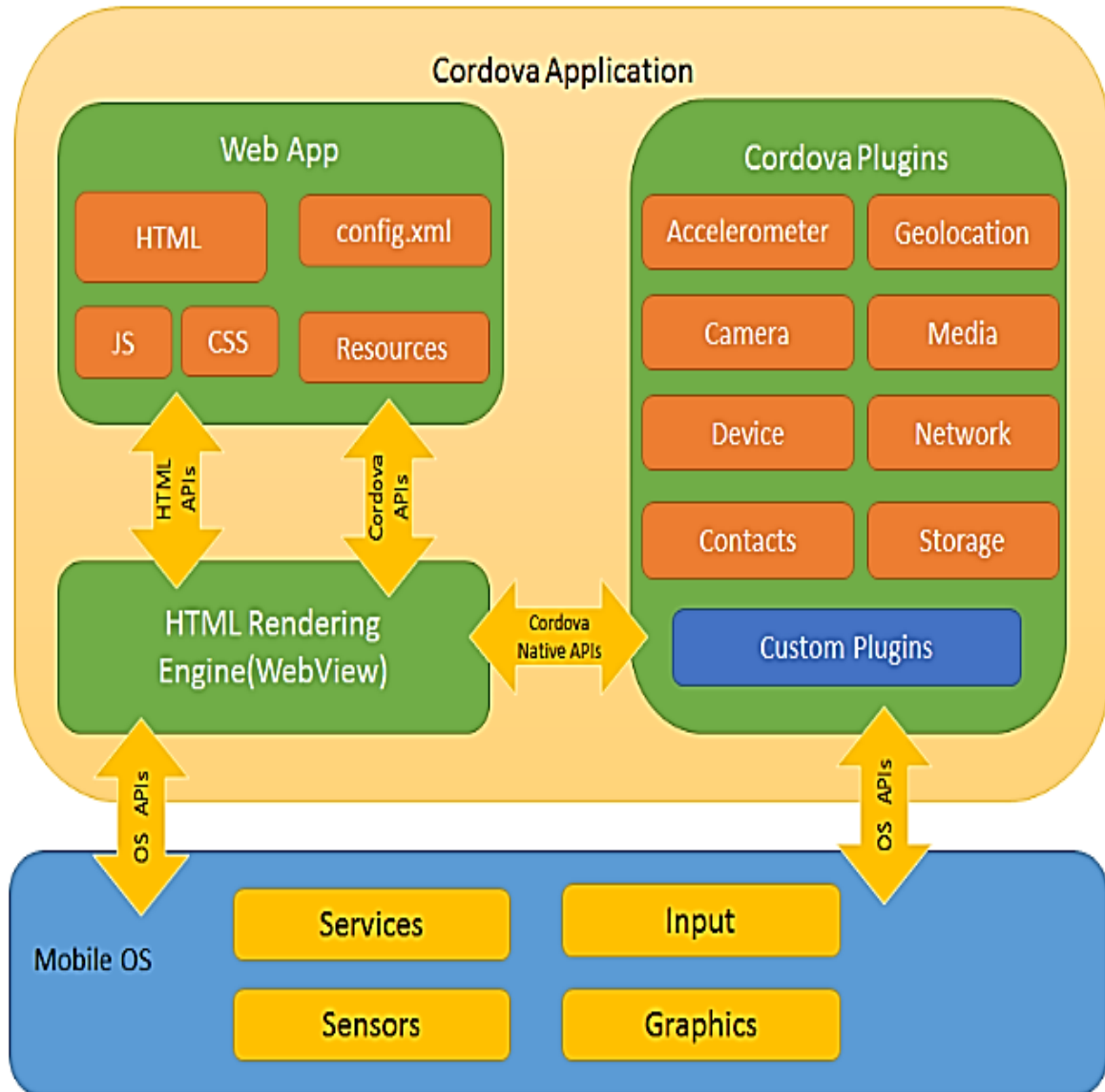
La arquitectura toma como base:

- **Vista Web:** proporciona a la aplicación una interfaz de usuario
- **Aplicación Web:** En este apartado se tiene el código de la aplicación ejecutado en una vista web, el cual hace referencia a recursos necesarios para su ejecución.
- **Plugins:** Son complementos de terceros, los cuales se obtienen de enlaces adicionales.

La arquitectura que se maneja dentro de Apache Cordova se muestran en la figura a continuación, en la figura 8.

Figura 8.

Arquitectura de Apache Cordova



Fuente: Apache Cordova, (2022)

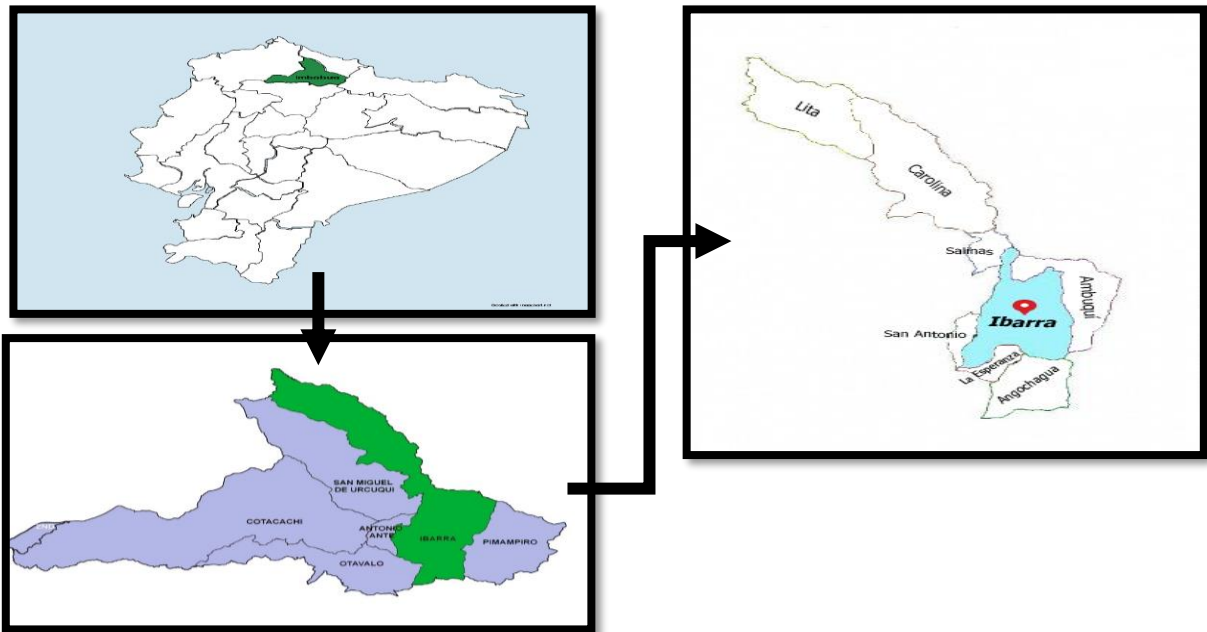
CAPÍTULO III: METODOLOGÍA

En este capítulo se presenta las etapas del diseño para el sistema de automatización de la recolección de residuos sólidos por medio de un prototipo utilizando la red Sigfox, siguiendo una metodología “Descriptiva – Experimental”, debido a que permite utilizar una serie de herramientas y técnicas que permitan explicar y resolver el fenómeno en estudio con la aplicación de diversos criterios y que permitan realizar el sistema planteado.

3.1. Macro Localización

Figura 9.

Macro Localización



Fuente: MapChart, (Autor)

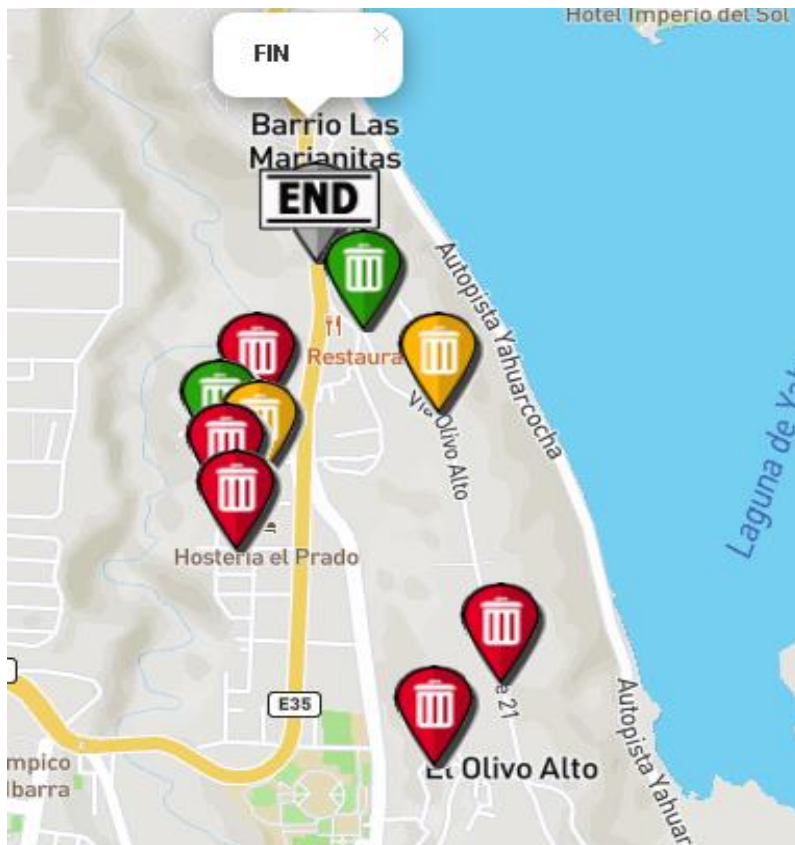
3.2. Micro Localización

El presente proyecto de titulación se desarrolla en Ecuador, provincia de Imbabura, cantón Ibarra, específicamente dentro de la parroquia priorato, Barrio el Olivo, en este

sector se toma en cuenta la entrevista realizada a una moradora del sector Sra. Mariana Realpe, para determinar las principales necesidades, con respecto al sistema de recolección actual.

Figura 10.

Micro Localización



Fuente: Aplicación Web, Autor

3.3. Población y muestra

En el sector del Olivo existen aproximadamente 100 contenedores, en dónde se toma como muestra representativa el 10% del total, determinando un valor de 10 contenedores como muestra. Se tomó como población la ubicada en el barrio El Olivo la población del sector es aproximadamente 100 familias, de acuerdo con el dirigente barrial

quien comenta que ese número es el de familias fijas que viven en el sector desde hace años, sin embargo, por temporada de clases varios estudiantes universitarios arriendan las viviendas esporádicamente.

3.3.1. Instrumentos de recolección de información

Como instrumento de recolección de información se realiza una entrevista dirigida a una moradora del sector, en este caso la señora Mariana Realpe, propietaria de una tienda ubicada en la arteria principal del sector, quien ayuda con información clave para el desarrollo y los requerimientos del sistema, como Anexo 2 se muestran la entrevista realizada.

3.3.2. Resultados de la recolección de información

Se muestran las respuestas obtenidas al realizar la entrevista a la señora Mariana Realpe, moradora de El Olivo, quien comenta que: “El sistema actual de recolección de residuos no es eficiente”. El audio de la entrevista realizada se encuentra en el link insertado en el anexo 2.

3.4. Diseño de prototipo

Se utilizará la metodología de cascada debido a que permite dividir los procesos para el desarrollo de forma sucesiva de un proyecto (Cortez, 2018). Las fases del modelo de cascada comprenden las siguientes etapas:

3.4.1. Requisitos del sistema

Mediante el análisis del proyecto y la recopilación de información aplicada al prototipo, se determinó los requerimientos a cumplir acorde con las necesidades planteadas. Estos requerimientos deben ser evaluados de manera que cumplan los requisitos enfocados al usuario, sistema y arquitectura del prototipo.

La construcción se elabora en base a tres requerimientos que son: Stakeholders, sistema y arquitectura; cada requerimiento debe ser verificable, que se pueda cumplir y que sea medible y limitable. Se identificó cada requerimiento mediante el uso de abreviaturas que permite un manejo organizado de los datos, como se muestra en la tabla 6

Tabla 6.

Abreviaturas de los requerimientos del sistema

REQUERIMIENTO	ABREVIATURA
Stakeholders	STSR
Sistema	SYSR
Arquitectura	SRSR

Fuente: Autor

3.4.1.1. Requerimientos De Stakeholders

Estos requerimientos ayudan a definir los requisitos que el sistema necesita para su uso, en base a las necesidades del usuario para la resolución de un problema planteado, como lo es el manejo de datos de la cantidad de basura presente en el tacho de basura y el diseño de una ruta de recolección de desechos ideal para optimización de recursos. En la tabla 7 se evaluarán los requerimientos operacionales y de usuarios.

Tabla 7.*Requerimientos De Stakeholders*

STSR					
#	REQUERIMIENTO	PRIORIDAD			
		Alta	Media	Baja	
Requerimientos operacionales					
STSR 1	El sistema debe tener la capacidad de funcionamiento durante al menos 12 horas diaria.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
STSR 2	El sistema debe tener alimentación eléctrica para el funcionamiento del sensor ultrasónico y envío de datos a la red Sigfox, puede ser mediante el uso de baterías	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STSR 3	El sistema debe tener un sensor ultrasónico capaz de captar objetos a una distancia de al menos 2,5 metros desde su ubicación hasta la base del contenedor de basura	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STSR 4	El sistema tendrá la capacidad de conexión y envío de datos a una red alterna, SigFox, en tiempo real para su análisis.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Requerimientos de usuarios					
STSR 5	Los usuarios visualizarán la información de una manera organizada y en tiempo real, mediante el acceso a una plataforma web (Sigfox).	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STSR 6	La plataforma web deberá contener la ubicación de cada uno de los dispositivos que envían datos a la red.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
STSR 7	Los datos serán visualizados en una interfaz web para la parte administrativa.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
STSR 8	Para los recolectores, se tendrá una aplicación móvil, la cual mostrará la ruta de recolección de manera visual.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Nota. Requisitos operacionales y de usuario de acuerdo a la prioridad. Fuente: Autor

Los requerimientos se plantearon mediante el análisis de las necesidades que se tiene para el desarrollo del sistema, el cual medirá la cantidad de desechos presentes dentro del contenedor de basura. Logrando una correcta interacción entre el dispositivo que envía los datos y el usuario que analiza los mismos. En base a la entrevista realizada al dirigente barrial, se determina que, son necesarias una interfaz web con usuarios para la parte administrativa y una aplicación móvil que muestre la ruta para los recolectores.

3.4.1.2. Requerimientos del sistema

Los requerimientos del sistema se determinan en base al tipo de sistema que se tiene y las características y recursos que debe poseer para su correcto funcionamiento para cumplir

una tarea. Estos requerimientos pueden ser interfaces, requerimientos de uso, performance y requerimientos físicos.

Tabla 8.

Requerimientos del sistema.

SYSR				
#	REQUERIMIENTO	PRIORIDAD		
		Alta	Media	Baja
Requerimientos de interfaz				
SYSR 1	Acceso al sistema de programación Arduino	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SYSR 2	Conectividad a red Sigfox	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SYSR 3	Recopilación de datos en la nube Sigfox	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SYSR 4	Manejo de Sistemas de Información Geográfica (SIG) con el Software ARCMAP 10.5.1.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SYSR 5	Manejo de algoritmo de Dijkstra por medio del programa Java, para determinación de ruta con la distancia más corta desde el nodo inicial de una red hasta el nodo final	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Requerimientos de uso				
SYSR 6	Dispositivo de recolección de datos flexible al trabajo en ambientes difíciles (contenedor de basura)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SYSR 7	Recepción de datos de manera inmediata	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requerimientos de performance				
SYSR 8	El dispositivo con el sensor debe ser colocado en la parte superior del contenedor de residuos para identificar su estado de capacidad (vacío o lleno)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SYSR 9	La recepción de datos debe ser de manera continua para observar la dinámica de llenado del contenedor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SYSR 10	La plataforma de Sigfox debe almacenar los datos recibidos y guardarlos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SYSR 11	el Software ARCMAP 10.5.1, identifica los puntos de recolección en donde se encuentran los sensores	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SYSR 12	Mediante el algoritmo de Dijkstra se determina la ruta de recolección optima	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requerimientos de modo y estado				
SYSR 13	El Software ARCMAP, con los datos obtenidos por los sensores en SigFox, determina la ruta de recolección óptima con los puntos necesarios de servicio de recolección detectados.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Requerimientos físicos				
SYSR 14	El sistema de recolección de datos (sensor y microcontrolador) deben estar protegidos por una estructura plástica para evitar daños ante cualquier golpe o situaciones climáticas que dañen el dispositivo.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Nota. Requerimientos necesarios para la interfaz, uso, performance y de estado físico de manera clara y específica para el correcto funcionamiento del sistema de recolección y transmisión de datos. Fuente: Autor

3.4.1.3. Requerimientos de arquitectura

Se hace referencia a los componentes de hardware, software y sus características en base al funcionamiento del sistema a implementar. Se toman en cuenta requisitos lógicos, de diseño, de hardware, software y eléctricos

Tabla 9.*Requerimientos de Arquitectura*

SRSH					
#	REQUERIMIENTO	PRIORIDAD			
		Alta	Media	Baja	
Requerimientos de diseño					
SRSH 1	El sistema de recolección de datos (sensor y microcontrolador) recolecta datos del estado de llenado del contenedor de basura	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 2	Los datos recolectados son receptados por la red SigFox para su almacenamiento en la nube	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 3	Los datos son descargados y analizados para determinar el estado de llenado del contenedor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 4	De acuerdo con el estado de los contenedores (vacío, medio o lleno) se planifica una ruta de recolección de desechos por medio de un software.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Requerimientos lógicos.					
SRSH 5	Software Arduino de programación para conexión con microcontrolador	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 6	Red Sigfox para recepción y almacenamiento de datos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 7	Software de análisis de datos para diseño de ruta de recolección	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Requerimientos de hardware					
SRSH 8	Sensor que transmita pulso ultrasónico, HC-SR04, hacia objetos que estén frente al módulo del sensor para su detección	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 9	Sensor que posea una distancia de detección de 2 hasta 250 cm	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 10	Sensor ultrasónico, HC-SR04 compatible con Arduino	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 11	Microcontrolador módulo de conexión compatible con la red SigFox. kit de desarrollo Thinxtra	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 12	Microcontrolador compatible con la red Sigfox en la región RCZ4 (Ecuador)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Requerimientos de software					
SRSH 13	Plataforma de programación Arduino	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 14	Red de almacenamiento de datos en la nube SigFox	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 15	Plataforma de programación Dijkstra Java	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 16	Plataforma Arc Map 10.5.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 17	Apache Cordova	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Requerimientos eléctricos					
SRSH 18	Batería recargable de 5V y 3 A para funcionamiento de microcontrolador Arduino Uno y Sensor ultrasónico HC-SR04	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Nota. Se describen los requerimientos para el diseño lógico, hardware, software y eléctricos para el sistema planteado, con un rango de prioridad alta, media y baja acorde a su necesidad. Fuente: Autor

3.5. Recursos Humanos

Se menciona a las personas que se involucran dentro del desarrollo del proyecto.

Tabla 10.

Recursos Humanos

Recursos Humanos	
Tutor	Ing. Carlos Vásquez
Asesor 1	Ing. Luis Suarez
Asesor 2	Ing. Marcelo Zambrano
Desarrollador	Sr. Jean Obando
Moradora El Olivo	Sra. Mariana Realpe

Fuente: Autor

3.6. Recursos económicos

Permite evaluar los costos aproximados del prototipo a desarrollar, los cuales se muestran en la tabla 11.

Tabla 11.

Recursos Económicos

#	Material	Costo Unitario	Costo total
1	Sensor ultrasónico HC-SR04	3.50	3.50
2	Microcontrolador Thinxtra Arduino Uno	45.00	45.00
3	Servidor valor mensual	10.00	10.00
4	Dominio	2.00	2.00
5	IP Pública	15.00	15.00
6	Cubierta protectora	30.00	30.00
		TOTAL	\$ 105.50

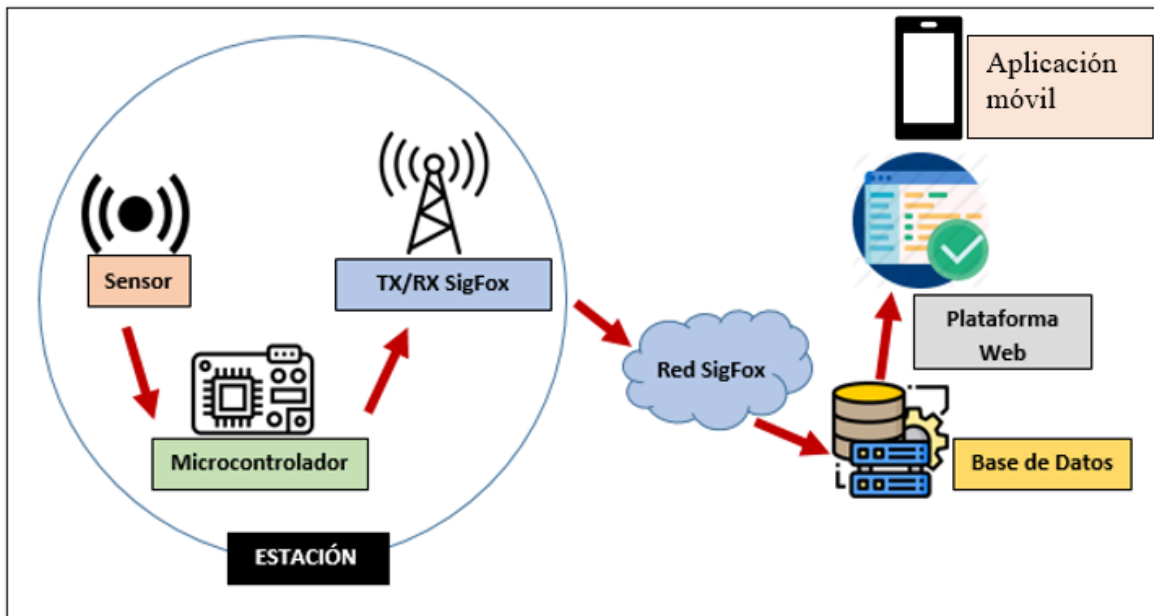
Fuente: Autor

3.7. Recurso tecnológico

Teniendo en cuenta los componentes requeridos para el sistema de recolección de basura, se realizó un diagrama general del sistema planteado identificando sus componentes y funcionamiento.

Figura 11.

Diagrama de Bloques del sistema de Recolección de Desechos



Fuente: Autor

- **Red de Sensores:** Este bloque está conformado por un sensor ultrasónico, el cual es el encargado de identificar el nivel actual de llenado del contenedor de basura y enviar la información al microcontrolador.
- **Microcontrolador:** este bloque está conformado por un Arduino UNO el cual administra la información recibida por el sensor y enviada a la red por medio del módulo de comunicación de Sigfox.
- **TX/RX Sigfox:** En este bloque está el módulo que se encarga de conectar a la red Sigfox y el microcontrolador, este chip tiene la función de mandar y recibir datos con la red.
- **Red Sigfox:** este bloque contiene toda la estructura de la red Sigfox encargada de transporte, acceso y conexión con el internet.

- **Base de Datos:** este bloque contiene la base de datos en el internet y los guarda creando una historia e indicando los datos actuales recibidos de la red de Sigfox
- **Plataforma web:** Este bloque muestra información proporcionada desde la base de datos a los usuarios que tengan acceso a dicha plataforma web, aquí se muestra el estado actual de los contenedores de basura y la ruta de recolección óptima de desechos.
- **Aplicación móvil:** Este bloque muestra la información de rutas para los recolectores, en este apartado se toma como prioridad los contenedores que se encuentran llenos, de tal manera que se elige la ruta más óptima.

A continuación, se detalla los costos de los softwares a utilizar dentro del sistema planteado para su correcto funcionamiento.

Tabla 12.

Recurso tecnológico

Software	Costo
Arduino Software	\$ 0
ArcMap 10.5.1	\$ 0
Apache Cordova	\$ 0
Total	\$ 0

Fuente: Autor

Se determinó los requisitos necesarios para el desarrollo de sistema automatizado de recolección de residuos sólidos mediante el desarrollo de un prototipo de sistema controlado. Se desarrolló el prototipo debido a que el servicio y la Red Sigfox actualmente no está disponible en la ciudad de Ibarra, por ello, se realizará en la ciudad de Quito que cuenta con el servicio para fines investigativos.

Tabla 13.

Componentes del sistema.

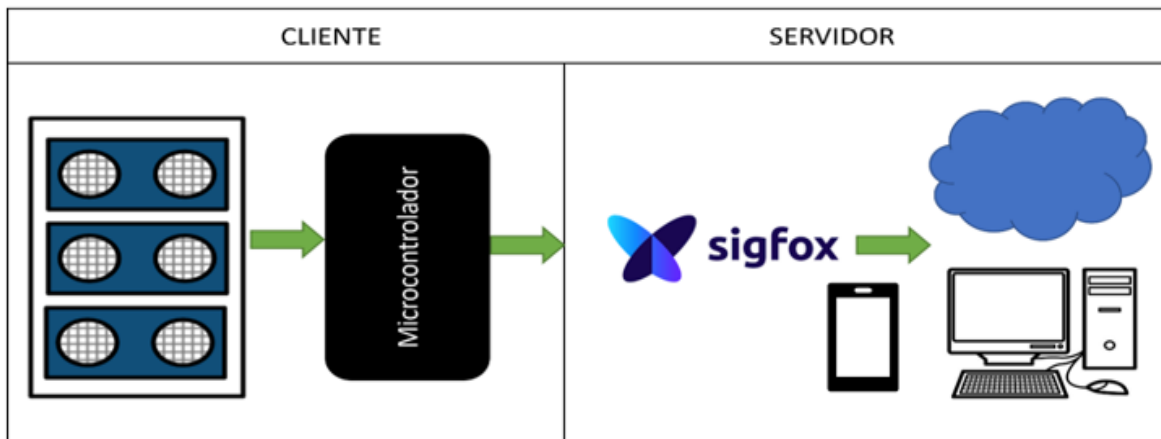
Componentes	
Electrónicos	Sensores ultrasónicos HC-SR04 Microcontrolador Thinextra Arduino Uno
Sigfox	Servicio Sigfox Cloud
Software	ArcMap 10.5.1 Java Apache Cordova

Fuente: Autor

Se Presenta la estructura que tendrá el prototipo en la transmisión y automatización de información por el uso de la Red Sigfox.

Figura 12.

Estructura del sistema de recolección de residuos.



Fuente: Autor.

Los sensores electrónicos son los encargados en medir el nivel de llenado para cada contenedor de basura. Estos sensores son conectados al microcontrolador el cual recibe la información y decodifica con el objeto de enviarlos en un lenguaje aceptado por el módulo de conexión a la Red Sigfox, el cual transmitirá hacia un servidor, conteniendo los datos en

el Cloud para la recuperación posterior por el cliente. La información de rutas se muestra a través de una aplicación móvil para los recolectores.

3.8. Modelación de rutas

Se determinó como herramienta para la modelación geográfica de las zonas de recolección en conjunto con las rutas óptimas los Sistemas de Información Geográfica (SIG) con el Software ARCMAP 10.5.1. Se modeló el barrio “El Olivo”, en la ciudad de Ibarra aplicando coordenadas UTM para el desarrollo de las rutas. Se utilizó la herramienta Vector Line para la delimitación de las principales vías del sector en conformidad con el documento de vías principales y secundarias, obtenido del Sistema Nacional de información (SIN) y del Instituto Geográfico Militar (IGM) del año 2018.

Se procedió a colocar 10 contenedores en el prototipo con las coordenadas de las ubicaciones con mayor porcentaje demográfico de la zona, situadas cerca de tiendas, lugares de estudio, vías principales, entre otros. Se georreferenció los contenedores y se procedió a realizar las rutas posibles que deben seguir en el proceso de recolección en base a los siguientes criterios:

- Vía principal o secundaria.
- Vía unidireccional o bidireccional.
- Cercanía a puntos de recolección.
- Puntos de descarga de residuos sólidos.
- Número de contenedores llenos.

Se modelaron las rutas en base a los criterios planteados del prototipo de 10 contenedores que serán seleccionadas en base al algoritmo desarrollado para determinar la distancia más corta entre puntos por el algoritmo de Dijkstra.

3.9. Algoritmo de rutas

Se aplicó el algoritmo de Dijkstra por medio del programa Java, el cual nos permite determinar la ruta con la distancia más corta desde el nodo inicial de una red hasta el nodo final para definir las rutas de recolección en base a la información obtenida de los sensores. Se debe colocar la distribución espacial de un punto de recolección o vértices (u) a otro punto de recolección (v). La suma de los arcos en base a los pesos determinados va a definir el costo o ruta óptima. El sistema parte desde un nodo de origen el cual es la dirección de llegada desde los puntos de recolección de los camiones. A continuación, se presentan los elementos del algoritmo:

- V: conjunto de vértices,
- E: conjunto de arcos,
- $G = (V, E)$
- S: conjunto de vértices cuyas rutas han sido determinadas por el camino más corto
- V-S: resto de vértices
- d: ARRAY de estimaciones de rutas más cortas a los vértices
- pr: ARRAY de predecesores para cada punto o vértice.

3.9.1. Modelo de ejecución en Java

Se presenta el modelo guía de ejecución utilizado para el desarrollo del algoritmo en base a las rutas de recolección de residuos sólidos en el prototipo planteados.

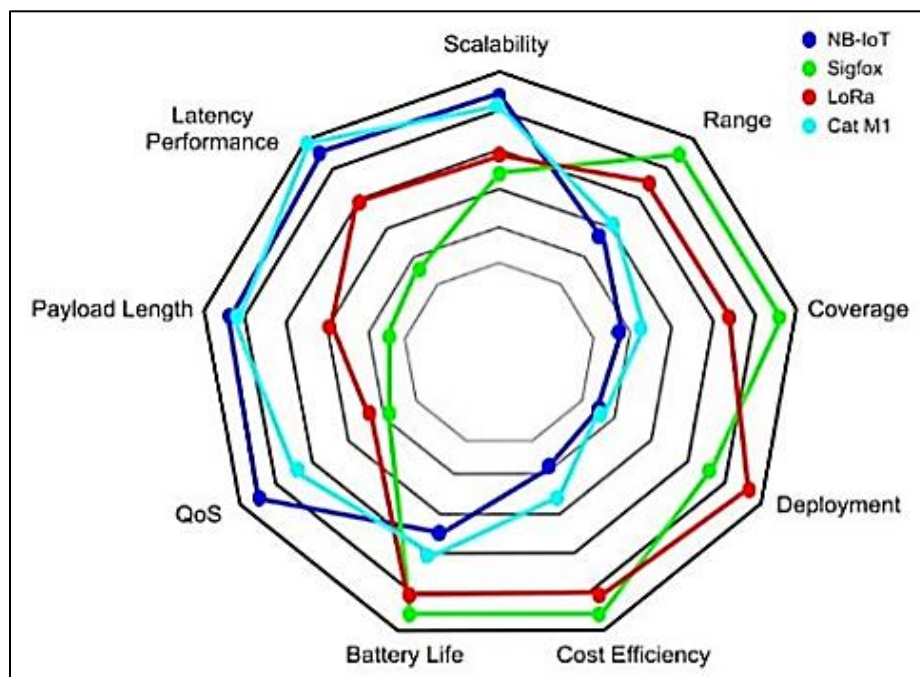
Pasos para el desarrollo del algoritmo:

- Inicializar pr y d
- Colocar $S = \mathcal{A}$
- While $\{V-S\} \neq \mathcal{A}$
- Organizar los vértices $V-S$ y analizar por distancia más eficiente
- Añadir vértices (u), el vértice más cercano en $V-S$, a $S - S = S + (u)$
- Se debe recalculer la distancia hacia todos los vértices en $V-S$

3.10. Selección de tecnología LPWAN

Figura 13.

Ventajas IoT de redes LPWAN



Fuente: López Vicario & Vilajosana Guillén, (2019)

De acuerdo al proyecto planteado, la mejor opción es la red inalámbrica LPWAN Sigfox, ya que cubre el rango de cobertura que tiene la red lo que permite la conexión de dispositivos o plataformas alejados de la estación base, además presenta un importante

desempeño en el ámbito de ahorro de energía y transmisión de datos. Sigfox permite la recolección y visualización de los datos obtenidos mediante sensores ubicados en los contenedores de basura los cuales permiten saber el estado de capacidad de basura en los mismo sabiendo identificar si estos requieren del servicio de recolección o no (López Vicario & Vilajosana Guillén, 2019).

3.11. Selección dispositivo Kit desarrollo ThinXtra

Se ha considerado el Kit ThinXtra RCZ4 para el presente proyecto debido a que es un conjunto de accesorios de desarrollo y pruebas de prototipos que cuenta con un conjunto completo de elementos que permite configurar fácilmente una solución de IoT. El kit cuenta con lo necesario para usar la red Sigfox disponible a nivel mundial (ThinXtra, 2021). El módulo cuenta con los siguientes sensores integrados:

- Acelerómetro de 3 ejes.
- Sensor digital de temperatura y presión.
- Interruptor de láminas.
- Sensor de luz.
- LED rojo y azul.
- Botón pulsador
- Cable USB.
- Soporte de batería
- Antena
- Clon de placa Arduino Uno R3
- Conectividad de 1 año a la red Sigfox.
- Cobertura en la región RCZ4 Sudamérica

CAPÍTULO IV: DESARROLLO Y PRUEBAS

En este apartado se muestra el desarrollo y pruebas realizados con el prototipo de sistema controlado con la instalación de sensores ultrasónicos para cada contenedor. Esta señal será transmitida hacia microcontroladores, los cuales por medio de la red Sigfox repetirán la información hacia la nube, en dónde se podrá recuperar los datos en el período y frecuencia que se desee. Una vez obtenida la información se analizará con el uso de un algoritmo de selección de la ruta óptima para la toma de decisiones.

4.1. Características del Sistema de recolección de desechos

Para el desarrollo e implementación del sistema de desechos se debe analizar todos los componentes que forman parte del sistema completo. Entre los componentes a nivel hardware se tiene:

- Kit de desarrollo ThinXtra (Arduino Uno y módulo Sigfox).
- 1 sensor ultrasónico.
- Batería 3.7V.
- Módulo de carga de batería.
- Convertidor DC-DC Step-up.

Entre los componentes a nivel software se tiene:

- Back- end de Sigfox.
- Base de datos.
- Página web
- Aplicación Móvil

Entre las características se tiene:

- Sigfox es la tecnología inalámbrica LPWAN seleccionada.
- Un microcontrolador está encargado de controlar la red de sensores y el TX/RX Sigfox.
- Un dispositivo de comunicación Sigfox está encargado de la comunicación del estado del contenedor de residuos sólidos.
- Cada sensor ultrasónico debe monitorear constantemente el estado del contenedor de basura
- Los estados de las plazas de estacionamiento son 3 (vacío, medio lleno, lleno)
- La base de datos está en la Internet
- El backend de Sigfox reenvía la información hacia nuestra base de datos cada vez que llega un nuevo mensaje.
- La plataforma web trae la información desde la base de datos y actualiza el estado de los contenedores de basura en tiempo real.
- Se traza una ruta de recolección con los contenedores que estén el estado “lleno”.
- La ruta de recolección se muestra en la aplicación móvil para el personal de recolección.

4.2. Implementación del Hardware del sistema

El hardware del sistema de recolección de desechos cuenta con una estación que contiene al sensor que a su vez está conectado al microcontrolador y transmisor de Sigfox.

La estación cuenta con los siguientes componentes:

- Placa Arduino UNO como microcontrolador.

- Batería 3.7V 4400 mAh.
- Tarjeta TX/RX Sigfox Wisol RC4.
- 1 sensor Ultrasónicos

El microcontrolador Arduino es la parte central de la estación, la cual está encargada de controlar al sensor ultrasónico que percibe el estado de llenado del contenedor de basura para poder identificar su estado, ya sea vacío, medio lleno y lleno; una tarjeta externa se une al Arduino UNO, esta tarjeta contiene al módulo Wisol el cual es el TX/Rx Sigfox que se encarga de la comunicación entre la red Sigfox y el microcontrolador. Se cuenta con una batería de 3.7V encargada de proveer de energía a la estación, entre la batería y el Arduino se cuenta con un convertidor DC-DC Step-up encargado de dar 5V estables para que el Arduino no sufra caída o picos de voltaje.

4.3. Implementación del Software del Sistema

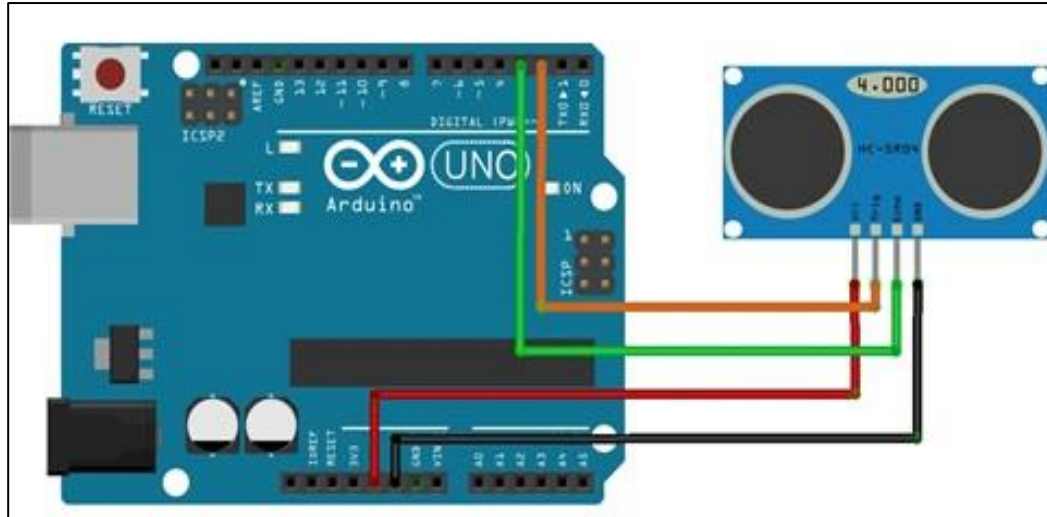
El software del sistema de recolección de desechos está distribuido en cuatro partes, las cuales se relacionan con las diferentes capas mencionadas anteriormente de una estructura IoT y hacen posible la configuración y el control para que el sistema pueda cumplir con lo requerido. Los bloques detallados anteriormente son:

- Capas de Dispositivos.
- Capa de red
- Capa de servicios
- Capa de aplicación

4.3.1. Capa de Dispositivos

Figura 14.

Diagrama de Conexión Arduino UNO y Sensor Ultrasónico



Fuente: Autor

Dentro de esta capa se encuentra el microcontrolador el cual es la parte central de la estación ya que controla y ejecuta todas las acciones de la estación. Las funciones que se debe considerar para la programación son:

- Controlar el sensor para que perciba el estado del contenedor de basura
- El sensor ultrasónico mide la distancia desde la parte superior del contenedor hacia abajo para determinar su estado.
- Se considera un estado “vacío” cuando la distancia medida es mayor a los $2/3$ de la altura del contenedor
- Se considera un estado “lleno” cuando la distancia medida es menor al $1/3$ de la altura del contenedor, si no cumple ninguna de estas alternativas se considera un estado “medio lleno”

Para la programación del microcontrolador Arduino UNO se descarga todas las librerías necesarias para ejercer la comunicación del sensor y el transmisor Sigfox, para esto se hace uso del medio de repertorio de Arduino (librerías math y LowPower) y mediante descarga desde el Internet para las librerías WISOL y Wire que son necesarias para la comunicación con la tarjeta que contiene el TX/RX Sigfox. El link de descarga de las librerías es proporcionado por el fabricante del Kit Thinextra. Cuando se cuenta con todas las librerías, se procede a declarar las variables para los pines y variables de cálculos como se muestra a continuación:

Figura 15.

Variables para pines y cálculos

```
// DECLARACION DE VARIABLES PARA PINES
const int pinecho = A2; // pin hecho de ultrasonico
const int pintrigger = 5; // pin trigger de ultrasonico
const int ali = 4; // alimentacion positivo de ultrasonico

// VARIABLES PARA CALCULOS
int altoB = 150;
unsigned int tiempo, distancia; // tiempo y distancia medidos
int espera=0;
int8_t dato=0;
```

Fuente: Autor

Seguidamente se inicializa el módulo de Sigfox y los pines utilizados por el sensor como se detalla a continuación:

Figura 16.

Comunicación entre módulo Sigfox y los pines del sensor

```
//CONFIGURACION DE SISTEMA
void setup() {
  // Iniciar comunicacion serial entre Arduino y Modulo Sigfox
  Wire.begin();
  Wire.setClock(100000);
  // Iniciar comunicacion serial
  Serial.begin(9600);
  // Prueba con el modulo WISOL
  Isigfox->initSigfox();
  Isigfox->testComms();
  // Configurar pines
  pinMode(pinecho, INPUT);
  pinMode(pintrigger, OUTPUT);
  pinMode(ali, OUTPUT);
}
```

Fuente: Autor

Una vez configurado los módulos y pines, se procede a programar la función principal con un bucle infinito, por lo que inicialmente se activan los sensores y se emite un pulso hacia la base del basurero; seguidamente se apaga el sensor y se procede a realizar los cálculos correspondientes para validar el estado de llenado del contenedor, para esto se divide el tiempo del pulso entre el tiempo que tarda en recorrer ida y vuelta un centímetro una onda sonora, luego se muestra en una consola la información resultante de los cálculos realizados.

Figura 17.

Función principal para monitorear los sensores

```
// Activar los sensores
digitalWrite(ali, HIGH);
delay(100);
//Obtener informacion de ultrasonico
// Enviar pulso de disparo en el pin "TRIGGER"
digitalWrite(pintrigger, LOW);
delayMicroseconds(4);
digitalWrite(pintrigger, HIGH);
// Espera 10us para duracion del pulso
delayMicroseconds(10);
digitalWrite(pintrigger, LOW);
// Medicion del periodo del pulso en el pin "ECHO"
tiempo = pulseIn(pinecho, HIGH);
delay(100);
// Apaga los sensores
digitalWrite(ali, LOW);
// LA VELOCIDAD DEL SONIDO ES DE 340 M/S O 29 MICROSEGUNDOS POR CENTIMETRO
// DIVIDIMOS EL TIEMPO DEL PULSO ENTRE 58, TIEMPO QUE TARDA RECORRER IDA Y VUELTA UN CENTIMETRO LA ONDA SONORA
//Calculamos la distancia en centimetros
distancia = tiempo / 58;
// Imprimimos por consola para validar la informacion del ultrasonico
Serial.println("inicio");
Serial.println(distancia);
Serial.println("end");

// Validamos la distancia y actualizamos el estado de la plaza
if(distancia > altoB*2/3){
  dato=1; // Contenedor vacio
}else if(distancia < altoB/3){
  dato=3; // Contenedor lleno
}else {
  dato=2; // Contenedor medio lleno
}
```

Fuente: Autor

A continuación, se envía los datos mediante una conexión a la red de Sigfox mediante la función `Send_Pload`. Finalmente se compila el código y se sube el programa al microcontrolador.

Figura 18.

Conexión con la red Sigfox

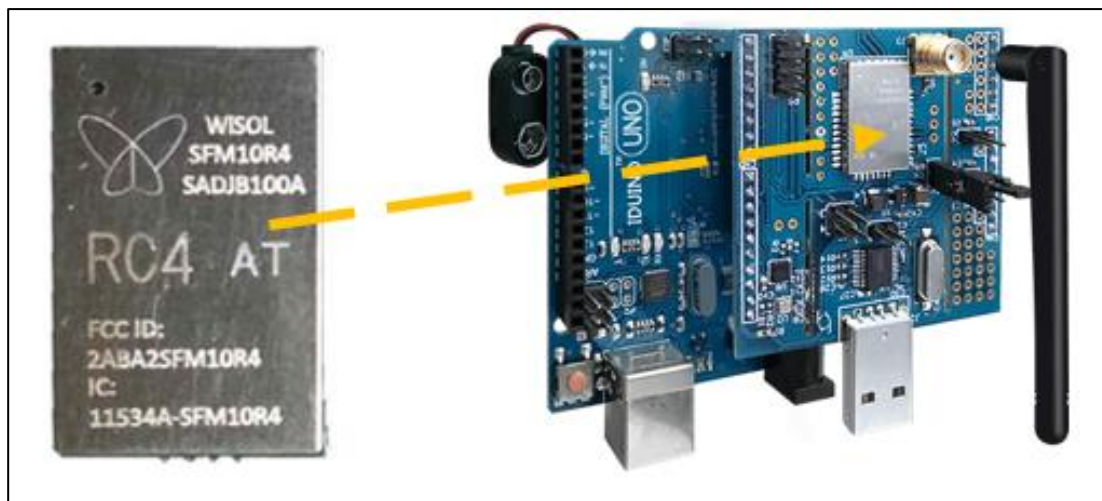
```
// Enviamos informacion a la nuba de Sigfox
Send_Pload((const char*)&dato, sizeof(dato)); // Envia datos
```

Fuente: Autor

4.3.2. Capa de red

Figura 19.

Módulo Wisol incorporado en microcontrolador



Fuente: Autor

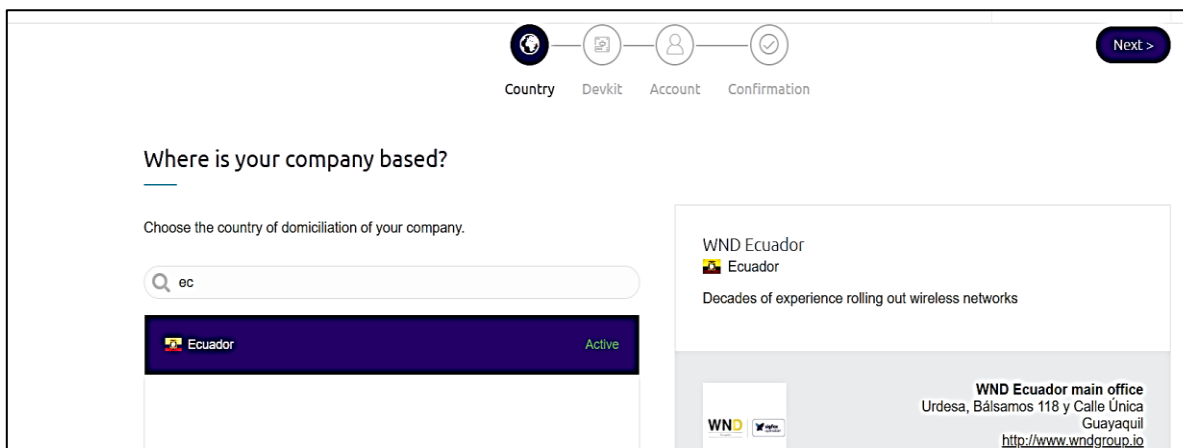
Para que la estación central pueda enviar los datos a la red de Sigfox se debe considerar que el comunicador Sigfox mediante el módulo Wisol como se indica en la figura 19, esté registrado en la red para poder configurar los callback hacia la base de datos creada.

4.3.2.1. Registro del dispositivo

Para el registro se ingresa a la página oficial de Sigfox en la opción Active my Devkit y selecciona el país, en este caso Ecuador.

Figura 20.

Registro Región Arduino en Sigfox



The screenshot shows the 'Country' selection step of the Sigfox registration process. At the top, there is a progress bar with four icons: a globe for 'Country', a DevKit for 'Devkit', a person for 'Account', and a checkmark for 'Confirmation'. A 'Next >' button is in the top right corner. The main heading is 'Where is your company based?'. Below it, the instruction says 'Choose the country of domiciliation of your company.' There is a search input field containing 'ec'. A search result for 'Ecuador' is highlighted in a dark purple box with the word 'Active' in green. To the right, there is a promotional box for 'WND Ecuador' with the text 'Decades of experience rolling out wireless networks' and contact information for the main office in Guayaquil, including the website <http://www.wndgroup.io>.

Fuente: Autor

Luego se ingresa los datos del comunicador, esto incluye el ID del dispositivo y el PAC proporcionado para cada dispositivo en la caja del kit. Además, de forma breve se describe el propósito del proyecto.

Figura 21.

Datos para registro en Sigfox



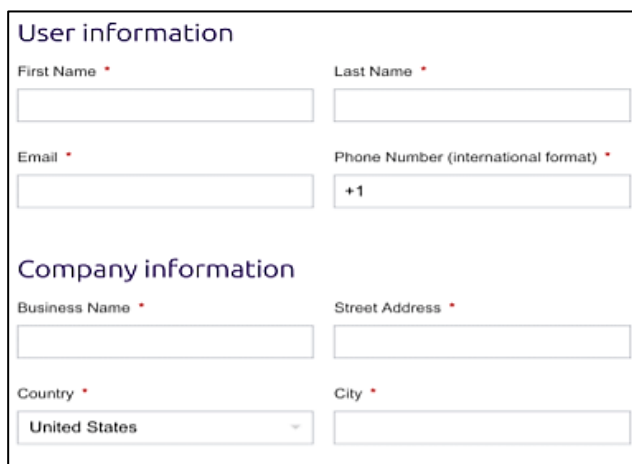
The screenshot shows the 'Provide your DevKit's details for identification' step. It features two input fields. The first is for 'Device ID' with a red asterisk, showing an example 'ex: 123AB' and a note 'Up to 8 numbers and letters (from A to F)'. The second is for 'PAC' with a red asterisk, showing an example 'ex: 1234567890ABCDEF' and a note 'Exactly 16 numbers and letters (from A to F)'. Below these is a section titled 'Tell us about your project' with a dropdown menu for 'Purpose of your project' showing 'Select...' and a close button.

Fuente: Autor

A continuación, se ingresa la información de la cuenta, datos de contacto del usuario y el grupo de pertenencia del dispositivo. El email proporcionado se utiliza para el ingreso al backend de Sigfox en donde se observa el registro del dispositivo. Para finalizar aparece un mensaje de confirmación.

Figura 22.

Datos de Usuario para Registro en Sigfox



The image shows a registration form titled "User information" and "Company information". The "User information" section includes fields for "First Name", "Last Name", "Email", and "Phone Number (international format)". The "Company information" section includes fields for "Business Name", "Street Address", "Country" (with a dropdown menu showing "United States"), and "City".

User information	
First Name *	Last Name *
<input type="text"/>	<input type="text"/>
Email *	Phone Number (international format) *
<input type="text"/>	+1 <input type="text"/>
Company information	
Business Name *	Street Address *
<input type="text"/>	<input type="text"/>
Country *	City *
United States	<input type="text"/>

Fuente: Autor

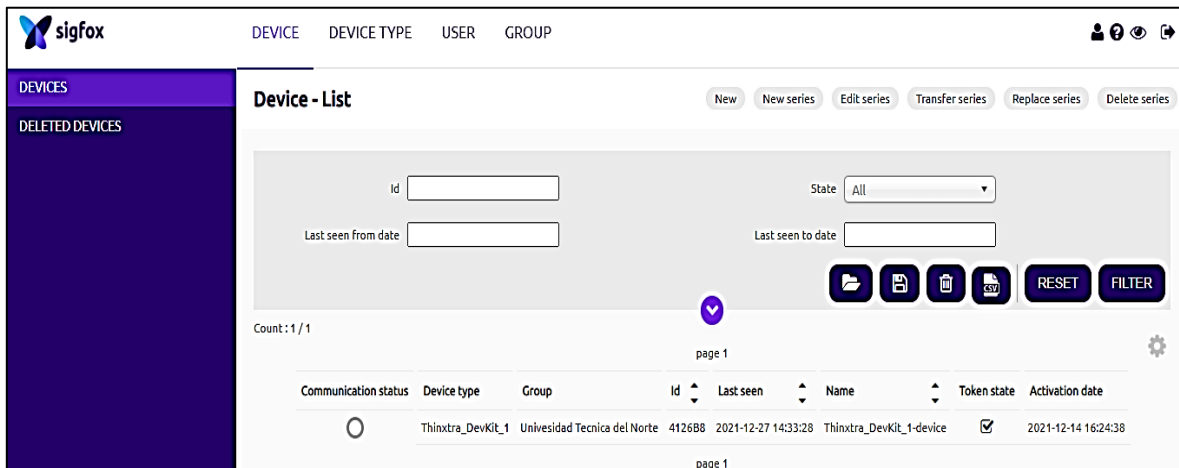
4.3.2.2. Backend Sigfox

Cuando se registra un dispositivo, se puede observar las opciones de configuración en la parte superior de panel. Dentro de estas configuraciones está la lista de dispositivos registrados, el tipo de dispositivos registrados, los usuarios registrados dentro de la cuenta, y los grupos creados dentro de la cuenta. Dentro de la opción de Dispositivo se puede encontrar la siguiente información:

- Tipo de dispositivo
- Estado de comunicación
- ID del dispositivo
- Nombre del dispositivo

Figura 23.

Pantalla Principal Backend



Fuente: Autor

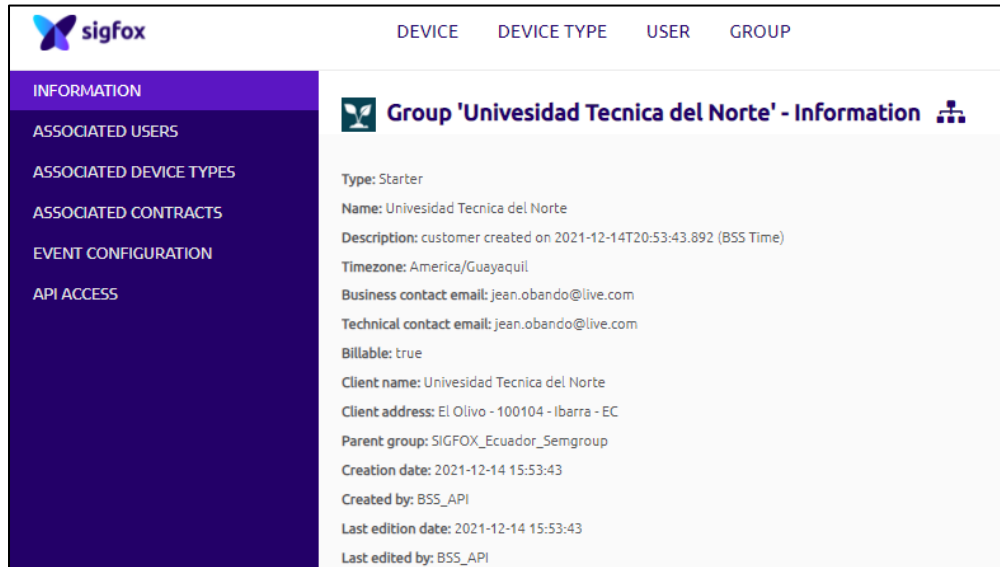
Se pulsa sobre el grupo registrado, aparece una nueva página en donde se detallan algunos datos como:

- Información: se despliega información general como: el nombre, descripción, la localización, nombre del cliente, datos de fecha y algunos códigos.
- Usuarios asociados: se despliega información de los usuarios que han registrado los dispositivos.
- Tipos de dispositivos asociados: los tipos de dispositivos que se han registrado, aquí hay varios entre los dispositivos desarrollados para prototipos de diferentes organizaciones, en nuestro caso Thinextra.
- Contrato asociado: el contrato que se tiene para cada dispositivo, es decir el tipo de suscripción que tiene y la fecha de inicio y de fin del contrato.
- Configuración de eventos: se pueden configurar eventos que se ejecuten cuando llegan nuevos mensajes a los dispositivos, es decir se puede reenviar la información al email o a un servidor externo.

- Acceso a API: se puede asociar a una API externa de algún proveedor IoT o de alguna plataforma compatible con Sigfox.

Figura 24.

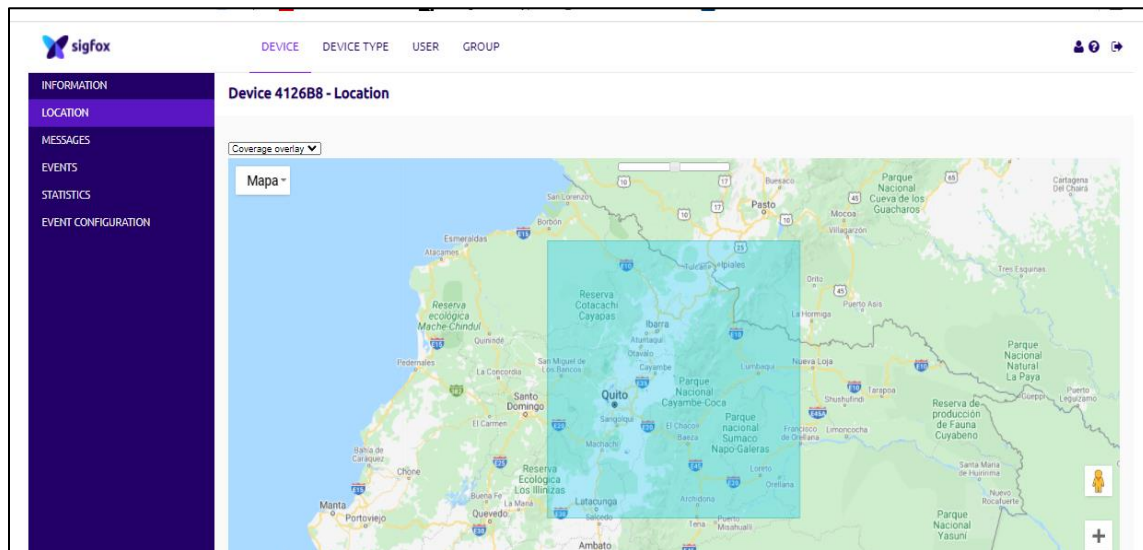
Pantalla del Grupo en Backend



Fuente: Autor

Figura 25.

Backend Sigfox, zona de operación del dispositivo



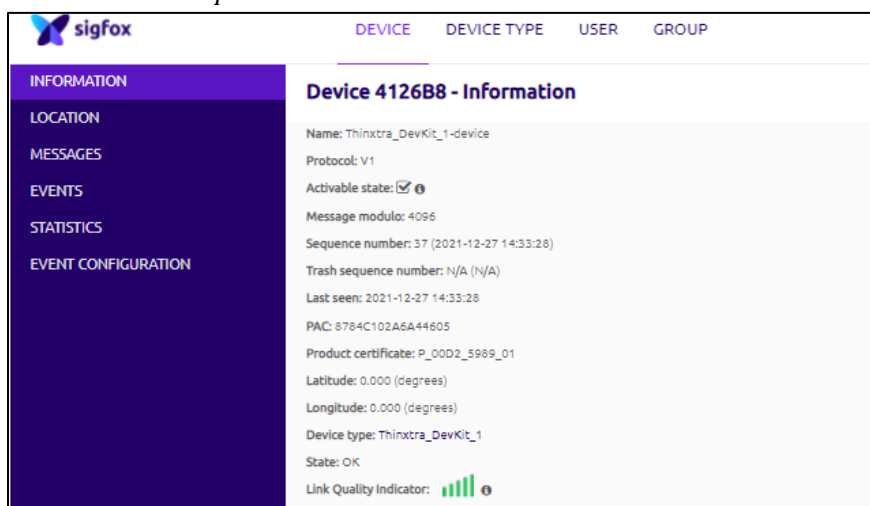
Fuente: Autor

De la misma manera, si se pulsa sobre el ID del dispositivo, se muestra una ventana como se indica en la figura 21, se detalla información diferente a la mencionada anteriormente y se muestra un panel lateral derecho con varias opciones como:

- **Información:** se indica información general del dispositivo, estado de activación, ubicación en longitud y latitud.
- **Localización:** se muestra un mapa para identificar la ubicación del dispositivo
- **Mensajes:** se despliega los mensajes recibidos los cuales son enviados por el TX/RX Sigfox ordenados cronológicamente.
- **Eventos:** son mensajes de error que envía la red en caso de detectar uno.
- **Estadísticas:** se muestra estadísticas relacionadas a el número de mensajes recibidos por unidad de tiempo, calidad de bytes por tiempo, SNR promedio (dB) por unidad de tiempo y RSSI (Dbm) promedio por unidad de tiempo.
- **Configuración de eventos:** se configura devoluciones de llamadas, es decir se puede reenviar la información de un evento al email o aun servidor externo.

Figura 26.

Pantalla de ID del Dispositivo



Fuente: Autor

Figura 27.

Backend Sigfox, zona de recepción de mensajes

Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2021-12-27 14:33:28	37	00a835e30706003ff9900	40	+	📍
2021-12-27 14:23:29	36	790a7f9e9f01ef00bffe3ff	40	+	📍
2021-12-27 14:13:30	35	f50a815eb201ef00bffe5ff	40	+	📍
2021-12-27 14:03:29	34	030b835eb201f000f5ffe8ff	40	+	📍
2021-12-27 13:53:29	33	000b855e1002f100f3fbebff	40	+	📍
2021-12-27 13:43:30	32	e30a875e0502f100f6ffe8ff	40	+	📍

Fuente: Autor

Figura 28.

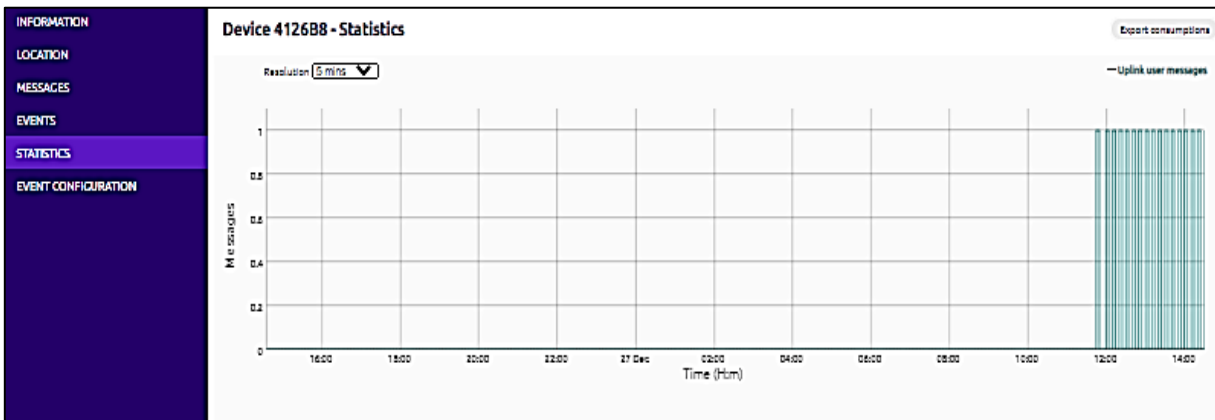
Backend Sigfox, zona de eventos

Time	Type	Severity	Description	Status
2021-12-14 16:24:38	Receive first payable message	INFO	Device #412688 has sent its first payable message	🔔
2021-12-14 15:53:48	On contract	INFO	Device #412688 has been associated to a contract	🔔

Fuente: Autor

Figura 29.

Backend Sigfox, zona de estadística



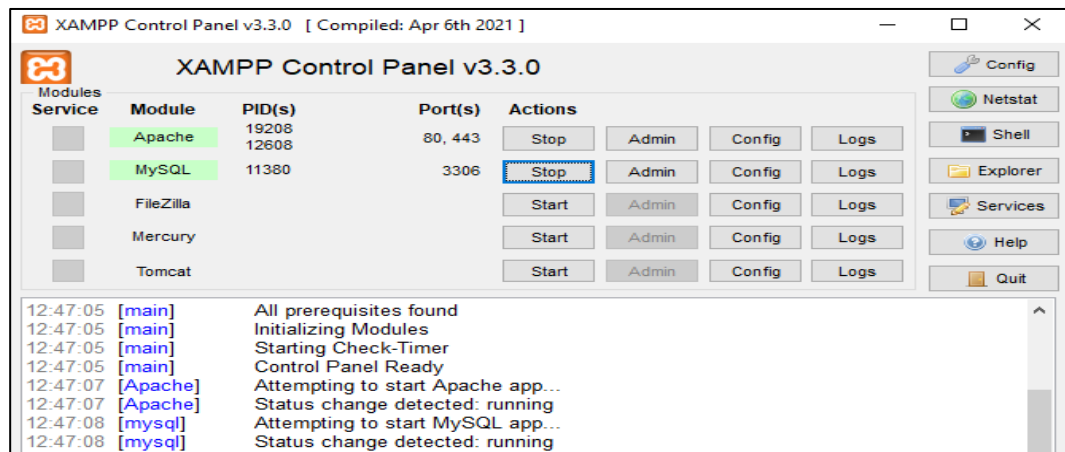
Fuente: Autor

4.3.3. Capa de servicios

En esta sección se encuentra la base de datos la cual se encarga de almacenar toda la información de los estados de llenado de los contenedores de basura para que la plataforma web pueda acceder a estos datos y mostrarlos al usuario, por lo que se requiere que la base de datos esté subida al internet utilizando un host gratuito.

Figura 30.

Servidor de base de Datos



Fuente: Autor

4.3.3.1. Tablas de datos

La base de datos está clasificada por 4 apartados:

- **Contenedores:** esta tabla es la encargada de guardar el estado actual de llenado de los contenedores de basura identificándolo en una escala del 1 al 3. Además, esta tabla muestra la ubicación de cada contenedor mostrando la latitud y la longitud.

Figura 31.

Tabla de Contenedores

id_contenedores	nombre_contenedores	latitud	longitud	estado	inicio	fin
1	CONTENEDOR 1	0.366382	-78.112900	2	0	0
2	CONTENEDOR 2	0.365749	-78.112879	1	0	0
3	CONTENEDOR 3	0.36114095927930573	-78.11313697450676	1	0	0
4	CONTENEDOR 4	0.36193487734194213	-78.11276117488964	1	0	0
27	CONTENEDOR 5	0.3653036640220642	-78.11370624632794	2	0	0
30	CONTENEDOR 11	0.36441318868434264	-78.11379201216155	2	0	0
31	CONTENEDOR 6	0.3640591442484949	-78.11300811464021	1	0	0
32	CONTENEDOR 7	0.3633081408537183	-78.11298663799582	3	0	0
33	CONTENEDOR 9	0.362760981198219	-78.11301885296243	1	0	0
34	CONTENEDOR 10	0.3633188694740786	-78.11214904886343	3	0	0
41	PUNTO DE INICIO	0.35905960619169736	-78.11358798403955	3	1	0
42	PUNTO FINAL	0.3675137590568327	-78.11177320758607	3	0	1

Fuente: Autor

- **Estado del contenedor:** se muestra y se define la escala de identificación de llenado de los contenedores de basura en donde 1 es vacío, 2 es medio lleno y 3 es lleno, los cuales son medidos por el sensor ultrasónico.

Figura 32.

Tabla de estado

id_estado	nombre_estado	imagen
1	VACIO	(Null)
2	MEDIO	(Null)
3	LLENO	(Null)

Fuente: Autor

- **Tipo de usuario:** este apartado muestra las diferentes cuentas para el acceso a las funciones de la plataforma web donde se muestra los resultados obtenidos del sensor ultrasónico y los estados de cada contenedor de basura. Cada cuenta puede tener habilitada o deshabilitada las distintas funciones de configuración de la plataforma web o aplicación móvil.

Figura 33.

Tabla de Tipo de Usuario

id_tipo	detalle_tipo
1	ADMINISTRADOR
2	USUARIO

Fuente: Autor

- **Usuario:** se muestra la información de los usuarios que han interactuado o que pueden interactuar con la plataforma web o la aplicación móvil

Figura 34.

Tabla de Información de Usuarios

id_usuario	nombre	direccion	ci_ruc	nick	pass	id_tipo
1	ADMINISTRADOR	DIRECCION DE ADMINISTRACION	111111111	ADMIN	ADMIN	1
2	PACO	CALDERON	1722221450	PACO	PEPE	2

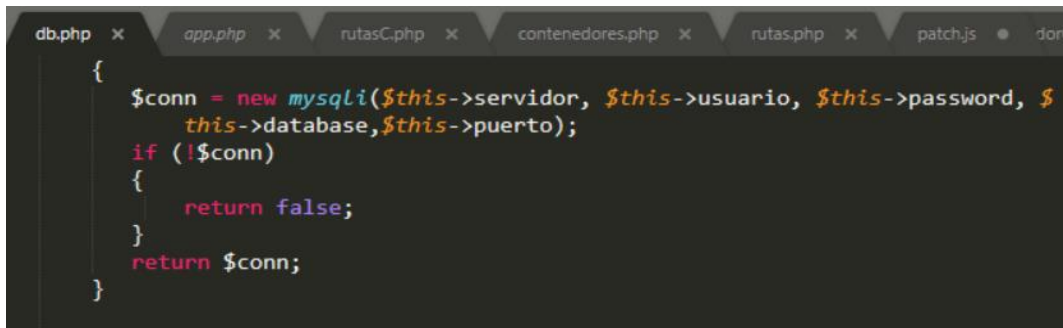
Fuente: Autor

4.3.3.2. Web Services

Cuando se tiene creada la base de datos con sus respectivas tablas y variables, se procede a configurar el web services mediante el lenguaje PHP, esto permitirá que los datos del backend de Sigfox se transmitan hacia la plataforma web diseñada para el usuario a su vez interactúe con ellos. Inicialmente se procede a realizar una conexión con la base de datos mediante:

Figura 35.

Conexión con la base de datos



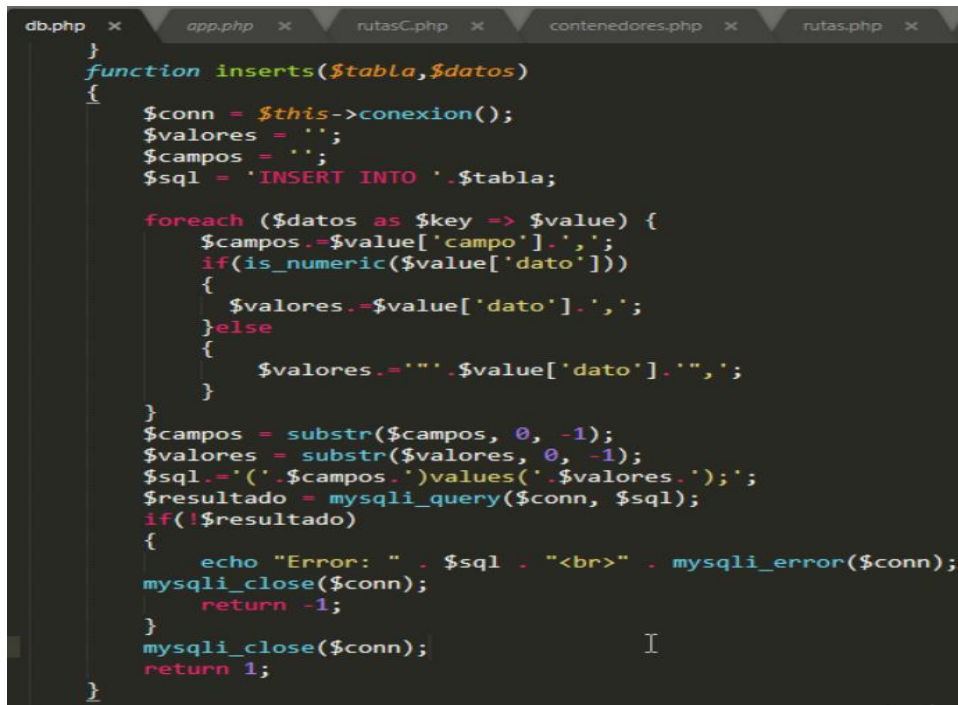
```
db.php x app.php x rutasC.php x contenedores.php x rutas.php x patch.js x dord
{
    $conn = new mysqli($this->servidor, $this->usuario, $this->password, $
        this->database,$this->puerto);
    if (!$conn)
    {
        return false;
    }
    return $conn;
}
```

Fuente: Autor

También se utiliza la función para insertar datos dentro de la base de datos, así como también la edición de los mismos, esto permite que el usuario interactúe con la base de datos cuando se lo requiera.

Figura 36.

Función para insertar datos en la Base de Datos



```
db.php x app.php x rutasC.php x contenedores.php x rutas.php x
}
function inserts($tabla,$datos)
{
    $conn = $this->conexion();
    $valores = '';
    $campos = '';
    $sql = 'INSERT INTO '.$tabla;

    foreach ($datos as $key => $value) {
        $campos.=$value['campo'].'.';
        if(is_numeric($value['dato']))
        {
            $valores.=$value['dato'].'.';
        }else
        {
            $valores.=''.$value['dato'].'.';
        }
    }
    $campos = substr($campos, 0, -1);
    $valores = substr($valores, 0, -1);
    $sql.='('.$campos.')values('.$valores.')';
    $resultado = mysqli_query($conn, $sql);
    if(!$resultado)
    {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
        mysqli_close($conn);
        return -1;
    }
    mysqli_close($conn);
    return 1;
}
```

Fuente: Autor

4.3.4. Capa de aplicación

4.3.4.1. Página Web

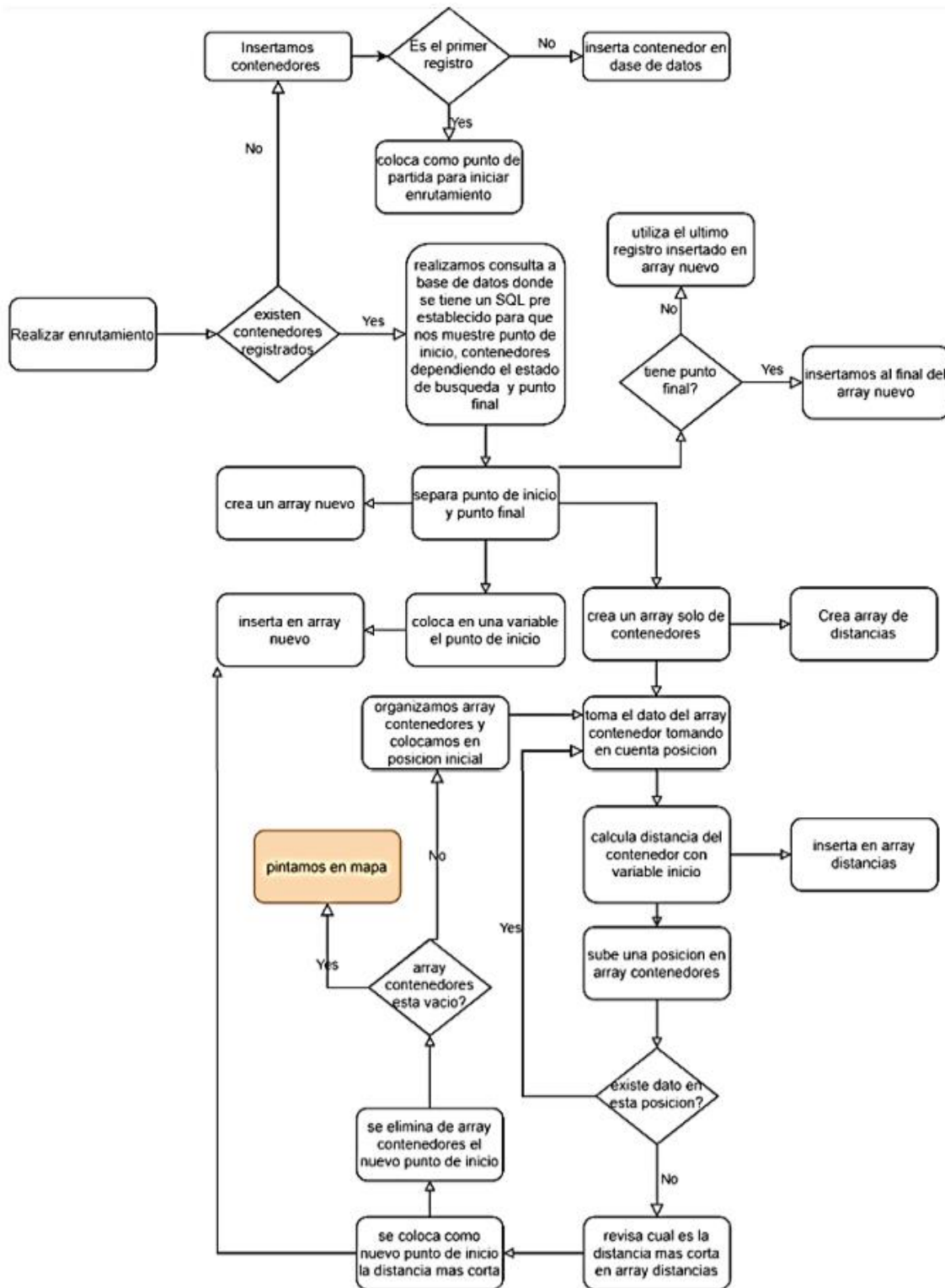
Mediante el uso de un local host, el cual es un servidor web almacenado en una red local en el cual no es necesario contratar un dominio de modo que el disco duro hace las funciones de hosting, se diseñó una plataforma para que el usuario pueda interactuar con los datos enviados por el sensor ubicado en el contenedor de basura y además poder identificar la ruta de recolección óptima de desechos sólidos. Las características principales del diseño de la plataforma web son las siguientes:

- Se tiene un login de acceso para los usuarios en donde se solicitará las credenciales de ingreso
- Se muestra una ventana en donde se encuentra un menú lateral en la parte izquierda con información acerca de los contenedores y rutas de recolección.
- Se muestra también un mapa interactivo en donde se muestra la ruta de recolección óptima con los contenedores en estado lleno.
- Para cada estado se diseñó un icono.
- La plataforma lleva como nombre Reco Desecho.

Para la creación de la interfaz donde se muestra el mapa y la ruta de recolección óptima, se realizó mediante la codificación del algoritmo que se muestra en la figura 30:

Figura 37.

Algoritmo de programación del sistema de Recolección de Desechos



Fuente: Autor

Figura 38.

Algoritmo de Dijkstra

```
db.php x app.php rutasC.php x contenedores.php x rutas.php x patch.js dorC
function buscar_puntos_app($parametros)
{
    // print_r($parametros);die();
    ini_set('max_execution_time', '300');
    $estado = $parametros['estado'];
    $puntos = $this->modelo->rutas_optima_app($estado);
    $inicio = '';
    $fin = '';
    $puntos_cont = array();
    $ruta_enviar = array();
    foreach ($puntos as $key => $value) {
        array_push($puntos_cont, array('lo' => $value['lo'], 'la' => $value['la'],
            'nombre' => $value['nombre'], 'estado' => $value['estado']));
    }
    $inicio = array('lo' => $parametros['lon'], 'la' => $parametros['lat'],
        'nombre' => 'mi posicion', 'estado' => '');
    array_push($ruta_enviar, $inicio);
    while (count($puntos_cont) > 0) {
        $posi = array();
        foreach ($puntos_cont as $key => $value) {
            $ini = $inicio['lo'].'.'.$inicio['la'];
            $nom = $inicio['nombre'];
            $pun = $value['lo'].'.'.$value['la'];
            $dis = $this->distancia($ini, $pun);
            array_push($posi, ['distancia' => $dis, 'pos' => $key]);
        }
        sort($posi); //ordena de manera acendente
        $pun = $puntos_cont[$posi[0]]['pos'];
        array_push($ruta_enviar, $pun);
        unset($puntos_cont[$posi[0]]['pos']);
        $inicio = $pun;
    }
    // print_r($ruta_enviar);die();
}
```

Fuente: Autor

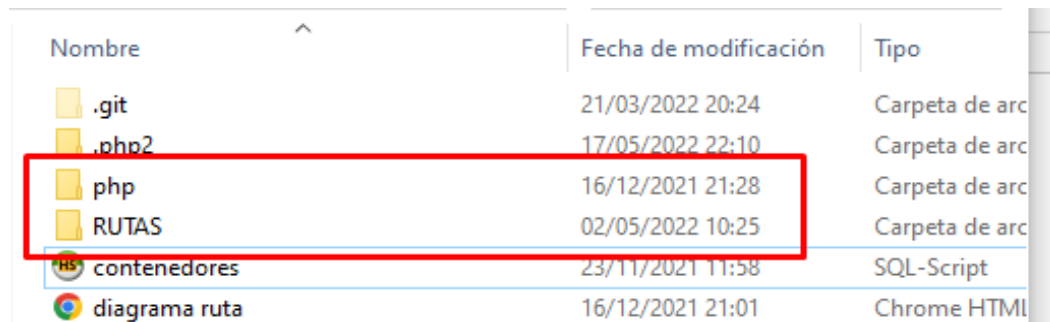
4.3.4.2. Aplicación Móvil

Para la aplicación móvil de este proyecto se ha realizado 2 archivos los cuales son:

- **PHP:** código fuente de aplicación web
- **Rutas:** código de aplicativo web realizado por **CORDOVA APACHE**

Figura 39.

Archivos para aplicación móvil



Nombre	Fecha de modificación	Tipo
.git	21/03/2022 20:24	Carpeta de arc
.php2	17/05/2022 22:10	Carpeta de arc
php	16/12/2021 21:28	Carpeta de arc
RUTAS	02/05/2022 10:25	Carpeta de arc
contenedores	23/11/2021 11:58	SQL-Script
diagrama ruta	16/12/2021 21:01	Chrome HTML

Fuente: Autor

Para la creación de la aplicación móvil, se tiene en cuenta que se va a comunicar con los archivos del aplicativo web, en este caso la carpeta PHP debe estar completamente configurada con respecto a la base de datos. En este puede estar las credenciales locales o credenciales de una base de datos externa.

Figura 40.

Comunicación con la base de datos

```
}  
  
function default_conexion()  
{  
    // $this->usuario = "root";  
    // $this->password = '';  
    // $this->servidor = "localhost";  
    // $this->database = "proyecto";  
    // $this->puerto = '3306';  
  
    $this->usuario = "diskcover";  
    $this->password = "XXXXXXXXXX";  
    $this->servidor = "mysql.diskcoversystem.com";  
    $this->database = "base_prueba";  
    $this->puerto = '13306';  
}
```

Fuente: Autor

Por la cantidad mínima de información, la aplicación se construyó con el fin de ser únicamente informativa y se comenzó a agregar campos en la base de datos. En la tabla de contenedores se agregaron campos para tener más detalles sobre los contenedores.

Figura 41.

Información agregada en los contenedores para mostrar en la aplicación móvil.

latitud	longitud	estado	inicio	fin	codigo	largo	alto	ancho	descripcion	foto
-0.0765376	-78.4336947	3 1	0	1		(Null)	(Null)	(Null)	(Null)	contenedor1.jpg
-0.0763982	-78.4332763	3 1	0	1		6	5	7	s	contenedor1.jpg
-0.0771843	-78.4343003	2 0	0	1		(Null)	(Null)	(Null)	(Null)	contenedor1.jpg
-0.0762748	-78.4324636	1 0	0	1		(Null)	(Null)	(Null)	(Null)	contenedor1.jpg
-0.0765001	-78.4324475	2 0	0	1		(Null)	(Null)	(Null)	(Null)	contenedor1.jpg
-0.0754463	-78.4333033	2 0	0	1		(Null)	(Null)	(Null)	(Null)	contenedor1.jpg
-0.0766557	-78.432603	3 0	0		br9					contenedor1.jpg
-0.0753164074475381	-78.43473315238954	1 0	0		BR1	2	3	1	contenedor nuevo	contenedor2.jpg

Fuente: Autor

De la misma forma, se agregó más campos informativos en usuario.

Figura 42.

Información agregada en los usuarios para mostrar en la aplicación móvil.

id_usuario	nombre	direccion	ci_ruc	nick	pass	id_tipo	foto	telefono	email
1	ADMINISTRADOR	DIRECCION DE ADMINISTRA	111111111	ADMIN	ADMIN	1	usuario1.jpg	99999999	javierexample.com
2	PACO	CALDERON	1722221450	PACO	PEPE	2	ss	09999999	(Null)

Fuente: Autor

Se procede al desarrollo de la aplicación móvil tomando en cuenta el framework de **APACHE CORDOVA**, el cual permite realizar aplicaciones híbridas tomando en cuenta el entorno de cada dispositivo y utilizando los componentes ya sea de Android o de ios.

Figura 43.

Apache Cordova página oficial



Fuente: Apache Cordova, (2022)

Para el correcto funcionamiento de la aplicación móvil, se coloca una ruta en donde se encuentra almacenada la carpeta php, esta ruta se coloca dentro del archivo **patch.js** el cual se encuentra en: **C:\xampp\htdocs\rutas\RUTAS\www\js\patch.js**. Esta ruta debería ser remplazada por el servidor en donde se encuentran los archivos.

Figura 44.

Ruta de archivos para la comunicación

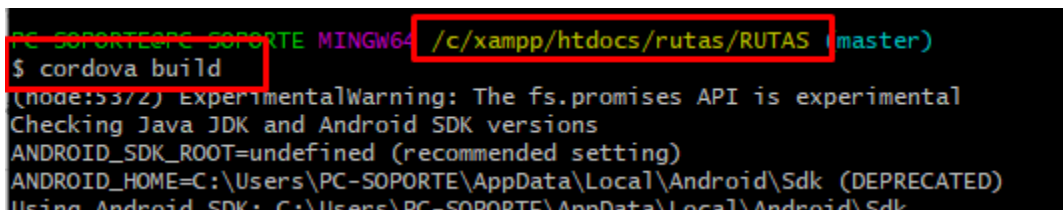
```
js\patch.js (rutas) - Sublime Text (UNREGISTERED)
File Tools Project Preferences Help
index.html patch.js db.php config.xml
1 var url_link = 'https://erp.diskcoversystem.com/~diskcover/php2/';
2 var url_img = url_link+'img/';
3 var url_img_u = url_link+'img/usuario/';
4 var url_img_c = url_link+'img/contenedores/';
5
6 // en caso de ser un servidor externo
7 //var server = 'http://corsinf.com:8087/';
8 //var carpeta = 'pruebas/rutas/php/';
9 //var url_link = server+carpeta;
10
11 //nuevo https
12 // var server = 'https://erp.diskcoversystem.com/';
13 // var carpeta = '~diskcover/php2/';
14 //var url_link = server+carpeta;
```

Fuente: Autor

Se debe comprobar que la ruta este bien escrita y con un comando netamente del framework apache Cordova se construye la aplicación móvil, se accede a la carpeta del archivo y se coloca el comando Cordova build.

Figura 45.

Comando para construir la aplicación móvil



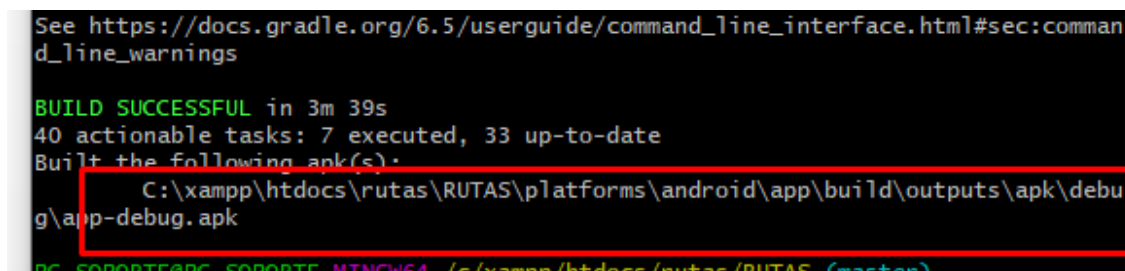
```
PC-SOPORTE@PC-SOPORTE MINGW64 /c:/xampp/htdocs/rutas/RUTAS (master)
$ cordova build
(node:5372) ExperimentalWarning: The fs.promises API is experimental
Checking Java JDK and Android SDK versions
ANDROID_SDK_ROOT=undefined (recommended setting)
ANDROID_HOME=C:\Users\PC-SOPORTE\AppData\Local\Android\Sdk (DEPRECATED)
Using Android SDK: C:\Users\PC-SOPORTE\AppData\Local\Android\Sdk
```

Fuente: Autor

Una vez terminado el proceso, se muestra una ruta en donde se encuentra la aplicación móvil.

Figura 46.

Ruta de la aplicación móvil



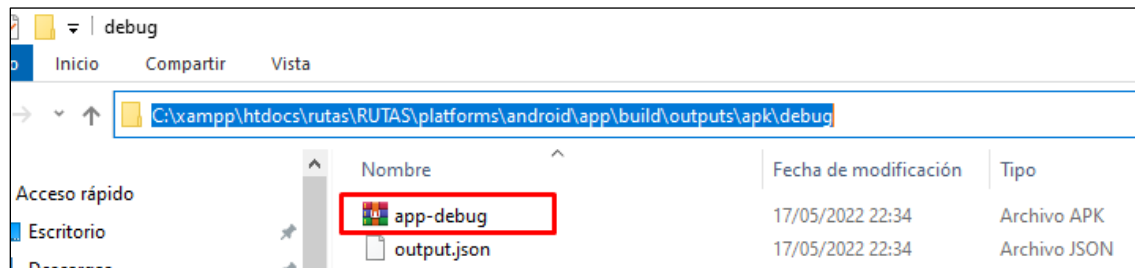
```
See https://docs.gradle.org/6.5/userguide/command_line_interface.html#sec:command_line_warnings
BUILD SUCCESSFUL in 3m 39s
40 actionable tasks: 7 executed, 33 up-to-date
Built the following apk(s):
  C:\xampp\htdocs\rutas\RUTAS\platforms\android\app\build\outputs\apk\debug\app-debug.apk
PC-SOPORTE@PC-SOPORTE MINGW64 /c:/xampp/htdocs/rutas/RUTAS (master)
```

Fuente: Autor

Se busca la ruta en el explorador de archivos, de tal manera que se localiza la carpeta comprimida en donde se aloja la aplicación móvil creada.

Figura 47.

Carpeta comprimida con la aplicación móvil.



Fuente: Autor

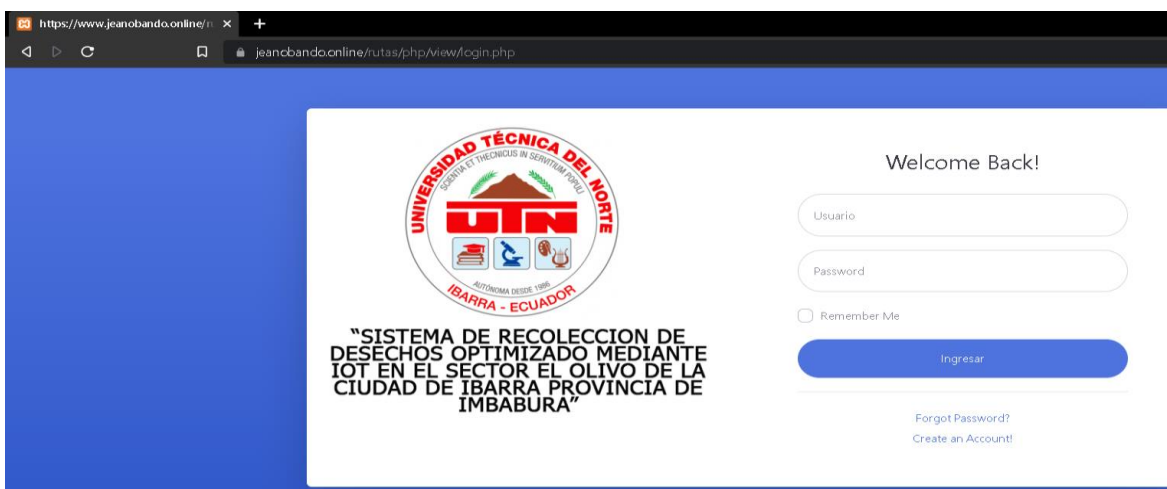
4.4. Funcionamiento del sistema y pruebas de desarrollo

4.4.1. Aplicación Web

Para ejecutar la aplicación del sistema de recolección de RSU, se debe seguir el manual de usuario del Anexo 1. Inicialmente se ingresa a la dirección electrónica <https://www.jeanobando.online/rutas/php/view/login.php>, el cual muestra una pestaña login para inicial el acceso al sistema.

Figura 48.

Pantalla de Inicio Aplicación Web

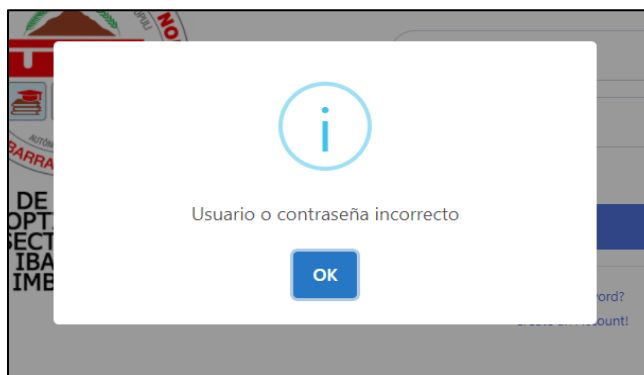


Fuente: Autor

En la parte derecha se coloca las credenciales del usuario y se hace clic en ingresar, en el caso de que las credenciales estén incorrectas aparece un mensaje de alerta.

Figura 49.

Error de ingreso de credenciales



Fuente: Autor

Si las credenciales de ingreso son correctas, el sistema muestra la parte home del sistema, en esta vista se puede observar al lado derecho un menú. En la parte derecha superior se encuentra el usuario, al pulsar este botón se puede seleccionar la opción de cerrar sesión.

Figura 50.

Menú Principal Aplicación Web



Fuente: Autor

Dentro del menú, dependiendo el tipo de usuario se muestra diferentes ítems. En el caso de que el usuario sea Administrador se muestra lo siguiente:

En contenedores

- Nuevo contenedor
- Ubicación
- Rutas

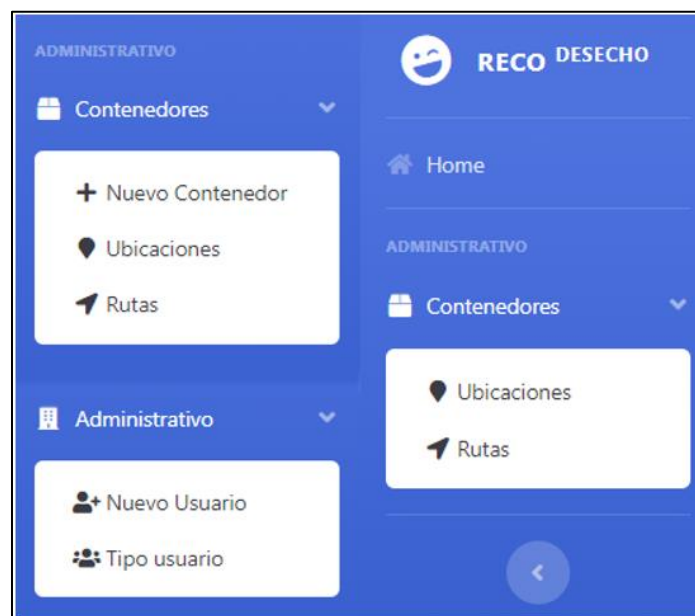
En Administrativo

- Nuevo usuario
- Tipo de usuario

En el caso de acceder con un usuario que no es administrador solo aparece el apartado de contenedores con las ubicaciones y rutas

Figura 51.

Menú de Administrador y Usuario

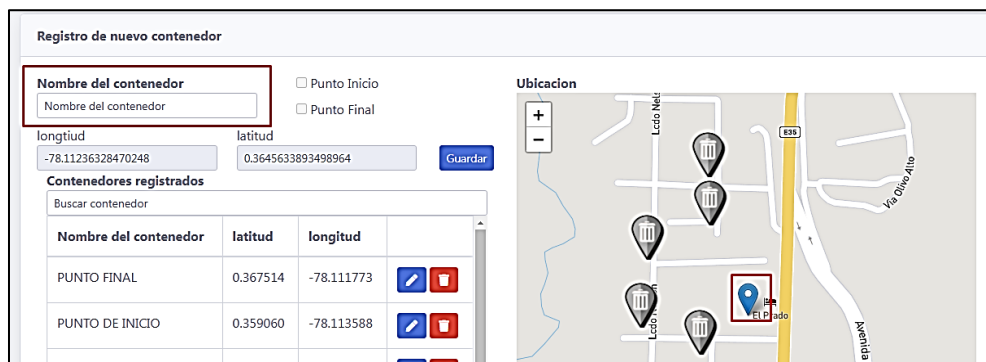


Fuente: Autor

Dentro del menú principal se encuentra la opción Contenedores, y dentro de ella la opción Nuevo Contenedor; esta opción permite ingresar nuevos contenedores, editar contenedores existentes y a su vez eliminar. Cuando se pulsa esta opción se muestra una pantalla en donde se puede observar en el lado izquierdo la lista de contenedores registrados y un mapa para ubicar geográficamente su posición. Para agregar un contenedor nuevo basta con escribir el nombre del nuevo contenedor y colocar la posición deseada en el mapa, finalmente se da clic en guardar.

Figura 52.

Pantalla de Registro de Contenedores

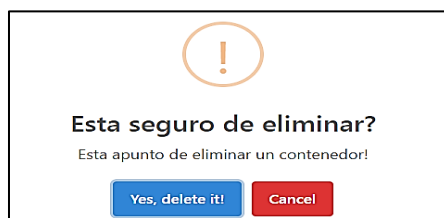


Fuente: Autor

En el caso de eliminar un contenedor se pulsa el botón rojo de la parte derecha de la tabla donde se muestra la lista de contenedores. Se aparecerá una confirmación, en caso de ser positivo se eliminará el contenedor registrado.

Figura 53.

Confirmación de Eliminación de Contenedores

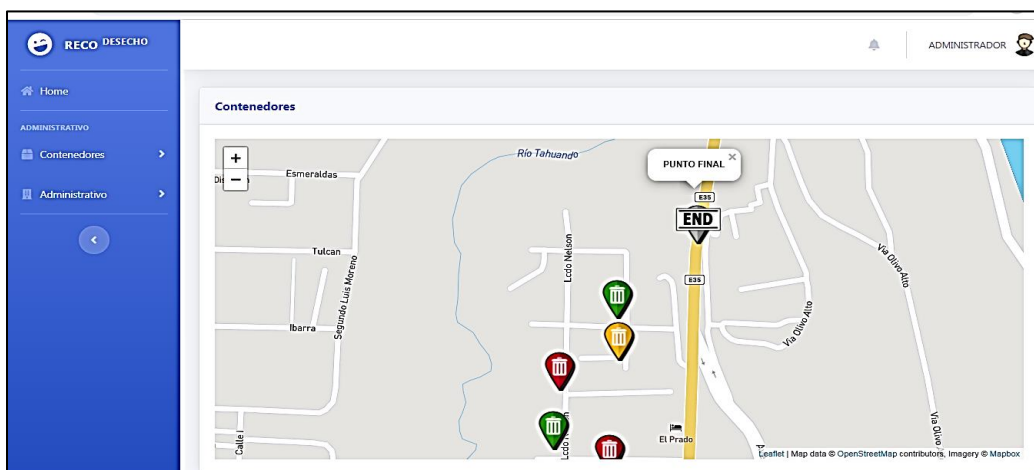


Fuente: Autor

Dentro del menú del administrador también se encuentra la opción Ubicaciones, este apartado sirve de manera informativa para saber dónde están ubicados los contenedores registrados y su estado. En el caso de necesitar el nombre del contenedor se da clic en el indicador o marker y aparece su nombre

Figura 54.

Pantalla de Ubicación de Contenedores



Fuente: Autor

Figura 55.

Simbología de Mapa de Ubicación de Contenedores



Indica el punto fin en el caso de registrarse un punto final



Indica el punto de partida para una ruta



Indica contenedores de estado lleno



Indica contenedores de estado medio lleno



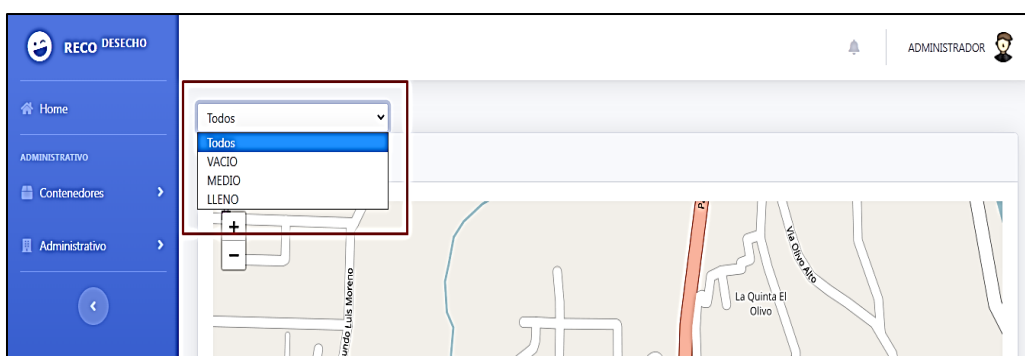
Indica contenedores de estado vacío

Fuente: Autor

El menú del usuario administrativo también consta con la opción rutas, al entrar en este apartado se muestra un mapa vacío, por lo que para visualizar la ruta se selecciona en la parte superior izquierda del mapa, la opción del tipo de ruta a trazar. Se calcula la ruta más óptima tomando en cuenta el punto de inicio y final. La ruta se traza dependiendo del estado de los contenedores.

Figura 56.

Tipo de Rutas

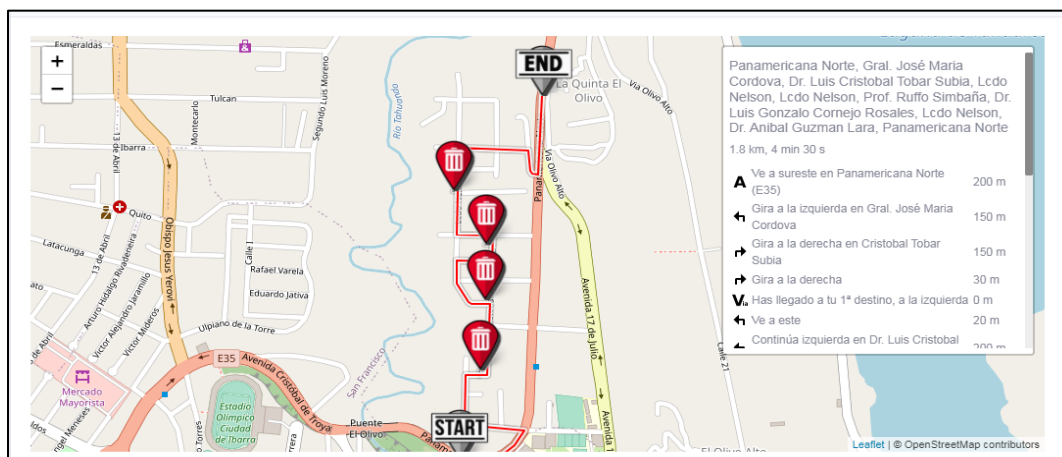


Fuente: Autor

Se calculará la ruta óptima tomando en cuenta el punto de inicio y final. La ruta se traza dependiendo del estado de los contenedores.

Figura 57.

Trazado de Ruta de Recolección



Fuente: Autor

La opción nuevo usuario permite ver la lista de todos los usuarios registrados, así como también la opción de agregar y eliminar los mismos. Para registrar un nuevo usuario se ingresa todos los datos requeridos en la parte izquierda, clic en guardar para almacenar los datos.

Figura 58.

Pantalla de registro de Usuarios

Nombre	CI / RUC	Nick	Password	Tipo usuario	Dirección	
JIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	[Edit/Delete]
ACO	1722221450	PACO	PEPE	USUARIO	CALDERON	[Edit/Delete]
ADMINISTRADOR	111111111	ADMIN	ADMIN	ADMINISTRADOR	DIRECCION DE ADMINISTRADOR	[Edit/Delete]

Fuente: Autor

4.4.2. Aplicación Móvil

- **Inicio:** En la pantalla de nuestro teléfono podremos observar la aplicación con el nombre de IOT recolector y con el logo de la universidad.

Figura 59.

Aplicación móvil



Fuente: Autor

- **Pantalla de login:** Al iniciar la aplicación móvil, se muestra una pantalla en donde se solicita iniciar sesión con un usuario y una contraseña.

Figura 60.

Pantalla de Login

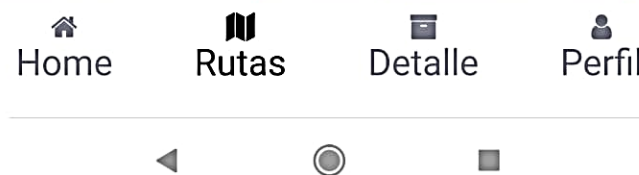


Fuente: Autor

Una vez iniciada la sesión, se muestra la pantalla de home, en la parte inferior se muestra un menú de navegación de las funcionalidades de la aplicación móvil, las cuales muestran los datos informativos dependiendo de la opción seleccionada.

Figura 61.

Barra de navegación

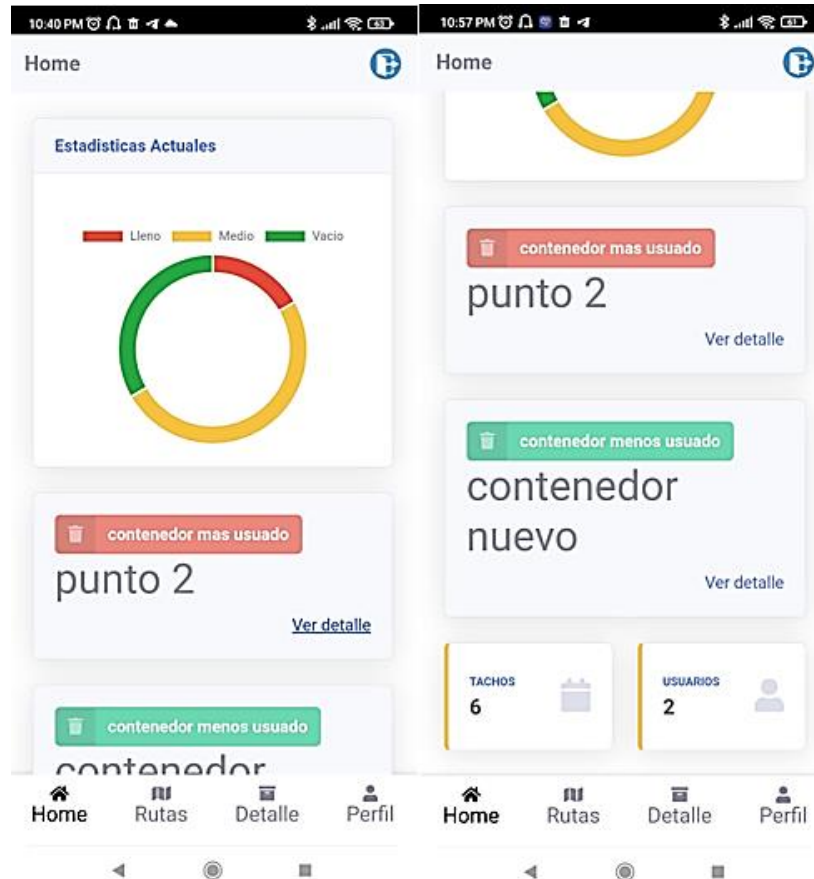


Fuente: Autor

- **Home:** En home encontraremos un cuadro estadístico de la cantidad de contenedores llenos, media y vacía. También el contenedor que más se llena y el que menos se llena, Número de contenedores y el número de usuario.

Figura 62.

Información de Home

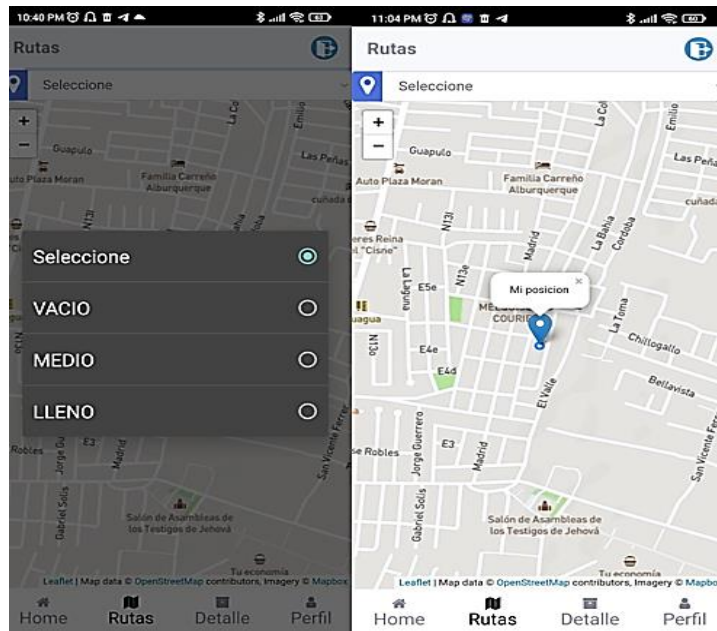


Fuente: Autor

- **Rutas:** Para la opción de Rutas este nos mostrara un mapa en donde nos señalara la ruta a seguir para tener encuentro con los contenedores de basura. Como pantalla inicial nos mostrara nuestra ubicación y en la parte superior una lista de opciones para generar las rutas y al dar click nos mostrara las opciones que se encuentran: vacío, medio y lleno

Figura 63.

Información de Rutas

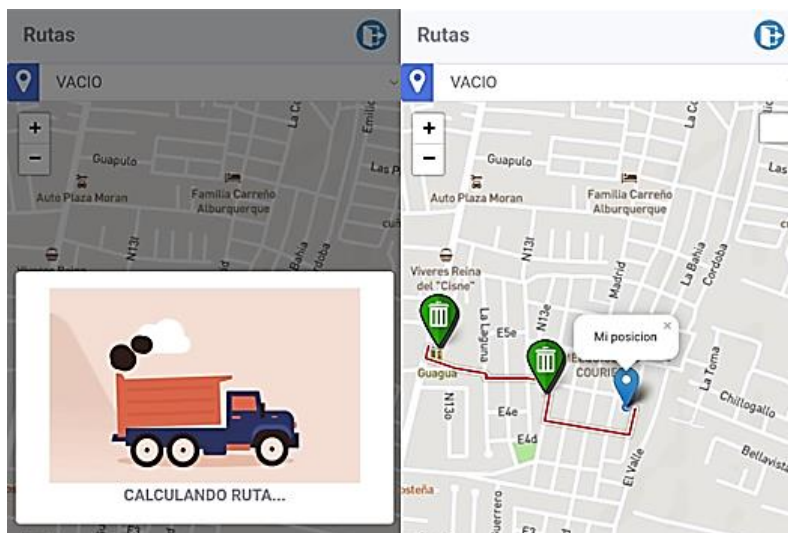


Fuente: Autor

Una vez seleccionado el tipo de contenedores a genera la ruta nos calculara la ruta mas optima. Una vez calculado este se muestra la ruta a tomar desde el punto actual.

Figura 64.

Cálculo de la ruta más óptima

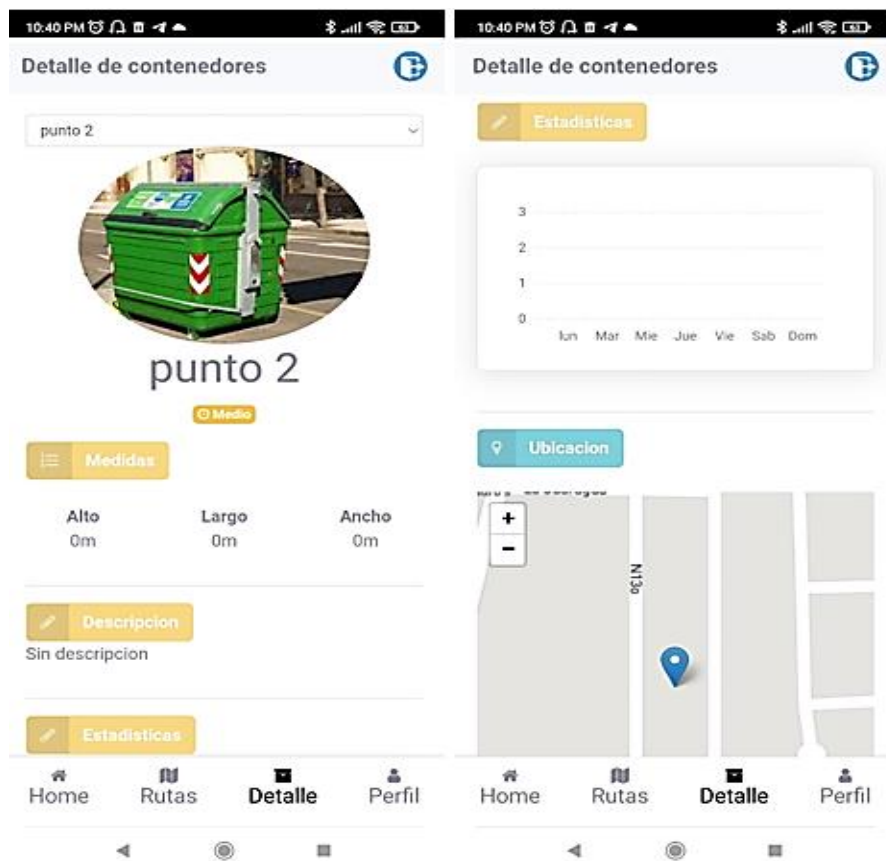


Fuente: Autor

- **Detalle:** En la pestaña de detalle se muestra los datos de los contenedores registrados en el sistema web. Esta pantalla muestra una foto del contenedor subida por el sistema web con datos como:
 - largo
 - alto
 - ancho
 - descripción u observación del contenedor
 - estadísticas de un promedio de su estado por día de la semana
 - posición en el mapa

Figura 65.

Información del detalle de los contenedores

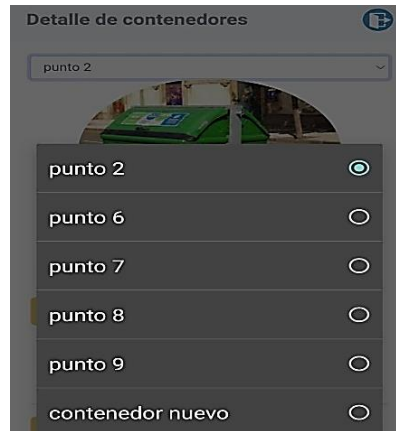


Fuente: Autor

En la parte superior se puede escoger el contenedor del cual se requiere ver su detalle.

Figura 66.

Lista de los contenedores

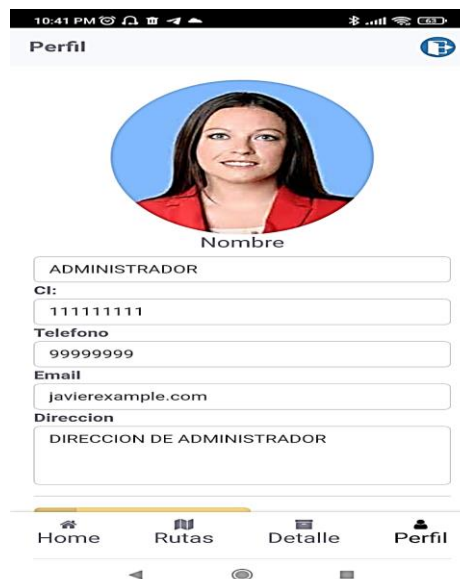


Fuente: Autor

- **Perfil:** En la ultima seccion tendremos la del perfil en donde mostraremos datos informativos de la persona que inicio sesion, datos basico que se guardaron des de la aplicación web previamente realizada

Figura 67.

Perfil de usuario

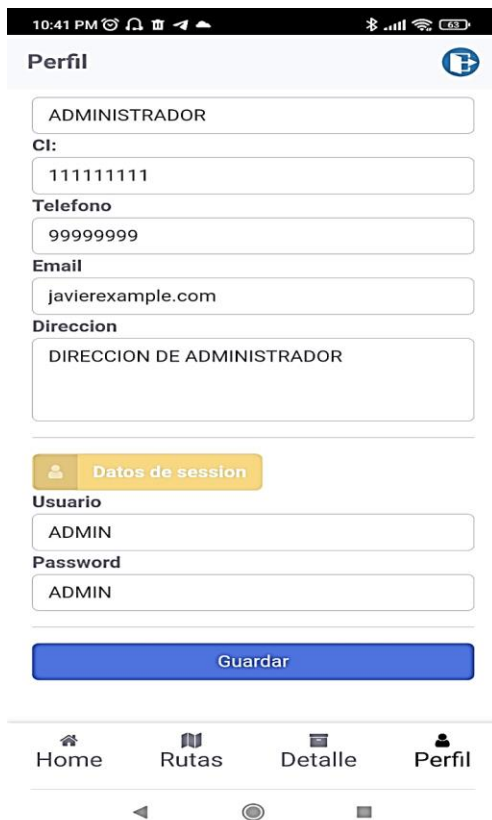


Fuente: Autor

Además se puede visualizar, las credenciales de inicio de session. Este apartado permite actualizar los datos desde la aplicación.

Figura 68.

Actualización de datos

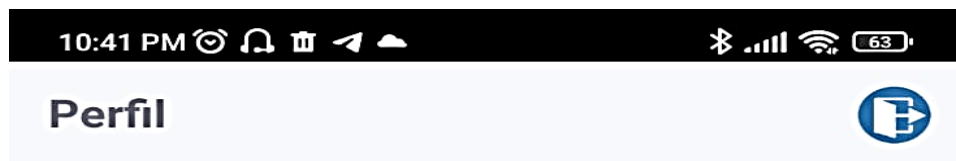


Fuente: Autor

En la parte superior derecha se encuentra un icono de salida. El cual permite cerrar la sesión.

Figura 69.

Cerrar sesión.

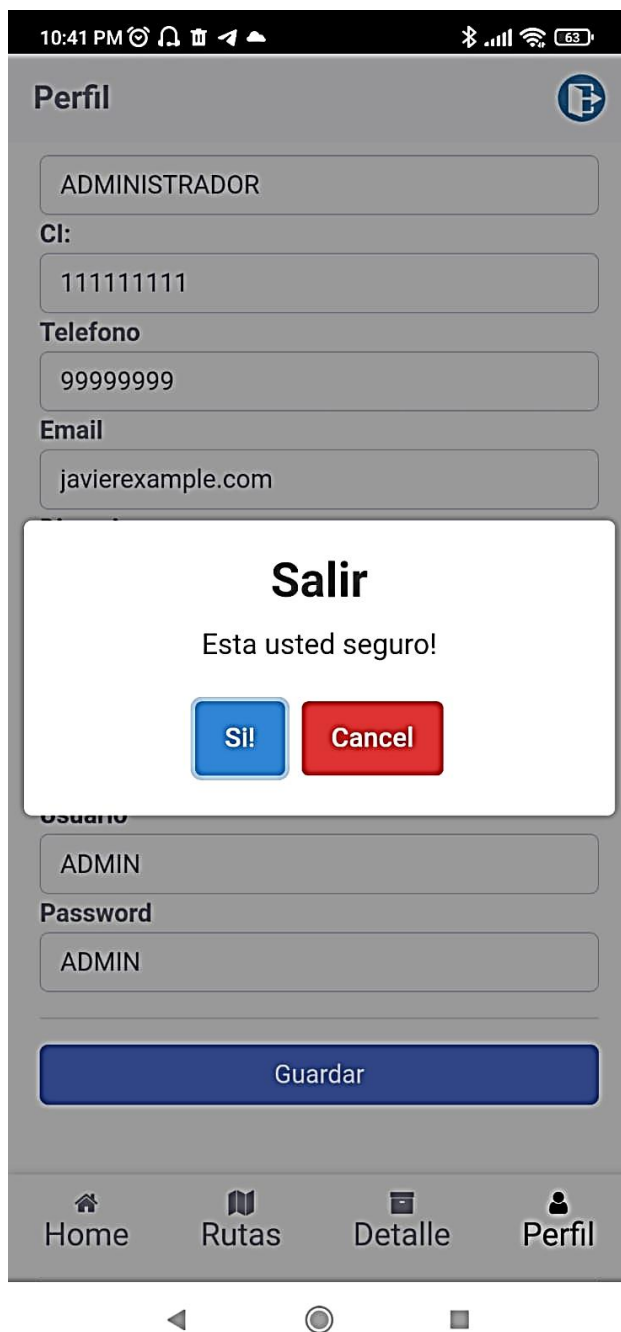


Fuente: Autor

Una vez presionado el icono de cerrar sesión, se muestra una alerta en donde se puede cancelar o salir definitivamente.

Figura 70.

Aceptar cerrar sesión.



Fuente: Autor

CONCLUSIONES Y RECOMENDACIONES

En este apartado se muestra las conclusiones obtenidas al realizar el presente trabajo de titulación.

5.1. Conclusiones

- El sistema actual de recolección presenta deficiencias, siendo estas evidenciadas de manera visual con el desbordamiento de los contenedores, por lo que es imperioso la implementación del **“sistema de recolección de desechos optimizado mediante IoT”**.
- Al eliminar la incertidumbre del sistema se logró la optimización de este, puesto que la ruta se centra únicamente en los contenedores en estado lleno, descartando los vacíos.
- Mediante la comparativa de tecnologías IoT se determina que la tecnología apropiada para el desarrollo del proyecto es la tecnología de transmisión SigFox, debido a sus prestaciones.
- El Backend de SigFox limita el tratamiento de los datos, por lo que es necesario que sean reenviados hacia un servidor propio.
- El sistema es adaptable a cualquier entorno que utilice contenedores y que cuente con cobertura SigFox.

5.2. Recomendaciones

- Utilizar el kit Thinxtra para la comunicación con Sigfox, debido a que basa su funcionamiento en Arduino y su costo de implementación es bajo.

- El módulo Wisol sea el RC-Z4, puesto que su frecuencia de trabajo es de 920 MHz, la cual es específica para trabajar en Ecuador y Sur América.
- En futuros trabajos, tomar en cuenta el tiempo para optimizar el algoritmo de selección de rutas.
- La estructura sea resistente a daños, debido a que se encontrará a la intemperie.
- Se impulse la utilización de la tecnología SigFox en futuros trabajos.

BIBLIOGRAFÍA

- Alcocer, P., Cevallos, O., & Knudsen, J. (2019). Mejoramiento de la gestión integral de los residuos sólidos urbanos en el cantón de Quevedo, Ecuador. *Universidad y Sociedad, 11*(5), 62-66.
- Alvear, V., Rosero, P., Peluffo, D., & Pijal, J. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE, 8*(1), 244-252.
- Apache Cordova. (26 de 04 de 2022). *Apache Cordova*. Obtenido de <https://cordova.apache.org/docs/en/11.x/guide/overview/index.html>
- Arduino . (2020). *Aprendiendo Arduino*. Obtenido de Estructura IoT: <https://aprendiendoarduino.wordpress.com/tag/modelos-de-capas-iot/>
- Arduino. (2021). *¿Que es Arduino?* Obtenido de <https://arduino.cl/que-es-arduino/>
- Arduino. (2022). *Tableros Arduino*. Obtenido de Store Arduino : <https://store-usa.arduino.cc/collections/boards>
- Asamblea Constituyente. (2008). *Constitución de la República del Ecuador*. Montecristi: Lexis.
- Barrera, O., & Ros, J. (2016). *Sistemas eléctricos y de seguridad y confortabilidad 2*. Madrid, España: Ediciones Paraninfo.
- Bermeo, J. (2019). *Diseño, desarrollo e implementación de una solución IoT para medir los niveles de radiación solar, usando una red (SIGFOX) de cobertura amplia de baja potencia (LPWAN) en la facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil*. Guayaquil: Universidad Católica de Santiago de Guayaquil.

- Cardozo, A., Yamin, A., & Souza, R. (2016). n Architecture Proposal to Distributed Sensing in Internet of Things. *INSCIT*, 67-72.
- Castillo, P. (2018). *Red pública, abierta y gratuita de Internet de las Cosas en la ciudad de Querétaro*. Universidad de Jaén, Escuela Politécnica Superior. Jaén, España: UJA.
- Castillo, R., & Espitia, J. (2020). Caracterización de zonas de riesgo por crecientes de ríos de bajo caudal, para la implementación de un sistema de alertas tempranas (SAT) con tecnología LoRa y LoRaWAN. *Información tecnológica*, 31(2), 47-53.
- Cortez, J. (2018). *Diseño de sistema de monitoreo para determinar el llenado de un contenedor de basura*. Universidad Mayor de San Andrés, Carrera de Electrónica y Telecomunicaciones. La Paz: RECLT.
- DSET Energy. (16 de 10 de 2021). *Sigfox*. Obtenido de Tecnología Sigfox:
<http://www.dset-energy.com/2019/06/05/tecnologia-sigfox/>
- EPA.gov. (2021). *Panorama nacional: datos y cifras sobre materiales, desechos y reciclaje*. Obtenido de <https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/national-overview-facts-and-figures-materials>
- Fazenda, A., & Tavares Russo, M. (2016). Caracterización de residuos sólidos urbanos en Sumbe: herramienta para gestión de residuos. *Ciencias Holguín*, 22(4), 1-12.
- GADP Imbabura. (2012). *Módulo de Gestión Integral de Residuos Sólidos para la Mancomunidad de los cantones de la provincia de Imbabura*. Ibarra: GADIP.
- García, F. (2018). *Ciencia de la Computación e Inteligencia Artificial*. Salamanca, España: Universidad de Salamanca.
- Gobierno Autónomo Descentralizado Ibarra. (2019). *Ordenanza que Regula la Actividad de Reciclaje y Gestores de Residuos Sólidos Comunes en la fuente en el Cantón*

Ibarra. Recuperado el 20 de Febrero de 2022, de

[http://documentos.ibarra.gob.ec/uploads/documentos/ORDENANZA/ORDENANZA_QUE_REGULA_LAS_ACTIVIDADES_DE_RECICLAJE\(12-06-2021_16_28_15\).pdf](http://documentos.ibarra.gob.ec/uploads/documentos/ORDENANZA/ORDENANZA_QUE_REGULA_LAS_ACTIVIDADES_DE_RECICLAJE(12-06-2021_16_28_15).pdf)

Hardyanto, W. (2016). Applying an MVC Framework for The System Development Life Cycle with Waterfall Model Extended. *Journal of Physics: Conference Series*, 1-8.

Higuera, L. (2019). *Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT*. Guayaquil: Universidad Católica de Santiago de Guayaquil.

IONOS. (Agosto de 5 de 2021). *Digital Guide*. Obtenido de Requisitos del sistema y pasos: <https://www.ionos.es/digitalguide/servidores/know-how/>

López Vicario, J., & Vilajosana Guillén, X. (2019). *Estudio en detalle de NB-IoT. Comparacion con otras tecnologías LPWAN considerando diferentes patrones de tráfico*. . Cataluña : Universitat Oberta de Catalunya.

Marmolejo, L., Torres, P., Oviedo, E., Bedoya, D., Amezquita, C., Klinger, R., & Diaz, L. (2009). Flujo de residuos: Elemento base para la sostenibilidad del aprovechamiento de residuos sólidos municipales. *Ingeniería y competitividad*, 11(2), 79-92.

Molano, I., Galarza, C., & Beainy, C. (2021). Sensor network communicated bidirectionally using sigfox technology. *Ingenio Magno*, 12(1), 122-132.

Mora, A., Sánchez, J., Sánchez, J., Chagolla, H., & Mora, M. (2016). Red pública, abierta y gratuita de Internet de las Cosas en la ciudad de Querétaro. *Handbook TI*, 33(1), 3-5.

- Morocho, V. (2018). Sistema multimodal que permita monitorear y controlar la recolección dedesechos sólidos en tiempo real para Smart Cities. *Revista de Tecnologías de la Informática y las Telecomunicaciones*, 60-73.
- Munera, A. (2018). *Modelado y evaluación de la tecnología Sigfox para NS3*. Valencia : Universitat Politècnica de València.
- Noguera, K., & Oliveros, J. (2010). Los rellenos sanitarios en Latinoamérica. *Revista Académica Colombiana de Ciencia*, XXXIV(132), 347-351.
- Ortiz, M. (2020). *Desarrollo de una red de sensores inalámbricos utilizando tecnología LoRa para el monitoreo de un sistema*. Quito: BS THES.
- Palma, D. (2011). Tarjetas de Desarrollo: Herramientas para el diseño. *Instituto Técnico Central* .
- Peralta, A., & Vélez, C. (2011). Diagnostico socioeconómico y ambiental del manejo de residuos sólidos domésticos en el municipio de Haina. *Ciencia y Sociedad*, XXXVI(2), 239-254.
- Pérez, E., Racero, J., & Villa, G. (2007). LOS SISTEMAS DE RECOLECCIÓN DE RESIDUOS SOLIDOS (LOS METODOS Y SUS APLICACIONES). *Ciencia UAT*, 1(4), 58-60.
- Pinto, A. (Noviembre de 2020). *UNIVERSIDAD TÉCNICA DEL NORTE*. Obtenido de INSTITUTO DE POSTGRADO:
<http://repositorio.utn.edu.ec/bitstream/123456789/10809/2/PG%20798%20TRABAJO%20GRADO.pdf>

- Royce, W. (2015 de Agosto de 2021). *Managing the Development of Large Software Systems*. Obtenido de Ideas That Created the Future:
<https://doi.org/10.7551/mitpress/12274.003.0035>
- Sáez, A., & Urdaneta, J. (2014). Manejo de residuos sólidos en América Latina y el Caribe. *Omnia*, 121-130.
- Sigfox. (19 de Octubre de 2020). *Coverage on Demand*. Obtenido de Conectivity Services
Sigfox: https://www.sigfox.com/en/services#id_connexion
- Thinextra. (2021). *Thinextra Sigfox Developer Xkit*. Obtenido de <https://thinextra.com/iot-connectivity/xkit/>
- Torres, Á., González, J., & Torres, A. (2017). Gestión de residuos sólidos domiciliarios en la ciudad de Villavicencio. Una mirada desde los grupos de interés: empresa, estado y comunidad. *Revista Luna Azul*, 44(1), 177-185.
- Torrubia, G., & Terrazas, V. (2012). Algoritmo de Dijkstra. Un tutorial interactivo. *VII Jornadas de Enseñanza Universitaria de la Informática*, 20(1), 12-17.

ANEXOS

Anexo 1. Manual de Usuario

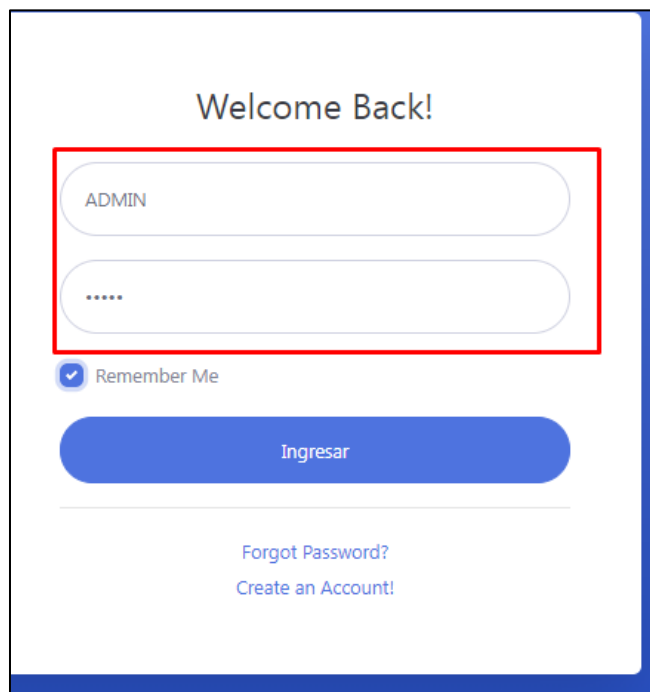
INGRESO AL SISTEMA

Se ingresa a la dirección electrónica:

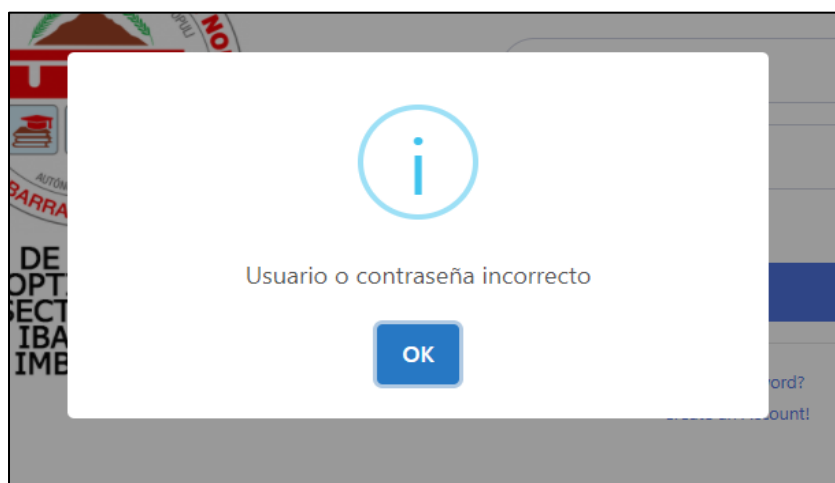
<https://www.jeanobando.online/rutas/php/view/login.php> , el cual nos mostrara una login para iniciar el acceso



En la parte derecha se coloca las credenciales de ingreso



Una vez colocado las credenciales se pulsa en ingresar, en el caso de que las credenciales estén incorrectas aparecerá una alerta



Si las credenciales son correctas se redireccionará a la parte home del sistema, en esta pantalla se puede observar en su lado izquierdo un menú lateral



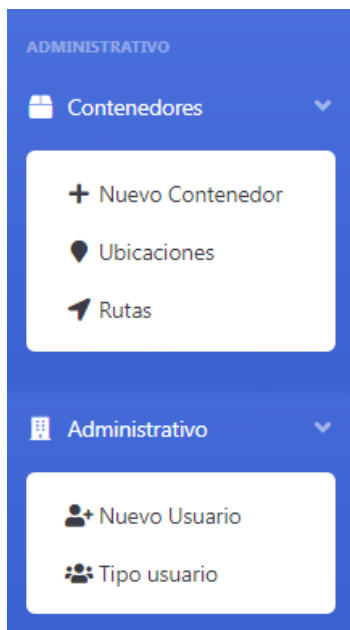
En este menú se desplegará según el tipo de usuario que ingrese al sistema, en el caso del administrador nos desplegará estos dos ítems

En donde encontraremos en contenedores

- Nuevo contenedor
- Ubicación
- Rutas

En el caso de Administrativo encontraremos

- Nuevo usuario
- Tipo de usuario



En el caso de acceder con un usuario que no es administrador

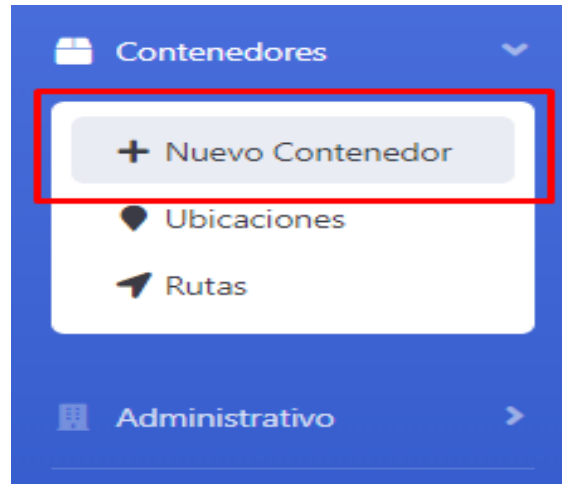
Solo nos parecerá el apartado de contenedores y dentro de el

- Ubicaciones
- Rutas

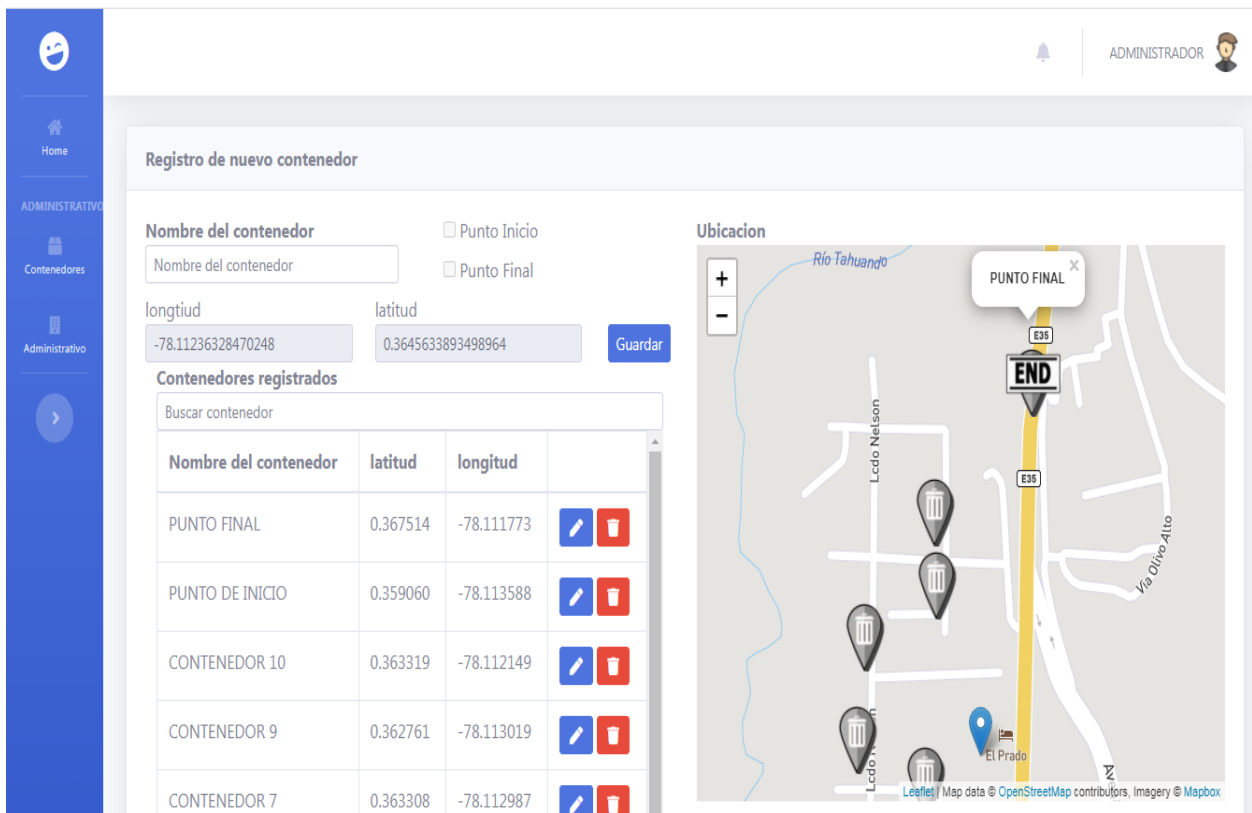


NUEVO CONTENEDOR

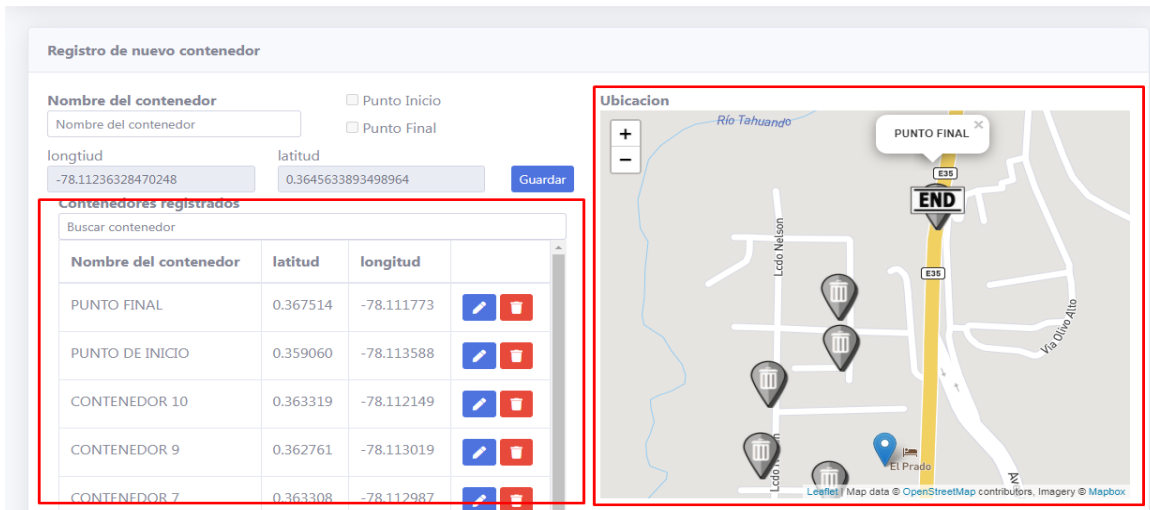
Este nos permitirá ingresar nuevos contenedores, editar contenedores existentes y a su vez eliminar, nos mostrará una lista de todos los contenedores registrados



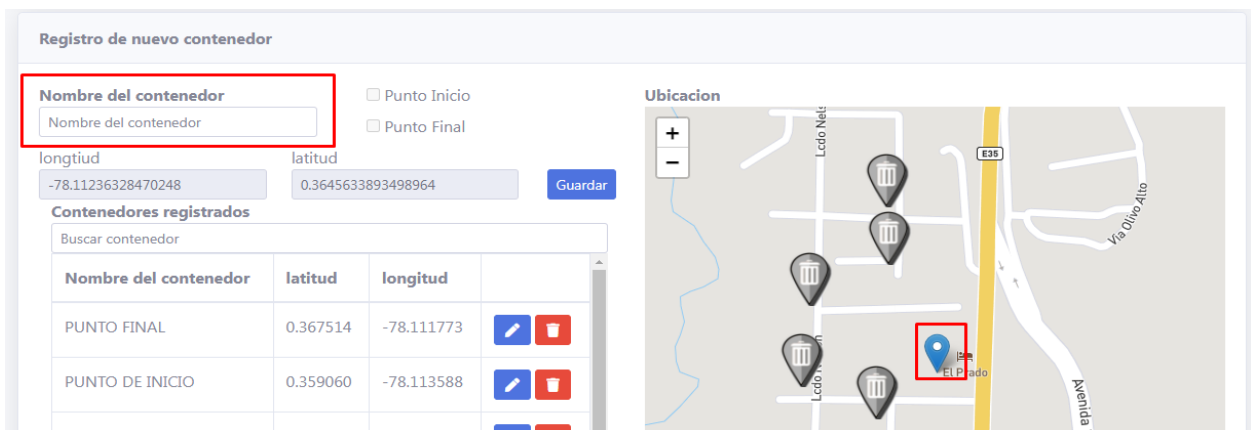
Una vez pulsado en nuevo contenedor nos enviara a una pantalla similar al de la imagen



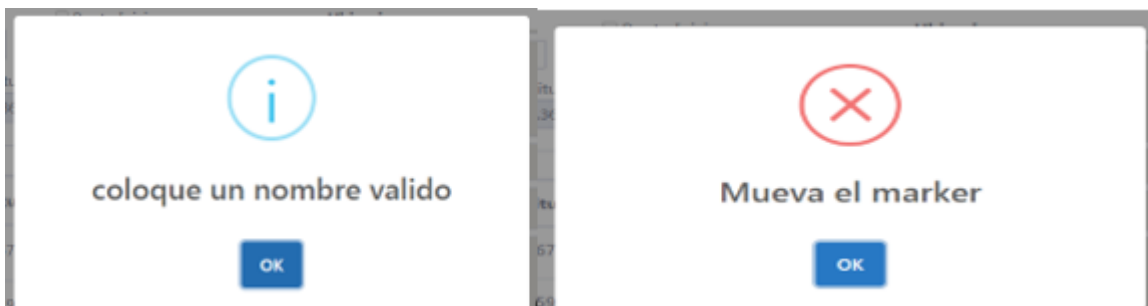
Aquí podremos observar en el lado izquierdo la lista de contenedores registrado y a su lado derecho un mapa para ubicar geográficamente su posición.





Para ingresar un contenedor nuevo basta con escribir el nombre del contenedor y mover el marker color azul y dar en el botón de guardar.



En el caso de no mover el marcador o no colocar un nombre valido este nos mostrara una alerta similar.



En el caso de editar un contenedor, pulsamos en el botón situado a la derecha de la tabla

CONTENEDOR 10	0.363319	-78.112149	 
---------------	----------	------------	---

Este nos cargara sus datos, en el caso de cambiar su ubicación, moveremos el marker azul a su nueva posición y guardaremos.











Registro de nuevo contenedor

Punto Inicio
 Punto Final

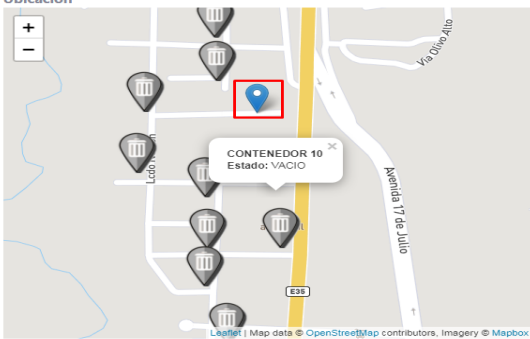
Nombre del contenedor
CONTENEDOR 10

longitud: -78.11214904886343 latitud: 0.3633188694740786 Guardar

Contenedores registrados



Nombre del contenedor	latitud	longitud	
PUNTO FINAL	0.367514	-78.111773	 
PUNTO DE INICIO	0.359060	-78.113588	 
CONTENEDOR 10	0.363319	-78.112149	 
CONTENEDOR 9	0.362761	-78.113019	 
CONTENEDOR 7	0.363308	-78.112987	 

Ubicación

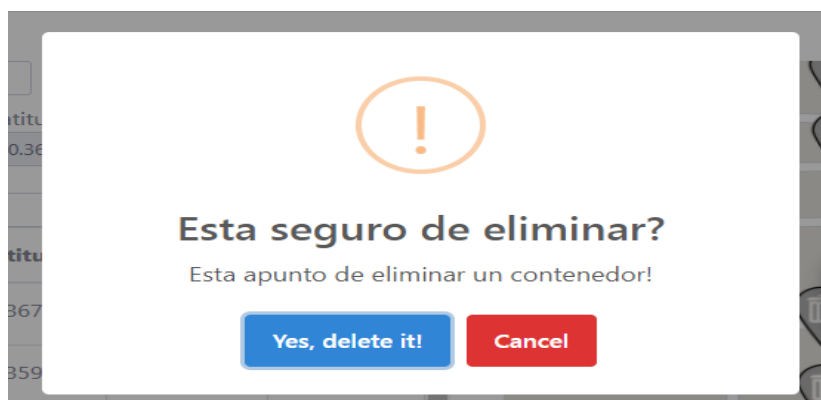


ELIMINAR UN CONTENEDOR

En el caso de eliminar un contenedor pulsaremos el botón rojo de la parte derecha de la tabla, buscando el contenedor a eliminar.

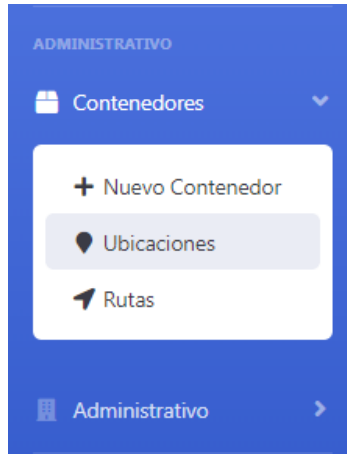
PUNTO DE INICIO	0.359060	-78.113588	 
-----------------	----------	------------	---

Este nos pedirá confirmación. Una vez confirmado este nos eliminara el contenedor registrado.

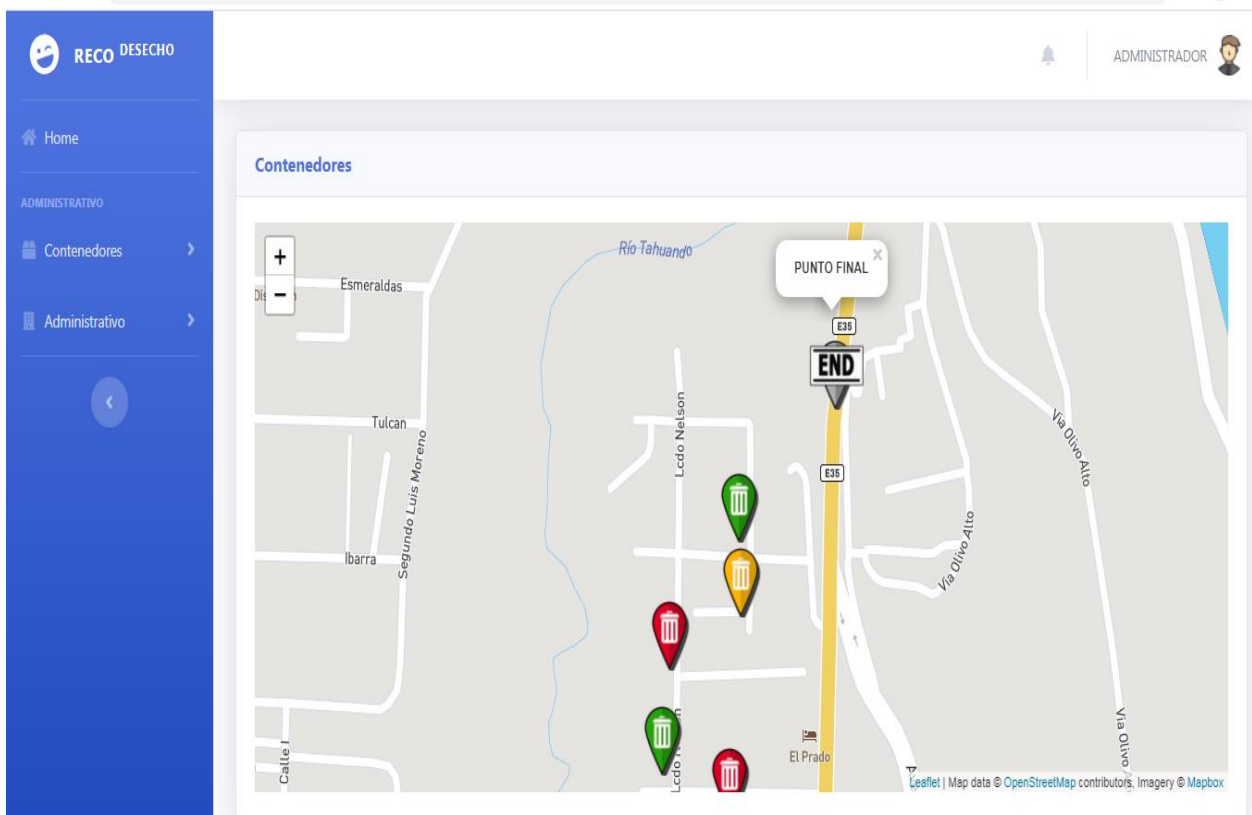


UBICACIONES






Este apartado nos servirá de manera informativa para saber dónde están ubicados los contenedores registrados y su estado.



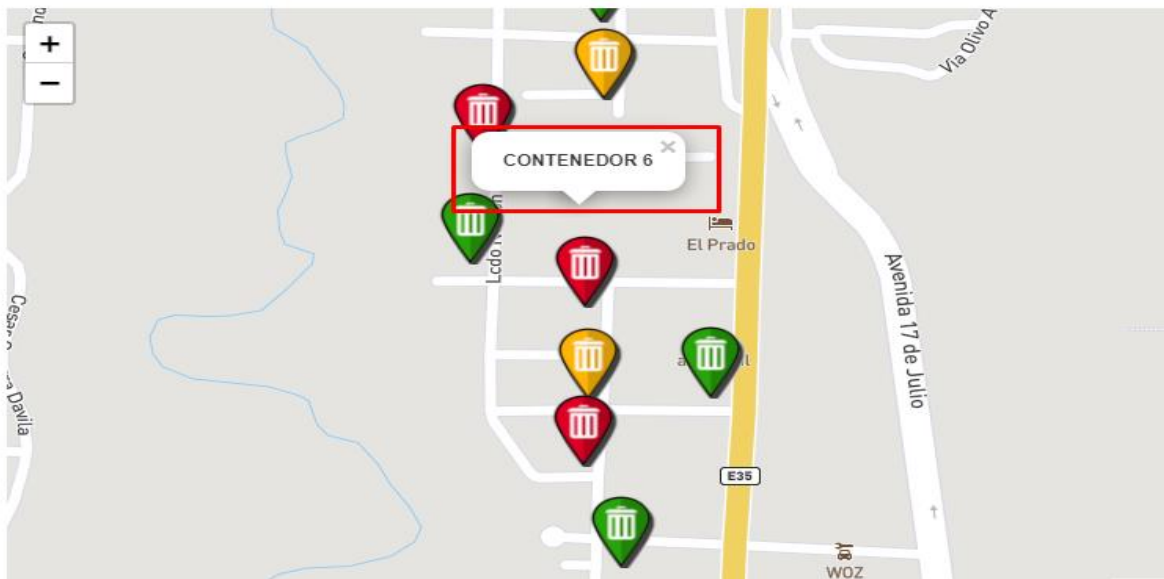
Una vez dentro nos aparecerá una pantalla similar a la imagen



Este nos mostrara los estados del contenedor con colores distinto

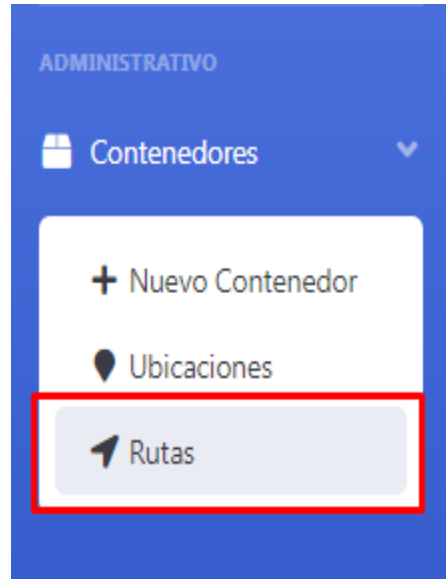
	Indica el punto fin en el caso de registrarse un punto final
	Indica el punto de partida para una ruta
	Indica contenedores de estado lleno
	Indica contenedores de estado medio lleno
	Indica contenedores de estado vacío

En el caso de querer saber el nombre del contenedor solo daremos clic en el indicador o marker y nos aparecerá su nombre

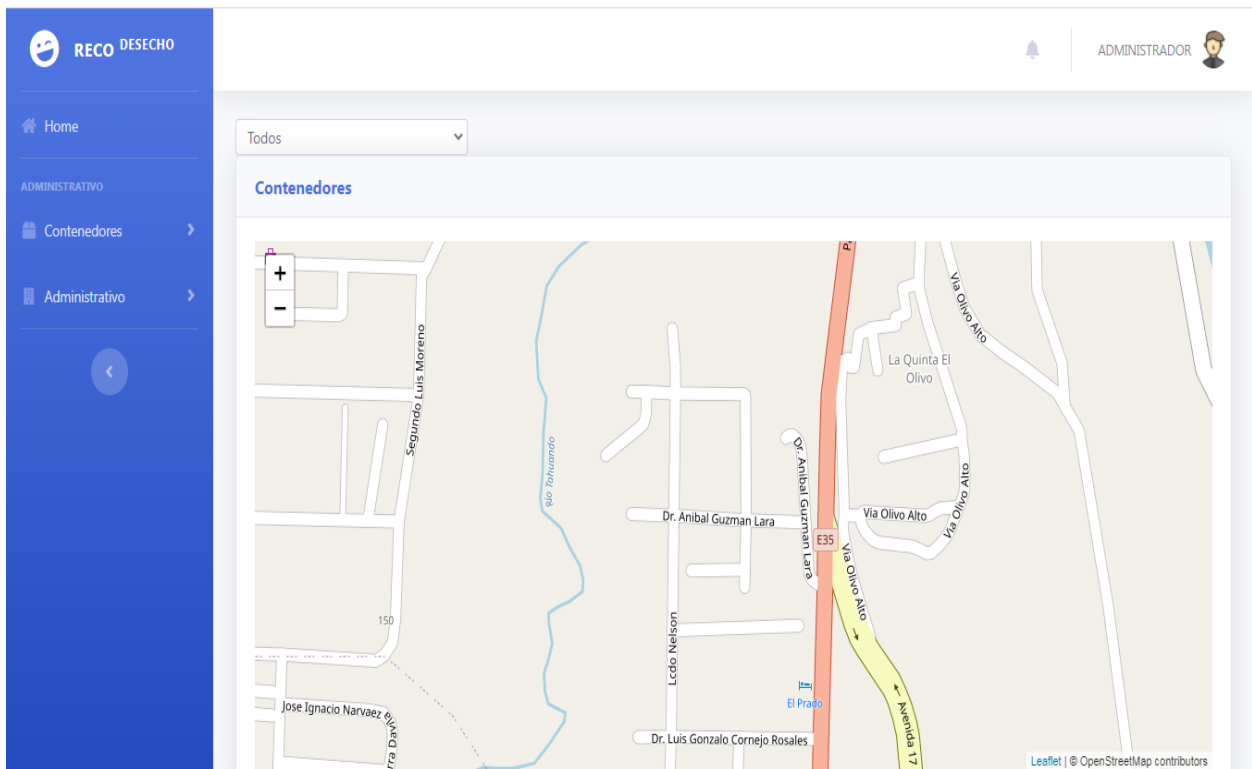


RUTAS

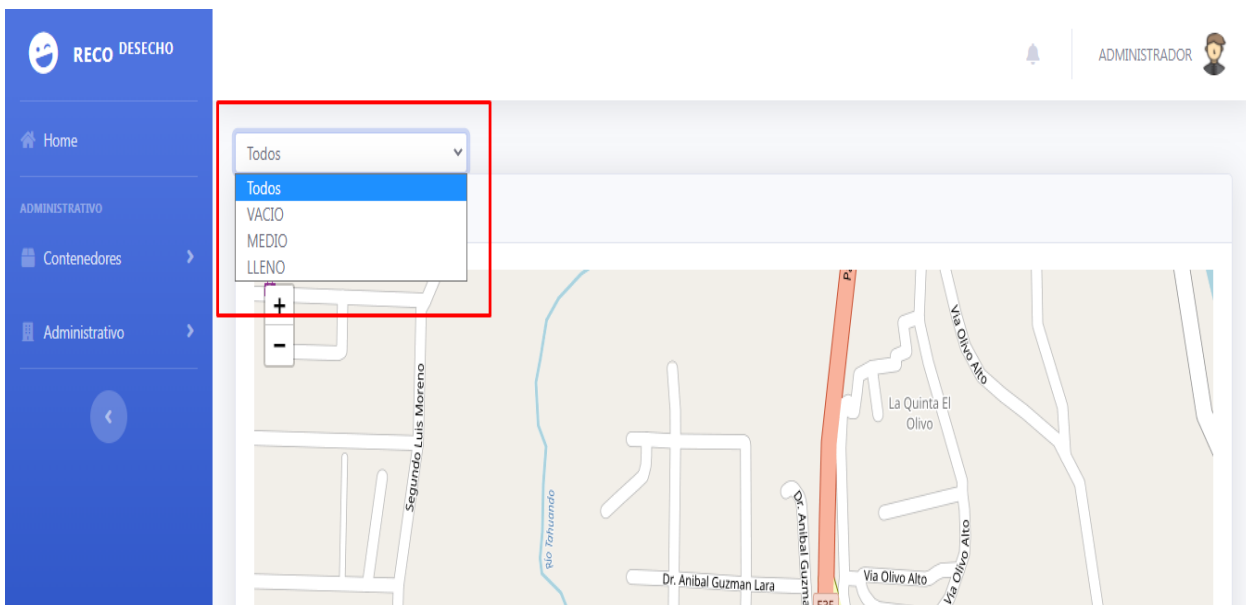
Este nos permitirá calcular la ruta más óptima entre contenedores para su recolección.



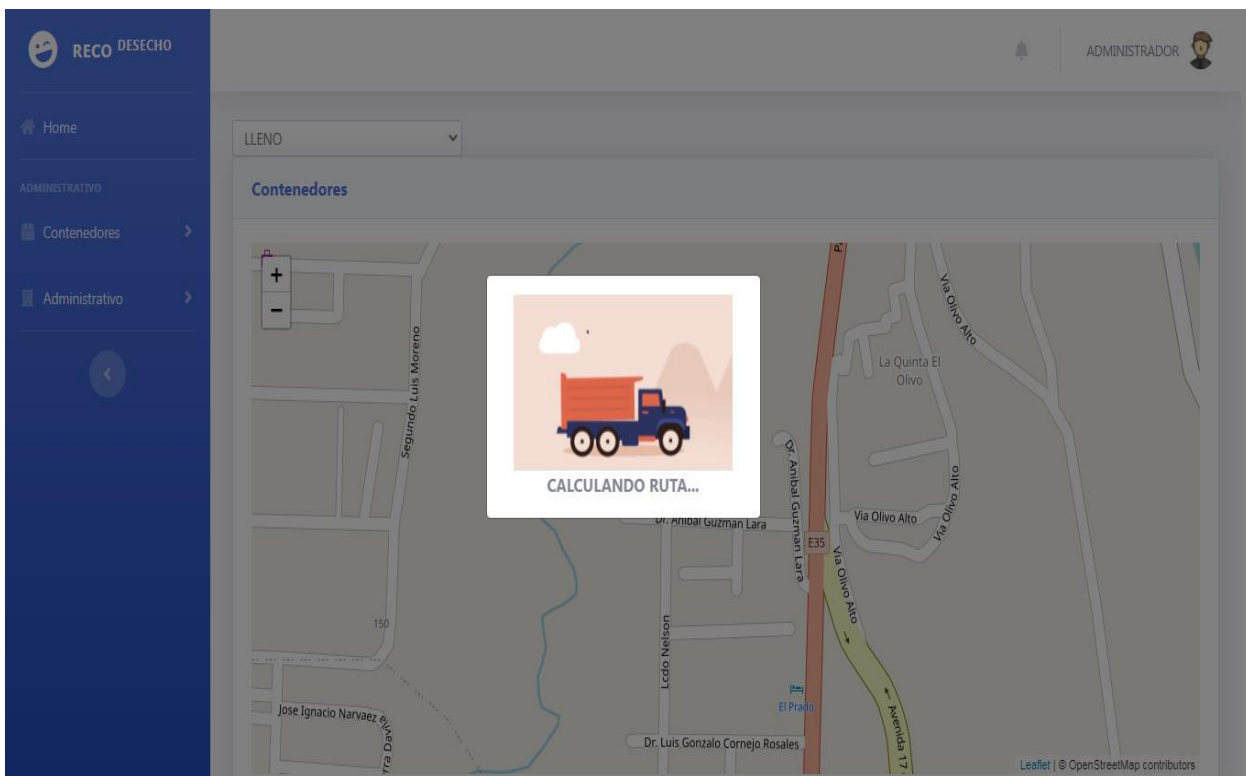
Al ingresar a este apartado nos aparecerá una pantalla similar a la imagen



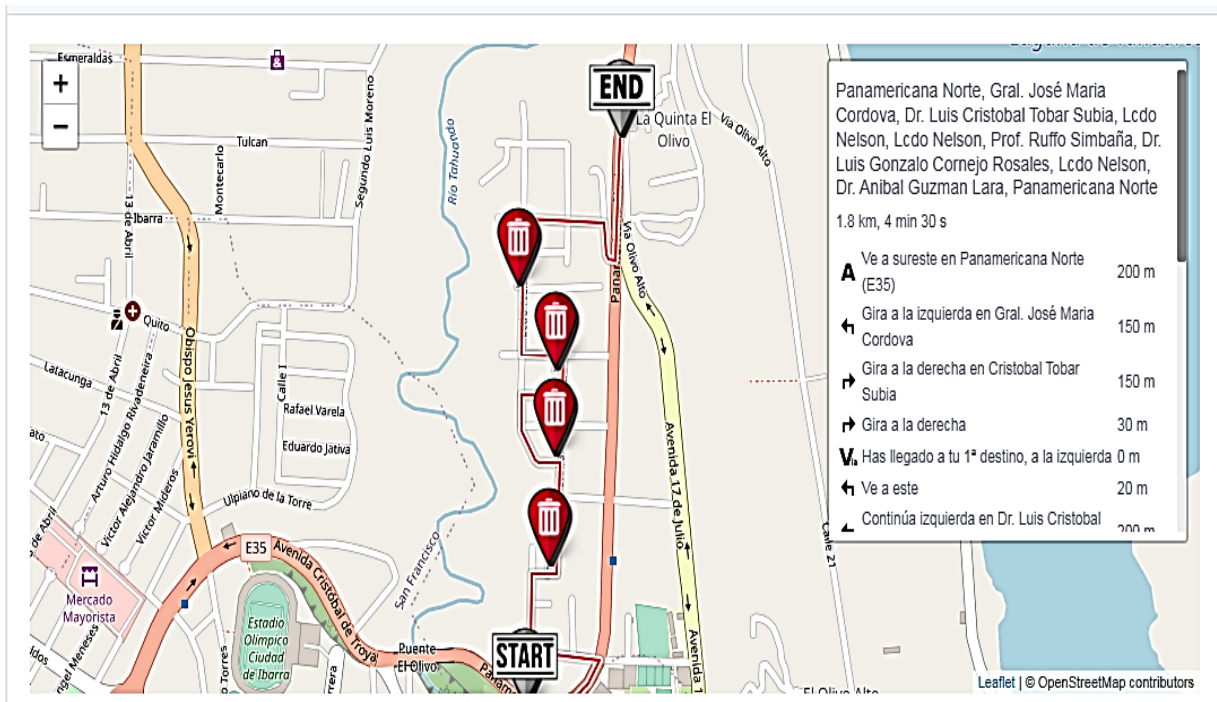
El mapa en esta ocasión nos aparecerá totalmente vacío, para visualizar las rutas nos dirigiremos a la parte superior en donde seleccionaremos el tipo de ruta a trazar.



Una vez seleccionado nos comenzara a calcular la ruta óptima tomando en cuenta el punto de inicio y el punto final.



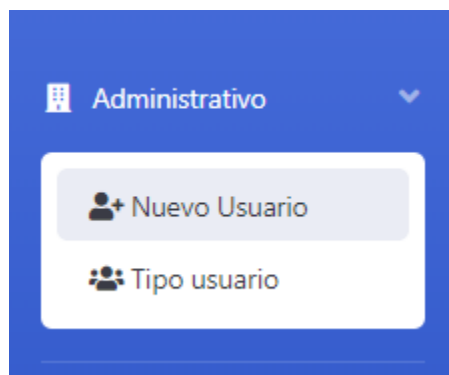
Nos trazara la ruta dependiendo el estado del contenedor seleccionado, en nuestro caso los contenedores llenos



Podremos observar además que nos aparecen indicaciones de cómo seguir el recorrido a su lado derecho.

NUEVO USUARIO

Este nos permitirá observar una lista de todos los usuarios registrados, así también agregar, editar y eliminar usuarios.



Al ingresar en esta opción observaremos una pantalla similar al de la imagen

Nombre: CI / RUC:

Nick: Password:

Tipo de usuario:

Direccion:

Nombre	CI / RUC	Nick	Password	Tipo usuario	Direccion	
LUIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
PACO	1722221450	PACO	PEPE	USUARIO	CALDERON	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
ADMINISTRADOR	111111111	ADMIN	ADMIN	ADMINISTRADOR	DIRECCION DE ADMINISTRADOR	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Para registrar un nuevo usuario ingresaremos todos los datos requeridos en la parte izquierda y daremos en guardar

Nombre: CI / RUC:

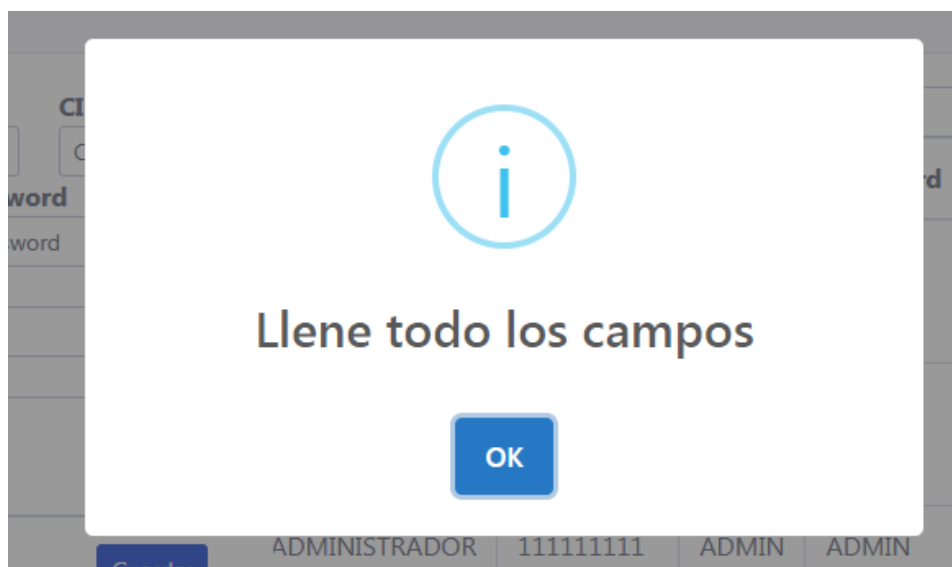
Nick: Password:

Tipo de usuario:

Direccion:

Nombre	CI / RUC	Nick	Password	Tipo usuario	Direccion	
LUIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
PACO	1722221450	PACO	PEPE	USUARIO	CALDERON	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
ADMINISTRADOR	111111111	ADMIN	ADMIN	ADMINISTRADOR	DIRECCION DE ADMINISTRADOR	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

En el caso de no llenar todos los campos este nos mostrara una alerta



Una vez lleno todos los campos este nos desplegara sus datos en el lado derecho en la tabla

Nombre CI / RUC Buscar usuario

Nombre CI / RUC

Nick Password

Nick Password

Tipo de usuario
Seleccione

Direccion

Nombre	CI / RUC	Nick	Password	Tipo usuario	Direccion	
LUIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
PACO	1722221450	PACO	PEPE	USUARIO	CALDERON	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
ADMINISTRADOR	111111111	ADMIN	ADMIN	ADMINISTRADOR	DIRECCION DE ADMINISTRADOR	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Para editar un registro pulsaremos en el botón de color azul de editar, situado a la derecha de la tabla

LUIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
-------------	-----------	------	------	---------	------------	--

Este cargara todos los datos en los mismos campos en donde registramos a nuevos usuarios







Usuario

Nombre **CI / RUC**



Nick **Password**

Tipo de usuario

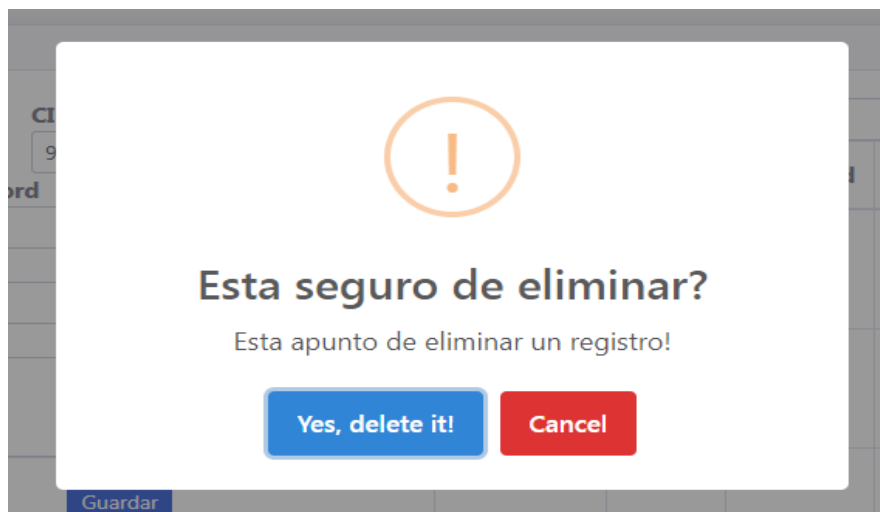
Direccion

Nombre	CI / RUC	Nick	Password	Tipo usuario	Direccion	
JIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	 
PACO	1722221450	PACO	PEPE	USUARIO	CALDERON	 
ADMINISTRADOR	111111111	ADMIN	ADMIN	ADMINISTRADOR	DIRECCION DE ADMINISTRADOR	 

Modificamos lo que necesite y le damos al botón de guardar. En el caso de eliminar un registro pulsaremos en el botón rojo situado a la derecha de la tabla

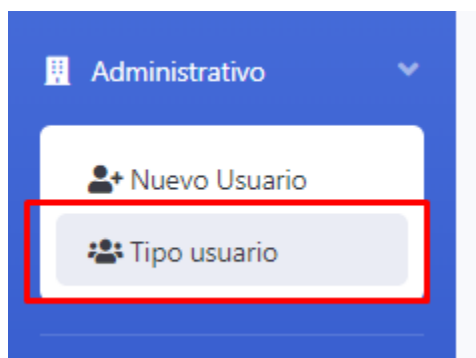
Nombre	CI / RUC	Nick	Password	Tipo usuario	Direccion	
JIS ORTEGA	999999990	LUIS	1234	USUARIO	NOTE OLIVO	 

Una vez pulsado este nos pedirá confirmación. Una vez confirmado este eliminara el usuario de la tabla.

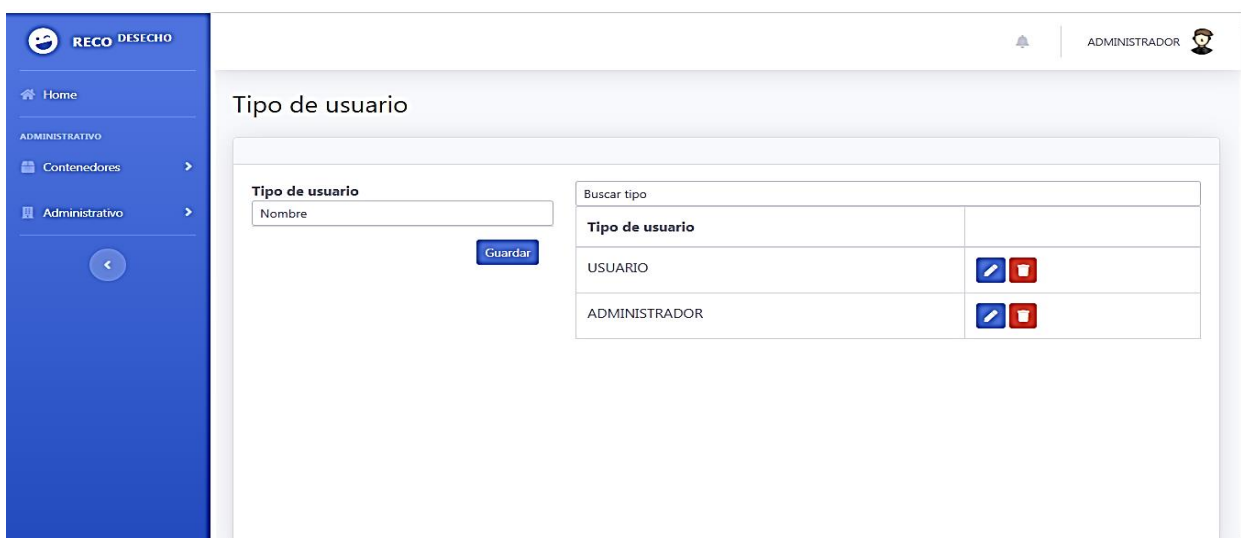


TIPO DE USUARIO

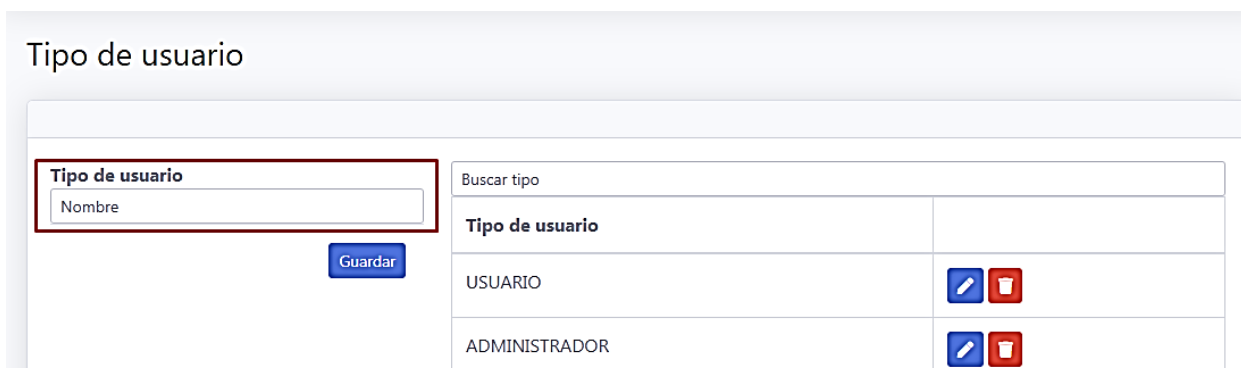
Este nos permitirá observar una lista de todos los tipos de usuarios registrados, así también agregar, editar y eliminar tipos de usuario.



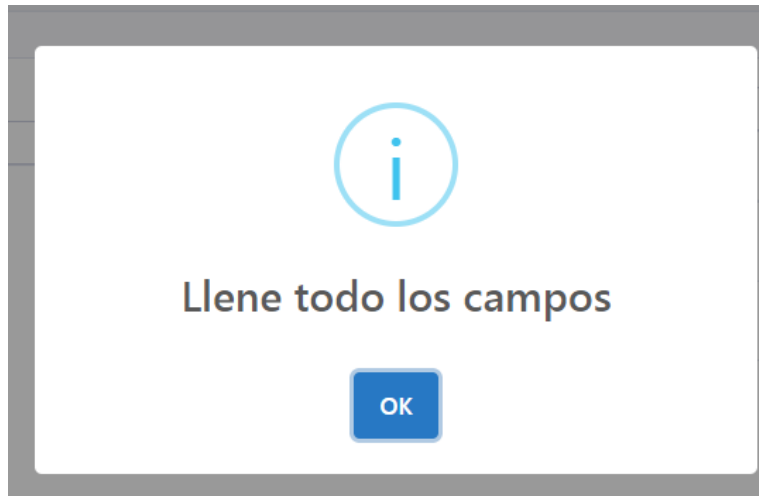
Al ingresar en esta opción observaremos una pantalla similar al de la imagen



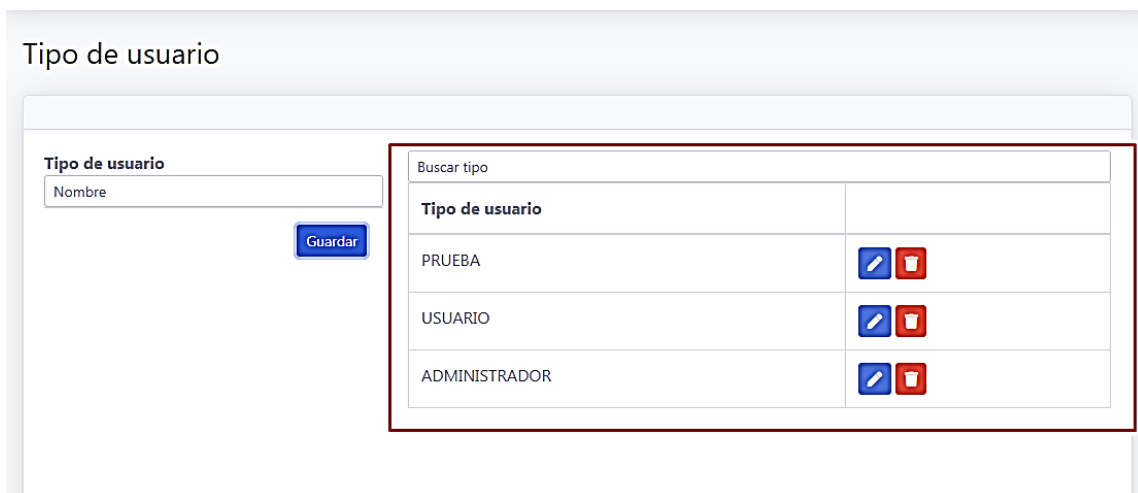
Para registrar un nuevo tipo de usuario ingresaremos un nombre para este en la parte izquierda y pulsaremos en guardar



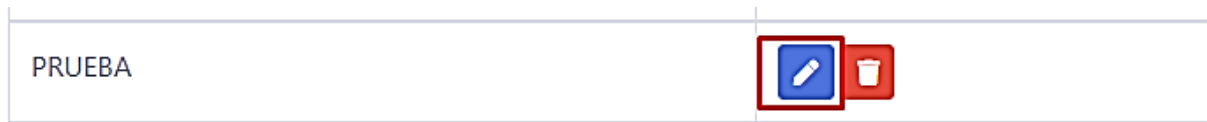
En el caso de no llenar todos los campos este nos mostrara una alerta



Una vez lleno todos los campos este nos desplegara sus datos en el lado derecho en la tabla









Para editar un registro pulsaremos en el botón de color azul de editar, situado a la derecha de la tabla



Este nos cargara todos los datos en los mismos campos en donde registramos a nuevos tipos de usuarios


Tipo de usuario

<input type="text" value="PRUEBA"/>	Buscar tipo
<input type="button" value="Guardar"/>	
Tipo de usuario	
PRUEBA	 
USUARIO	 
ADMINISTRADOR	 

Modificamos lo que necesite y le damos al botón de guardar. En el caso de eliminar un registro pulsaremos en el botón rojo situado a la derecha de la tabla

Tipo de usuario	
PRUEBA	 

Una vez pulsado este nos pedirá confirmación



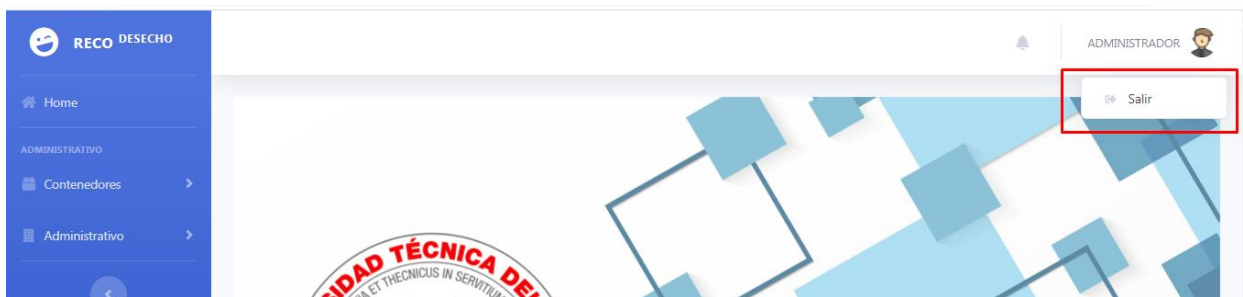
Esta seguro de eliminar?
Esta apunto de eliminar un registro!

Una vez confirmado este eliminara el tipo de usuario de la tabla.

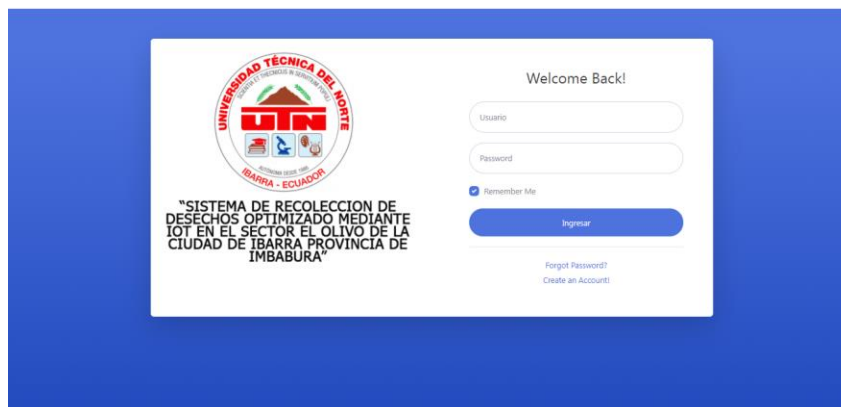
Terminado con el menú lateral podremos observar en la parte superior derecha el usuario que ingreso



Y al pulsar en este nos desplegara la opción de salir



El cual nos permitirá salir del sistema devolviéndonos al punto de inicio.



Anexo 2. Entrevista

A continuación, se muestra las preguntas de la entrevista realizada a la señora Mariana Realpe, moradora de El Olivo.



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERIA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN

1. ¿Cree usted que el sistema de recolección de residuos actual es eficiente?
2. ¿Cree usted que la cantidad de contenedores que posee el barrio es adecuada para la población existente en la zona?
3. ¿Con que frecuencia se realiza la recolección de residuos actualmente?
4. ¿Actualmente existe una manera de informar que un contenedor se encuentra colapsado?
5. ¿Cómo actúan los moradores del sector al observar que los contenedores se encuentran colapsados?
6. ¿Cómo afecta a la población de la zona el hecho de que los contenedores se encuentren colapsados?
7. ¿Cree usted que es adecuado implementar un sistema que indique en tiempo real el nivel de desechos existentes dentro del contenedor?
8. ¿Cree usted que sería útil la implementación de una aplicación móvil en donde se muestre el nivel de desechos existentes del contenedor al personal de recolección?
9. ¿Cree usted que la implementación del sistema de recolección de residuos optimizada con IoT contribuirá con el desarrollo de la zona?
10. ¿Tiene usted alguna sugerencia que le gustaría que se implemente dentro del sistema?

En el enlace a continuación, se encuentra un audio de la entrevista realizada a la señora Mariana Realpe, moradora de El Olivo.

https://utneduec-my.sharepoint.com/:v:/g/personal/jpobandos_utn_edu_ec/EaAoHp_I1jJj2WMK--UVt0BXdiuKbX7ni7cbixQV1-VRw?e=U76ud9

Anexo 3. Presentación Final del Sistema

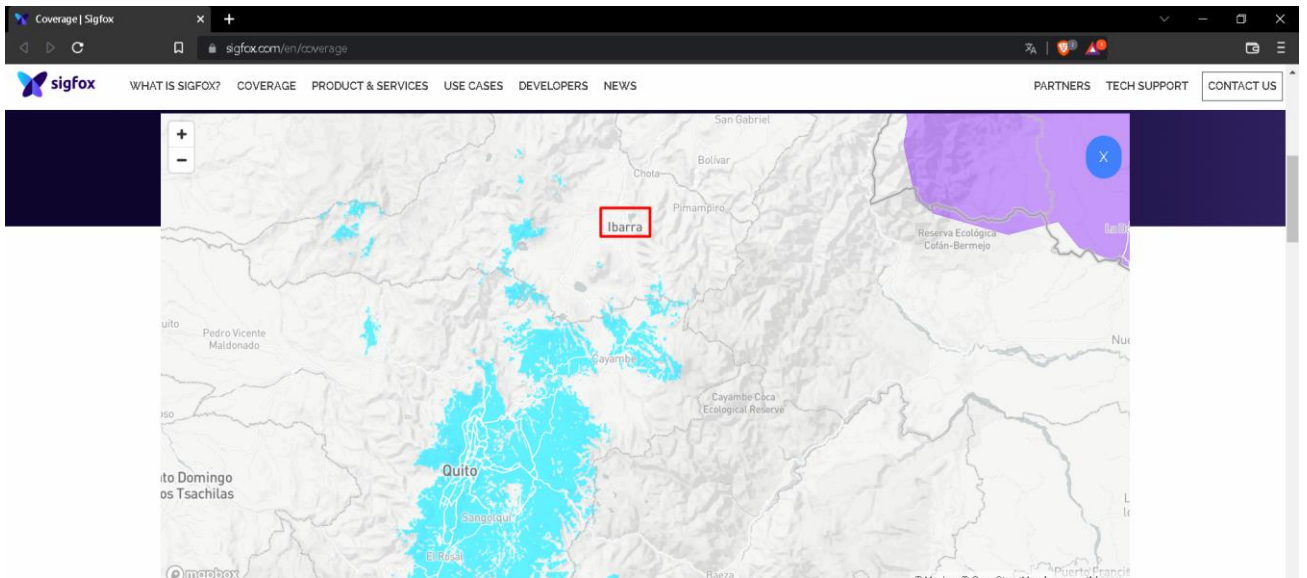
Para la presentación final se monta todos los elementos dentro de una cubierta protectora, la cual se realiza tomando en cuenta las medidas necesarias. En las figuras a continuación se puede observar la presentación final.



Anexo 4. Justificación del desarrollo del prototipo en Quito

El trabajo de grado previo a la obtención del título de ingeniería en electrónica y redes de comunicación, denominado: **Sistema de recolección de desechos optimizado, mediante IOT en el sector el olivo de la ciudad de Ibarra provincia de Imbabura.**

- No se realiza en la ciudad de Ibarra, debido a que la tecnología SigFox no se encuentra disponible actualmente en la zona. La parte en azul es en donde se tiene cobertura, de acuerdo con la página oficial de SigFox. En la parte de antecedentes se menciona que se realizará un prototipo del sistema.



- El tema y los objetivos, se establecen en la ciudad de Ibarra, debido a que, de acuerdo con el reglamento de la Universidad, al momento de la aprobación del tema de tesis, se debe dar soluciones a problemáticas del sector en el que se encuentra la Institución.