



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE TECNOLOGÍAS DE LA  
INFORMACIÓN Y COMUNICACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS**

**SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE DATOS  
REFERENTE A LA RECOLECCIÓN DE DESECHOS DOMICILIARIOS  
APLICANDO GEOLOCALIZACIÓN EN EL GAD MUNICIPAL DEL  
CANTÓN AZOGUES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN SISTEMAS**

**AUTOR: DARIO GONZÁLEZ MIRANDA, JONATHAN PEÑA PEÑA**

**DIRECTOR: ING. ANDRÉS SEBASTIÁN QUEVEDO SACOTO, MSC**

**AZOGUES - ECUADOR**

**2021**

*Yo me gradué en los  
50 años de La Cato!*



# **UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

## **UNIDAD ACADÉMICA DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN**

### **CARRERA DE INGENIERÍA DE SISTEMAS**

**SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE DATOS  
REFERENTE A LA RECOLECCIÓN DE DESECHOS DOMICILIARIOS  
APLICANDO GEOLOCALIZACIÓN EN EL GAD MUNICIPAL DEL CANTÓN  
AZOGUES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERÍA DE SISTEMAS**

**AUTOR:** JONATHAN PEÑA PEÑA, DARIO GONZÁLEZ MIRANDA

**DIRECTOR:** ING. ANDRÉS SEBASTIÁN QUEVEDO SACOTO, MSC

**AZOGUES - ECUADOR**

**2021**

*Yo me gradúe en los  
50 años de La Cato!*

**DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD**

**Dario Xavier González Miranda** portador(a) de la cédula de ciudadanía N° **0301879649**. Declaro ser el autor de la obra: **“Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, 28 de julio del 2021

**Estudiante:**

---

**Dario Xavier González Miranda**

**C.I. 0301879649**

**DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD**

**Jonathan Alexander Peña Peña** portador(a) de la cédula de ciudadanía N° **2100917992**. Declaro ser el autor de la obra: **“Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, 28 de julio del 2021

**Estudiante:**

---

**Jonathan Alexander Peña Peña**

**C.I. 2100917992**

## **CERTIFICACION DEL TUTOR**

Certifico que el presente trabajo fue desarrollado por los estudiantes Dario Xavier González Miranda y Jonathan Alexander Peña Peña, bajo la supervisión del tutor designado (Ing. Sebastián Quevedo); la investigación propuesta sirve como requisito previo a la obtención del título de Ingeniero de Sistemas, el tema “**Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues**” cumple con todas las observaciones realizadas por el tribunal evaluador, por lo que las ideas, opiniones vertidas en el presente, son de exclusiva responsabilidad de los autores.

**Ing. Sebastián Quevedo Sacoto**

**Director**

## **DEDICATORIA**

El presente trabajo de Titulación con gran satisfacción y ahínco en quiero dedicarla a mis padres por su entrega sacrificio y amor todas y cada una de las personas que de una u otra manera me han sabido apoyar no solo durante la elaboración del presente sino durante la vida, de manera especial a mis padres por su entrega amor y sacrificio, de todo corazón muchas gracias.

De la misma manera quiero dedicarla a mis amigos, todos y cada uno de ellos han sido pilares fundamentales durante mi crecimiento no solo como profesional sino como persona, muchas gracias.

Dario Xavier González Miranda

Mi tesis va dedicada con mucho cariño y amor a mi abuelita: Laura Peña, quien con todo su esfuerzo y paciencia me ha sabido guiar por el camino del bien, y haberme permitido cumplir con este objetivo tan importante en mi vida.

A mi mamá, que a pesar de la distancia siempre estuvo junto a mi durante toda mi carrera estudiantil, aún más durante mi formación universitaria le agradezco por todo el apoyo y consejos brindados.

Para mi esposa quien con su presencia ilumina cada segundo de mi existencia y me da fuerzas de seguir adelante en cada momento difícil de la vida, cada segundo que comparto con ella es una eternidad de enseñanzas.

.

Jonathan Alexander Peña Peña

## **AGRADECIMIENTOS**

Quiero expresar mis más sentidos agradecimientos para aquellas personas que hicieron posible la conclusión no solo de este trabajo de tesis, sino desde el inicio de esa aventura llamada “Universidad”, gracias a mis padres Luis González y Zaida Miranda, sin su incondicional apoyo la conclusión de los mismos no hubiese sido posible.

De la misma manera quiero agradecer a mi tutor que sin su aporte y guía no hubiese sido posible el realizar el presente de manera exitosa.

A mis catedráticos de la Universidad Católica de Cuenca, y en si todo el personal que integran la misma, que más allá sus roles dentro de las instalaciones han brindado siempre un apoyo incondicional cuando la situación así lo ameritaba.

Por último y no por ello menos importante a mis compañeros que más de una vez me han sacado sonrisas por muy difícil que se presentaban las situaciones.

Dario Xavier González Miranda

En primer lugar, agradezco a Dios por haberme guiado en los momentos más difíciles y brindarme la fuerza y la sabiduría para llevarme por el camino de la vida.

Quiero agradecer a mi abuelita, a mi mamá y a toda mi familia por haberme apoyado siempre en las buenas y en las malas, por sus consejos y guías para lograr mis objetivos propuestos y sobre todo por la paciencia que me han sabido brindar.

Quiero también agradecer a mi tutor de trabajo de titulación por la ayuda brindada, su orientación, conocimiento y sus valiosas sugerencias para permitirme llegar a mi objetivo planteado, De igual manera, mi agradecimiento a Cristian Guillen, por su conocimiento y gran apoyo que me supo brindar en cada momento

Por último y no menos importante también quiero agradecer a todos los catedráticos de la Universidad Católica de Cuenca y todo el personal que integran la misma por haberme brindado siempre su apoyo incondicional cuando lo necesitaba.

Jonathan Alexander Peña Peña

## RESUMEN

El departamento de Higiene y Limpieza de la Ilustre Municipalidad de Azogues es el encargado de la recolección de desechos domiciliarios. Servicio que se brinda sin interrupciones desde hace varios años atrás.

Debido al incremento del número de habitantes y la expansión propia que ha tenido la ciudad se vuelve complicado administrar correctamente datos referentes a la recolección, por ejemplo, la ciudadanía no sabe exactamente cuál es la ruta de recolección que recorre un vehículo o la ubicación de los contenedores para facilitar la evacuación de sus desechos domiciliarios. Toda esta información se mantiene archivada de manera rustica, lo cual no aporta ningún valor al departamento.

Dicha situación da como consecuencia una comunicación inexistente del servicio de recolección generando molestia en la ciudadanía. A partir de esta problemática, se decide desarrollar un Sistema de Información Web que utiliza geolocalización para mejorar los procesos que maneja el Departamento de Higiene y Limpieza enfocado en la recolección de desechos domiciliarios del Cantón Azogues.

El desarrollo de este trabajo de titulación se divide en cuatro capítulos: En el Capítulo 1 se describe el Diagnostico Situacional. En el Capítulo 2 se encuentra la Fundamentación Teórica. En el Capítulo 3 se aplica la metodología descrita en el capítulo previo para el sistema de información planteado. Finalmente, en el Capítulo 4, se desglosan conclusiones y recomendaciones relacionadas al presente trabajo de titulación.

*Palabras clave:* Sistema de información web, SCRUM, Geolocalización, Recolección de desechos domiciliarios



## **ABSTRACT**

The Department of Hygiene and Cleanliness of the Municipality of Azogues is in charge of the collection of household waste. This service has been provided without interruption for several years.

Due to the increase in the number of inhabitants and the expansion of the city, it has been complicated to handle correctly the data, regarding collection. For example, citizens do not know exactly which collection route a vehicle follows or the location of the containers to facilitate the evacuation of their household waste. All this information is kept on file in a rustic manner, which is of no value to the department.

This situation results in a non-existent communication of the collection service, generating annoyance among the citizens. Based on this problem, it was decided to develop a Web Information System that uses geolocation to improve the processes managed by the Department of Hygiene and Cleanliness focused on the collection of household waste in the Azogues Canton.

The development of this degree work is divided into four chapters: Chapter 1 describes the Situational Diagnosis. Chapter 2 contains the Theoretical Foundation. Chapter 3 applies the methodology described in the previous chapter to the proposed information system. Finally, in Chapter 4, conclusions and recommendations related to this degree work are given.

*Keywords:* Web information system, SCRUM, Geolocation, Household waste collection.

## TABLA DE CONTENIDO

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD .....	I
CERTIFICACION DEL TUTOR.....	III
DEDICATORIA .....	IV
AGRADECIMIENTOS .....	V
RESUMEN .....	VI
ABSTRACT.....	VII
CAPITULO 1 .....	13
1.1. Antecedentes .....	13
1.2. Descripción del problema.....	13
1.3. Justificación.....	14
1.4. Objetivos .....	14
Objetivo general.....	14
Objetivos específicos .....	14
1.5. Alcance.....	14
Estado del arte.....	15
1.6. Contribuciones.....	17
CAPITULO 2.....	18
2. Marco teórico .....	18
2.1. Desarrollo de la Metodología.....	18
2.2. Metodologías para el desarrollo de software.....	18
2.3. SCRUM.....	19
2.3.1. Equipos Scrum.....	19
2.3.2. Artefactos de un Scrum.....	20
2.3.3. Proceso Scrum .....	20
2.4. Geolocalización.....	22
2.4.1. Google Maps .....	23
2.5. Desarrollo de aplicaciones web.....	23
2.6. Arquitectura cliente-servidor.....	24
2.7. Editor de código .....	25
2.7.1. Visual Studio Code .....	26
2.8. Frontend .....	26

1.10.1.	Angular 11 .....	28
2.9.	Backend.....	30
2.9.1.	Node JS .....	31
2.9.2.	Mongo DB .....	33
2.10.	Git .....	34
2.11.	GitHub .....	35
CAPITULO 3.....		36
3.	Desarrollo.....	36
3.1.	Análisis de Requerimientos.....	36
3.2.	Implementación de la Metodología.....	37
3.3.	Definición de los Sprints.....	38
3.4.	Planificación de los Sprint.....	39
3.5.	Desarrollo del sistema .....	42
3.5.1.	Sprint 1 .....	42
3.5.2.	Sprint 2.....	48
3.5.3.	Sprint 3.....	54
3.5.4.	Sprint 4.....	60
3.6.	Entregables finales del proyecto .....	68
CAPITULO 4.....		70
4.	Conclusiones y Recomendaciones.....	70
4.1.	Conclusiones .....	70
4.2.	Recomendaciones.....	70
REFERENCIAS BIBLIOGRÁFICAS.....		71
Anexos 73		
	Sectores provistos del servicio de recolección de desechos en la ciudad de Azogues. ....	74
	Estructura del BackEnd.....	79
	FrontEnd .....	89

## LISTA DE FIGURAS

Fig. 1 SCRUM.....	21
Fig. 2 Modelo cliente-servidor.....	24
Fig. 3 Web Framework.....	27
Fig. 4 Arquitectura de Angular.io.....	29
Fig. 5 Servidores más usados en el 2020.....	30
Fig. 6 Base de datos más usados en el 2020.....	32
Fig. 7 Burndown inicial.....	42
Fig. 8 Burndown del Sprint 1.....	43
Fig. 9 Colecciones MongoDB Atlas.....	44
Fig. 10 Burndown del Sprint 1.....	45
Fig. 11 Control de recursos humanos.....	46
Fig. 12 Registro de Empleados.....	46
Fig. 13 Empleados registrados.....	47
Fig. 14 Burndown final del Sprint 1.....	48
Fig. 15 Burndown del Sprint 2.....	49
Fig. 16 Inventario.....	49
Fig. 17 Registro de vehículos.....	50
Fig. 18 Vehículos registrados.....	50
Fig. 19 Burndown del Sprint 2.....	51
Fig. 20 Control de rutas.....	52
Fig. 21 Registro de rutas.....	52
Fig. 22 Rutas registradas.....	53
Fig. 23 Ruta trazada en el mapa.....	53
Fig. 24 Burndown final del Sprint 2.....	54
Fig. 25 Burndown del Sprint 3.....	55
Fig. 26 Control de contenedores.....	56
Fig. 27 Registro de contenedores.....	57
Fig. 28 Contenedores registrados.....	57
Fig. 29 Burndown del Sprint 3.....	58
Fig. 30 Reporte.....	59
Fig. 31 Burndown final del Sprint 3.....	60
Fig. 32 Burndown del Sprint 4.....	61
Fig. 33 Login.....	61
Fig. 34 Burndown del Sprint 4.....	62
Fig. 35 Página publica.....	63
Fig. 36 Sectores.....	63
Fig. 37 Contenedores.....	64
Fig. 38 Burndown del Sprint 4.....	65
Fig. 39 Producción.....	65
Fig. 40 GitHub Pages.....	66
Fig. 41 Página pública.....	66
Fig. 42 Burndown final del Sprint 4.....	67

## ÍNDICE DE TABLAS

Tabla I. Metodologías para el desarrollo de software .....	18
Tabla II Editores de código.....	26
Tabla III. Framework de desarrollo web .....	27
Tabla IV Lenguajes de programación del servidor.....	30
Tabla V Bases de datos No SQL.....	32
Tabla VI Servicios de MongoDB .....	33
Tabla VII Requerimientos del Aplicativo de información web.....	36
Tabla VIII Product Backlog .....	37
Tabla IX Sprint 1 .....	38
Tabla X Sprint 2 .....	38
Tabla XI Sprint 3 .....	39
Tabla XII Sprint 4 .....	39
Tabla XIII Planificación del Sprint 1 .....	40
Tabla XIV Planificación del Sprint 2.....	40
Tabla XV Planificación del Sprint 3.....	40
Tabla XVI Planificación del Sprint 4.....	41
Tabla XVII Taskboard inicial .....	41
Tabla XVIII Taskboard del Sprint 1 .....	42
Tabla XIX Taskboard del Sprint 1 .....	45
Tabla XX Taskboard del Sprint 1 .....	47
Tabla XXI Taskboard del Sprint 2 .....	48
Tabla XXII Taskboard del Sprint 2 .....	51
Tabla XXIII Taskboard del Sprint 2 .....	54
Tabla XXIV Taskboard del Sprint 3 .....	55
Tabla XXV Taskboard del Sprint 3 .....	58
Tabla XXVI Taskboard del Sprint 3 .....	59
Tabla XXVII Taskboard del Sprint 4 .....	60
Tabla XXVIII Taskboard del Sprint 4 .....	62
Tabla XXIX Taskboard del Sprint 4 .....	64
Tabla XXX Taskboard del Sprint 4 .....	67
Tabla XXXI Informe final de Entregables.....	69

## LISTA DE ANEXOS

Anexo A Entrevista.....	73
Anexo B Distribución del servicio de recolección de desechos sólidos inorgánicos.....	75
Anexo C Sectores periféricos .....	76
Anexo D Distribución de los sectores .....	77
Anexo E Recolección contenedores.....	77
Anexo F Vehículos .....	78
Anexo G Vehículos .....	78
Anexo H Vehículos .....	79
Anexo I Servidor de Node JS .....	80
Anexo J Database.js .....	80
Anexo K Employee.js.....	81
Anexo L Vehículos.js.....	81
Anexo M User.js.....	81
Anexo N Route.js.....	82
Anexo O Container.js .....	82
Anexo P Employee.controller.js.....	83
Anexo Q Vehicle.controller.js .....	84
Anexo R User.controller.js .....	85
Anexo S Route.controller.js .....	86
Anexo T Container.controller.js .....	87
Anexo U Employee.routes.js.....	88
Anexo V Vehicle.routes.js .....	88
Anexo W User.routes.js .....	88
Anexo X Route.routes.js.....	89
Anexo Y Container.routes.js .....	89
Anexo Z Servidor Express en funcionamiento .....	89
Anexo AA admin-layout.module.ts.....	90
Anexo BB rrhh.component.ts .....	92
Anexo CC inventario.component.ts .....	94
Anexo DD users.component.ts .....	96
Anexo EE rutas.component.ts.....	98
Anexo FF container.component.ts.....	101
Anexo GG Informe Final de Tutoría de Tesis .....	102
Anexo HH Certificado de Turnitin .....	103
Anexo II Certificados de No Adeudar en Biblioteca .....	106

# **CAPITULO 1**

## **1.1.Antecedentes**

El departamento de Higiene y Limpieza fue creado en agosto de 1997 por la ilustre municipalidad de Azogues, teniendo como misión “El manejo, recolección y disposición final de los residuos”. La planta nació en la Parroquia Luis Cordero ubicada en la zona oriental de la ciudad, debido al gran crecimiento poblacional, en el año 2004 se tomó la decisión de reubicar la planta de tratamiento al sector de Chavay ubicado en la autopista Cuenca-Azogues.

Dentro de las actividades tiene como objetivo el recolectar desechos domiciliarios generados por la ciudadanía, dicho proceso se ha venido realizando en forma no técnica, por ejemplo, el manejo de inventario, gestión de rutas (vehículo y personal asignado) se sigue realizando manualmente.

Para el correcto desarrollo de las actividades referente a la recolección de residuos se cuenta con 24 trabajadores, dividiós de la siguiente manera: 7 choferes encargados del manejo de los vehículos y completar el recorrido, y 17 ayudantes encargados de depositar los residuos en el vehículo recolector.

Actualmente el cambio más importante que se ha hecho en referencia al Servicio de Recolección de Desechos Domiciliarios consta en dividir al Cantón Azogues en sectores: Sectores Central Comercial (SC1, SC2), Sector Central Residencial (SR1, SR2, SR3) y Sectores Periféricos (SP1, SP2, SP3, SP4, SP5, SP6, SP7, SP8), además en los Sectores Comerciales y Residenciales se realiza recolección de desechos orgánicos e inorgánicos.

## **1.2.Descripción del problema**

El Departamento de Higiene y Limpieza de la ilustre Municipalidad de Azogues, actualmente no dispone de un sistema de información web que apoye a la gestión de procesos, y permita comunicar a la ciudadanía sobre el servicio de recolección de desechos domiciliarios; como resultado se ha producido lo siguiente:

- Los ciudadanos provenientes de otros cantones, parroquias, sectores o cambios de domicilio desconocen las rutas y horarios de los recolectores de desechos.
- La ciudadanía desconoce la ubicación de los contenedores que existen en los diferentes sectores del Cantón Azogues.
- El supervisor o jefe del departamento encargado debe realizar los procesos de gestión manualmente.
- La información de rutas, horarios y contenedores, se guarda en hojas de papel y las mismas son almacenadas en folder o carpetas en un archivo físico.

### **1.3.Justificación**

El problema del manejo de información para la recolección de desechos domiciliarios en el cantón Azogues se da al no contar con un sistema que automatice estos procesos, con el desarrollo de AzoRecolecta se conseguirá solucionar mediante un aplicativo web que agilite la organización, búsqueda y gestión de procesos relacionados a este servicio. Obteniendo como principales beneficiados: la ciudadanía, el personal, y sirviendo de apoyo al jefe departamental permitiéndole obtener reportes detallados, además como se puede observar por el momento no se ha elaborado un aplicativo del tema planteado por lo que será un buen comienzo de la idea a ilustrar.

El presente trabajo surge de la necesidad de contar con un Sistema de Información Web que se acople a las necesidades que se han generado con el paso del tiempo, de igual manera proporcionar un medio eficaz y confiable para informar a la ciudadanía sobre ciertos aspectos puntuales (rutas, horarios y contenedores) del servicio de recolección brindado por este departamento.

### **1.4.Objetivos**

#### **Objetivo general**

Desarrollar un Sistema de Información Web, utilizando metodologías de desarrollo de software ágiles, para automatizar los procesos de gestión del Departamento de Higiene y Limpieza del Cantón Azogues referente a la recolección de desechos domiciliarios.

#### **Objetivos específicos**

- Desarrollar el marco teórico de la problemática propuesta, revisando conceptos y tecnologías usadas actualmente, para demostrar lo que se quiere construir.
- Levantar los requerimientos del sistema, recopilando información necesaria por medio del Departamento de Higiene y Limpieza referente a las rutas y horarios de recolección de desechos domiciliarios a través de una entrevista, para describir los procesos del sistema.
- Construir el prototipo del sistema de información web, cumpliendo todas las fases de la metodología de desarrollo ágil, para generar un software que se acople a las necesidades.

### **1.5.Alcance**

Se plantea realizar un sistema de información web para el Departamento de Higiene y Limpieza del GAD municipal del Cantón Azogues, el cual se dividirá en dos partes; la primera siendo exclusivamente para a gestión de información (inventario, reportes, recursos humanos, contenedores, rutas y horarios), la segunda servirá para mantener informada a la ciudadanía sobre el servicio de recolección de desechos domiciliarios, este sistema estará disponible desde cualquier lugar mientras se disponga de un dispositivo inteligente que tenga acceso a internet y permita ejecutar un navegador web.



La implementación del sistema de información web se realizará en un periodo de prueba, utilizando un host y un dominio (gratuito) que permita demostrar el normal funcionamiento del sistema, cumpliendo con los requerimientos establecidos por el Departamento de Higiene y Limpieza.

## **Estado del arte**

En el año 2012, en el municipio de la ciudad de Madrid se plantea estrategias para optimizar el proceso de recolección de residuos sólidos urbanos, mejorando tiempos de movilidad de los vehículos de recolección. Debido a que la problemática es que no se cuenta con un sistema de recolección lo que se pretende es transformar el actual sistema en una verdadera gestión integral de residuos sólidos considerando los componentes que son las rutas, vehículos y zonas de recolección, para solucionar esta problemática se analizó el estado actual de la recolección, los vehículos, sus capacidades, la distancia que existe entre el municipio y el relleno sanitario para lo cual se formularan planes para optimizar el proceso de recolección de residuos sólidos urbanos en las zonas de difícil acceso, además mejorando las rutas actuales y reduciendo los tiempos de recolección.[1]

En el año 2013, se realizó un estudio con la finalidad de localizar la cantidad de contenedores sobre un conjunto de nodos candidatos, que sean capaces de cubrir la demanda total de basura generada por los clientes, de tal manera que cada contenedor sea asignado a la persona más cercana y a su vez determinar la ruta óptima de cada uno de los contenedores, para resolver esta problemática se tuvo que considerar restricciones de capacidad tanto de los camiones como de los contenedores, en definitiva el modelo propuesto permite minimizar los costes totales del sistema de recolección de residuos domiciliarios, utilizando herramientas de programación y geolocalización, donde los resultados ayudaron a planificar las rutas que los camiones deben seguir para recolectar los desechos de estos contenedores en lugar de pasar por cada una de sus casas ya que su objetivo es minimizar el costo del transporte y también minimizar la distancia que deben caminar los usuarios para llegar al contenedor.[2]

En el año 2015, en la ciudad de Cuenca la empresa EMAC EP no cuenta con un diseño técnico de rutas de recolección y transporte de recolección de residuos sólidos debido a que se realizó un sistema de forma empírica, motivo por el cual el servicio carece de un estudio técnico, para solucionar esta problemática se utilizaron herramientas SIG, puesto que su objetivo principal era mejorar el aspecto financiero, como solución a este problema se rediseñaron las rutas de recolección, lo cual significó la reducción de costos operativos de recolección y también acortar los viajes que realizan los camiones desde su salida hasta su llegada al relleno sanitario.[3]

En el año 2015, en la municipalidad de Esmeraldas se realizó un estudio de la recolección actual de residuos sólidos urbanos, esta propuesta se enfoca en la búsqueda de la cantidad adecuada de depósitos (contenedores), su ubicación actual utilizando herramientas LINGO, Sistema de Información Geográfica como Google Maps, y Waze, ya que la propuesta se enfoca en resolver la

ubicación actual de los contenedores, en la cual como resultado se obtuvo un alto porcentaje de cobertura en lo que fue la asignación de los contenedores en lugares estratégicos.[4]

En el año 2017 Icaza Alvares en su trabajo "Sistema de seguridad ciudadana por geolocalización y georreferenciación para zonas rurales del Cantón Cuenca incorporados el SIS ECU-911", diseña un sistema de seguridad apoyados por los modernos sistemas de geolocalización y georreferenciación que dispone el centro integrado del ECU-911, el sistema se amplía a las zonas rurales del Cantón Cuenca, además se considera el botón de auxilio para las camionetas que están legalmente facultadas para prestar los servicios en las zonas urbanas. A estos sistemas se incorporara cámaras de monitoreo desde el centro integrado del ECU-911 al igual que receptor llamadas de auxilio y accionamiento de botones de pánico desde sitios estratégicos para ubicar a los vehículos se utilizara el subsistema GPS y herramientas SIG donde se pueda visualizar en el mapa el vehículo más cercano y la ruta más óptima para enviar a los recursos, bomberos, cruz roja...como resultado el sistema de telecomunicaciones propuesto en las zonas rurales se acopla al ya existente ya que los SIG son de apoyo bastante importante en las plataformas de seguridad donde la información debe ser veras para la propuesta oportuna e identificar las coordenadas exactas y que el sistema funcione de manera correcta y eficaz.[5]

En el año 2017, se realizó una propuesta de rutas óptimas para la recolección de desechos sólidos en la zona centro norte de la Parroquia Sangolquí, mediante la extensión network analyst del software ArcGIS, para ello se realizó una investigación de campo con la ayuda de herramientas de los sistemas de información, se pueden generar bases de datos con información georreferenciada y así determinar las rutas óptimas para la recolección, permitiendo así que los camiones de recolección no ocasionen tráfico ni malestar a la población, para solucionar esta problemática se utilizó las herramientas ArcGIS con la finalidad de controlar el movimiento de los vehículos ahorrando gastos y ayudar a la planificación de las rutas.[6]

En el año 2018, en el distrito de San Jerónimo, no existen estrategias eficientes que optimicen sus recursos ni implantan políticas tecnológicas (TIC) que ayuden a mejorar el proceso de recolección, por otro lado, los municipios y la ciudadanía tienen la responsabilidad que este servicio sea más eficiente, ya que requiere un trabajo en conjunto donde cada uno tenga una participación activa, actualmente la ciudadanía por las múltiples actividades que realizan en las horas de la mañana olvidan evacuar sus desechos en las horas y días establecidos, olvidando también estar pendientes del vehículo recolector de residuos. Por lo que para solucionar esta investigación se plantea desarrollar un sistema de geolocalización de alerta de recojo de residuos sólidos, las herramientas que se utilizó son raspBerry Ip y geolocalización. El sistema que se desarrolló será capaz de mostrar en un mapa el recorrido en tiempo real del recolector de residuos como también notificar cuando el vehículo este acercándose al domicilio del ciudadano que esté utilizando la aplicación.[7]

En el año 2019, se realizó un estudio en el que el Cantón Naranjal se enfrenta a la ineficiencia de la recolección de los residuos sólidos debido al mal manejo de los recursos que dispone, las rutas que siguen los recolectores no tienen un estudio previo tanto de los días y horas fijas de recolección, para la solución se realizó en levantamiento de rutas actuales con la ayuda de una aplicación Android ORUXMAPS, y ARCMAP para poder realizar la simulación, una vez que se generó la información base para la red de transporte de la zona urbana del Cantón, se estableció la demanda del servicio de recolección de residuos sólidos para determinar la propuesta de mejora, se contemplaron tres escenarios de simulación para los recorridos de los camiones, tomando en cuenta la capacidad de cada vehículo, como resultado se obtuvo que al realizar la simulación con VehicleRoute se obtuvo un modelo de rutas mejoradas para brindar un mejor servicio.[8]

En el año 2019, en la ciudad de Ibarra no cuentan con un sistema que ayude a la optimización de rutas para la recolección de desechos sólidos ni un medio para determinar capacidad, peso y volumen de los contenedores que se encuentran ubicados en la ciudad. Para resolver esta problemática se utilizó herramientas libres como Framework y Base de Datos PostgreSQL con la metodología XP y como lenguaje de programación se utilizó PHP Y LARAVEL, el objetivo es realizar un diseño e implementación de los módulos de registro de los usuarios, registro de rutas, contenedores y dar reportes detallados de cada uno de los contenedores y vehículos de recolección, y como solución a esta problemática es obtener un mapa temático que permita identificar con claridad cada uno de los elementos geográficos como son los contenedores, predios y vías, a través del modelo de ruteo vehicular permitirá determinar el conjunto de rutas de los vehículos de recolección, mejorando de esta manera la organización de este servicio.[9]

En el desarrollo del estado del arte se plantean diferentes problemáticas referentes a los recorridos de recolección de desechos que se realiza, al no tener una correcta planificación de rutas se genera un gran impacto económico (combustible, mantenimiento de los vehículos ...) sobre las empresas que brindan este servicio. Al aplicar un sistema de información que utilice geolocalización y permita comunicar a la ciudadanía también se soluciona la problemática de desinformación.

## **1.6.Contribuciones**

En el presente trabajo de titulación, se da a conocer una alternativa para la gestión de datos respecto a la recolección de desechos domiciliarios. De tal manera que los usuarios finales puedan conocer exactamente todas las diferentes vías que tienen para evacuar sus desechos, consultando en la página web las diferentes rutas de recolección, así mismo la ubicación más cercana de los contenedores. Además, que sirva para el apoyo óptimo en la toma de decisiones y control de rutas por parte de la gerencia.

## CAPITULO 2

### 2. Marco teórico

En el desarrollo del estado del arte, se obtuvo una visión general sobre el uso de herramientas de programación, almacenamiento de datos y geolocalización, por lo que, en el presente capítulo se determinará cada una de las tecnologías que se usarán para el desarrollo del proyecto, toda la información recopilada será de gran ayuda para que los lectores tengan un mayor entendimiento de las áreas tales como: Entornos de desarrollo web y una de las áreas más importantes para el almacenamiento de datos, metodología de desarrollo, y el uso de servicios para el alojamiento de repositorios de software gestionados por un sistema de control de versiones.

#### 2.1.Desarrollo de la Metodología

Para el desarrollo del presente trabajo se ha planteado una investigación descriptiva; en donde detallamos todo el proceso de construcción del prototipo del sistema de información web “AzoRecolecta”. La información que detalla la situación actual, se obtendrá a través de las técnicas de la observación y la entrevista no estructurada y libre para obtener información relacionada al proceso de recolección de desechos domiciliarios, los datos proporcionados por el jefe departamental, con datos cuantitativos, precisos y fidedignos.

El diseño experimental, para la construcción del prototipo, empleará la metodología ágil en el desarrollo del software.

#### 2.2.Metodologías para el desarrollo de software

Hoy en día existen diversas alternativas a la hora de elegir una estrategia de trabajo para desarrollar software, la estructura, proceso, principios, ciclo de vida, artefactos y roles entre otras características que diferencian cada una de estas metodologías. Sin embargo, algunas de estas tienen un enfoque iterativo para la especificación, el desarrollo y la entrega del producto final, en cuanto a la captura de requerimientos y el proceso de desarrollo del sistema también tienen diferencias que son notables, debido a lo antes mencionado a continuación nos enfocamos en dos específicas:

Tabla I. Metodologías para el desarrollo de software

	<i><b>SCRUM</b></i>	<i><b>RUP</b></i>
<i><b>Roles</b></i>	Si	Si
<i><b>Enfoque iterativo</b></i>	Si	Si
<i><b>Ciclo</b></i>	Sprint	4 fases
<i><b>Alcance</b></i>	Se evalúa al final de cada iteración (Sprint).	Definido al comienzo del Proyecto.

<b>Entregables</b>	Software operativo	Documento de alcance, visión, caso de negocio, lista de riesgos, Plan de desarrollo, plan de iteraciones, lista de principales casos de uso.
<b>Producto</b>	Requisitos cambiantes o mejoras rápidas.	Proyectos de largo alcance y tiempo.
<b>Conclusión</b>	Metodología que se adapta a las necesidades cambiantes del cliente, en base a revisiones periódicas.	Metodología utilizada para trabajo en equipo, puesto que si cambian las necesidades del cliente se debe reformular desde un inicio.

Fuente: Autor

Analizando la comparativa de la Tabla I se ha elegido la metodología SCRUM, ya que logra de manera exitosa la integración de todas las partes involucradas para el desarrollo de un proyecto, debido a las revisiones periódicas permite acoplarse a los cambios en la construcción de la solución. Por lo antes señalado hemos decidido desarrollar la propuesta de tesis basándonos en esta metodología.

## 2.3.SCRUM

Scrum permite el desarrollo de sistemas web y móviles a través de un entorno funcional, colaborativo, flexible y adaptable al cambio, está basado en entregas parciales y regulares del producto final. Esta metodología funciona de la siguiente manera.

### 2.3.1. Equipos Scrum

El equipo Scrum está formado por tres roles:

- **Producto Owner o Dueño del Producto:**

Es la persona que toma las decisiones y es la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog.[10]

- **Scrum Master:**

Es el encargado de comprobar que el modelo y la metodología funciona. Eliminará todos los inconvenientes que impidan que el proceso fluya, e interactuara con el cliente y con los gestores.[10]

- **Equipo de Desarrollo:**

Suele ser un equipo pequeño de unas 5 a 9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del Backlog.[10]

Un equipo de desarrollo debe ser multifuncional: es decir debe tener la capacidad de realizar cada uno de los elementos del Backlog.

### 2.3.2. Artefactos de un Scrum

Los artefactos de un Scrum permiten al equipo encargado del proyecto tener la misma visión del proyecto:

- **Backlog de Producto / Product Backlog:** Lista las necesidades del cliente, es decir, los requisitos que tendrá el producto o los que irán adquiriendo en sucesivas iteraciones.[10]
- **Backlog del Sprint/Sprint Backlog:** Lista de las tareas y planificaciones que se realizan en un Sprint. Se asignan las tareas a cada persona y el tiempo estimado para terminirlas.[10]
- **Incremento:** Parte añadida o desarrollada en un sprint, es una parte terminada y totalmente operativa. Dependiendo de los resultados que se vayan obteniendo el cliente puede ir haciendo los cambios necesarios del proyecto.[10]

### 2.3.3. Proceso Scrum

Todo proyecto que se elabora mediante SCRUM se ejecuta por medio de ciclos repetitivos con duración fija (iteraciones que varía entre 2 y 4 semanas, siendo la ultima el tiempo máximo para un feedback<sup>1</sup> de producto real). Cada iteración debe ser entregada con un resultado completo, el mismo que agrega valor al producto final.

A continuación, se puede observar en la Figura 1, como se manejan los procesos Scrum:

---

<sup>1</sup> Feedback: Retroalimentación



Fig. 1 SCRUM.

Fuente: <https://bimer.cl/metodologia-scrum-ver-los-beneficios-fundamentos-y-requisitos/>

## • Planificación del Spring

Definición del Product Backlog. Se divide en dos secciones.

- **Selección de requisitos:** El cliente muestra una lista de requerimientos del producto. Se interroga al cliente sobre dudas y cuáles son los requisitos importantes para cada iteración, de esta manera se garantiza la entrega en caso de ser solicitada.
- **Planificación de la iteración:** En base a los requisitos seleccionados se genera una nómina de tareas con el fin de dividir las cargas de trabajo para garantizar la resolución de los objetivos.

En caso de tener tareas muy extensas o complejas es posible organizar grupos de trabajo con tal de garantizar su resolución.

## • Ejecución del Sprint

Se realiza reuniones conjuntas entre todos los integrantes del equipo de trabajo por un máximo de 15 minutos, normalmente apoyándose de una pizarra o tablero físico. Estas reuniones se realizan con la finalidad de evaluar, sugerir y acoplar los cambios que permitan cumplir con los objetivos planteados. En cada sesión los integrantes del equipo deben responder a las siguientes preguntas:

- ¿Qué se ha terminado desde la última reunión?

- ¿Qué se va a realizar a partir de ese instante?
- ¿Cuáles son las dificultades que no permiten cumplir los objetivos?

- **Revisión del Sprint o Reunión Demo**

Se realiza al final de cada iteración o Sprint en un espacio de tiempo de 4 horas para un sprint de un mes, en dicha reunión solo se muestran las historias terminadas, esto sirve para actualizar el Product Backlog. Desde la revisión del primer sprint el proyecto puede ser re planificado.

### ¿Por qué utilizar Scrum?

Algunas de las principales ventajas de utilizar Scrum:

- **Adaptabilidad:** Entrega iterativa hace que los proyectos sean adaptables y abiertos a cambios.
- **Mejora continua:** Los proyectos se mejoran progresivamente Sprint por Sprint, siempre manteniendo prioridad de los sprint pendientes.
- **Entregables efectivos:** La lista de pendientes del producto y las revisiones periódicas aseguran entregas efectivas para el cliente.

## 2.4.Geolocalización

La geolocalización nos permite conocer nuestra ubicación en el espacio utilizando un objeto que esté conectado a internet, por ejemplo, una computadora, o nuestro celular a través del GPS integrado. También podemos mencionar el uso de la geolocalización en las redes sociales, como por ejemplo en Facebook cuando compartimos la ubicación actual del lugar en que nos encontramos, otro uso significativo se da cuando utilizamos Google Maps, al permitir a esta aplicación acceder al GPS en nuestros dispositivos nos muestra nuestra posición referente a algún punto de interés, sin embargo no es su única funcionalidad, también nos brinda información geolocalizada (rutas más corta, distancia, tiempo...) en un mapa.

Características:

- Un dispositivo hardware, que actuara como plataforma en la que se desarrollara el proceso de geolocalización.[11]
- Un programa de software que ejecutara el proceso de geolocalización según la implementación.[11]
- Una conexión a internet que actuara como medio de obtención, interpretación e intercambio de información geográfica.[11]
- Localizar: conocer el lugar donde se encuentra una persona u objeto en el mapa.[11]



### 2.4.1. Google Maps

El servicio de mapas de Google nos brinda la opción visualizar imágenes de mapas desplazables, de la misma manera nos ofrece fotografías de los lugares o puntos considerados de interés para su previsualización, y de esta manera tener una idea más clara de lo que se puede visitar, además de todo esto Google Maps nos permite interactuar con nuestro GPS (En un dispositivo inteligente) de una manera intuitiva y así poder llegar a un punto en específico o establecer un recorrido para entre otras cosas por ejemplo hacer turismo.

- Direccctions API

Para el desarrollo del presente trabajo se utilizará esta API, este servicio permite trazar la ruta entre los puntos de interés que se le indique. Esta ruta trazada nos permitirá mostrar cual es el recorrido que debe realizar cada uno de los vehículos recolectores.

### 2.5.Desarrollo de aplicaciones web

Web e internet son términos que a menudo se confunden y se tiende a creer que son los mismos, internet es una gran red mundial de ordenadores conectados entre sí, que nos permiten comunicarnos con personas ubicadas en distintos lugares del mundo. La web fue creada en 1989 por Tim Berners Lee, la misma que consistía en una forma de conectar la información de internet con un medio físico, a través del protocolo HTTP<sup>2</sup> (*Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto*) en los navegadores, y se utiliza para realizar peticiones a los servidores web y recibir respuestas de ellos; desde su origen este protocolo ha sufrido cambios que si bien son minúsculos al día de hoy nos garantiza mayor seguridad (*HTTPS*) y mejora nuestra velocidad de navegación. Los elementos indispensables en internet para poder tener conexión a una página web desde cualquier parte del mundo son los siguientes:

- Servidor web: Es un servidor que está preparado para servir paginas las 24 horas al día. Se está ejecutando continuamente y atiende a las peticiones que hacen los clientes desde los navegadores. [12]
- Dominio: Es un sitio web al que queremos acceder, se introduce su nombre en la barra de direcciones del navegador. [12]
- Servidor DNS: Es el encargado de transformar el nombre del dominio de un servidor web en la dirección IP.[12]

También podemos mencionar algunas características de la web que vamos a utilizar para el desarrollo de esta propuesta:

- Contenido accesible a través de múltiples dispositivos inteligentes. [12]

---

<sup>2</sup> HTTP. – Protocolo utilizado para comunicar con el servidor web, para acceder a una página web

- La web geoespacial: es la que combina la información geográfica disponible de los usuarios, con información abstracta que predomina en la web, al usar dispositivos GPS se genera conceptos que permiten realizar búsquedas u ofertar servicios en base a la localización, gracias a ello se logra ubicar una tienda, el lugar donde se toma una foto o buscar la ruta más óptima para llegar a su destino. [12]

Al hablar de una aplicación web, esta tiene un comportamiento casi igual al de una página web con todas las tecnologías implícitas en la misma, las cuales son las siguientes:

- HTML: lenguaje de etiquetado, encargado de mostrar en el navegador web la información de la aplicación, nos permite interactuar con las diferentes funciones programadas.
- CSS: lenguaje de estilos en cascada, es el encargado de la representación visual del sitio web. Complemento perfecto para HTML.
- JavaScript: lenguaje de programación destinado a la ejecución de acciones en un entorno web. Es un lenguaje no compilado, funciona del lado cliente donde los navegadores web son los encargados de interpretar su funcionalidad. No es obligatorio su uso, ya que existen diferentes lenguajes compilados, pero normalmente compilan su código a JavaScript.

## 2.6.Arquitectura cliente-servidor

Una arquitectura cliente- servidor funciona únicamente con un computador central que maneja las peticiones de los clientes hacía el servidor, otorgando una respuesta a cada una de las solicitudes. Este equipo servidor recibe las solicitudes HTTP por parte de los clientes, procesa los datos recibidos y envía una respuesta de vuelta hacia el cliente (usando funciones preprogramadas en el servidor), este proceso no le toma más que una fracción de segundo, incluso con varios clientes al mismo tiempo mientras la aplicación se esté ejecutando.

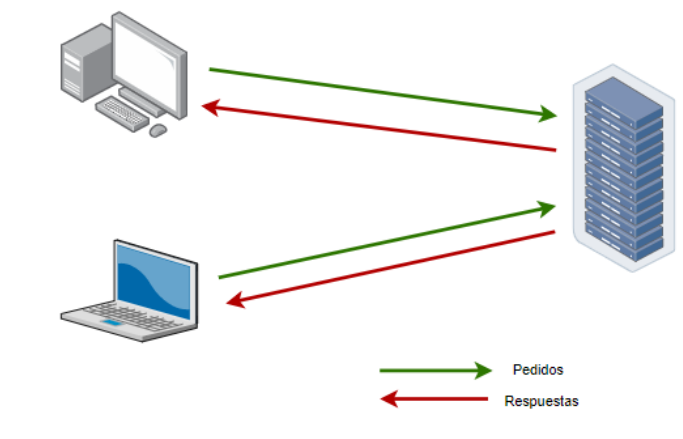


Fig. 2 Modelo cliente-servidor.

Fuente: Autor

Este modelo de arquitectura es el más popular en la región, puesto que permite centralizar la información en un solo equipo, de esta manera los clientes se ahorran coste computacional en la ejecución de los procesos, puesto que los mismos se ejecutan en el servidor. Esta arquitectura nos permite distribuir las funciones acordes al servicio que se va a ejecutar. Adicionalmente, este tipo de arquitectura presenta ciertas ventajas:

- Segmentar procesos de una red de servidores, evitando sobrecargar a uno solo.
- Escalabilidad de la aplicación, permitiendo una escalabilidad tanto horizontal como vertical, ya sea mediante el agregado de nuevos equipos, clientes o pudiendo migrar a servidores más potentes y de mayor velocidad respectivamente.
- Acceso a los datos, sin importar la ubicación de los usuarios.
- El cliente, así como el servidor pueden existir en una misma máquina.

## **2.7.Editor de código**

Actualmente disponemos de muchos editores de código, partiendo desde los editores de pago, gratuitos de escritorio y online. Estas herramientas nos ayudan a facilitar el trabajo y aumentar productividad, ya que, si bien podemos escribir código desde un archivo de texto con el bloc de notas, estos editores nos ayudan marcando posibles errores referencias inconclusas ... además que estos nos ayudan a entender mejor el lenguaje de programación en el que estamos trabajando.

Al momento de hablar de editores no debemos confundirlos con los IDE<sup>3</sup> (Entorno de Desarrollo Integrado) ya que los editores son programas ligeros, que nos ofrecen solo lo necesario para ser productivos y tener una experiencia de desarrollo acorde a los requerimientos, eso sí evitando complicaciones.

A continuación, detallamos características de entre los más populares que se han considerado previo al desarrollo del software.

---

<sup>3</sup> IDE: Entorno de Desarrollo Integrado

Tabla II Editores de código

	VSC (Visual Studio Code)	WebStrom	ATOM
Open source	Si	No	Si
Multiplataforma	Si	Si	Si
Documentación	Si	Si	Si
Terminal Integrado	Si	Si	No
Lenguajes de extensión (JavaScript, TypeScript)	Si	Si	Si (media)
Compatibilidad con otros lenguajes de programación	Si	Si	Si
Total	6	5	5

Fuente: Autor

Con los resultados obtenidos en la tabla II se decide utilizar a visual studio code como editor de código para el desarrollo de la solución.

### 2.7.1. Visual Studio Code

Visual Studio Code es un editor de código fuente, desarrollado por la empresa Microsoft, puede trabajar en cualquier plataforma existente en la actualidad (MacOS, Windows, Linux) está escrito en TypeScript, JavaScript y hoja de estilos en cascada, es un editor relativamente ligero y muy potente permitiendo ejecutar en el escritorio, viene con soporte incorporado para JavaScript, TypeScript y Node JS. Además, integra un gran soporte (nativo o con extensiones) para muchos lenguajes C++, C#, Java, Python, PHP... para poder ampliar su funcionalidad sin que pierda ligereza que lo caracteriza, cuenta con soporte para bibliotecas JavaScript como Angular, Vue js o React.

### 2.8.Frontend

El Frontend o GUI (Graphical User Interface, Interfaz Gráfica del Usuario en español), es la parte del desarrollo web que se dedica al diseño de sitios web, desde la estructura del mismo hasta los colores, fondos, animaciones y efectos.

Para el desarrollo de aplicaciones Frontend se utiliza lenguajes del lado del cliente como HTML, CSS para darle la estructura al sitio web, y JavaScript para darle dinamismo al mismo, mediante el uso de algunos framework o librerías que expanden sus capacidades para desarrollar cualquier tipo de interfaces de usuarios.

Un framework de aplicaciones web es un tipo de framework (React, Vue, Redux, Angular, ionic, ...) que permite desarrollo de sitios web dinámicos, como los webs services y aplicaciones web que además promueven la reutilización de código, conectividad de componentes, y la

implementación de bibliotecas para el acceso a base de datos. El patrón más conocido para el diseño de aplicaciones web es la arquitectura Model- View-Controller MVC (Modelo Vista Controlador).

Tabla III. Framework de desarrollo web

	Angular	React JS	Vue JS
<b>Desarrollado por:</b>	Google	Facebook	Comunidad
<b>Open Source</b>	Si	Si	Si
<b>Código reutilizable</b>	Si	Solo CSS	CSS y HTML
<b>Sencillez de aprendizaje</b>	Si (TypeScript)	Medio	Si
<b>Documentación</b>	Si	Si	Si
<b>Enlace de datos</b>	Bidireccional <sup>4</sup>	Unidireccional	Bidireccional
<b>Modelo de desarrollo</b>	MVC <sup>5</sup>	DOM virtual	DOM virtual
<b>Empresas</b>	Netflix, Google	Facebook, WhatsApp	Alibaba, GitLab
<b>Total</b>	4	2	3

Fuente: Autor

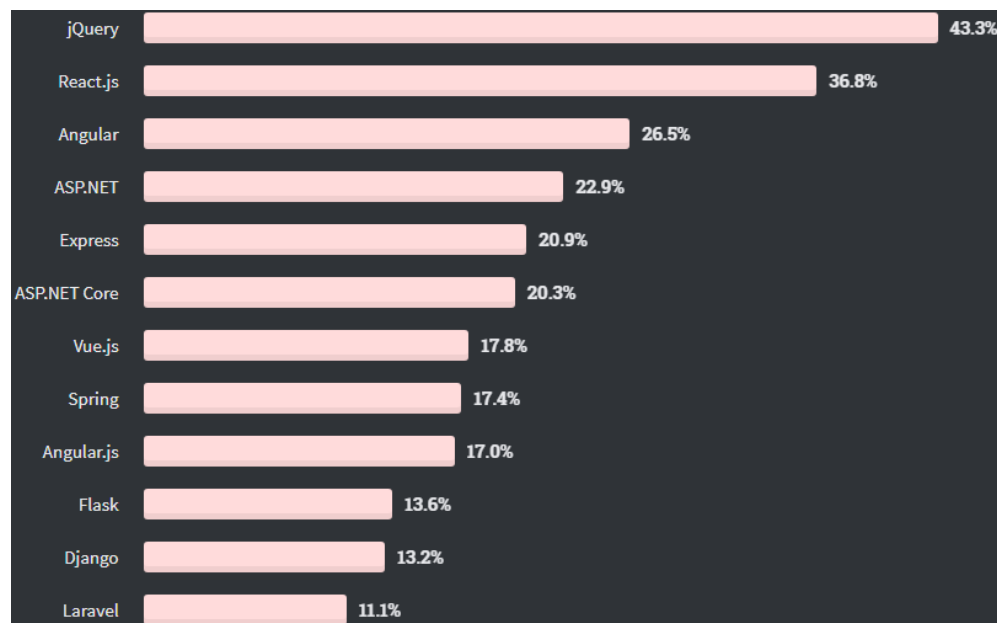


Fig. 3 Web Framework

Fuente: <https://insights.stackoverflow.com/survey/2020#technology>

<sup>4</sup> Bidireccional. - Si modificamos el valor de una propiedad en la vista el valor también se actualiza en el modelo de datos.

<sup>5</sup> MVC. – Modelo vista controlador

En la figura 3, se representa el porcentaje de la encuesta anual realizada por StackOverflow a profesionales y aficionados que presentan un interés por el desarrollo web, en la cual se observa que jQuery sigue estando es los primeros lugares, pero va perdiendo terreno frente a React.js y Angular, por lo que al ubicar a Angular en el tercer puesto de la lista indica lo importancia de trabajar con el mismo, de esta manera angular tiene documentación suficientemente extensa para solventar cualquier tipo de duda o inquietud que suceda en el desarrollo

En base a los resultados de la tabla II y apoyados en la Fig. 3, se decidido utilizar Angular como framework de desarrollo web para el desarrollo de la propuesta.

### 1.10.1. Angular 11

Se conoce como Angular 11, a todas las versiones existentes que dan continuidad a la primera versión de Angular, conocida como Angular JS, en la actualidad siendo la versión más estable la 11. Angular está escrito en TypeScript<sup>6</sup> y es desarrollado por la gigante Google, destinada al desarrollo de aplicaciones web y móviles. Definiendo Angular es una plataforma y un marco para crear aplicaciones basado en componentes permitiendo la reutilización de los mismos en varios lugares del sitio Web, de esta manera la aplicación se comporta en una sola página (SPA<sup>7</sup>), brindando una mayor eficiencia y escalabilidad en el desarrollo de una manera más sencilla. Para empezar a trabajar con el mismo no se necesita más conocimientos intermedios en cuanto al uso de las herramientas web comunes como HTML, CSS y JavaScript.[13]

**SPA:** Single page application o en español Aplicación de una sola página, significa que la página web completa se carga al inicio, y luego las sucesivas actualizaciones que se producen como consecuencia de la interacción de los usuarios con el navegador son realizadas sin la necesidad de recargar la página completa, de esta manera ahorrando coste computacional y de red para las diferentes operaciones que se puedan ejecutar.

Angular CLI (ng), es una interface de línea de comandos que se utiliza para:

- Crear nuevos proyectos.
- Mantener una aplicación ya creada.
- Compilar, ejecutar y probar.
- Generar un paquete que luego se utilizara para publicar la aplicación en un ambiente productivo.

---

<sup>6</sup> TypeScript. - lenguaje de desarrollo libre y de código abierto, superconjunto de JavaScript, desarrollado por Microsoft orientado al desarrollo web.

<sup>7</sup> SPA. - Single Page Application

## Arquitectura de angular

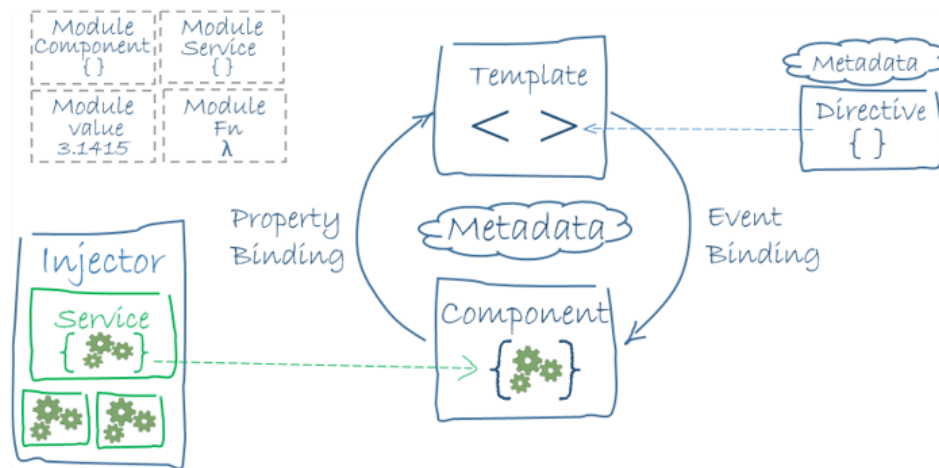


Fig. 4 Arquitectura de Angular.io

Fuente: <https://angular.io/generated/images/guide/architecture/overview2.png>

Angular está diseñado con una arquitectura orientada a módulos y componentes.

- Módulos: estos proporcionan un contexto o dominio de la aplicación para los componentes y servicios que contiene..[13]
- Componentes: este representa una porción de la aplicación o página y esté contenido dentro de un módulo, está dividido en la parte lógica (código) y la vista (HTML/CSS) .[13]
- Servicios: este es un componente lógico de código reutilizable y su principal objetivo es proveer funcionalidad extra a un componente.[13]
- Directiva: permite añadir, manipular, o eliminar los elementos del DOM del HTML.[13]
- Inyección de dependencias: son los servicios que una clase necesita para ejecutar una determinada función.[13]

Angular como framework de desarrollo web, gracias a la compatibilidad con varias de las tecnologías existentes, permite alcanzar perfectamente lo que se desea desarrollar, por las siguientes características:

- Es universal, con compatibilidad con las tecnologías de servidor más comunes como PHP, Node.js y .NET.
- Desarrollo basado en componentes, permitiendo cargar solo los componentes que queremos mostrar, ahorrando recursos, tiempos de carga y agilizando el desarrollo.
- Actualizaciones constantes por parte del equipo de trabajo.

## 2.9.Backend

Se encuentra del lado del servidor, es decir nos referimos a la arquitectura interna del sitio que asegura que todos los elementos desarrollen la función correcta. No está visible a ojos del usuario y no incluye ningún tipo de elemento gráfico. Esta área es la encargada, además de la funcionalidad del sitio, de la seguridad y la optimización de los recursos de la aplicación que se ejecuta fuera del dispositivo del cliente (servidor físico, la nube). Los lenguajes de programación para servidores como ASP.NET, Ruby, PHP, JavaScript utilizado en el entorno Node.js con la estructura Express, Python, C#, java.

Tabla IV Lenguajes de programación del servidor

	Node Js	Deno	.NET
<b>Documentación</b>	Si	Si	Si
<b>Multiplataforma</b>	Si	Si	Si
<b>Es escalable</b>	Si		Si
<b>Código reutilizable</b>	Si		
<b>Asincrónico</b>	Si	Si	Si
<b>Módulos</b>	Si		Si
<b>Formato Json</b>	Si		Si
<b>Total</b>	7	3	6

Fuente: Autor

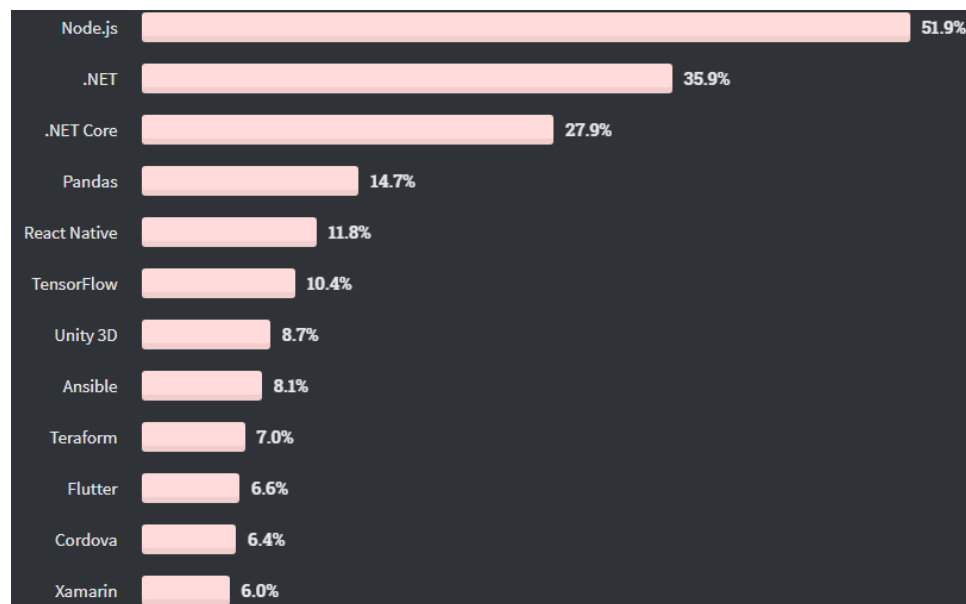


Fig. 5 Servidores más usados en el 2020

Fuente: <https://insights.stackoverflow.com/survey/2020#technology>



En la Figura 4 se puede visualizar los porcentajes de la encuesta realizada por StackOverflow, por lo que al ubicar a Node JS en el primer puesto de la lista esto indica la importancia de trabajar con el mismo.

Apoyados en la tabla III y la figura 5 se ha decidido utilizar un servidor basado en node js.

### 2.9.1. Node JS

Se conoce a Node JS como un entorno de ejecución que puede ejecutarse en Mac OS X, Windows y Linux, es de código abierto orientado a eventos asincrónicos. El objetivo de Node js es crear aplicaciones web o aplicaciones real time de una manera que soporte concurrencias muy altas, es decir ejecutar dos o más procesos simultáneamente. Actualmente tenemos la versión de Node JS v15.6.0.[14]

Está basado en el motor V8 este es el motor de ejecución de JavaScript que se creó inicialmente para Google Chrome, también podemos decir que V8 compila el código fuente de JavaScript al código de maquina nativo en tiempo de ejecución. Además, cuenta con el gestor de paquetes preinstalado para la plataforma de Node JS. Admite datos JSON como algunas bases de datos (Postgres, mongo DB...).

Ventajas de Node JS:

- Permite compartir código entre el navegador y el servidor.
- Menor tiempo de ejecución.
- Facilidad de hacer test unitarios.
- Escalabilidad: esto es importante para proyectos de medio y gran tamaño.
- Node JS es muy fácil de integrar con HTTP, lo que lo hace perfecto para montar aplicaciones web que tengan mucha entrada/salida.

**NPM:** (Nodo Package Manager) es un gestor de paquetes, que viene integrada para instalar y administrar los módulos de Node que se estén utilizando, ya que podemos decir que npm es una forma fácil de expandir la funcionalidad de Node sin tener que preocuparse en desconfigurar la instalación.

Algunos módulos npm más populares de hoy en día:

- Express: Es una infraestructura web rápida, minimalista y flexible para las aplicaciones de Node.js. En general se prefiere Express a Express.js aunque también es usado. Express está inspirado en el framework de desarrollo web para Node.js y estándar de facto para la mayoría de aplicaciones Node.js de hoy en día.
- Mongoose: es utilizado para proporcionar la API para la base de datos de objetos MongoDB en Node.js.

- Socket.io: Es un componente del servidor de los dos componentes de Websockets más comunes en la actualidad.

Sin embargo, debemos tener en cuenta que del lado del servidor se interactúa con la base de datos, ya que toda aplicación web debe almacenar datos de alguna manera para los cual se ha elegido trabajar con una base de datos No SQL.

Tabla V Bases de datos No SQL

	<b>Mongo DB</b>	<b>Firebase</b>
<b>Desarrollado</b>	Mongo db	Google
<b>Rendimiento</b>	Alto	Medio
<b>Seguridad</b>	Alta	Media
<b>Aplicaciones</b>	Aplicaciones de gran escala	Aplicaciones de pequeña escala
<b>Empresas que utilizan</b>	Adobe, Sega, eBay	Duolingo, Alibaba, Shazam

Fuente: Autor

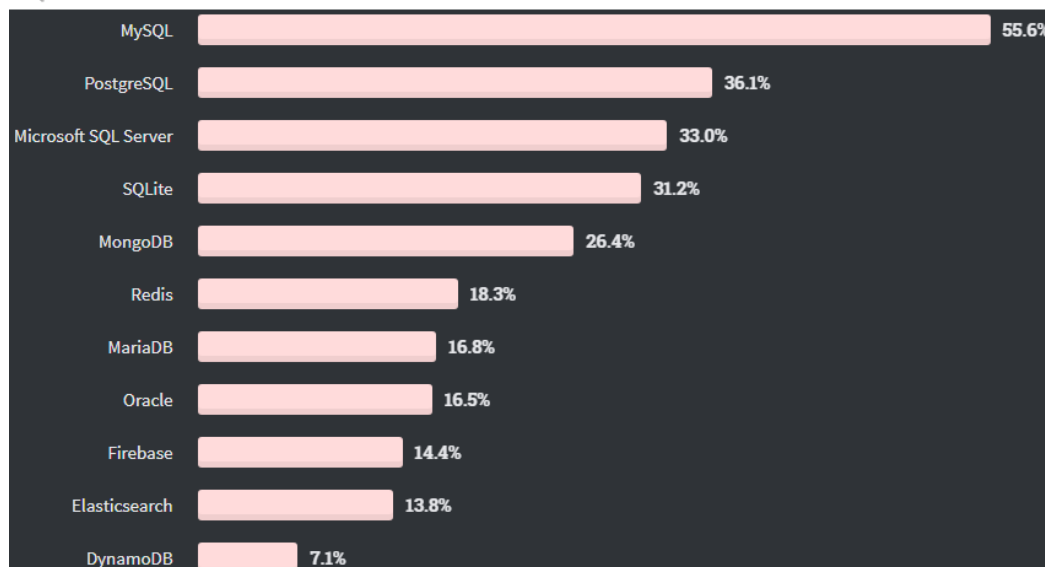


Fig. 6 Base de datos más usados en el 2020

Fuente: <https://insights.stackoverflow.com/survey/2020#technology>

Analizando la Figura 6, de la encuesta realizada cada año por StackOverflow vemos que Mongo está en el Top 5 de la lista de preferencias, esta información sirve de apoyo para escoger este sistema como Base de datos para nuestro Sistema Web Informativo, apoyado esto por la gran cantidad de alternativas que nos expresa mongo db (almacenamiento local y en la nube) así mismo como su extensa documentación.

Basados en la tabla IV y en la figura 6, se decide utilizar Mongo DB puesto que presenta algunas alternativas gratuitas.

## 2.9.2. Mongo DB

Es una base de datos No SQL<sup>8</sup>, de código abierto que está desarrollada en C++ y es multiplataforma. Mongo viene del término Humongous que significa gigante, es una base de datos escalable quiere decir que podemos añadir más recursos a la base de datos simplemente añadiendo más servidores. Es soportado por muchos lenguajes de programación para conectarse a través de driver<sup>9</sup> como lo hacen las bases de datos típicas.[15]

Mongo Db es una base de datos documental, lo que significa que almacena datos en forma de documentos de tipo JSON.

Una de las ventajas de utilizar Mongo DB es que cuenta con varias extensiones, una de ellas es MongoDB Atlas como servicio en la Nube, es la base de datos que permite implementar, utilizar y escalar una base de datos de MongoDB con tan solo unos pocos clics, pero podemos decir que no es el único servicio que ofrece Mongo DB.

Tabla VI Servicios de MongoDB

	MongoDB Atlas	Amazon DocumentDB	API de emulación Cosmos DB
<b>Admite todas las funciones de Mongo Db</b>	Si (Ejecuta Mongo DB real)	No	No
<b>Nivel Libre</b>	Si (Alm. 512 MB de RAM)	No	No (Limitado)
<b>Migración en vivo</b>	Si	No	No
<b>Elección de proveedores en la nube</b>	Si (AWS, Azure, GCP)	Media (AWS)	No
<b>Soporte de tipo de datos JSON</b>	BSON (JSON Binario)	JSON primitivo	BSON
<b>Escalado automático y almacenamiento</b>	Si	No	Si
<b>Copias de seguridad consultables</b>	Si	No	No

<sup>8</sup> No SQL. - sistemas propios para información con paquetes flexibles. Optimizados para aplicaciones con grandes volúmenes de datos.

<sup>9</sup> Driver. - JDBC (equivalente a ODBC) nos permiten efectuar conexiones con las bases de datos.

<b>Control de acceso basado en roles</b>	Si	Limitado	No
<b>Cifrado a nivel de campo del lado del cliente</b>	Si	No	No
<b>Total</b>	8	6	1

Fuente: Autor

### 2.9.2.1.Mongo DB Atlas

MongoDB Atlas es un servicio de base de datos en la nube, administrada y desarrollada por las mismas personas que crean MongoDB. Atlas maneja toda la complejidad de implementar, administrar y reparar sus implementaciones en el proveedor de servicios en la nube que elija (AWS<sup>10</sup>, Azure<sup>11</sup> y GCP<sup>12</sup>). La mejor automatización y practicas comprobadas de su clase garantizan la disponibilidad, la escalabilidad y el cumplimiento de los estándares de seguridad y la privacidad de datos más exigente.[16]

Características de Mongo DB Atlas:

- Automatización: facilidad de crear, lanzar y escalar aplicaciones en mongo DB.
- Flexibilidad: la única base de datos como servicio con todo lo necesario para las aplicaciones modernas.
- Seguridad: disponibles altos niveles de seguridad.
- Escalabilidad: gran escalabilidad sin interrumpir la actividad.
- Alta disponibilidad: implementaciones con tolerancia a errores y auto-reparación predeterminadas.
- Dispone de copias de seguridad automáticas.
- Puesta en marcha de Clúster<sup>13</sup> en segundos (configuración automática) lo que quiere decir que puede estar funcionando en pocos minutos.
- Protección y cifrado (encriptado de manera nativa los archivos en el que solo el personal con credencial tiene acceso) de los datos.

## 2.10. Git

<sup>10</sup> AWS. - Amazon Web Servicio.

<sup>11</sup> Azure. – Es un servicio de computación en la nube creado por Microsoft.

<sup>12</sup> GCP. – Servicio de almacenamiento de Google.

<sup>13</sup> Clúster. - es una agrupación de ordenadores, a menudo llamados nodos. Los nodos se conocen como conjunto de réplica.

Git es un sistema de control de versiones de código abierto y multiplataforma creado por Linus Torvalds en el 2005.

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.[17]

Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente, es decir Git nos proporciona las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas.

Git permite comparar el código de un archivo para ver las diferencias entre las versiones, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones. También permite trabajar con distintas ramas de un proyecto, como la de desarrollo para meter nuevas funciones al programa o la de producción para depurar los bugs.

## **2.11. GitHub**

GitHub es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, que fue comprado por Microsoft en junio del 2018 es de código abierto. También permite que los desarrolladores alojen proyectos de forma gratuita.

GitHub es el mayor proveedor de alojamiento de repositorios Git, y es el punto de encuentro para que millones de desarrolladores colaboren en el desarrollo de sus proyectos. Un gran porcentaje de los repositorios Git se almacenan en GitHub, y muchos proyectos de código abierto lo utilizan para hospedar su Git, realizar su seguimiento de fallos, hacer revisiones de código y otras tareas.[18]

## CAPITULO 3

### 3. Desarrollo

Este capítulo describe el desarrollo del sistema de información web, el cual pretende mejorar la experiencia del usuario final (personas beneficiadas por la recolección de desechos), y del mismo modo de los encargados de la gestión de información referente a la recolección de desechos domiciliarios del Departamento de Higiene y Limpieza; aquí se describen los procesos y fases necesarias por las cuales debe pasar el proyecto previo a su finalización, cabe aclarar que para ello se ha basado en la metodología de desarrollo de software “SCRUM”, la cual ayuda aumentar la productividad y potencia el compromiso del equipo.

#### 3.1. Análisis de Requerimientos

En una reunión conjunta entre los estudiantes y el Ing. Fausto Quevedo, llevada a cabo para recolectar información asociada a la visión de usuario se llegó al siguiente acuerdo, lo detallado a continuación sirve para tener una idea clara del aplicativo web a desarrollar:

Tabla VII Requerimientos del Aplicativo de información web

REQUERIMIENTOS FUNCIONALES	REQUERIMIENTOS NO FUNCIONALES
Almacenar la información asociada al proceso de recolección	Los modelos deben contener todos los campos necesarios y la nomenclatura adecuada
Crear la página pública	Interfaz intuitiva y que muestre las rutas trazadas en el mapa, horarios y la ubicación de los contenedores.
Mantenimiento de recursos humanos (Crear, Eliminar, Modificar)	Mantenimiento de fácil acceso e intuitivo, solo tendrá acceso la persona autorizada
Mantenimiento de Inventario (Crear, Eliminar, Modificar)	
Mantenimiento de Rutas (Crear, Eliminar, Modificar)	
Mantenimiento de Contenedores (Crear, Eliminar, Modificar)	
Generar el Reporte	Imprimir el reporte de las rutas, horarios y contenedores
Crear el Login	Interfaz intuitiva para el acceso al sistema (solo personal autorizado)
Publicar la página web informativa	Disponible en Cualquier dispositivo inteligente

Fuente: Autor

### 3.2.Implementación de la Metodología

Previo al desarrollo es necesario especificar el Product Backlog, en esta sección se enlista los requerimientos del cliente, conocidos como historias de usuarios, las mismas fueron realizadas en conjunto con los usuarios involucrados en los procesos seleccionados.

Cada historia de usuario tiene:

- Código
- Nombre de la historia
- Estimación
- Importancia y Comentario

A continuación, en la tabla VIII se puede observar los requerimientos del Proyecto.

**Tabla VIII Product Backlog**

Id	Nombre de la Historia	Importancia	Estimación	Comentario
1	Crear la Base de datos	5	2	Se crea la base de datos para ingresar la información adquirida a través del cliente.
2	Mantenimiento de Recursos Humanos	5	2	Ingreso del personal que labora en el Departamento de Higiene y Limpieza.
3	Mantenimiento Inventario	5	2	Ingresar al sistema los Vehículos que se usan para la recolección de desechos.
4	Mantenimiento Rutas	5	2	Ingresar y trazar las rutas que realizan los recolectores de desechos, asignación de recursos.
5	Mantenimiento de contenedores	5	2	Insertar el contenedor según la ubicación y asignación de recursos.
6	Generar el Reporte	4	2	Imprimir el reporte de las rutas, horarios y contenedores
7	Crear el login	4	1	Crear un login de acceso a la página para administrar la información del sistema.
8	Crear la página pública	4	2	Se crea la página pública para que la ciudadanía pueda visualizar la ruta trazada con sus respectivos horarios y la ubicación de los contenedores.
9	Publicar la página web informativa	4	1	Publicar la página web informativa.

**Fuente: Autor**

### 3.3. Definición de los Sprints.

Una vez concluida la definición de los requerimientos en el Product Backlog se agrupan las historias de usuario de tal manera que se puedan concluir en el espacio de tiempo estimado.

En cada sprint y su estimación de horas para trabajo se espera contar con distracciones o impedimentos que retrasen la conclusión de los mismos, sin embargo, al ya estar contemplados en los espacios de tiempo (estimación), se asigna un 90% de dedicación del total de horas comprendido para el desarrollo de los mismos.

Cada requerimiento o historia de usuario tiene detallado la estimación de esfuerzo y además la importancia que requiere cumplir el objetivo planteado. Estos procesos están definidos según:

- Importancia: Rangos de 1 y 5.
- Estimación: Rangos de 1 y 4 Semanas (20 horas c/u)

A continuación, en la tabla IX se observa las historias de usuario del primer Sprint.

**Tabla IX Sprint 1**

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
1	Crear la Base de datos	5	2	Se crea la base de datos para ingresar la información adquirida a través del cliente.
2	Mantenimiento de Recursos Humanos	5	2	Ingreso del personal que labora en el Departamento de Higiene y Limpieza.

**Fuente: Autor**

A continuación, en la tabla X se observa las historias de usuario del segundo Sprint.

**Tabla X Sprint 2**

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
3	Mantenimiento Inventario	5	2	Ingresar al sistema los Vehículos que se usan para la recolección de desechos.
4	Mantenimiento Rutas	5	2	Ingresar y trazar las rutas que realizan los recolectores de desechos, asignación de recursos.

**Fuente: Autor**

A continuación, en la tabla XI se observa las historias de usuario del tercer Sprint.



**Tabla XI Sprint 3**

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
5	Mantenimiento de contenedores	5	2	Insertar el contenedor según la ubicación y asignación de recursos.
6	Generar el Reporte	4	2	Imprimir el reporte de las rutas y contenedores

**Fuente: Autor**

A continuación, en la tabla XII se observa las historias de usuario del cuarto Sprint.

**Tabla XII Sprint 4**

<b>Id</b>	<b>Nombre</b>	<b>Importancia</b>	<b>Estimación</b>	<b>Comentario</b>
7	Crear el login	4	1	Crear un login de acceso a la página para administrar la información del sistema.
8	Crear la página pública	4	2	Se crea la página pública para que la ciudadanía pueda visualizar la ruta trazada con sus respectivos horarios y la ubicación de los contenedores.
9	Publicar la página web informativa	4	1	Publicar la página web informativa.

**Fuente: Autor**

Una vez agrupadas las historias de usuario de tal manera que nos genere Sprint realizables contemplando la estimación de tiempo, se obtiene un total de 4 Sprint, los mismos han sido organizados según su nivel de importancia para el desarrollo del proyecto.

### **3.4. Planificación de los Sprint**

Para el desarrollo de cada sprint se planifican revisiones para validar los avances, de esta manera se genera retroalimentación (en caso de ser necesario), o en su defecto se siguen acciones de mejora para las futuras entregas del desarrollo.

Por cada desarrollo de Sprint se evidenciarán los avances a través del TaskBoard, en el mismo se aprecian de manera clara las tareas finalizadas, en desarrollo o pendientes por cada historia de usuario; además de mostrar el Burndown para constatar la velocidad de desarrollo que está teniendo el proyecto, esto nos sirve para evidenciar aquellas tareas o historias de usuario que estén demandando mucho tiempo de desarrollo.

Una vez concluido cada Sprint se procede a evidenciar los entregables de cada uno. A continuación, se procede a detallar la planificación para cada sprint del proyecto.

A continuación, en la tabla XIII se observa las historias de usuario con el tiempo estimado para el desarrollo del primer Sprint.

**Tabla XIII Planificación del Sprint 1**

Sprint 1	
Delegado/Autor: Jonathan Peña, Dario González	
Fecha de Inicio	14/ Septiembre /2020
Fecha de fin	12/ Octubre / 2020
Duración	4 semanas (80 HRS)
Tareas a desarrollar	Crear la Base de datos Mantenimiento de Recursos Humanos

**Fuente: Autor**

A continuación, en la tabla XIV se observa las historias de usuario con el tiempo estimado para el desarrollo del segundo Sprint.

**Tabla XIV Planificación del Sprint 2**

Sprint 2	
Delegado/Autor: Jonathan Peña, Dario González	
Fecha de Inicio	12/ Octubre / 2020
Fecha de fin	09 / Noviembre / 2020
Duración	4 semanas (80 HRS)
Tareas a desarrollar	Mantenimiento Inventario Mantenimiento Rutas

**Fuente: Autor**

A continuación, en la tabla XV se observa las historias de usuario con el tiempo estimado para el desarrollo del tercer Sprint.

**Tabla XV Planificación del Sprint 3**

Sprint 3	
Delegado/Autor: Jonathan Peña, Dario González	
Fecha de Inicio	09 / Noviembre / 2020
Fecha de fin	07 / Diciembre / 2020
Duración	4 semanas (80 HRS)
Tareas a desarrollar	Mantenimiento de contenedores Generar reporte

**Fuente: Autor**

A continuación, en la tabla XVI se observa las historias de usuario con el tiempo estimado para el desarrollo del ultimo Sprint.

**Tabla XVI Planificación del Sprint 4**

Sprint 4	
Delegado/Autor: Jonathan Peña, Dario González	
Fecha de Inicio	07 / Diciembre / 2020
Fecha de fin	05 / Enero/ 2021
Duración	4 semanas (80 HRS)
Tareas a desarrollar	Crear el login Crear la página pública Publicar la página web informativa

**Fuente: Autor**

### **TaskBoard inicial y BurnDown Chart inicial**

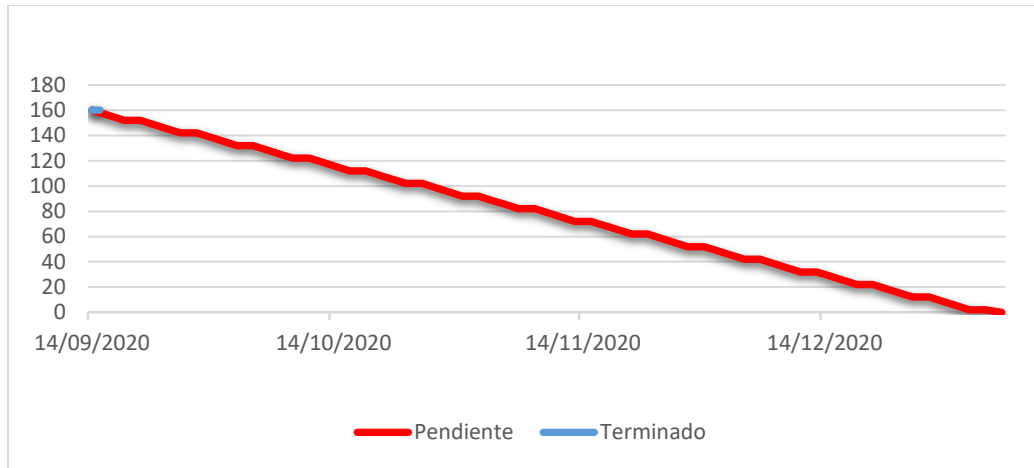
En la tabla XVII se presenta el Taskboard de desarrollo inicial del proyecto con todas las historias y la condición inicial de cada uno de los sprint.

**Tabla XVII Taskboard inicial**

N.- Sprint	Inicio: 14/ Septiembre /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 05 / Enero/ 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos	✓		
	Mantenimiento de Recursos Humanos	✓		
Sprint 2	Mantenimiento Inventario	✓		
	Mantenimiento Rutas	✓		
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la Figura 7 donde se muestra el Burndown inicial de todas las historias de usuario.



**Fig. 7 Burndown inicial**

**Fuente: Autor**

### 3.5.Desarrollo del sistema

#### 3.5.1. Sprint 1

- Crear la Base de Datos
- Mantenimiento de Recursos Humanos

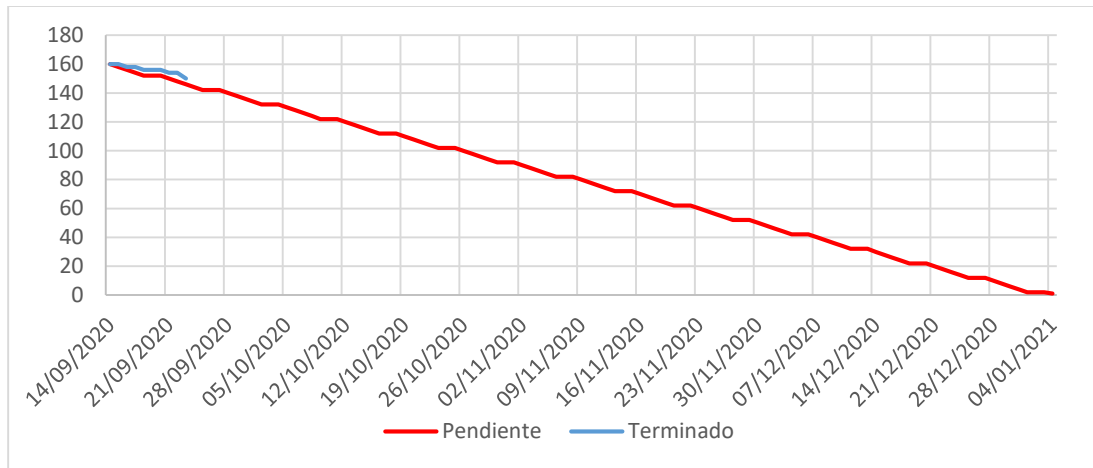
En la tabla XVIII se muestra el Taskboard del Sprint 1 y la historia de usuario “Creación de la Base de Datos” se encuentra en curso.

**Tabla XVIII Taskboard del Sprint 1**

N.- Sprint	Inicio: 14/ Septiembre /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 12/ Octubre / 2020	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos		✓	
	Mantenimiento de Recursos Humanos	✓		
Sprint 2	Mantenimiento Inventario	✓		
	Mantenimiento Rutas	✓		
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 8 se muestra el avance del primer Sprint, en donde se aprecia que al estar las actividades pendientes y en curso aun no generan impacto negativo al Burndown, pero aún están dentro del cronograma de desarrollo.



**Fig. 8 Burndown del Sprint 1**

**Fuente: Autor**

## Crear la base de datos

En la figura 9 se puede visualizar cada una de las colecciones de la base de datos en MongoDB Atlas.

Usuarios	Recursos Humanos (Employee)
<pre> _id: ObjectId("5fbf01e227349827984ca63b") nickname: "Alexander95" name: "Jonathan Alexander" surname: "Peña Peña" email: null password: null rol: "JDepartamento" __v: 0 </pre>	<pre> _id: ObjectId("5ff8a04b61c81d198c330e21") name: "Fausto" apellido: "Quevedo" cargo: "JDepartamento" telefono: "0987071234" direccion: "Calle Bolívar y 4 de Noviembre" __v: 0 </pre>
Rutas	Container
<pre> _id: ObjectId("5ff8b1cf61c81d198c330e2b") ✓ schedule_days_runs: Array   0: "thu"   1: "fri" name: "Sector Comercial 1" des: "Desechos Organicos" schedule_begin: "16:25" schedule_end: "17:25" ✓ gps: Object   &gt; origin: Object   &gt; destination: Object   &gt; waypoints: Array &gt; vehicle: Object &gt; employee: Object __v: 0 </pre>	<pre> _id: ObjectId("5ff8b78961c81d198c330e2d") ✓ schedule_days_runs: Array   0: "Lunes"   1: "Martes"   2: "Miércoles"   3: "Jueves" location: "Mercado Sucre" cantidad: 6 schedule_begin: "08:00" schedule_end: "11:00" lat: -2.732279667584203 lng: -78.85162353515625 &gt; vehicle: Object &gt; employee: Object __v: 0 </pre>
Inventario (Vehículos)	
<pre> _id: ObjectId("5ff8a9fd61c81d198c330e28") descripcion: "Recolector" disco: 2 marca: "Hino" anio: 2008 carga: "Posterior" capacidad: 20 combustible: "Diesel" estado: "Operable" __v: 0 </pre>	

Fig. 9 Colecciones MongoDB Atlas

Fuente: Autor

En la tabla XIX se muestra el Taskboard del sprint 1 la historia de usuario “Creación de la Base de Datos “se encuentra finalizada y la historia “Mantenimiento de recursos humanos” se encuentra en curso.

Tabla XIX Taskboard del Sprint 1

N.- Sprint	Inicio: 14/ Septiembre /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 12/ Octubre / 2020	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Creación de Base de datos			✓
	Mantenimiento de Recursos Humanos		✓	
Sprint 2	Mantenimiento Inventario	✓		
	Mantenimiento Rutas	✓		
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

Fuente: Autor

En la figura 10 se muestra el avance del primer Sprint, en donde se aprecia que al estar las actividades en curso aun no generan impacto negativo al Burndown de desarrollo, pero se mantiene los tiempos esperados para culminar con el desarrollo del proyecto.

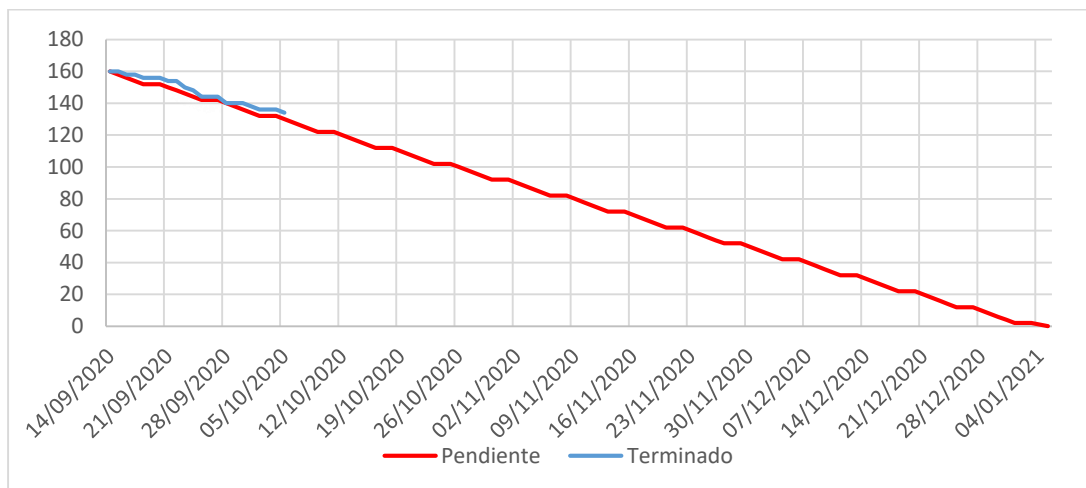


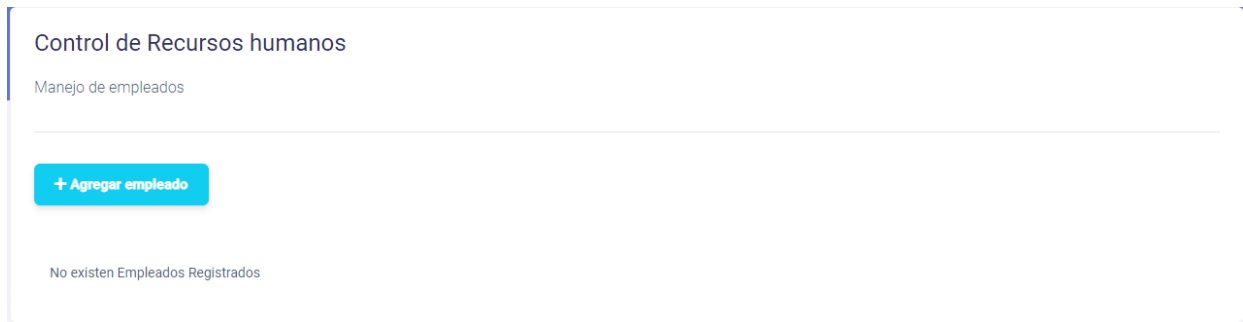
Fig. 10 Burndown del Sprint 1

Fuente: Autor

## Mantenimiento de Recursos Humanos

Control de recursos humanos

En la figura 11 se muestra la pantalla de inicio de control de recursos humanos para el posterior registro de empleados que estén laborando en el Departamento de Higiene y Limpieza.



Control de Recursos humanos

Manejo de empleados

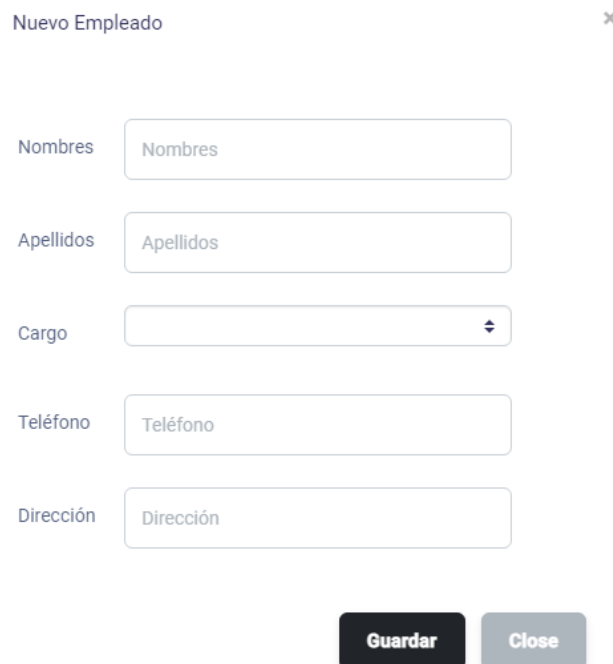
+ Agregar empleado

No existen Empleados Registrados

**Fig. 11 Control de recursos humanos**

**Fuente: Autor**

En la figura 12 se muestra los datos necesarios para el registro de los empleados.



Nuevo Empleado

Nombres

Apellidos

Cargo

Teléfono

Dirección

Guardar Close

**Fig. 12 Registro de Empleados**

**Fuente: Autor**



En la figura 13 se muestra un listado de los empleados ya registrados para el control de recursos

## Control de Recursos humanos

Manejo de empleados

+ Agregar empleado

#	EMPLEADO	DIRECCIÓN	
1	Fausto Quevedo JDepartamento	Calle Bolívar y 4 de Noviembre TIF: 0987071234	
2	Dario Calle Supervisor	Calle Ambato y Cotopaxi TIF: 0987071235	
3	Pablo Pesantes Conductor	Charasol TIF: 0987071236	
4	Andres Ormaza ARecolección	San Francisco TIF: 0987074561	

**Fig. 13 Empleados registrados**

**Fuente: Autor**

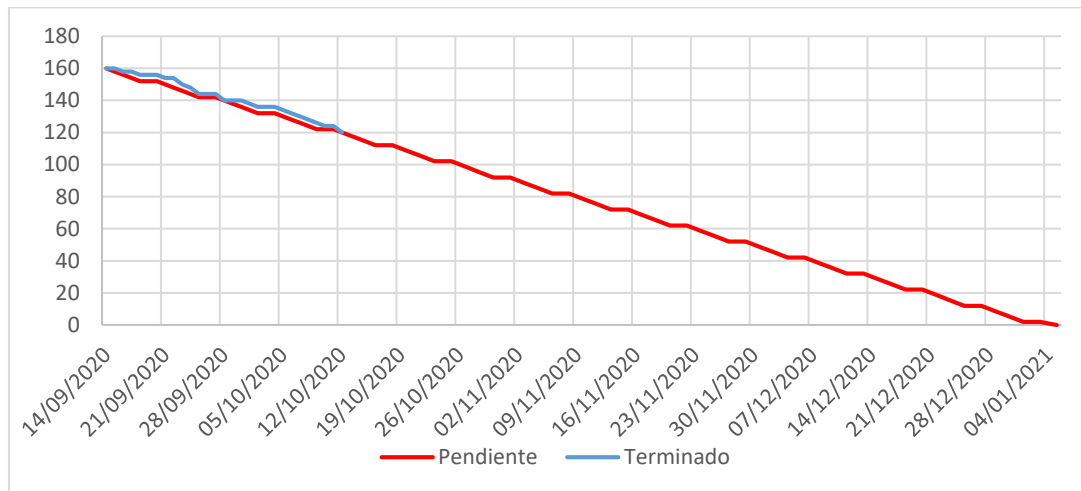
En la tabla XX se muestra el Taskboard del sprint 1 la historia de usuario “Mantenimiento de recursos humanos “se encuentra finalizada.

**Tabla XX Taskboard del Sprint 1**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario	✓		
	Mantenimiento Rutas	✓		
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 14 se muestra que el primer Sprint y la historia de usuario “Mantenimiento de Recursos humanos” se encuentra Finalizada.



**Fig. 14 Burndown final del Sprint 1**

Fuente: Autor

### 3.5.2. Sprint 2

- ✓ Mantenimiento de Inventario y rutas

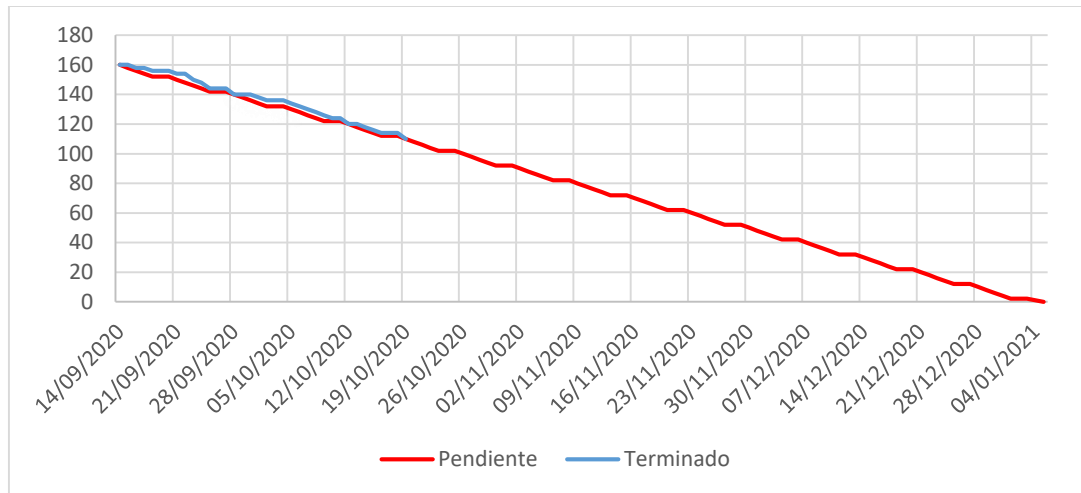
En la tabla XXI se muestra el Taskboard del Sprint 2 y la historia de usuario “Mantenimiento Inventario” se encuentra en curso y se encuentra dentro del rango de desarrollo estipulado para el proyecto.

**Tabla XXI Taskboard del Sprint 2**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario		✓	
	Mantenimiento Rutas	✓		
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

Fuente: Autor

En la figura 15 se muestra el avance del segundo Sprint, en donde se aprecia que al estar las actividades pendientes y en curso aun no generan impacto negativo al Burndown, pero aún están dentro del cronograma de desarrollo.

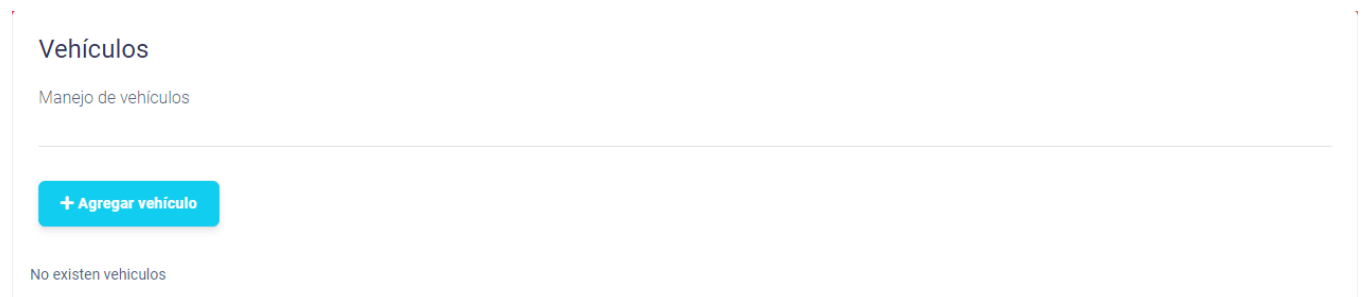


**Fig. 15 Burndown del Sprint 2**

Fuente: Autor

## Mantenimiento de Inventario

En la figura 16 se muestra la pantalla de inicio de vehículos para su posterior registro.



**Fig. 16 Inventario**

Fuente: Autor

En la figura 17 se muestra los datos necesarios para el registro de vehículos.

Nuevo vehículo

Descripción

Descripción

Disco

0

Marca

Año

0

Carga

carga

Capacidad

0

Combustible

Estado

Guardar

Close

Fig. 17 Registro de vehículos

Fuente: Autor

En la figura 18 se muestra el listado de los vehículos ya registrados para el control de inventario.

Vehículos

Manejo de vehículos

+ Agregar vehículo






#	MARCA	AÑO	ESTADO	
1	Hino Carga: Posterior	2008 Capacidad: 20 (yds3)	Operable Combustible: Diesel	 
2	Nissan Carga: Posterior	2008 Capacidad: 16 (yds3)	Parcialmente Operable Combustible: Diesel	 
3	Internacional Carga: Posterior	2004 Capacidad: 20 (yds3)	No Operable Combustible: Diesel	 

Fig. 18 Vehículos registrados

Fuente: Autor

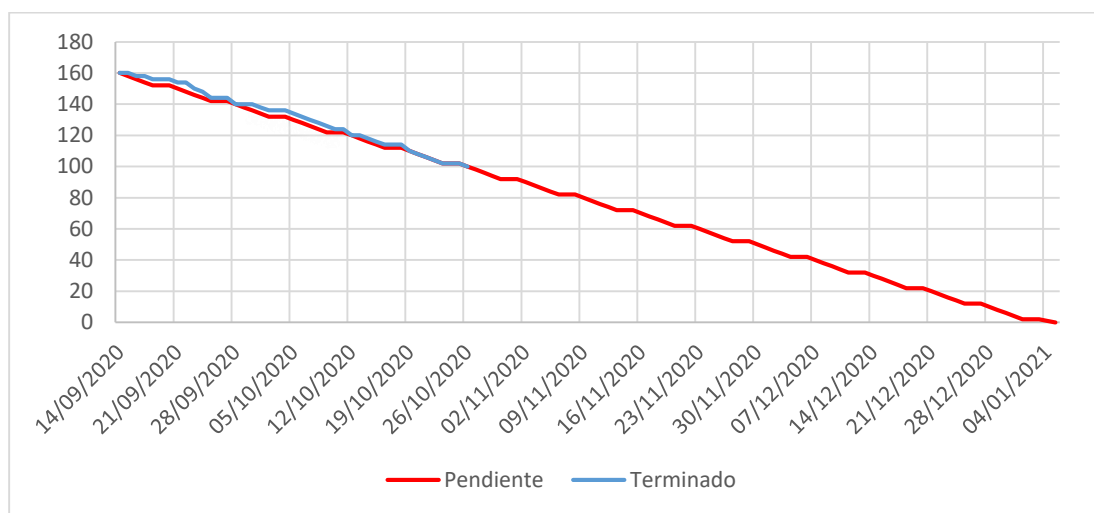
En la tabla XXII se muestra el Taskboard del Sprint 2 y la historia de usuario” Mantenimiento de Inventario” se encuentra Finalizada y la historia “Mantenimiento de rutas” se encuentra en curso.

**Tabla XXII Taskboard del Sprint 2**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas		✓	
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 20 se muestra el avance del segundo Sprint, en donde se aprecia que al estar las actividades en curso aun no generan impacto negativo al Burndown de desarrollo, pero se mantiene los tiempos esperados para culminar con el desarrollo del proyecto.



**Fig. 19 Burndown del Sprint 2**

**Fuente: Autor**

## Mantenimiento de Rutas

### Control de Rutas

En la figura 20 se muestra la pantalla de inicio de control de Rutas para el posterior registro de rutas de recolección de desechos domiciliarios.

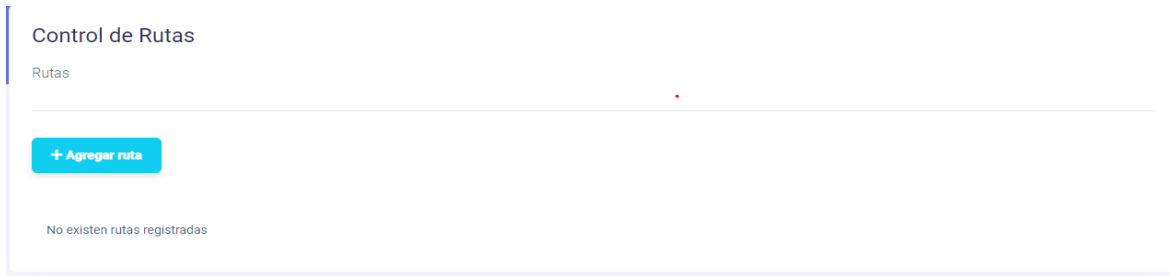


Fig. 20 Control de rutas

Fuente: Autor

En la figura 21 se muestra la pantalla en la cual se va a realizar el control de rutas, para ello se muestra todos los controles necesarios para su registro además de asignar un vehículo y empleado.

The screenshot shows a form titled 'Nueva Ruta' for registering a new route. The form includes several input fields and checkboxes. At the top, there are dropdown menus for 'Empleado' and 'Vehículo'. Below these are dropdowns for 'Sector' and 'Desechos'. Further down are time pickers for 'Hora Inicio' and 'Hora Fin'. A section for 'Días de la semana' contains checkboxes for 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado', and 'Domingo'. The bottom half of the form features a Google Map showing a street view of a neighborhood with labels like 'Iglesia de Charasol', 'Banco del Austro', and 'Bomberos Charasol'. At the bottom right of the map area are two buttons: 'Guardar' (Save) and 'Close'.

Fig. 21 Registro de rutas

Fuente: Autor

En la figura 22 se muestra el listado de las rutas ya registrados para el control de rutas.

Control de Rutas

Rutas

+ Agregar ruta

#	EMPLEADO	SECTOR	VER LA RUTA TRAZADA EN EL MAPA	
1	Vehículo:	Sector Comercial 1 (Desechos Organicos) Días: (thu,fri) H. Inicio: (16:25) H. Fin: (17:25)	<div>Ver en Mapa</div>	<div></div> <div></div>
2	Vehículo:	Sector Residencial 2 (Desechos Inorganicos) Días: (mon,wed) H. Inicio: (10:26) H. Fin: (14:26)	<div>Ver en Mapa</div>	<div></div> <div></div>

Fig. 22 Rutas registradas

Fuente: Autor

En la figura 23 se muestra la ruta trazada en el mapa con los respectivos puntos de interés.

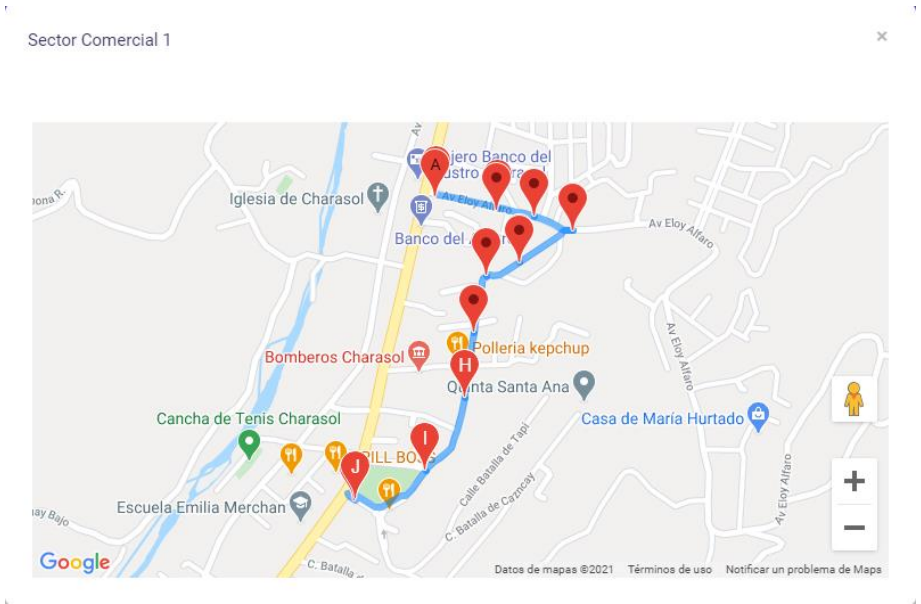


Fig. 23 Ruta trazada en el mapa

Fuente: Autor

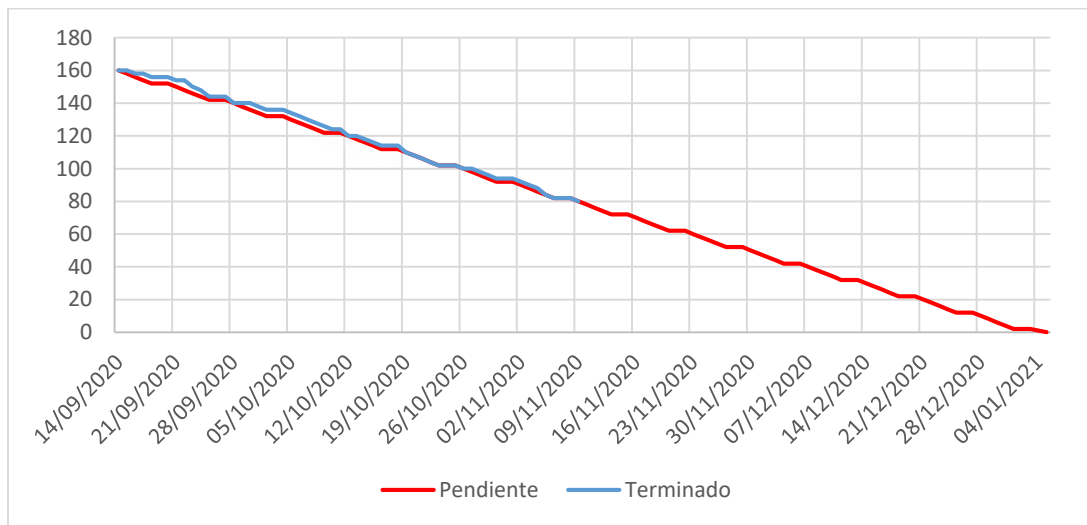
En la tabla XXIII se muestra el Taskboard del Sprint 2 y la historia de usuario” Mantenimiento de rutas” se encuentra Finalizada.

**Tabla XXIII Taskboard del Sprint 2**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores	✓		
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 24 se muestra que el segundo Sprint y la historia de usuario “Mantenimiento de rutas” se encuentra Finalizada.



**Fig. 24 Burndown final del Sprint 2**

**Fuente: Autor**

### 3.5.3. Sprint 3

- ✓ Mantenimiento de contenedores
- ✓ Generar reporte



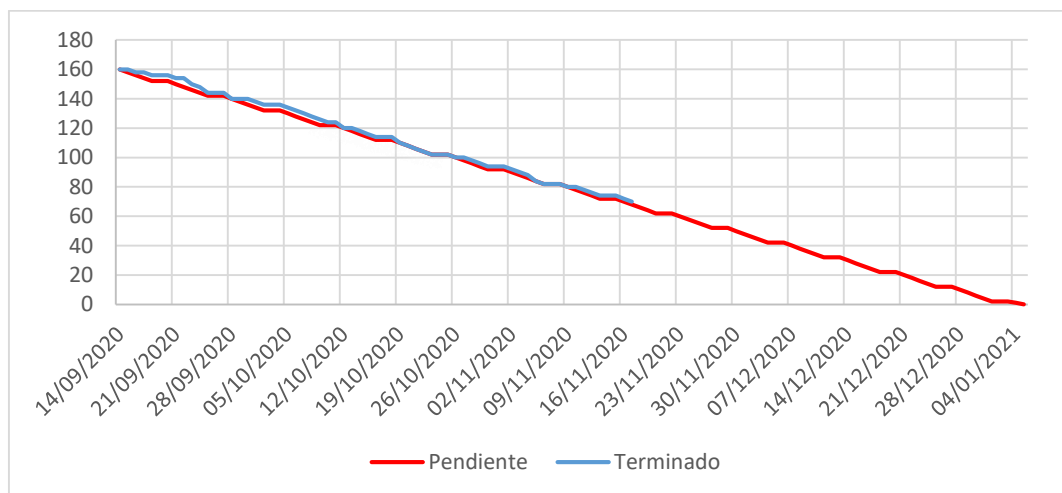
En la tabla XXIV se muestra el Taskboard del Sprint 3 y la historia de usuario “Mantenimiento de contenedores” se encuentra en curso y se encuentra dentro del rango de desarrollo estipulado para el proyecto.

**Tabla XXIV Taskboard del Sprint 3**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores		✓	
	Generar reporte	✓		
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 25 se muestra el avance del tercer Sprint, en donde se aprecia que al estar las actividades pendientes y en curso aun no generan impacto negativo al Burndown, pero aún están dentro del cronograma de desarrollo.



**Fig. 25 Burndown del Sprint 3**

**Fuente: Autor**

## Mantenimiento de Contenedores

### Control de contenedores

En la figura 26 se muestra la pantalla de inicio de control de Contenedores, usada para el registro de la ubicación de los mismos.

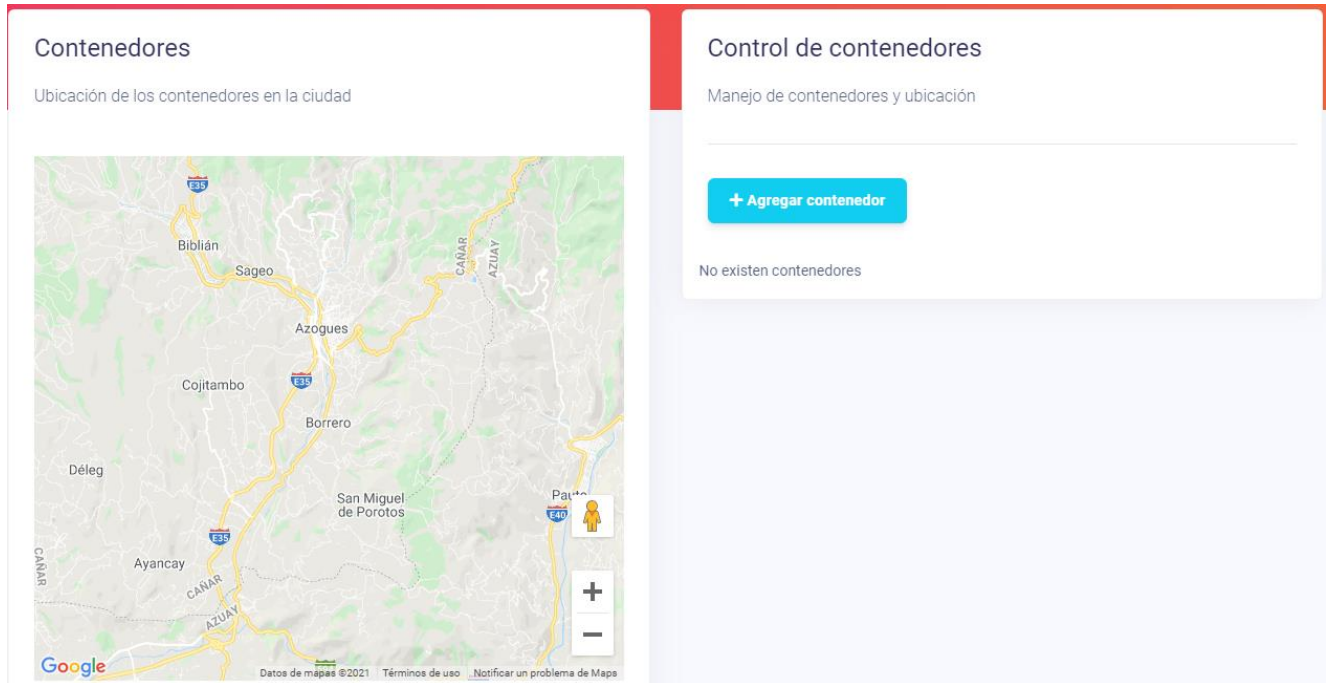


Fig. 26 Control de contenedores

Fuente: Autor

En la figura 27 se muestran los datos necesarios para el registro de contenedores y a su vez asignar el empleado y vehículo que deberán realizar la recolección.

Nuevo contenedor ✕

Empleado

Vehículo

Ubicación

Cantidad

Hora Inicio

Hora Fin

Días de la semana

☐ Lunes
☐ Martes
☐ Miércoles
☐ Jueves
☐ Viernes
☐ Sábado
☐ Domingo

Guardar

Close

**Fig. 27 Registro de contenedores**

**Fuente: Autor**

En la figura 28 se muestra el listado de los contenedores ya registrados y a su vez la ubicación de los mismos en la ciudad.

**Contenedores**  
Ubicación de los contenedores en la ciudad

**Control de contenedores**  
Manejo de contenedores y ubicación

+ Agregar contenedor

#	EMPLEADO	CONTAINER	
1	Vehículo:	Mercado Sucre   (6) Días: (Lunes,Martes,Miércoles,Jueves) H. Inicio: (08:00) H. Fin: (11:00)	
2	Vehículo:	Ceenterio   (1) Días: (Lunes,Jueves) H. Inicio: (08:00) H. Fin: (11:00)	

**Fig. 28 Contenedores registrados**

**Fuente: Autor**

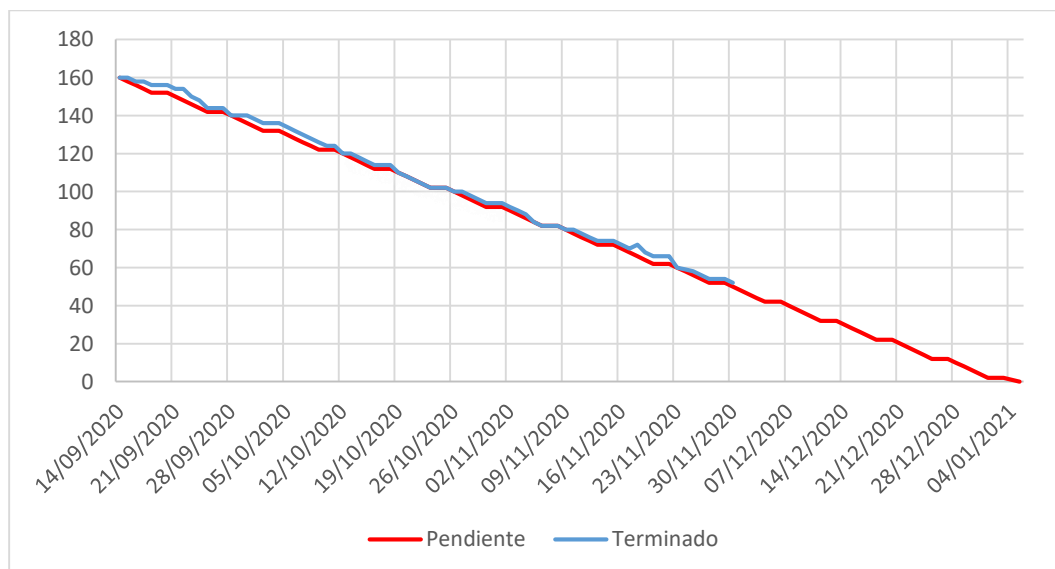
En la tabla XXV se muestra el Taskboard del Sprint 3 y la historia de usuario” Mantenimiento de contenedores” se encuentra Finalizada y la historia “Generar reporte” se encuentra en curso.

**Tabla XXV Taskboard del Sprint 3**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores			✓
	Generar reporte		✓	
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 29 se muestra el avance del Sprint 3, en donde se aprecia que al estar las actividades en curso aun no generan impacto dentro del Burndown de desarrollo, pero se mantiene los tiempos esperados para culminar con el desarrollo del proyecto.



**Fig. 29 Burndown del Sprint 3**

**Fuente: Autor**

## Generar reporte

### Reporte

En la figura 30 se muestra la pantalla de inicio de control de reportes, en la cual se visualiza los datos de rutas, horarios y contenedores para su posterior impresión.

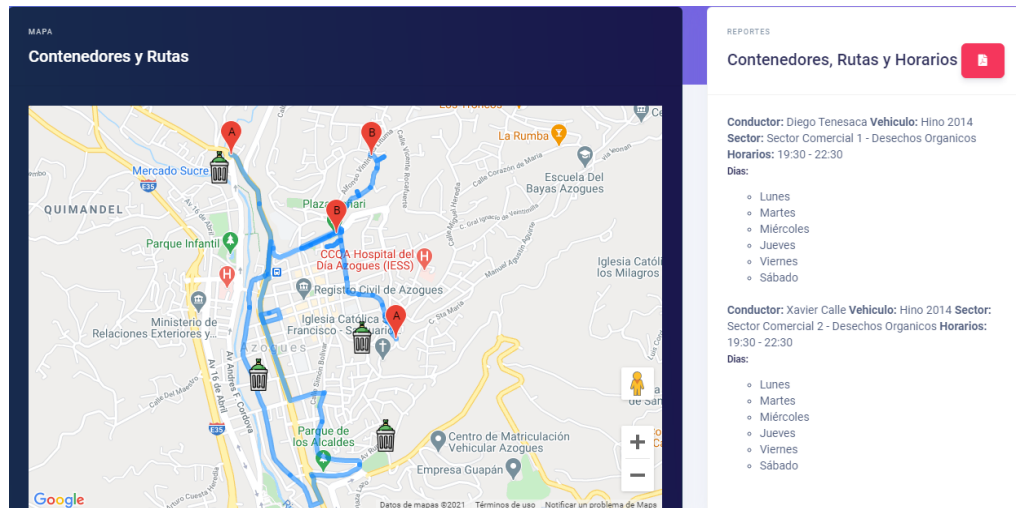


Fig. 30 Reporte

Fuente: Autor

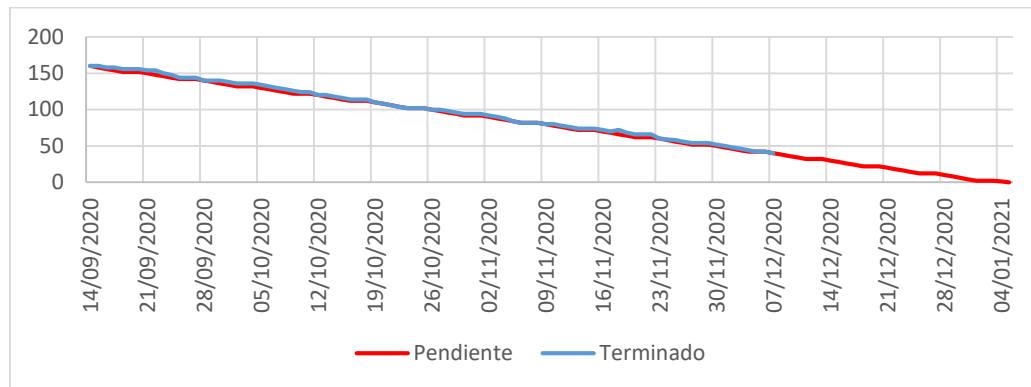
En la tabla XXVI se muestra el Taskboard del Sprint 3 y la historia de usuario” Generar reporte” se encuentra Finalizada

Tabla XXVI Taskboard del Sprint 3

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores			✓
	Generar reporte			✓
Sprint 4	Crear el login	✓		
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

Fuente: Autor

En la figura 31 se muestra que el Tercer Sprint y la historia de usuario “Generar reporte” se encuentra Finalizada.



**Fig. 31 Burndown final del Sprint 3**

**Fuente: Autor**

### 3.5.4. Sprint 4

- ✓ Creación de login
- ✓ Crear la página pública
- ✓ Publicar la página web informativa

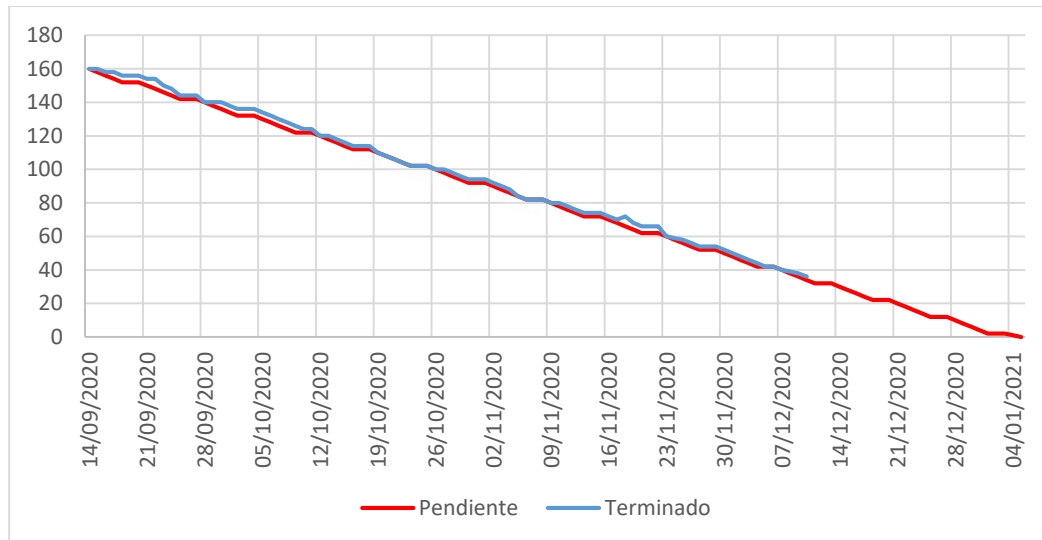
En la tabla XXVII se muestra el Taskboard del Sprint 4 y la historia de usuario “Creación del login” se encuentra en curso y se encuentra dentro del rango de desarrollo estipulado para el proyecto.

**Tabla XXVII Taskboard del Sprint 4**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores			✓
	Generar reporte			✓
Sprint 4	Crear el login		✓	
	Crear la página pública	✓		
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 32 se muestra el avance del Sprint 4, en donde se aprecia que al estar las actividades pendientes y en curso aun no generan impacto negativo al Burndown, pero aún están dentro del cronograma de desarrollo.



**Fig. 32 Burndown del Sprint 4**

Fuente: Autor

## Crear el Login

En la figura 33 se muestra la pantalla de inicio de sesión.

The login form is displayed within a light blue container. It features two input fields: 'Usuario' (User) and 'Contraseña' (Password). The 'Usuario' field has an envelope icon, and the 'Contraseña' field has a lock icon. Below these fields is a checkbox labeled 'Recordarme' (Remember me). At the bottom of the form is a blue button labeled 'Iniciar Sesión' (Login).

**Fig. 33 Login**

Fuente: Autor

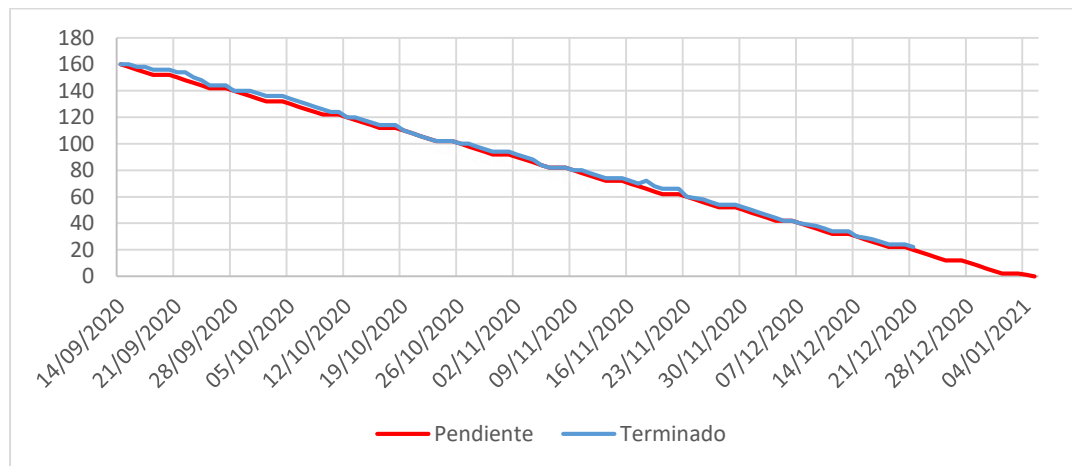
En la tabla XXVIII se muestra el Taskboard del Sprint 4 y la historia de usuario” Creación de Login” se encuentra Finalizada y la historia “Crear la página pública” se encuentra en curso.

**Tabla XXVIII Taskboard del Sprint 4**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores			✓
	Generar reporte			✓
Sprint 4	Crear el login			✓
	Crear la página pública		✓	
	Publicar la página web informativa	✓		

**Fuente: Autor**

En la figura 34 se muestra el avance del Sprint 4, en donde se aprecia que al estar las actividades en curso aun no generan impacto dentro del Burndown de desarrollo, pero se mantiene los tiempos esperados para culminar con el desarrollo del proyecto.



**Fig. 34 Burndown del Sprint 4**

**Fuente: Autor**

**Crear la página pública**



En la figura 35 se muestra la pantalla de inicio en la que se puede visualizar los sectores, contenedores.

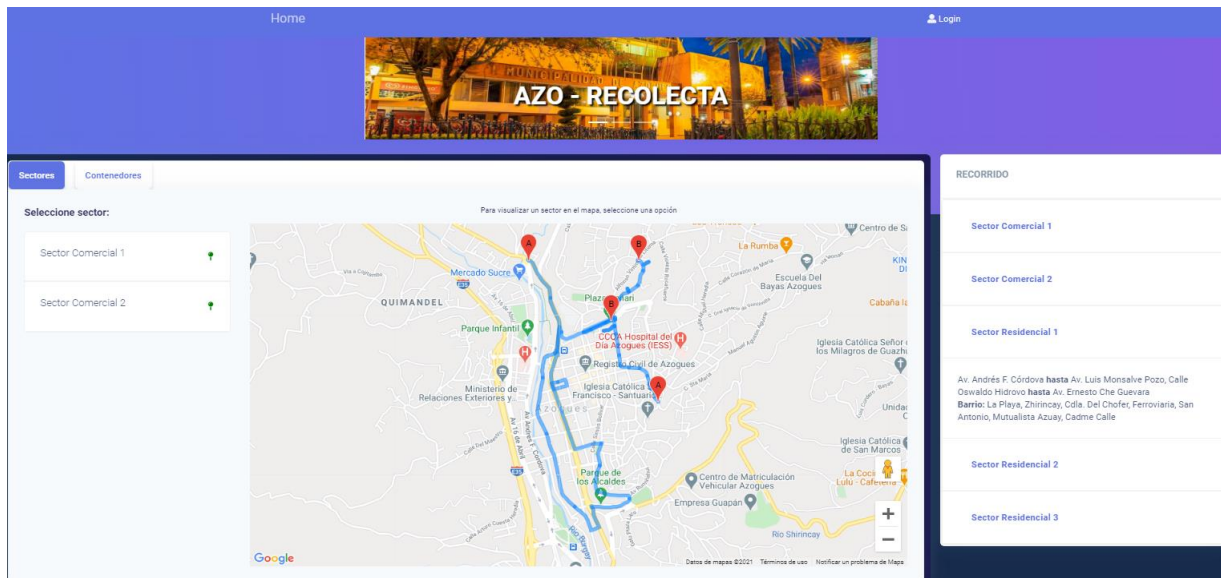


Fig. 35 Página publica

Fuente: Autor

En la figura 36 se muestra la pantalla con el listado de los sectores con sus respectivos horarios y la ruta de recolección trazada en el mapa.

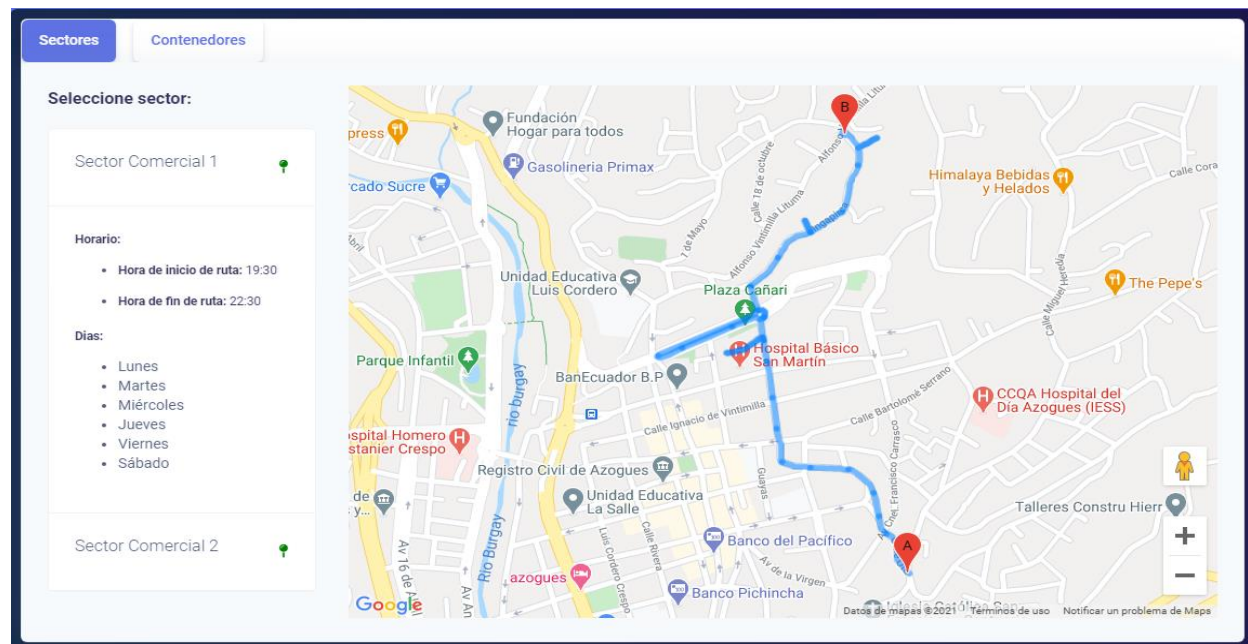
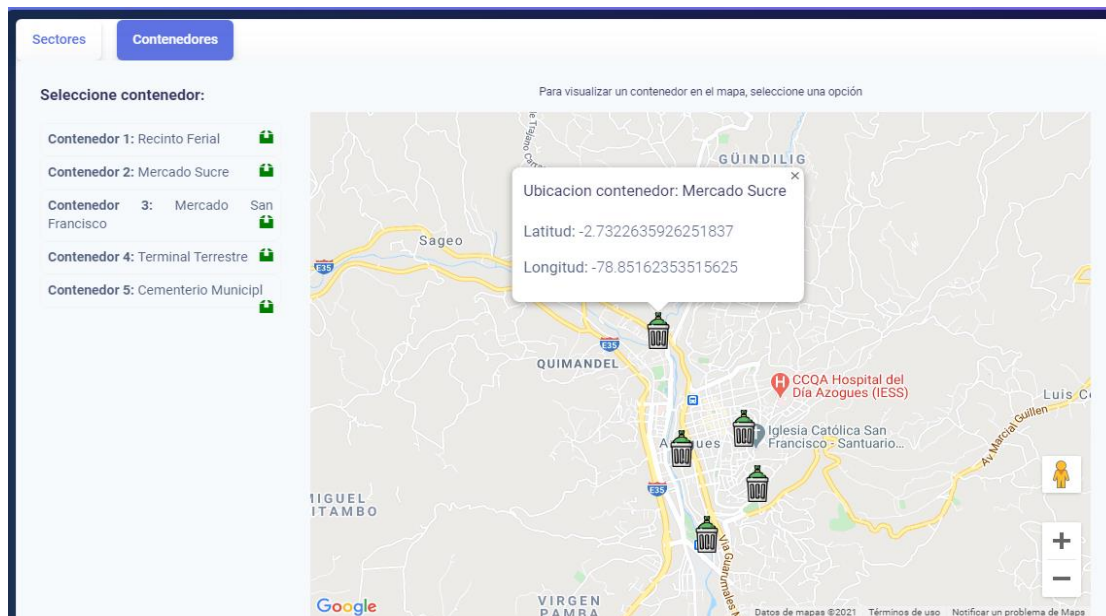


Fig. 36 Sectores

Fuente: Autor

En la figura 37 se muestra la pantalla con el listado de los contenedores y también podemos visualizar la ubicación en el mapa.



**Fig. 37 Contenedores**

**Fuente: Autor**

En la tabla XXIX se muestra el Taskboard del Sprint 4 y la historia de usuario” Crear la página pública” se encuentra Finalizada y la historia “Publicar la página web informativa” se encuentra en curso.

**Tabla XXIX Taskboard del Sprint 4**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores			✓
	Generar reporte			✓
Sprint 4	Crear el login			✓
	Crear la página pública			✓
	Publicar la página web informativa		✓	

**Fuente: Autor**

En la figura 38 se muestra el avance del Sprint 4, en donde se aprecia que al estar las actividades en curso aun no generan impacto dentro del Burndown de desarrollo, pero se mantiene los tiempos esperados para culminar con el desarrollo del proyecto.

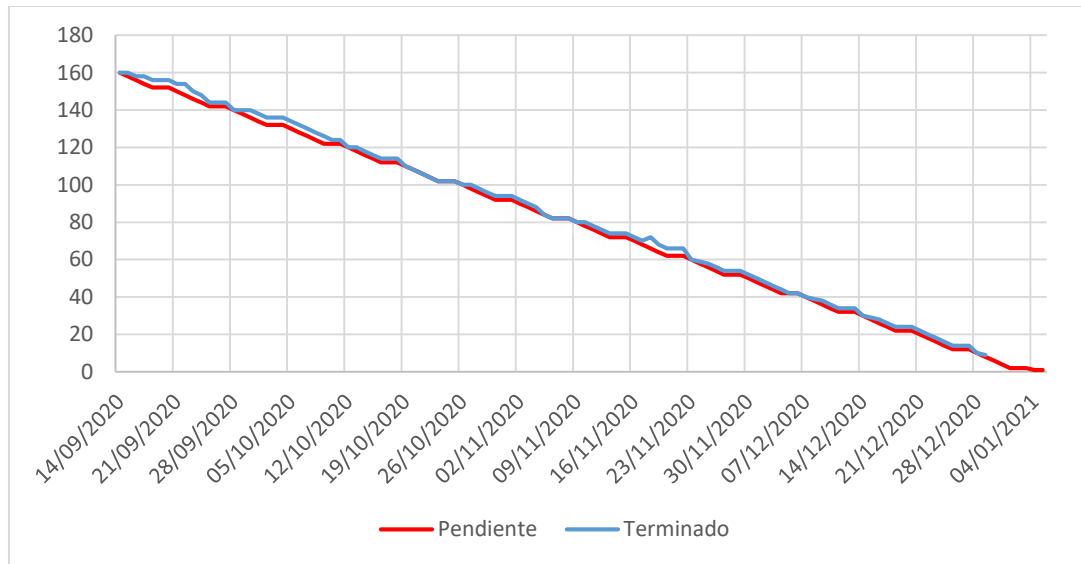


Fig. 38 Burndown del Sprint 4

Fuente: Autor

## Publicar la página web informativa

En la figura 39 se procede a poner el servidor de FrontEnd en producción para posteriormente subir los cambios y habilitar GitHub pages.

```

PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jjalex\Documents\Visual Studio Code\Recolector\FrontEnd> ng build --prod
Your global Angular CLI version (11.0.7) is greater than your local Angular CLI version (9.0.5). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".

chunk {} runtime.2608b02daea62672021a.js (runtime) 2.3 kB [entry] [rendered]
chunk {1} 1.fbfdd6359f0637bc2ffb.js () 118 kB [rendered]
chunk {2} 2.ca4325f2395eba4020f9.js () 172 kB [rendered]
chunk {3} main.2bfd7817f8deb26fda0f.js (main) 848 kB [initial] [rendered]
chunk {4} polyfills.07dc418898ae27f04d48.js (polyfills) 58.6 kB [initial] [rendered]
chunk {5} polyfills-es5.4f6dd560d5777051c8b4.js (polyfills-es5) 131 kB [initial] [rendered]
chunk {6} styles.bd4b292c95d09b667dad.css (styles) 315 kB [initial] [rendered]
chunk {7} 7.5fd61031111e302c1114.js () 141 kB [rendered]
chunk {8} 8.af233ef2654f089efe8b.js () 15.6 kB [rendered]
chunk {scripts} scripts.2db4cbc668a10a0c2a68.js (scripts) 165 kB [entry] [rendered]
Date: 2021-01-19T15:02:05.577Z - Hash: b6d9ff1ca989270404c6 - Time: 257819ms

```

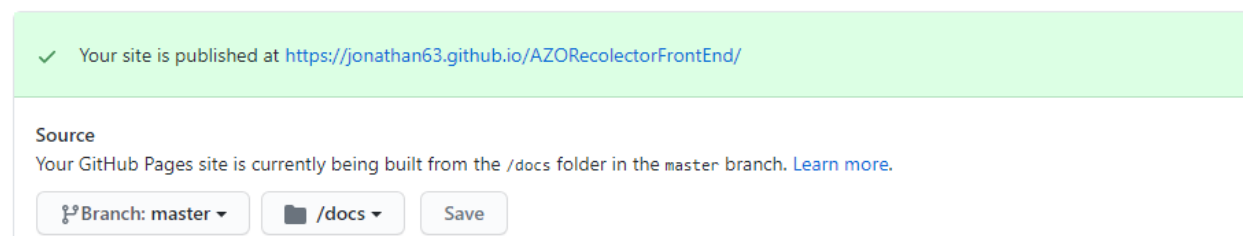
Fig. 39 Producción

Fuente: Autor

En figura 40 una vez configurado GitHub Pages se observa el acceso a través del siguiente link (<https://jonathan63.github.io/AZORecolectorFrontEnd/>).

## GitHub Pages

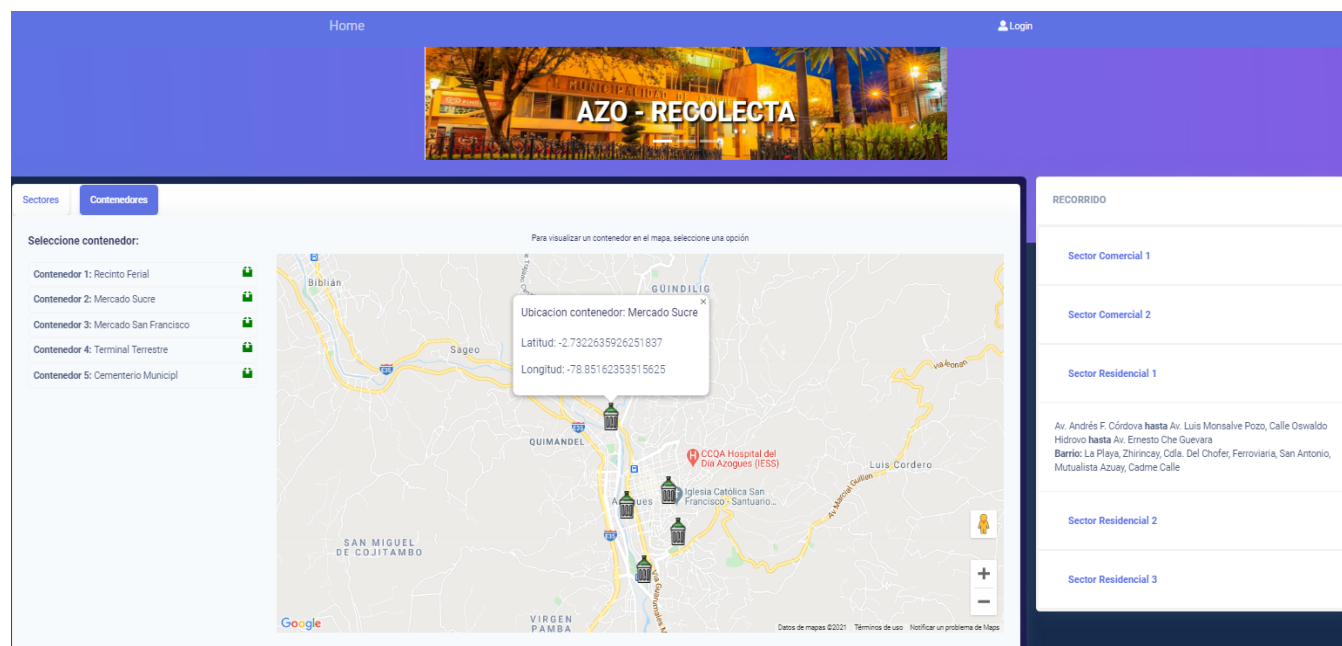
GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.



**Fig. 40 GitHub Pages**

**Fuente: Autor**

En la figura 41 se visualiza la pantalla de inicio de las rutas trazadas en el mapa con sus respectivos horarios y los sectores con sus respectivas calles y barrios, cabe recalcar que es una captura de pantalla tomada desde un computador.



**Fig. 41 Página pública**

**Fuente: Autor**

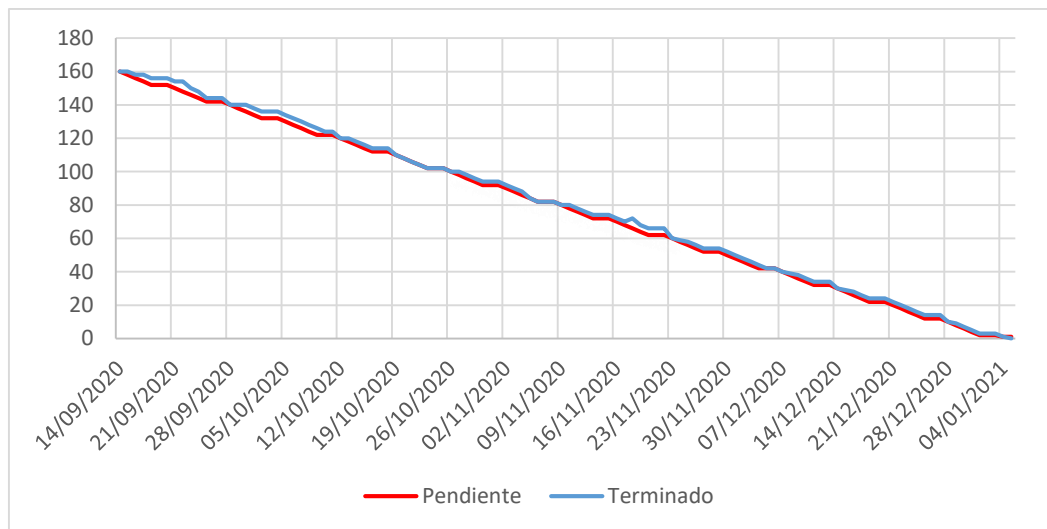
En la tabla XXX se muestra el Taskboard del Sprint 4 y la historia de usuario” Publicar la página web informativa” se encuentra Finalizada.

**Tabla XXX Taskboard del Sprint 4**

N.- Sprint	Inicio: 13/ Jul /2020	Delegado/Autor: Jonathan P. y Dario G.		
	Fin: 22 / Ene / 2021	Desarrollo del sistema		
	Historias de Usuario	Pendiente	En Curso	Finalizado
Sprint 1	Crear la Base de datos			✓
	Mantenimiento de Recursos Humanos			✓
Sprint 2	Mantenimiento Inventario			✓
	Mantenimiento Rutas			✓
Sprint 3	Mantenimiento de contenedores			✓
	Generar reporte			✓
Sprint 4	Crear el login			✓
	Crear la página pública			✓
	Publicar la página web informativa			✓

**Fuente: Autor**

En la figura 42 se muestra que el Sprint 4 y la historia de usuario “Publicar la página web informativa” se encuentra Finalizada.



**Fig. 42 Burndown final del Sprint 4**

**Fuente: Autor**

### **3.6. Entregables finales del proyecto**

Una vez se concluye el ultimo Sprint se da por terminado el desarrollo de la aplicación, el desarrollo del mismo está dentro de lo establecido en la estimación de tiempo, así pues, los retrasos surgidos en las tareas de cada sprint no tuvieron impacto significativo para la entrega del Producto final, esto gracias a que se planifico un 90% del tiempo en cada semana de desarrollo como trabajo optimo y el 10% restante implicaba la existencia de retrasos inesperados que puedan surgir en la construcción del Sistema de Información.

Una vez concluido el desarrollo de la solución se facilita una URL en la cual se puede visualizar la funcionalidad del mismo: <https://jonathan63.github.io/AZOREcolectorFrontEnd/>.

En la Tabla XXXI a continuación se puede observar el informe de Entregable Final, además una vez concluido los sprint se puede comprobar la funcionalidad del Sistema de Información Web.

**Tabla XXXI Informe final de Entregables**

<b>INFORME DE ENTREGABLE FINAL</b>						
<b>Tarea</b>	<b>Respuesta esperada de la Aplicación</b>	<b>Coincide</b>		<b>Respuesta del Sistema</b>	<b>Porcentaje de Avance</b>	<b>Observación</b>
		Si	No			
Crear la Base de datos	Permite ejecutar métodos HTML como REST API	<i>X</i>		Usando Postman permite comprobar esta funcionalidad.	<i>15%</i>	Ninguna
Mantenimiento de RRHH	Carga la información Solicitada	<i>X</i>		Muestra las Paginas requeridas para el listado y creación.	<i>25%</i>	Ninguna
Mantenimiento de Inventario	Carga la información Solicitada	<i>X</i>		Muestra las Paginas requeridas para el listado y creación.	<i>35%</i>	Ninguna
Mantenimiento de Rutas	Carga la información Solicitada	<i>X</i>		Muestra las Paginas requeridas para el listado y creación.	<i>50%</i>	Necesita información cargada en los anteriores.
Mantenimiento de contenedores	Carga la información Solicitada	<i>X</i>		Muestra las Paginas requeridas para el listado y creación.	<i>65%</i>	Necesita información cargada en los anteriores
Generar reporte	Carga la información Solicitada	<i>X</i>		Muestra un reporte listo para ser utilizado	<i>75%</i>	Necesita información cargada en los anteriores
Crear el login	Verifica autenticación	<i>X</i>		Una vez se realice el login permite acceder al sistema.	<i>80%</i>	Verifica autenticación simple
Crear la página pública	Permite visualizar información relevante para el beneficiario final	<i>X</i>		No requiere que el usuario realice autenticación previo visualización.	<i>90%</i>	Ninguna
Publicar la página web informativa	Permite acceder al sistema a través de Internet	<i>X</i>		Se puede visualizar el sistema usando un navegador web.	<i>100%</i>	Está disponible para cualquier dispositivo inteligente.

**Fuente: Autor**

## **CAPITULO 4**

### **4. Conclusiones y Recomendaciones**

#### **4.1.Conclusiones**

Al trabajar con SCRUM, la carga de trabajo se divide a los diferentes integrantes, de esta manera se saca el mayor provecho a las habilidades de cada uno, generando un impacto mínimo en los costes y tiempo. Además, al momento de generar las entregas de cada Sprint se interactúa con el cliente, pudiendo este último sugerir cambios con la finalidad de mejorar el producto.

Se identificaron los procesos del sistema mediante la recolección de información que nos sirvió para planificar de mejor manera el Backlog, y asignar porcentajes de estimación de esfuerzos aceptables para la entrega oportuna de las tareas.

Se desarrolló un sistema de información web que se acopla a las necesidades del departamento de higiene y limpieza; dicho prototipo permitirá al momento de su implementación la automatización de los procesos manuales, también servirá como un medio informativo hacia la colectividad sobre los servicios brindados.

#### **4.2.Recomendaciones**

Implementar y utilizar el software desarrollado para mejorar las actividades productivas del departamento de higiene y limpieza del Gad Municipal del Cantón Azogues. Además, se recomienda la creación de un equipo SCRUM (interno o externo) que se encargue de dar mejoras continuas al software.

Llevar a efecto el desarrollo de una aplicación móvil y uso de placas programables que permitan geo localizar en tiempo real la ubicación de los vehículos recolectores.

Utilizar Mongo DB por su alto rendimiento en datos no estructurados, debido a su utilidad en entornos que requieran escalabilidad, facilidad de uso, configuración sencilla y su compatibilidad con AWS, que nos ayudará a conseguir sistemas de escala horizontal.

Utilizar la metodología SCRUM para el desarrollo de proyectos de software similares, debido a su facilidad de integración y manejo de datos.



## REFERENCIAS BIBLIOGRÁFICAS

- [1] O. J. V. Zamudio y J. A. G. Herreño, «Formular Estrategias para optimizar la recolección de residuos sólidos domiciliarios del municipio de Madrid», Universidad Piloto de Colombia, 2012.
- [2] Mohr Lagos Marucio Alberto, «Un Modelo de Programación Lineal Entera para Resolver el problema de Recolección de Residuos Domiciliarios», Universidad del Bio-Bio, 2013.
- [3] J. Cusco y K. Picón, «Optimización de rutas de recolección de desechos sólidos domiciliarios mediante uso de herramientas SIG», Universidad de Cuenca, 2015.
- [4] C. C. G. Maribel, «Una mirada en el futuro : “ Propuesta de mejora para el manejo de desechos sólidos en el centro de la Municipalidad de Esmeraldas , Ecuador ”», Universidad de San Francisco de Quito, 2015.
- [5] D. I. Álvarez, «EJE 07-10 Sistemas de seguridad ciudadana por georeferenciación y geolocalización para zonas rurales del Cantón Cuenca incorporados al SIS ECU 9-1-1 del Ecuador», pp. 413-418, 2017, [En línea]. Disponible en: <http://revistas.uazuay.edu.ec/index.php/memorias/article/view/88>.
- [6] A. D. E. Aguilar, «Propuesta de rutas óptimas para la recolección de desechos sólidos en la zona centro norte de la parroquia sangolquí mediante la extensión network analyst del software ARCGIS», Pontificia Universidad Católica del Ecuador, 2017.
- [7] Q. O. E. Alejandrina, «Desarrollo De Un Sistema De Geolocalización De Alerta De Recojo De Residuos Sólidos En El Distrito De San Jerónimo, 2018.», Universidad NacionalL Jose Maria Arguedas, 2018.
- [8] E. A. M. Vera, «Propuesta para mejorar la recolección de desechos sólidos urbanos del Canton Narnajal Mediante Sistemas de Informacion Geografica», Universidad de Guayaquil, 2019.
- [9] F. B. L. Chavez, «Optimización de rutas para la recolección de desechos sólidos en la ciudad de Ibarra provincia de Imbabura», Universidad Técnica del Norte, 2019.
- [10] M. Trigas Gallego y A. C. Domingo Troncho, «Gestión de Proyectos Informáticos. Metodología Scrum.», *Openaccess.Uoc.Edu*, p. 56, 2012, [En línea]. Disponible en: <http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf%5Cnhttp://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>.
- [11] Gersón Beltrán López, «Geomarketing: geolocalización, redes sociales y turismo - Google

- Libros», 2014.  
[https://books.google.com.ec/books?id=n9eLBQAAQBAJ&pg=PT48&dq=geolocalizacion&hl=es&sa=X&ved=0ahUKEwixhODLrPXpAhXkmuAKHS\\_GCUEQ6AEIJzAA#v=onepage&q=geolocalizacion&f=false](https://books.google.com.ec/books?id=n9eLBQAAQBAJ&pg=PT48&dq=geolocalizacion&hl=es&sa=X&ved=0ahUKEwixhODLrPXpAhXkmuAKHS_GCUEQ6AEIJzAA#v=onepage&q=geolocalizacion&f=false) (accedido nov. 29, 2020).
- [12] Ramos Martin Alicia y Ramos Martin Martin Jesus, «Aplicaciones Web 2.<sup>a</sup> edición - Google Libros», 2014.  
<https://books.google.es/books?hl=es&lr=&id=43G6AwAAQBAJ&oi=fnd&pg=PA1#v=onepage&q&f=false> (accedido nov. 29, 2020).
- [13] Powered by Google, «Angular - Setting up the local environment and workspace», 2010.  
<https://angular.io/guide/setup-local> (accedido nov. 29, 2020).
- [14] Foundation NodeJS, «About | Node.js», 2020. <https://nodejs.org/en/about/> (accedido nov. 29, 2020).
- [15] MongoDB, «Introducción a MongoDB - Manual de MongoDB», 2008.  
<https://docs.mongodb.com/manual/introduction/> (accedido nov. 29, 2020).
- [16] Mongo DB, «MongoDB Atlas — MongoDB Atlas», 2008.  
<https://docs.atlas.mongodb.com/> (accedido nov. 29, 2020).
- [17] Git is a member of Software Freedom Conservancy, «Git - Acerca del Control de Versiones». <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones> (accedido nov. 29, 2020).
- [18] Git is a member of Software Freedom Conservancy, «Git - Account Setup and Configuration». <https://git-scm.com/book/en/v2/GitHub-Account-Setup-and-Configuration> (accedido nov. 29, 2020).

## Anexos

### Entrevista dirigida al Ing. Fausto Quevedo

Ing. Fausto Quevedo buenos días, soy estudiante de la Universidad Católica de Cuenca, estoy realizando mi tema de tesis en la que voy a realizar un sistema informativo sobre la recolección de desechos domiciliarios me podría ayudar con una pequeña entrevista.

1. Con cuantos vehículos recolectores de desechos cuenta el municipio.

a. Cuantos vehículos están asignados a la recolección de desechos inorgánicos

6 vehículos recolectores con compactación y 1 camión sin compactación.

b. Cuantos vehículos están asignados a la recolección de desechos orgánicos.

2. Con cuanto personal cuenta para realizar la recolección.

c. Con cuantos conductores 7 chóferes.

d. Ayudantes 17 trabajadores.

3. Para la recolección de desechos Uds. disponen de rutas establecidas; cada que tiempo se actualiza.

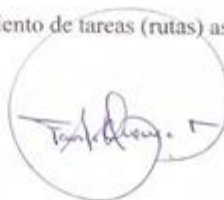
Las rutas se actualizan cada año.

4. Dispone de un formato específico de rutas y horario de recolección de desechos.

Si.

5. Cuentan con algún registro de cumplimiento de tareas (rutas) asignadas

Si.



### Anexo A Entrevista

Además de la entrevista, el jefe departamental nos brindó la siguiente información:

## **Sectores provistos del servicio de recolección de desechos en la ciudad de Azogues.**

Los sectores provistos del servicio de recolección son 13, mismos que se detallan a continuación:

- **Sector Central Comercial (SCC1):** Comprende desde la calle Oriente hasta la Av. Ignacio Neira
- **Sector Central Comercial (SCC2):** Calle Juan Bautista Cordero, Gasolinera Abad y Calle Ingapirca hasta la calle Juan Montalvo y Redondel Sur.
- **Sector Central Residencial 1 (SR1):** Comprende desde la Av. Andres F. Cordova hasta la Av. Luis Monsalve Pozo y desde la calle Oswaldo Hidrovo hasta la Av. Ernesto Che Guevara. En este sector se encuentran incluidos el barrio La Playa, Zhirincay, Cdla. Del Chofer, Ferroviaria, San Antonio, Mutualista Azuay y Cadme Calle.
- **Sector Central Residencial 2 (SR2):** Comprende desde la calle Padre Armando Fajardo Parra y calle Miguel Heredia hasta la calle Oriente y desde la calle Coronel Francisco Carrasco hasta el Cementerio Municipal.
- **Sector Central Residencial 3 (SR3):** Comprende el sector del estadio municipal, sector del ex camal municipal y la Parroquia Borrero hasta el sector de el Corte.
- **Sector Periférico 1 (SP1):** Comprende la Comunidad de Guarangos Grande y Parroquia Javier Loyola hasta Zhullin.
- **Sector Periférico 2 (SP2):** Comprende Chacapamba, Guapán centro, Parroquia Bayas y Uchupucun.
- **Sector Periférico 3 (SP3):** Comprende los sectores de Cachipamba, Guapán Quinua, Pucun Quinua, Zhindilig, Buil Chacapamba y Parroquia Cojitambo.
- **Sector Periférico 4 (SP4):** Comprende el sector de Bolivia, Pampa Vintimilla, Cruz Loma, Guarangos Chico, San Pedro y Parroquia San Miguel de Porotos.
- **Sector Periférico 5 (SP5):** Comunidad de Aguilan.
- **Sector Periférico 6 (SP6):** Comunidad de Guindilig.
- **Sector Periférico 7 (SP7):** Parroquia Luis Cordero, Comunidad de Quillopungo y Comunidad de Leonan.
- **Sector Periférico 8 (SP8):** Parroquias orientales de Taday, Pindilig y Rivera.
- **Recolección de residuos orgánicos:** Comprende la recolección de residuos orgánicos en los sectores comercial 1 y 2 y en los sectores residenciales 1 y 2.

**Anexo B Distribución del servicio de recolección de desechos sólidos inorgánicos**

<b>SECTORES</b>	<b>HORARIO</b>	<b>FRECUENCIAS</b>	<b>VEHICULO</b>	<b>PERSONAL</b>
SCC1	19:30 – 22:30	Lunes a sábado	HINO	1 chofer
SCC2	19:30 – 22:30	Lunes a sábado	HINO	1 chofer
SR1	08:00 – 12:15	Lunes – miércoles Y viernes	HINO	1 chofer
SR2	08:00 – 12:15	Martes – jueves y Sábados	HINO	1 chofer
SR3	08:00 – 12:15	Lunes – miércoles Y viernes	NISSAN	1 chofer
SP1	08:00 – 13:00	Lunes y jueves	NISSAN	1 chofer
SP2	08:00 – 13:00	Martes y viernes	NISSAN	1 chofer
SP3	08:00 – 13:00	Miércoles	NISSAN	1 chofer

**Fuente: Autor**

Tareas del vehículo recolector de marca HINO de 12 yds<sup>3</sup> en los sectores periféricos 8 a 12 son los siguientes:

**Anexo C Sectores periféricos**

<b>DÍA</b>	<b>HORARIO</b>	<b>VEHÍCULO</b>	<b>RECORRIDO</b>
Lunes	08H00-12H00	Compactador HINO	Parroquia Luis cordero, comunidades de Quilopungo y Leonan. Urb. Gómez Arreaga 2, barrio el Mirador de Zhizhiquin, barrio Macas La Playa, Tío Tom, Opara Paccha, Sector De Mururco Y Bellavista, sector de Zhapacal alto de las Antenas y otras tareas
	13H00-17H00	Compactador HINO	Agüilan, autopista desde Sageo hasta el Descanso, escuela 9 de Octubre, Quimandel y otras tareas
Martes	08H00-12H00	Compactador HINO	Comunidad de Güindilig, sector Leg Abuga y barrio el Paraíso y los troncos de Uchupucun, Polígono del Tiro, Vía a Magüarcay, Cachipamba, Guartanchun vía a Saguin y otras tareas.
	13H00-17H00	Compactador HINO	Comunidades de Buil Tabacay y Buil Guapán. Urbanización del Puente Sucre, Cementerio y otras tareas
Miércoles	08H00-12H00	Compactador HINO	Autopista sector el Descanso-Azogues-Sageo, comunidades la Merced, la Dolorosa, comunidades de Zumbahuayco, comunidades de Portete, la Unión, Pedregal y Rayoloma, el Paraíso en Rumihurco, el Pampero, Ayancay, San José, La Villa, el Descanso, parte baja y la Victoria, sector de Zhindilig (concejal Ortiz) y otras tareas.
	13H00-17H00	Compactador HINO	Quisquis, San Miguel, Chabay alto y bajo, sector de la Vieja Posada, Comunidad de Chapte, Macas, San Jacinto, San José de la Parroquia Cojitambo y otras tareas.
Jueves	08H00-12H00	Compactador HINO	Parroquia Luis cordero, comunidades de Quilopungo y Leonan. Urb. Gómez Arreaga 2, barrio el Mirador de Zhizhiquin, barrio Macas La Playa, Tío Tom, sector de Güintul de Luis Cordero, comunidades de las Antenas, Cruz Loma, puente de Ingaloma, Zhapacal Alto, Local de Lomaverde, F. Castanier, sector Guadalupe y otras tareas.
	13H00-17H00	Compactador HINO	Comunidades de Bolivia, Pampa Vintimilla, Guarangos Chico, San Pedro. Parroquia San Miguel de Porotos, Sector Zhapacal Alto, Alto y Cruz blanca (vía que conduce a Luis Cordero) y otras tareas.
Viernes	08H00-12H00	Compactador HINO	Barrios Macas la playa, Sector Leg Abuga, barrio el Paraíso Uchupucun, parroquias orientales de Taday, Pindilig y Zhoray, y otras tareas.
	13H00-17H00	Compactador HINO	Limpieza de vías. - Recolección de restos de vegetación: Reciclaje de grandes productores, recolección de chatarra y electrodomésticos en desuso y otras tareas.

**Fuente: Autor**

Distribución en los sectores comercial, residenciales y periféricos 1,2 y 3.

- Chofer nro. 1: horario a
- Chofer nro. 2: horario b
- Chofer nro. 3: horario c

**Anexo D Distribución de los sectores**

<b>HORARIOS</b>			
<b>DIAS</b>	<b>A</b>	<b>B</b>	<b>C</b>
LUNES	SR1 08:00 A 12:15 SC1 19:30 A 22:30 SR3 08:00 A 13:00	SC2 19:30 A 22:30	SP1 08:00 A 13:00
MARTES	SR2 08:00 A 12:15 SC1 19:30 A 22:30	SC2 19:30 A 22:30	SP2 08:00 A 13:00
MIÉRCOLES	SR1 08:00 A 12:15 SC1 19:30 A 22:30 SR3 08:00 A 13:00	SC2 19:30 A 22:30	SP3 08:00 A 13:00
JUEVES	SR2 08:00 A 12:15 SC1 19:30 A 22:30	SC2 19:30 A 22:30	SP1 08:00 A 13:00
VIERNES	SR1 08:00 A 12:15 SC1 19:30 A 22:30 SR3 08:00 A 13:00	SC2 19:30 A 22:30	SP2 08:00 A 13:00
SABADO	SR2 08:00 A 12:15 SC1 19:30 A 22:30	SC2 18:30 A 22:30	MERC. 18:00 A 22:00

**Fuente: Autor**

Recolección de residuos orgánicos y contenedores (vehículo recolector de marca Hino de 12 yds3).

**Anexo E Recolección contenedores**

<b>SECTORES</b>	<b>HORARIO</b>	<b>FRECUENCIAS</b>	<b>VEHICULO</b>	<b>PERSONAL</b>
SCC1	19:30 – 22:30	Lunes a Sábado	Hino	1 Chofer
SCC2	19:30 – 22:30	Lunes a Sábado	Hino	1 Chofer
SR1	08:00 – 12:15	Lunes – Miércoles y viernes	Hino	1 Chofer
SR2	08:00 – 12:15	Martes – Jueves y Sábados	Hino	1 Chofer
3 cont. Recinto Ferial	08:00 – 13:00	Lunes a domingos	Hino	1 Chofer
6 cont. M. Sucre	08:00 – 13:00	Lunes a sábados	Hino	1 Chofer
1 cont. M. San Francisco	08:00 – 13:00	Lunes a sábados	Hino	1 Chofer
2 cont. T. Terrestre	13:00 – 17:00	Lunes a domingos	Hino	1 Chofer
1 cont. Cementerio	08:00 – 13:00	Lunes y Jueves	Hino	1 Chofer
Autopista desde la bomba de Zumbahuaico hasta el relleno	08:00 – 15:00	Miércoles	Hino	1 Chofer

**Fuente: Autor**

Descripción de los equipos que dispone la Dirección de Gestión Ambiental del GAD Municipal de la parroquia Azogues.

El Equipo que dispone la Dirección de Gestión Ambiental del GAD Municipal de Azogues se presenta en el siguiente cuadro y su estado se los clasifica de acuerdo al siguiente orden:

- A: operable;
- B: Parcialmente operable;
- C: No operable

**Anexo F Vehículos**

<b>Nro. DE ORDEN</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Descripción	Recolector	Recolector	Recolector	Recolector
Disco nro.	2	2	2	2
Chasis marca	Nissan	Nissan	Internacional	Dimex
Chasis año	2008	2008	2001	2001
Tipo de carga	Posterior	Posterior	Posterior	Posterior
Capacidad (yds3)	20	20	16	17
Combustible	Diesel	Diesel	Diesel	Diesel
Estado	A	A	B	C

**Fuente: Autor**

**Anexo G Vehículos**

<b>Nro. DE ORDEN</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
Descripción	Recolector	Recolector	Volquete	Camioneta
Disco nro.	2	2	2	1
Chasis marca	Hino	Mitsubishi	Hino	Chevrolet
Chasis año	2008	2001	2008	2008
Tipo de carga	Posterior	Posterior	Balde	Doble cabina
Capacidad (yds3)	10	10	8 M3	5 pasajeros
Combustible	Diesel	Diesel	Diesel	Gasolina
Estado	B	C	A	A

**Fuente: Autor**



#### Anexo H Vehículos

Nro. DE ORDEN	9	10	11	12	13
Descripción	Camión parroquias orientales	Recolector	Recolector	Recolector	Recolector
DISCO Nro.	1	2	2	2	2
Chasis Marca	Hino	Hino	Hino	Hino	Hino
Chasis Año	2012	2014	2014	2014	2014
Tipo de carga	Cajón	Posterior	Posterior	Posterior	Posterior
Capacidad (yds3)	6 m3	20	20	12	12
Combustible	Diesel	Diesel	Diesel	Diesel	Diesel
Estado	A	A	A	A	A

Fuente: Autor

### Estructura del Sistema de Información Web

#### Estructura del BackEnd

**Package.json**, es un archivo de formato JSON, donde se listan los paquetes utilizados en el proyecto.

**Node\_modules**, contiene las librerías necesarias instaladas para el correcto funcionamiento de la aplicación.

**Gitignore**, es un archivo que nos permite el control de las versiones, es el encargado de ignorar las carpetas y archivos que se van a subir al repositorio.

**Index.js**, Archivo central del Servidor, junta el resto de archivos y los módulos necesarios para que funcione el servidor.

A continuación, se incluye el código fuente del desarrollo del servidor, se incluirá el código ya compilado a JavaScript.

```

1  const express = require('express');
2  const morgan = require('morgan');
3  const cors = require('cors');
4  const app = express();
5
6  const {mongoose} = require('./database');
7
8  // Settings
9  app.set('PORT', process.env.PORT || 3000);
10
11 // Middlewares
12 app.use(morgan('dev'));
13 app.use(express.json());
14 app.use(cors({origin: 'http://localhost:4200'}));
15
16 // Routes
17 app.use('/api/users', require('./routes/user.routes'));
18 app.use('/api/containers', require('./routes/container.routes'));
19 app.use('/api/vehicles', require('./routes/vehicle.routes'));
20 app.use('/api/materials', require('./routes/material.routes'));
21 app.use('/api/employees', require('./routes/employee.routes'));
22 app.use('/api/routes', require('./routes/route.routes'));
23
24
25 // Starting server
26 app.listen(app.get('PORT'), () => {
27   console.log(`Server on port: ${app.get('PORT')}`)
28 });

```

## Anexo I Servidor de Node JS

Fuente: Autor

**Database.js** En este Archivo se encuentra la cadena de conexión hacia nuestra base de datos en la Nube y el módulo necesario para interpretarla

```

1  const mongoose = require('mongoose');
2
3  const URI = "mongodb+srv://admin:admin@recoleccion.wivae.mongodb.net/recoleccion?retryWrites=true&w=majority"
4
5  mongoose.connect(URI, {useNewUrlParser: true, useUnifiedTopology: true, useFindAndModify: false})
6    .then(db => console.log('Conectado a DB'))
7    .catch(e => console.log('Error:', e));
8
9  module.exports = mongoose;

```

## Anexo J Database.js

Fuente: Autor

## Models.js

Dentro de esta carpeta se encuentran declarados todos y cada una de las colecciones de base de datos necesarias para el correcto funcionamiento del Software. A continuación, el listado de las colecciones de base de Datos:

```

1  const mongoose = require('mongoose');
2  const {Schema} = mongoose;
3
4  const EmployeeSchema = new Schema({
5    name: {type: String, required: true},
6    apellido: {type: String, required: true},
7    cargo: {type: String, required: true},
8    telefono: {type: String, required: true},
9    direccion: {type: String, required: true}
10  }, {
11    collection: 'employees'
12  });
13
14  module.exports = mongoose.model('Employee', EmployeeSchema);

```

## Anexo K Employee.js

Fuente: Autor

```

1  const mongoose = require('mongoose');
2  const {Schema} = mongoose;
3
4  const VehicleSchema = new Schema({
5    descripcion: {type: String, required: true},
6    disco: {type: Number, required: true},
7    marca: {type: String, required: true},
8    anio: {type: Number, required: true},
9    carga: {type: String, required: true},
10   capacidad: {type: Number, required: true},
11   combustible: {type: String, required: true},
12   estado: {type: String, required: true}
13  }, {
14    collection: 'vehicles'
15  });
16
17  module.exports = mongoose.model('Vehicle', VehicleSchema);

```

## Anexo L Vehículos.js

Fuente: Autor

```

1  const mongoose = require('mongoose');
2  const {Schema} = mongoose;
3
4  const UserSchema = new Schema({
5    nickname: {type: String, required: true, unique: true},
6    name: {type: String, required: true},
7    surname: {type: String, required: true},
8    email: {type: String, required: true},
9    password: {type: String, required: true},
10   rol: {type: String, required: true}
11  }, {
12    collection: 'users'
13  });
14
15  module.exports = mongoose.model('User', UserSchema);

```

## Anexo M User.js

Fuente: Autor

```

1  const mongoose = require('mongoose');
2  const {Schema} = mongoose;
3
4  const RouteSchema = new Schema({
5    name: {type: String, required: true},
6    des: {type: String, required: true},
7    schedule_begin: {type: String, required: true},
8    schedule_end: {type: String, required: true},
9    schedule_days_runs: {type: [String], required: true},
10   gps: {type: Schema.Types.Mixed, required: true},
11   vehicle: Schema.Types.Mixed,
12   employee: Schema.Types.Mixed,
13 }, {
14   collection: 'routes'
15 });
16
17 module.exports = mongoose.model('Route', RouteSchema);

```

#### Anexo N Route.js

Fuente: Autor

```

1  const mongoose = require('mongoose');
2  const {Schema} = mongoose;
3
4  const ContainerSchema = new Schema({
5    location: {type: String, required: true},
6    cantidad: {type: Number, required: true},
7    schedule_begin: {type: String, required: true},
8    schedule_end: {type: String, required: true},
9    schedule_days_runs: {type: [String], required: true},
10   lat: {type: Number, required: true},
11   lng: {type: Number, required: true},
12   vehicle: Schema.Types.Mixed,
13   employee: Schema.Types.Mixed,
14 }, {
15   collection: 'containers'
16 });
17
18 module.exports = mongoose.model('Container', ContainerSchema);

```

#### Anexo O Container.js

Fuente: Autor

## Controllers

Dentro de esta carpeta están los archivos necesarios para especificar la funcionalidad de cada una de las colecciones de la Base de Datos, estas especificaciones son los procesos CRUD necesarios para el correcto funcionamiento del Software con sus respectivas peticiones que la misma va a realizar para ello se establece lo siguiente.

- GET: esta petición nos permite listar los datos de MongoDB Atlas.
- POST: esta petición nos permite guardar los datos en MongoDB Atlas.
- PUT: esta petición permite actualizar los datos de MongoDB Atlas.
- DELETE: esta petición nos permite eliminar los datos en MongoDB Atlas.

A continuación, se enlistan los Controllers de cada colección de la Base de Datos.

```

1  const Employee = require('../models/employee');
2
3  const employeeCtrl = {};
4
5  employeeCtrl.getEmployees = async (req, res) => {
6    await Employee.find((err, Employees) => {
7      if (err) {
8        console.log('Error:', err);
9        return
10       }
11       res.json(Employees);
12     });
13   }
14
15   employeeCtrl.createEmployee = async (req, res) => {
16     const employee = new Employee({
17       name: req.body.name,
18       apellido: req.body.surname,
19       cargo: req.body.cargo,
20       telefono: req.body.telefono,
21       direccion: req.body.direccion
22     });
23     await employee.save((err) => {
24       if (err) {
25         console.log('Error:', err);
26         return;
27       }
28       res.json({
29         status: 'Empleado guardado'
30       });
31     });
32   }
33
34   employeeCtrl.getEmployee = async (req, res) => {
35     await Employee.findById(req.params.id, (err, employee) => {
36       if (err) {
37         console.log('Error:', err);
38         return;
39       }
40
41       res.json({
42         status: employee
43       });
44     });
45   }
46
47   employeeCtrl.editEmployee = async (req, res) => {
48     const {id} = req.params;
49     const employee = {
50       name: req.body.name,
51       apellido: req.body.surname,
52       cargo: req.body.cargo,
53       telefono: req.body.telefono,
54       direccion: req.body.direccion
55     }
56     await Employee.findByIdAndUpdate(id, {$set: employee}, {new: true}, (err) => {
57       if (err) {
58         console.log('Error:', err);
59         return;
60       }
61
62       res.json({
63         status: 'Empleado actualizado'
64       });
65     });
66   }
67
68   employeeCtrl.deleteEmployee = async (req, res) => {
69     await Employee.findByIdAndRemove(req.params.id, (err) => {
70       if (err) {
71         console.log('Error:', err);
72         return;
73       }
74
75       res.json({
76         status: 'Empleado eliminado'
77       });
78     });
79   }
80
81   module.exports = employeeCtrl;

```

## Anexo P Employee.controller.js

Fuente: Autor

```

1  const Vehicle = require('../models/vehicle');
2  const vehicleCtrl = {};
3
4  vehicleCtrl.getVehicles = async (req, res) => {
5    await Vehicle.find((err, vehicles) => {
6      if (err) {
7        console.log('Error:', err);
8        return
9      }
10     res.json(vehicles);
11   });
12 }
13
14 vehicleCtrl.createVehicle = async (req, res) => {
15   const vehicle = new Vehicle({
16     descripcion: req.body.descripcion,
17     disco: req.body.disco,
18     marca: req.body.marca,
19     anio: req.body.anio,
20     carga: req.body.carga,
21     capacidad: req.body.capacidad,
22     combustible: req.body.combustible,
23     estado: req.body.estado
24   });
25   await vehicle.save((err) => {
26     if (err) {
27       console.log('Error:', err);
28       return;
29     }
30     res.json({
31       status: 'Vehiculo guardado'
32     });
33   });
34 }
35
36 vehicleCtrl.getVehicle = async (req, res) => {
37   await Vehicle.findById(req.params.id, (err, vehicle) => {
38     if (err) {
39       console.log('Error:', err);
40       return;
41     }
42     res.json({
43       status: vehicle
44     });
45   });
46 }
47
48 vehicleCtrl.editVehicle = async (req, res) => {
49   const {id} = req.params;
50   const vehicle = {
51     descripcion: req.body.descripcion,
52     disco: req.body.disco,
53     marca: req.body.marca,
54     anio: req.body.anio,
55     carga: req.body.carga,
56     capacidad: req.body.capacidad,
57     combustible: req.body.combustible,
58     estado: req.body.estado
59   }
60   await Vehicle.findByIdAndUpdate(id, {$set: vehicle}, {new: true}, (err) => {
61     if (err) {
62       console.log('Error:', err);
63       return;
64     }
65     res.json({
66       status: 'Vehiculo actualizado'
67     });
68   });
69 }
70
71 vehicleCtrl.deleteVehicle = async (req, res) => {
72   console.log(req.params.id);
73   await Vehicle.findByIdAndRemove(req.params.id, (err) => {
74     if (err) {
75       console.log('Error:', err);
76       return;
77     }
78     res.json({
79       status: 'Vehiculo eliminado'
80     });
81   });
82 }
83
84 module.exports = vehicleCtrl;

```

## Anexo Q Vehicle.controller.js

Fuente: Autor

```

1  const User = require('../models/user');
2  const userCtrl = {};
3
4  userCtrl.getUsers = async (req, res) => {
5    await User.find((err, users) => {
6      if (err) {
7        console.log('Error:', err);
8        return
9      }
10     res.json(users);
11   });
12 }
13
14 userCtrl.createUser = async (req, res) => {
15   const user = new User({
16     nickname: req.body.nickname,
17     name: req.body.name,
18     surname: req.body.surname,
19     email: req.body.email,
20     password: req.body.password,
21     rol: req.body.rol
22   });
23   await user.save((err) => {
24     if (err) {
25       console.log('Error:', err);
26       return;
27     }
28     res.json({
29       status: 'User guardado'
30     });
31   });
32 }
33
34 userCtrl.getUser = async (req, res) => {
35   await User.findById(req.params.id, (err, user) => {
36     if (err) {
37       console.log('Error:', err);
38       return;
39     }
40
41     res.json({
42       status: user
43     });
44   });
45 }
46
47 userCtrl.editUser = async (req, res) => {
48   const {id} = req.params;
49   const user = {
50     nickname: req.body.nickname,
51     name: req.body.name,
52     surname: req.body.surname,
53     email: req.body.email,
54     password: req.body.password,
55     rol: req.body.rol
56   }
57   await User.findByIdAndUpdate(id, {$set: user}, {new: true}, (err) => {
58     if (err) {
59       console.log('Error:', err);
60       return;
61     }
62
63     res.json({
64       status: 'User actualizado'
65     });
66   });
67 }
68
69 userCtrl.deleteUser = async (req, res) => {
70   await User.findByIdAndRemove(req.params.id, (err) => {
71     if (err) {
72       console.log('Error:', err);
73       return;
74     }
75
76     res.json({
77       status: 'User eliminado'
78     });
79   });
80 }
81
82 module.exports = userCtrl;

```

## Anexo R User.controller.js

Fuente: Autor

```

1  const Route = require('../models/route');
2
3  const routeCtrl = {};
4
5  routeCtrl.getRoutes = async (req, res) => {
6    await Route.find((err, routes) => {
7      if (err) {
8        console.log('Error:', err);
9        return
10      }
11
12      res.json(routes);
13    });
14  }
15
16  routeCtrl.createRoute = async (req, res) => {
17    const route = new Route({
18      name: req.body.name,
19      des: req.body.des,
20      schedule_begin: req.body.schedule_begin,
21      schedule_end: req.body.schedule_end,
22      schedule_days_runs: req.body.schedule_days_runs,
23      gps: req.body.gps,
24      vehicle: req.body.vehicle,
25      employee: req.body.employee
26    });
27    await route.save((err) => {
28      if (err) {
29        console.log('Error:', err);
30        return;
31      }
32
33      res.json({
34        status: 'Ruta guardada'
35      });
36    });
37  }
38
39  routeCtrl.getRoute = async (req, res) => {
40    await Route.findById(req.params.id, (err, route) => {
41      if (err) {
42        console.log('Error:', err);
43        return;
44      }
45
46      res.json({
47        status: route
48      });
49    });
50  }
51
52  routeCtrl.editRoute = async (req, res) => {
53    const {id} = req.params;
54    const route = {
55      name: req.body.name,
56      des: req.body.des,
57      schedule_begin: req.body.schedule_begin,
58      schedule_end: req.body.schedule_end,
59      schedule_days_runs: req.body.schedule_days_runs,
60      gps: req.body.gps,
61      vehicle: req.body.vehicle,
62      employee: req.body.employee
63    }
64    await Route.findByIdAndUpdate(id, {$set: route}, {new: true}, (err) => {
65      if (err) {
66        console.log('Error:', err);
67        return;
68      }
69
70      res.json({
71        status: 'Ruta actualizada'
72      });
73    });
74  }

```

## Anexo S Route.controller.js

Fuente: Autor



```

1  const Container = require('../models/container');
2
3  const containerCtrl = {};
4
5  containerCtrl.getContainers = async (req, res) => {
6    await Container.find((err, containers) => {
7      if (err) {
8        console.log('Error:', err);
9        return
10       }
11       res.json(containers);
12     });
13   }
14
15
16   containerCtrl.createContainer = async (req, res) => {
17     const container = new Container({
18       location: req.body.location,
19       cantidad: req.body.cantidad,
20       schedule_begin: req.body.schedule_begin,
21       schedule_end: req.body.schedule_end,
22       schedule_days_runs: req.body.schedule_days_runs,
23       lat: req.body.lat,
24       lng: req.body.lng,
25       vehicle: req.body.vehicle,
26       employee: req.body.employee
27     });
28     console.log(container)
29     await container.save((err) => {
30       if (err) {
31         console.log('Error:', err);
32         return;
33       }
34       res.json({
35         status: 'Contenedor guardado'
36       });
37     });
38   }
39
40   containerCtrl.getContainer = async (req, res) => {
41     await Container.findById(req.params.id, (err, container) => {
42       if (err) {
43         console.log('Error:', err);
44         return;
45       }
46
47       res.json({
48         status: container
49       });
50     });
51   }
52
53   containerCtrl.editContainer = async (req, res) => {
54     const {id} = req.params;
55     const container = {
56       location: req.body.location,
57       cantidad: req.body.cantidad,
58       schedule_begin: req.body.schedule_begin,
59       schedule_end: req.body.schedule_end,
60       schedule_days_runs: req.body.schedule_days_runs,
61       lat: req.body.lat,
62       lng: req.body.lng,
63       vehicle: req.body.vehicle,
64       employee: req.body.employee
65     }
66     await Container.findByIdAndUpdate(id, {$set: container}, {new: true}, (err) => {
67       if (err) {
68         console.log('Error:', err);
69         return;
70       }
71
72       res.json({
73         status: 'Contenedor actualizado'
74       });
75     });
76   }
77
78   containerCtrl.deleteContainer = async (req, res) => {
79     await Container.findByIdAndRemove(req.params.id, (err) => {
80       if (err) {
81         console.log('Error:', err);
82         return;
83       }
84
85       res.json({
86         status: 'Contenedor eliminado'
87       });
88     });
89   }
90 }
91
92 module.exports = containerCtrl;

```

## Anexo T Container.controller.js

Fuente: Autor

## Routes

En esta carpeta se especifica los archivos necesarios para la navegación dentro del Servidor, esto permite que nuestra RESAPI entienda las diferentes operaciones CRUD. A continuación, se enlista los archivos de route de cada colección de la base de datos:

```
1  const express = require('express');
2  const router = express.Router();
3
4  const employee = require('../controllers/employee.controller');
5
6  router.get('/', employee.getEmployees);
7  router.post('/', employee.createEmployee);
8  router.get('/:id', employee.getEmployee);
9  router.put('/:id', employee.editEmployee);
10 router.delete('/:id', employee.deleteEmployee);
11
12 module.exports = router;
```

### Anexo U Employee.routes.js

Fuente: Autor

```
1  const express = require('express');
2  const router = express.Router();
3
4  const vehicle = require('../controllers/vehicle.controller');
5
6  router.get('/', vehicle.getVehicles);
7  router.post('/', vehicle.createVehicle);
8  router.get('/:id', vehicle.getVehicle);
9  router.put('/:id', vehicle.editVehicle);
10 router.delete('/:id', vehicle.deleteVehicle);
11
12 module.exports = router;
```

### Anexo V Vehicle.routes.js

Fuente: Autor

```
1  const express = require('express');
2  const router = express.Router();
3
4  const user = require('../controllers/user.controller');
5
6  router.get('/', user getUsers);
7  router.post('/', user.createUser);
8  router.get('/:id', user.getUser);
9  router.put('/:id', user.editUser);
10 router.delete('/:id', user.deleteUser);
11
12 module.exports = router;
```

### Anexo W User.routes.js

Fuente: Autor

```
1  const express = require('express');
2  const router = express.Router();
3
4  const route = require('../controllers/route.controller');
5
6  router.get('/', route.getRoutes);
7  router.post('/', route.createRoute);
8  router.get('/:id', route.getRoute);
9  router.put('/:id', route.editRoute);
10 router.delete('/:id', route.deleteRoute);
11
12 module.exports = router;
```

## Anexo X Route.routes.js

Fuente: Autor

```
1  const express = require('express');
2  const router = express.Router();
3
4  const container = require('../controllers/container.controller');
5
6  router.get('/', container.getContainers);
7  router.post('/', container.createContainer);
8  router.get('/:id', container.getContainer);
9  router.put('/:id', container.editContainer);
10 router.delete('/:id', container.deleteContainer);
11
12 module.exports = router;
```

## Anexo Y Container.routes.js

Fuente: Autor

Una vez configurado en el anexo U se observa el servidor levantado correctamente y a la espera de las solicitudes del cliente.

```
PS C:\Users\jalex\Documents\Visual Studio Code\Recoleccion> npm run dev
> MeanStack@1.0.0 dev C:\Users\jalex\Documents\Visual Studio Code\Recoleccion
> nodemon Backend/index.js

[nodemon] 2.0.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node Backend/index.js`
Server on port: 3000
(node:13308) DeprecationWarning: collection.ensureIndex is deprecated. Use createIndexes instead.
Conectado a DB
```

## Anexo Z Servidor Express en funcionamiento

Fuente: Autor

## FrontEnd

A continuación, se incluye el código fuente del desarrollo del FrontEnd, para esto se incluirá el código TypeScript.

En el Anexo X se establecen las rutas a utilizar en el panel administrativo, así como también la API de Google Maps (Directions Service).

```

1  import {NgModule} from '@angular/core';
2  import {HttpClientModule} from '@angular/common/http';
3  import {RouterModule} from '@angular/router';
4  import {CommonModule} from '@angular/common';
5  import {FormsModule, ReactiveFormsModule} from '@angular/forms';
6  import {ClipboardModule} from 'ngx-clipboard';
7  import {AdminLayoutRoutes} from './admin-layout.routing';
8
9  import {NgbModule} from '@ng-bootstrap/ng-bootstrap';
10 /* Componentes */
11 import {DashboardComponent} from './dashboard/dashboard.component';
12 import {ContainersComponent} from './containers/containers.component';
13 import {UsersComponent} from './users/users.component';
14 import {RrhhComponent} from './rrhh/rrhh.component';
15 import {RutasComponent} from './rutas/rutas.component';
16 import {InventarioComponent} from './inventario/inventario.component';
17 import {AgmCoreModule} from '@agm/core';
18 import {AgmDirectionModule} from 'agm-direction';
19
20 // import { ToastrModule } from 'ngx-toastr';
21
22 You, 27 minutes ago | 1 author (You)
23 @NgModule({
24   imports: [
25     CommonModule,
26     RouterModule.forChild(AdminLayoutRoutes),
27     FormsModule,
28     HttpClientModule,
29     NgbModule,
30     ClipboardModule,
31     AgmCoreModule.forRoot({apiKey: 'AIzaSyBP_oi2VSYvva-KeAcjuTa2oxM9PT8c_AI'}),
32     AgmDirectionModule,
33     ReactiveFormsModule,
34   ],
35   declarations: [
36     DashboardComponent,
37     ContainersComponent,
38     UsersComponent,
39     RrhhComponent,
40     RutasComponent,
41     InventarioComponent
42   ]
43 })
44
45 export class AdminLayoutModule {
46 }

```

## Anexo AA admin-layout.module.ts

Fuente: Autor

```
1 import {Component, OnInit} from '@angular/core';
2 import {NgbModal} from '@ng-bootstrap/ng-bootstrap';
3 import {RrhhService} from "../../services/rrhh.service";
4 import {Employee} from "../../models/employee";
5 import Swal from "sweetalert2";
6 import {NgForm} from "@angular/forms";
7 import {HttpClient} from "@angular/common/http";
8
9 @Component({
10   selector: 'app-rrhh',
11   templateUrl: './rrhh.component.html',
12   styles: []
13 })
14 export class RrhhComponent implements OnInit {
15
16   loading: boolean;
17   isEdit: boolean;
18   cargos: any[] = [];
19
20   constructor(private modalService: NgbModal,
21               private _rS: RrhhService,
22               public http: HttpClient) {
23     this.getRrhh().then(() => {
24       this.loading = false;
25     });
26   }
27
28   async getRrhh() {
29     this.loading = true;
30     await this._rS.getRrhh().subscribe(resp => {
31       this._rS.employees = resp as Employee[];
32     }, error => {
33       console.log(`Error: ${error}`);
34     });
35   }
36
37   openWindowCustomClass(content3, isEdit: boolean) {
38     this.modalService.open(content3);
39     if (!isEdit) {
40       this._rS.selectedEmployee = new Employee();
41     }
42     this.isEdit = isEdit;
43   }
44
45   ngOnInit(): void {
46     this.http.get('../../assets/docs/cargos.json')
47       .subscribe((cargos: any[]) => {
48         this.cargos = cargos
49       })
50   }
```

```

52 addEmployee(employeeForm: NgForm) {
53   if (employeeForm.invalid) {
54     Object.values(employeeForm.controls).forEach(control => {
55       control.markAsTouched();
56     });
57     return;
58   }
59
60   if (employeeForm.value._id) {
61     this._rS.putEmployee(employeeForm.value).subscribe(data => {
62       this.getRrhh().then(() => {
63         Swal.fire(
64           'Ok!',
65           'Empleado actualizado correctamente',
66           'success'
67         ).then(() => this.loading = false)
68       ));
69     });
70   } else {
71     this._rS.postEmployee(employeeForm.value).subscribe(data => {
72       this.getRrhh().then(() => {
73         Swal.fire(
74           'Ok!',
75           'Empleado registrado correctamente',
76           'success'
77         ).then(() => this.loading = false)
78       ));
79     });
80   }
81
82   this._rS.selectedEmployee = new Employee();
83   this.modalService.dismissAll();
84 }
85
86 editEmployee(employee: Employee) {
87   this._rS.selectedEmployee = employee;
88   this.isEdit = true;
89 }
90
91 deleteEmployee(_id: string) {
92   Swal.fire({
93     title: 'Desea eliminar el empleado?',
94     text: "No podrá revertir este proceso",
95     icon: 'warning',
96     showCancelButton: true,
97     confirmButtonColor: '#3085d6',
98     cancelButtonColor: '#d33',
99     confirmButtonText: 'Eliminar'
100   }).then((result) => {
101     if (result.isConfirmed) {
102       this._rS.deleteEmployee(_id)
103         .subscribe(res => {
104           this.getRrhh().then(() => {
105             Swal.fire(
106               'Ok!',
107               'Empleado eliminado correctamente',
108               'success'
109             ).then(() => this.loading = false)
110           ));
111         });
112     }
113   })
114 }
115
116 }

```

Anexo BB rrhh.component.ts

Fuente: Autor

```

1 import { Component, OnInit } from '@angular/core';
2 import { NgbModal } from '@ng-bootstrap/ng-bootstrap';
3 import { InventarioService } from './../../services/inventario.service';
4 import Swal from 'sweetalert2';
5 import { NgForm } from '@angular/forms';
6 import { Vehicle } from './../../models/vehicle';
7
8 You, 34 minutes ago | 1 author (You)
9 @Component({
10   selector: 'app-inventario',
11   templateUrl: './inventario.component.html',
12   styleUrls: ['./inventario.component.css']
13 })
14 export class InventarioComponent implements OnInit {
15
16   loading: boolean;
17   isEdit: boolean;
18
19   constructor(private modalService: NgbModal,
20               private _is: InventarioService) {
21     this.getVehicles().then(() => {
22       this.loading = false;
23     });
24   }
25
26   async getVehicles() {
27     await this._is.getVehicles().subscribe(resp => {
28       this._is.vehicles = resp as Vehicle[];
29     }, error => {
30       console.log('Error: ${error}');
31     });
32   }
33
34   openWindowCustomClass(content3, isEdit: boolean) {
35     this.modalService.open(content3);
36     if (!isEdit) {
37       this._is.selectedVehicle = new Vehicle();
38     }
39     this.isEdit = isEdit;
40   }
41
42   ngOnInit(): void {
43   }
44
45   addVehicle(vehicleForm: NgForm) {
46     if (vehicleForm.invalid) {
47       Object.values(vehicleForm.controls).forEach(control => {
48         control.markAsTouched();
49       });
50       return;
51     }

```

```

52  > if (vehicleForm.value._id) {
53  >   this._is.putVehicle(vehicleForm.value).subscribe(data => {
54  >     this.getVehicles().then(() => {
55  >       Swal.fire(
56  >         'Ok!',
57  >         'Vehiculo actualizado correctamente',
58  >         'success'
59  >       ).then(() => this.loading = false)
60  >     });
61  >   });
62  > } else {
63  >   this._is.postVehicle(vehicleForm.value).subscribe(data => {
64  >     this.getVehicles().then(() => {
65  >       Swal.fire(
66  >         'Ok!',
67  >         'Vehiculo registrado correctamente',
68  >         'success'
69  >       ).then(() => this.loading = false)
70  >     });
71  >   });
72  > }
73
74   this._is.selectedVehicle = new Vehicle();
75   this.modalService.dismissAll();
76
77 }
78
79  > editVehicle(vehicle: Vehicle) {
80  >   this._is.selectedVehicle = vehicle;
81  >   this.isEdit = true;
82  > }
83
84  > deleteVehicle(_id: string) {
85  >   Swal.fire({
86  >     title: 'Desea eliminar el vehiculo?',
87  >     text: 'No podrá revertir este proceso',
88  >     icon: 'warning',
89  >     showCancelButton: true,
90  >     confirmButtonColor: '#3085d6',
91  >     cancelButtonColor: '#d33',
92  >     confirmButtonText: 'Eliminar'
93  >   }).then((result) => {
94  >     if (result.isConfirmed) {
95  >       this._is.deleteVehicle(_id)
96  >         .subscribe(res => {
97  >           this.getVehicles().then(() => {
98  >             Swal.fire(
99  >               'Ok!',
100  >               'Vehiculo eliminado correctamente',
101  >               'success'
102  >             ).then(() => this.loading = false)
103  >           });
104  >         });
105  >     }
106  >   })
107  > }

```

Anexo CC inventario.component.ts

Fuente: Autor



```

1  import { Component, OnInit } from '@angular/core';
2  import { NgbModal } from '@ng-bootstrap/ng-bootstrap';
3  import { UserService } from '../../services/user.service';
4  import { User } from '../../models/user';
5  import { NgForm } from '@angular/forms';
6  import { AuthService } from "../../services/auth.service";
7  import Swal from 'sweetalert2';
8  import { HttpClient } from "@angular/common/http";
9
10 @Component({
11   selector: 'app-users',
12   templateUrl: './users.component.html',
13   styleUrls: []
14 })
15 export class UsersComponent implements OnInit {
16
17   loading: boolean;
18   isEdit: boolean;
19   roles: any[] = [];
20
21   constructor(private modalService: NgbModal,
22               private _us: UserService,
23               public auth: AuthService,
24               public http: HttpClient) {
25     this.getUsers().then(() => {
26       this.loading = false;
27     });
28   }
29
30   async getUsers() {
31     this.loading = true;
32     await this._us.getUsers().subscribe(resp => {
33       this._us.users = resp as User[];
34     }, error => {
35       console.log(`Error: ${error}`);
36     });
37   }
38
39   openWindowCustomClass(content3, isEdit: boolean) {
40     this.modalService.open(content3);
41     if (!isEdit) {
42       this._us.selectedUser = new User();
43     }
44     this.isEdit = isEdit;
45   }
46
47   ngOnInit(): void {
48     this.http.get('../../assets/docs/roles.json')
49       .subscribe( (roles: any[]) => {
50         this.roles = roles
51       })
52   }

```

```

54   addUser(userForm: NgForm) {
55     if (userForm.invalid) {
56       Object.values(userForm.controls).forEach(control => {
57         control.markAsTouched();
58       });
59       return;
60     }
61
62     if (userForm.value._id) {
63       this._us.putUser(userForm.value).subscribe(data => {
64         this.getUsers().then(() => {
65           Swal.fire(
66             'Ok!',
67             'User actualizado correctamente',
68             'success'
69           ).then(() => this.loading = false)
70         ));
71       });
72     } else {
73       const userToRegister = userForm.value;
74       this.auth.SignUp(userToRegister.email, userToRegister.password)
75         .then( resp => {
76           this._us.postUser(userToRegister).subscribe(data => {
77             this.getUsers().then(() => {
78               Swal.fire(
79                 'Ok!',
80                 'User registrado correctamente',
81                 'success'
82               ).then(() => this.loading = false)
83             ));
84           });
85         })
86         .catch(e => {
87           Swal.fire(
88             'Error!',
89             e.message,
90             'error'
91           ).then(() => this.loading = false)
92         })
93       );
94
95       this._us.selectedUser = new User();
96       this.modalService.dismissAll();
97     }
98
99     editUser(user: User) {
100       this._us.selectedUser = user;
101       this.isEdit = true;
102     }
103
104     deleteUser(_id: string) {
105       Swal.fire({
106         title: 'Desea eliminar el usuario?',
107         text: "No podrá revertir este proceso",
108         icon: 'warning',
109         showCancelButton: true,
110         confirmButtonColor: '#3085d6',
111         cancelButtonColor: '#d33',
112         confirmButtonText: 'Eliminar'
113       }).then((result) => {
114         if (result.isConfirmed) {
115           this._us.deleteUser(_id)
116             .subscribe(res => {
117               this.getUsers().then(() => {
118                 Swal.fire(
119                   'Ok!',
120                   'User eliminado correctamente',
121                   'success'
122                 ).then(() => this.loading = false)
123               ));
124             });
125         }
126       })
127     }
128
129   }

```

Anexo DD users.component.ts

Fuente: Autor

```

1 import {Component, OnInit} from '@angular/core';
2 import {NgModel} from '@ng-bootstrap/ng-bootstrap';
3 import {RutasService} from "../../services/rutas.service";
4 import {NgForm} from "@angular/forms";
5 import Swal from "sweetalert2";
6 import {Route} from "../../models/route";
7
8 import {InventarioService} from "../../services/inventario.service";
9 import {Vehicle} from "../../models/vehicle";
10 import {RrhhService} from "../../services/rrhh.service";
11 import {Employee} from "../../models/employee";
12

```

You, 41 minutes ago | 1 author (You)

```

13 @Component({
14   selector: 'app-rutas',
15   templateUrl: './rutas.component.html',
16   styles: []
17 })
18 export class RutasComponent implements OnInit {
19

```

```

20   loading: boolean;
21   days = [
22     {
23       value: 'Lunes',
24       des: 'Lunes'
25     },
26     {
27       value: 'Martes',
28       des: 'Martes'
29     },
30     {
31       value: 'Miércoles',
32       des: 'Miércoles'
33     },
34     {
35       value: 'Jueves',
36       des: 'Jueves'
37     },
38     {
39       value: 'Viernes',
40       des: 'Viernes'
41     },
42     {
43       value: 'Sábado',
44       des: 'Sábado'
45     },
46     {
47       value: 'Domingo',
48       des: 'Domingo'
49     }
50   ];
51   lat = -2.7705791;
52   lng = -78.8464126;
53   zoom = 17;
54   origin = {};

```

```

55   waypoints = [];
56   selectedRoute: Route;
57   isEdit: boolean;
58
59   constructor(private modalService: NgModel,
60               private _rt: RutasService,
61               private _rh: RrhhService,
62               private _is: InventarioService) {
63   }
64
65   ngOnInit(): void {
66     this.getRoutes();
67     this.getEmployees();
68     this.getVehicles();
69   }
70
71   onChange(e) {
72     if (e.target.checked) {
73       this._rt.selectedRoute.schedule_days_runs.push(e.target.value);
74     } else {
75       let i: number = 0;
76       this._rt.selectedRoute.schedule_days_runs.forEach((item: string) => {
77         if (item == e.target.value) {
78           this._rt.selectedRoute.schedule_days_runs.splice(this._rt.selectedRoute.schedule_days_runs.indexOf(item), 1);
79           return;
80         }
81         i++;
82       });
83     }
84   }
85
86   assignRoute(route: Route) {
87     this.selectedRoute = route;
88   }
89
90   async getRoutes() {
91     await this._rt.getRoutes().subscribe(resp => {
92       this._rt.routes = resp as Route[];
93     }, error => {
94       console.log('Error: ' + error);
95     });
96   }
97
98   async getEmployees() {
99     await this._rh.getRrhh().subscribe(resp => {
100       this._rh.employees = resp as Employee[];
101     }, error => {
102       console.log('Error: ' + error);
103     });
104   }
105

```

```

110 ~ async getVehiculos() {
111 ~   await this._isGetVehiculos().subscribe(resp => {
112 ~     this._isVehiculos = resp as Vehicle[];
113 ~   }, error => {
114 ~     console.log('Error: $({error})');
115 ~   })
116 ~ }
117
118 ~ openWindowCustomClass(content, isEdit: boolean) {
119 ~   this.modalService.open(content, { size: 'lg' });
120 ~   if (!isEdit) {
121 ~     this._rt.selectedRoute = new Route();
122 ~   }
123 ~   this.isEdit = isEdit;
124 ~ }
125
126 ~ openWindowMap(content) {
127 ~   this.modalService.open(content, { size: 'lg' })
128 ~ }
129
130 ~ /* Agregar marcador al mapa */
131 ~ addMarker(event: any) {
132 ~   const coords: { lat: number, lng: number } = event;
133 ~   const nuevoMarcador = new Marcador(event.coords.lat, event.coords.lng);
134 ~   if (this._rt.selectedRoute.gps.origen.lat === 0 && this._rt.selectedRoute.gps.destination.lat === 0) {
135 ~     this._rt.selectedRoute.gps.origen = nuevoMarcador;
136 ~   } else if (this._rt.selectedRoute.gps.origen.lat != 0 && this._rt.selectedRoute.gps.destination.lat === 0) {
137 ~     this._rt.selectedRoute.gps.destination = nuevoMarcador;
138 ~   } else {
139 ~     this.waypoints.push({ location: { lat: this.destination['lat'], lng: this.destination['lng'] }});
140 ~     this._rt.selectedRoute.gps.waypoints.push({ location: { lat: this._rt.selectedRoute.gps.destination.lat, lng: this._rt.selectedRoute.gps.destination.lng }});
141 ~     this._rt.selectedRoute.gps.destination = nuevoMarcador;
142 ~   }
143 ~ }
144
145 ~ addRoute(routeForm: NgForm) {
146 ~   if (routeForm.invalid) {
147 ~     Object.values(routeForm.controls).forEach(control => {
148 ~       control.markAsTouched();
149 ~     });
150 ~     return;
151 ~   }
152
153 ~   const routes = {
154 ~     ...routeForm.value,
155 ~     employee: this._rt.selectedEmployee,
156 ~     vehicle: this._is.selectedVehicle,
157 ~     gps: this._rt.selectedRoute.gps,
158 ~     schedule_days_runs: this._rt.selectedRoute.schedule_days_runs
159 ~   };
160
161 ~   if (routeForm.value._id) {
162 ~     this._rt.putRoutes(auxItem).subscribe(data => {
163 ~       this.getRoutes().then(() => {
164 ~         Swal.fire(
165 ~           'Ok!',
166 ~           'Ruta actualizado correctamente',
167 ~           'success'
168 ~         ).then(() => this.loading = false)
169 ~       );
170 ~     });
171 ~   } else {
172 ~     this._rt.postRoutes(auxItem).subscribe(data => {
173 ~       this.getRoutes().then(() => {
174 ~         Swal.fire(
175 ~           'Ok!',
176 ~           'Ruta registrada correctamente',
177 ~           'success'
178 ~         ).then(() => this.loading = false)
179 ~       );
180 ~     });
181 ~   }
182
183 ~   this._rt.selectedRoute = new Route();
184 ~   this.modalService.dismissAll();
185 ~ }
186
187 ~ editRoute(route: Route) {
188 ~   this._rt.selectedRoute = route;
189 ~ }
190 ~ deleteRoute(_id: string) {
191 ~   Swal.fire({
192 ~     title: 'Desea eliminar la ruta?',
193 ~     text: 'No podrá revertir este proceso',
194 ~     icon: 'warning',
195 ~     showCancelButton: true,
196 ~     confirmButtonColor: '#3085d6',
197 ~     cancelButtonColor: '#d33',
198 ~     confirmButtonText: 'Eliminar'
199 ~   }).then((result) => {
200 ~     if (result.isConfirmed) {
201 ~       this._rt.deleteRoute(_id)
202 ~         .subscribe(res => {
203 ~           this.getRoutes().then(() => {
204 ~             Swal.fire(
205 ~               'Ok!',
206 ~               'Ruta eliminada correctamente',
207 ~               'success'
208 ~             ).then(() => this.loading = false)
209 ~           );
210 ~         });
211 ~     }
212 ~   })
213 ~ }
214
215 ~ }

```

Anexo EE rutas.component.ts

Fuente: Autor

```
1 import {Component, OnInit} from '@angular/core';
2 import {NgbModal} from "@ng-bootstrap/ng-bootstrap";
3 import {ContainerService} from "../../services/container.service";
4 import {Container} from "../../models/container";
5 import {NgForm} from "@angular/forms";
6 import Swal from "sweetalert2";
7
8 import {InventarioService} from "../../services/inventario.service";
9 import {Vehicle} from "../../models/vehicle";
10 import {RrhhService} from "../../services/rrhh.service";
11 import {Employee} from "../../models/employee";
12
13 @Component({
14   selector: 'app-containers',
15   templateUrl: './containers.component.html',
16   styles: []
17 })
18 export class ContainersComponent implements OnInit {
19   days = [
20     {
21       value: 'Lunes',
22       des: 'Lunes'
23     },
24     {
25       value: 'Martes',
26       des: 'Martes'
27     },
28     {
29       value: 'Miércoles',
30       des: 'Miércoles'
31     },
32     {
33       value: 'Jueves',
34       des: 'Jueves'
35     },
36     {
37       value: 'Viernes',
38       des: 'Viernes'
39     },
40     {
41       value: 'Sábado',
42       des: 'Sábado'
43     },
44     {
45       value: 'Domingo',
46       des: 'Domingo'
47     }
48   ];
49   lat = -2.7705791;
50   lng = -78.8464126;
51   zoom = 15;
52   loading: boolean;
53   isEdit: boolean;
```

```

55 constructor(private modalService: NgbModal,
56             public _cS: ContainerService,
57             public _rh: RrhhService,
58             public _iS: InventarioService) {
59     this.getContainers().then(() => {
60         this.loading = false;
61     });
62 }
63
64 ngOnInit(): void {
65     this.getEmployees();
66     this.getVehicles();
67 }
68
69 async getContainers() {
70     this.loading = true;
71     this._cS.getContainers().subscribe(resp => {
72         this._cS.containers = resp as Container[];
73         console.log(this._cS.containers)
74     }, error => {
75         console.log("Error: ${error}");
76     });
77 }
78
79 async getEmployees() {
80     await this._rh.getRrhh().subscribe(resp => {
81         this._rh.employees = resp as Employee[];
82     }, error => {
83         console.log("Error: ${error}");
84     })
85 }
86
87 async getVehicles() {
88     await this._iS.getVehicles().subscribe(resp => {
89         this._iS.vehicles = resp as Vehicle[];
90     }, error => {
91         console.log("Error: ${error}");
92     })
93 }
94
95 openWindowCustomClass(content3, isEdit: boolean) {
96     this.modalService.open(content3, {size: <any>'lg'});
97     if (!isEdit) {
98         this._cS.selected_container = new Container();
99     }
100     this.isEdit = isEdit;
101 }
102
103 onCheckboxChange(e) {
104
105     if (e.target.checked) {
106         this._cS.selected_container.schedule_days_runs.push(e.target.value);
107     } else {

```

```

108         let i: number = 0;
109         this._cS.selected_container.schedule_days_runs.forEach((item: string) => {
110             if (item == e.target.value) {
111                 this._cS.selected_container.schedule_days_runs.splice(this._cS.selected_container.schedule_days_runs.indexOf(item), 1);
112             }
113             return;
114         });
115         i++;
116     }
117 }
118
119 }
120
121 placeMarker($event) {
122     this._cS.selected_container.lat = $event.coords.lat;
123     this._cS.selected_container.lng = $event.coords.lng;
124 }
125
126 addContainer(containerForm: NgForm) {
127     if (containerForm.invalid) {
128         Object.values(containerForm.controls).forEach(control => {
129             control.markAsTouched();
130         });
131         return;
132     }
133
134     let auxForm = containerForm.value;
135
136     let aux = {
137         ...auxForm,
138         ...this._cS.selected_container
139     }
140
141     if (containerForm.value.id) {
142         this._cS.putContainer(aux).subscribe(data => {
143             this.getContainers().then(() => {
144                 Swal.fire(
145                     'OK!',
146                     'Contenedor actualizado correctamente',
147                     'success'
148                 ).then(() => {
149                     this._cS.selected_container = new Container();
150                     this._rh.selectedEmployee = new Employee();
151                     this._iS.selectedVehicle = new Vehicle();
152                 })
153             })
154         });
155     } else {
156         this._cS.postContainer(aux).subscribe(data => {
157             this.getContainers().then(() => {
158                 Swal.fire(
159                     'OK!',
160                     'Contenedor registrado correctamente',
161                     'success'
162                 ).then(() => {

```

```

162         ).then(() => {
163             this._cS.selected_container = new Container();
164             this._rh.selectedEmployee = new Employee();
165             this._is.selectedVehicle = new Vehicle();
166         })
167     });
168 });
169 }
170
171 this.loading = false
172 this.modalService.dismissAll();
173 }
174
175 editContainer(container: Container) {
176     this._cS.selected_container = container;
177     this.isEdit = true;
178 }
179
180 deleteContainer(_id: string) {
181     Swal.fire({
182         title: 'Desea eliminar el contenedor?',
183         text: 'No podrá revertir este proceso',
184         icon: 'warning',
185         showCancelButton: true,
186         confirmButtonColor: '#3085d6',
187         cancelButtonColor: '#d33',
188         confirmButtonText: 'Eliminar'
189     }).then((result) => {
190         if (result.isConfirmed) {
191             this._cS.deleteContainer(_id)
192                 .subscribe(res => {
193                     this.getContainers().then(() => {
194                         Swal.fire(
195                             'Ok!',
196                             'Contenedor eliminado correctamente',
197                             'success'
198                         ).then(() => this.loading = false)
199                     ));
200                 })
201         }
202     })
203 }
204
205 }

```

## Anexo FF container.component.ts

Fuente: Autor

**Anexo GG Informe Final de Tutoría de Tesis**

Azogues, 09 de marzo de 2021

**Asunto:** Informe final de Tutoría de Tesis

Señora Economista  
Nancy Peralta Idrovo

**Auxiliar de Secretaría de la Unidad Académica de Tecnologías de la Información y Comunicación.**

**UNIVERSIDAD CATÓLICA DE CUENCA SEDE AZOGUES.**

Su despacho. –

De mi consideración:

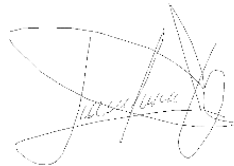
Por medio del presente me permito saludarle y a la vez indicarle que una vez culminada la revisión del trabajo de titulación de los alumnos Jonathan Alexander Peña Peña y Darío Xavier González Miranda, es mi deber conferir la nota de 50/50 al trabajo de titulación en mención, cumpliendo de esta manera con los parámetros establecidos por parte de nuestra Alma Mater como Tutor de la misma, dentro de la Unidad de Titulación.

Dicho proyecto lleva por nombre “Prototipo de un sistema de rutas para la recolección de desechos aplicando geolocalización en el GAD Municipal de la parroquia Azogues””, Previo a la obtención del título de Ingeniero de Sistemas, de la Unidad Académica de Tecnologías de la información y Comunicación.

Es menester informar que el presente trabajo de titulación tiene coincidencia del 4% de similitud de contenidos con otras fuentes, según reporte del sistema anti-plagio (Turnitin) de nuestra Universidad, reporte que se adjunta.

Por la atención que dé a la presente suscribo de Ud. Agradecido.

Atentamente.



Ing. Sebastián Quevedo Sacoto.

**TUTOR**



## Anexo HH Certificado de Turnitin

Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues

### INFORME DE ORIGINALIDAD

4%

INDICE DE SIMILITUD

4%

FUENTES DE  
INTERNET

0%

PUBLICACIONES

2%

TRABAJOS DEL  
ESTUDIANTE

### FUENTES PRIMARIAS

1

[www.xataka.com](http://www.xataka.com)

Fuente de Internet

1%

2

[dawcbtis223guadalupe.blogspot.com](http://dawcbtis223guadalupe.blogspot.com)

Fuente de Internet

<1%

3

[es.scribd.com](https://es.scribd.com)

Fuente de Internet

<1%

4

Submitted to Escuela de Posgrado PNP

Trabajo del estudiante

<1%

5

[github.com](https://github.com)

Fuente de Internet

<1%

6

Submitted to Universidad Distrital FJDC

Trabajo del estudiante

<1%

7

[repositorio.usfq.edu.ec](http://repositorio.usfq.edu.ec)

Fuente de Internet

<1%

8

Submitted to Fundación Universitaria del Area Andina

<1%

9	Submitted to Universidad de Cundinamarca Trabajo del estudiante	<1 %
10	www.ibm.com Fuente de Internet	<1 %
11	uvadoc.uva.es Fuente de Internet	<1 %
12	www.mongodb.com Fuente de Internet	<1 %
13	dspace.unitru.edu.pe Fuente de Internet	<1 %
14	doczz.es Fuente de Internet	<1 %
15	www.gestiopolis.com Fuente de Internet	<1 %
16	zenodo.org Fuente de Internet	<1 %
17	grupo-deco.com Fuente de Internet	<1 %
18	es.itpedia.nl Fuente de Internet	<1 %
19	issuu.com Fuente de Internet	<1 %

20	<a href="http://www.dspace.espol.edu.ec">www.dspace.espol.edu.ec</a> Fuente de Internet	<1%
21	<a href="http://www.floristeriamuguet.com">www.floristeriamuguet.com</a> Fuente de Internet	<1%
22	<a href="http://www.gestiontek.com">www.gestiontek.com</a> Fuente de Internet	<1%
23	<a href="http://www.flotecpump.com">www.flotecpump.com</a> Fuente de Internet	<1%
24	<a href="http://doku.pub">doku.pub</a> Fuente de Internet	<1%

Excluir citas

Apagado

Excluir coincidencias

Apagado

Excluir bibliografía

Activo

Anexo II Certificados de No Adeudar en Biblioteca

 <b>Universidad Católica de Cuenca</b>	<b>CERTIFICADO DE NO ADEUDAR LIBROS EN BIBLIOTECA</b>	CÓDIGO: F – DB – 31 VERSION: 01 FECHA: 2021-07-28 Página <b>106</b> de <b>109</b>
---	---	--

El Bibliotecario de la Sede Azogues

**CERTIFICA:**

Que, **Dario Xavier González Miranda** portador de la cédula de ciudadanía N° 0301879649 de la Carrera de **Ingeniería de Sistemas**, Sede Azogues, Modalidad de estudios presencial no adeuda libros, a esta fecha.

Azogues, **28 de julio de 2021**



Biblioteca Universitaria  
MONS. "FROILAN POZO QUEVEDO"

**Eco. Fabián Rodríguez Herrera**



Universidad  
Católica  
de Cuenca

**CERTIFICADO DE NO ADEUDAR LIBROS EN  
BIBLIOTECA**

CÓDIGO: F – DB – 31  
VERSION: 01  
FECHA: 2021-07-28  
Página **107** de **109**

El Bibliotecario de la Sede Azogues

**CERTIFICA:**

Que, **Jonathan Alexander Peña Peña** portador de la cédula de ciudadanía N° 2100917992 de la Carrera de **Ingeniería de Sistemas**, Sede Azogues, Modalidad de estudios presencial no adeuda libros, a esta fecha.

Azogues, **28 de julio de 2021**

Biblioteca Universitaria  
MONS. "FROILAN POZO QUEVEDO"

**Eco. Fabián Rodríguez Herrera**

**Dario Xavier Gonzalez Miranda** portador(a) de la cédula de ciudadanía N° **0301879649**. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación “**Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues**” de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, 28 de julio de 2021

F: .....

**DARIO XAVIER GONZALEZ MIRANDA**

**0301879649**



**Jonathan Alexander Peña Peña** portador(a) de la cédula de ciudadanía N° **2100917992**. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación “**Sistema de información web para la gestión de datos referente a la recolección de desechos domiciliarios aplicando geolocalización en el GAD Municipal del Cantón Azogues**” de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, 28 de julio de 2021

F: .....

**JONATHAN ALEXANDER PEÑA PEÑA**

**2100917992**