

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE APLICACIÓN WEB Y MÓVIL PARA PROVEER
INFORMACIÓN DE LA RECOLECCIÓN Y MANEJO DE DESECHOS
EN QUITO
BACKEND**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

CINDY KATHALINA YAZÁN CHUQUIMARCA

DIRECTORA: ING. MAYRA ISABEL ALVAREZ JIMENEZ

DMQ, septiembre 2022

CERTIFICACIONES

Yo, Cindy Kathalina Yazán Chuquimarca declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Cindy Yazán

cindy.yazan@epn.edu.ec

cindyk.20@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Cindy Kathalina Yazán Chuquimarca, bajo mi supervisión.



ING. MAYRA ISABEL ALVAREZ JIMENEZ

DIRECTOR

mayra.alvarez@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CINDY KATHALINA YAZÁN CHUQUIMARCA

DEDICATORIA

El presente trabajo es dedicado a Dios, quien me dio la sabiduría, paciencia, fuerza para alcanzar mis objetivos y siempre estuvo a mi lado. A mis padres que me apoyaron desde el inicio y me enseñaron que el estudio es la mejor herencia que me pueden dejar, agradezco que al igual que Dios estuvieron conmigo frente toda adversidad.

CINDY KATHALINA YAZÁN CHUQUIMARCA

AGRADECIMIENTO

Agradezco nuevamente a Dios, por llevarme hasta el final del arduo camino que elegí cursar y darme el amor que necesitaba en mi vida. Agradezco a mi mamá, quien me dio su comprensión en los momentos en los que me quería rendir y enseñarme que con fe en Dios y uno mismo se puede alcanzar cualquier meta. También mi papá, la figura paterna que me enseñó a tener un carácter fuerte y me dio la valentía para realizar cualquier cosa. Gracias a los dos por darme la oportunidad de aprender mi carrera y tener un título profesional. Agradezco a mis hermanas, que igualmente me brindaron su apoyo y me ayudaron en todo momento y mi perrito Tomi que me ayudo a aliviar los momentos de estrés. Además, un agradecimiento a los profesores de la ESFOT de la Escuela Politécnica Nacional, por sus excelentes conocimientos y por saber cómo comunicarlos a sus estudiantes.

CINDY KATHALINA YAZÁN CHUQUIMARCA

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
2 METODOLOGÍA.....	8
2.1 Metodología de Desarrollo.....	8
Roles	8
2.1 Diseño interfaces.....	12
2.2 Diseño de la arquitectura.....	12
Patrón arquitectónico.....	12
2.3 Herramientas de desarrollo.....	13
Librerías	14
3 RESULTADOS	15
3.1 Sprint 0. Configuración del ambiente de desarrollo.....	15
3.2 <i>Sprint</i> 1. Establecimiento de transacciones para el módulo de autenticación.....	17
3.3 <i>Sprint</i> 2. Establecimiento de transacciones para el módulo informativo.....	21
3.4 <i>Sprint</i> 3. Establecimiento de transacciones para el módulo complementario.....	31
3.5 <i>Sprint</i> 4. Pruebas y despliegue del sistema.....	34
4 Conclusiones	38
5 Recomendaciones	39
6 REFERENCIAS BIBLIOGRÁFICAS.....	40
7. ANEXOS.....	42
ANEXO I	43
ANEXO II	44

Recopilación de requerimientos	44
Historias de usuarios.....	45
Product Backlog.....	49
Sprint Backlog.....	50
Diseño de prototipos	1
Esquema de la base de datos.....	1
Pruebas	4
Pruebas de seguridad	4
Pruebas de compatibilidad	5
Pruebas de carga.....	7
ANEXO III	9
ANEXO IV.....	10

RESUMEN

El incremento de habitantes ha sido directamente proporcional al aumento de desechos y contaminación. El espacio para acumular desperdicios es limitado y estos desembocan en la contaminación del medio ambiente. Los barrios de Quito son afectados por los desechos que se generan a diario, por tanto, es necesario brindar información a los ciudadanos que propicie la mitigación de problemas de acumulación y contaminación debido a la desinformación.

Dados estos antecedentes, se plantea informar a la comunidad sobre la recolección y manejo de desechos en la ciudad de Quito. En esta sección nos centramos en el desarrollo del backend de un sistema con temática de reciclaje, horarios y rutas de los camiones recolectores de basura en Quito. El backend aquí propuesto es la plataforma de datos de los sistemas web y móvil que proveerán información sobre el manejo de desechos, consejos, noticias, videos y un apartado de comentarios o sugerencias sobre el servicio de recolección de basura o acerca de la aplicación en uso.

Este documento está estructurado de la siguiente forma: La primera sección, contiene los antecedentes, objetivos y alcance del proyecto. La segunda sección aborda la metodología elegida (*Scrum*), artefactos de recopilación de requerimientos, diseño de interfaces y arquitectura (MVC) aplicada en el proyecto. La tercera sección contiene el desarrollo de tareas y resultados acorde a los requerimientos, además de pruebas realizadas. Por último, se encuentran conclusiones y recomendaciones que se obtuvieron en este proyecto.

PALABRAS CLAVE: Sistema, Móvil, Reciclaje, *Scrum*, Web, Medio Ambiente.

ABSTRACT

The increase in inhabitants has been directly proportional to the increase in waste and pollution. The space to accumulate waste is limited and these lead to environmental pollution. The neighborhoods of Quito are affected by the waste that is generated daily, therefore, it is necessary to provide information to citizens that promotes the eradication of accumulation and pollution problems due to misinformation.

Given this background, it is proposed to inform the community about the collection and management of waste in the city of Quito. In this section we focus on the development of the backend of a recycling-themed system, schedules, and routes of garbage trucks in Quito. The backend proposed here is the data platform of the web and mobile systems that will provide information about waste management, advice, news, videos and a section for comments or suggestions about the garbage collection service or about the application in use.

This document is structured as follows: The first section contains the background, objectives, and scope of the project. The second section deals with the chosen methodology (Scrum), requirements gathering artifacts, interface design and architecture (MVC) applied in the project. The third section contains the development of tasks and results according to the requirements, as well as tests performed. Finally, conclusions and recommendations that were obtained in this project are found.

KEYWORDS: System, Mobile, Recycling, Scrum, Web, Environment.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El manejo de desperdicios generados por los hogares en todo el municipio de Quito es administrado por la entidad pública EMASEO EP (Empresa Pública Metropolitana de Aseo de Quito) [1]. Esta empresa brinda servicios como la recolección domiciliaria y una gestión ambiental para tratar la basura.

La generación de basura ha ido en incremento en el tiempo de pandemia por la compra de productos a domicilio [2] y con ello las envolturas, fundas y plásticos que son usados comúnmente para preservar los productos, pero esta no es la única fuente de creación de desechos dado a que las familias compran más para abastecerse en la cuarentena, a esto se suma los desechos médicos como mascarillas y guantes quirúrgicos para la protección del COVID-19. Esto solo puede significar que el incremento en la generación de residuos es causado por los hogares [2].

Hemos identificado que una problemática presente en la comunidad quiteña está relacionada al escaso conocimiento de la población sobre horarios y rutas de recolección de basura, consejos para separar los desechos comunes, ayudas visuales que aporten conciencia al medio ambiente e información de contacto sobre recicladores individuales.

La gestión de los servicios de recolección, información sobre el trato de residuos es de suma importancia debido a que la generación de desperdicios es diaria y los actores principales somos los residentes de Quito que necesitamos el conocimiento de reciclado y puntos de recolección de basura para un mejor cuidado del ambiente.

Existen varias opciones para comenzar a adquirir conocimientos, en el mercado encontramos una aplicación para dispositivos móviles llamada iRecycle de Earth911 que informa a las personas acerca del proceso de reciclaje, a encontrar sitios de reciclaje en la zona con su dirección, hora de apertura y número telefónico. Además, ofrece una sección de noticias con las últimas novedades del sector del reciclaje [3]. Sin embargo, al ser una aplicación extranjera no se han registrado sitios de reciclado en Ecuador, su interfaz y artículos están en el idioma inglés.

Siendo este tipo de aplicaciones parte de una solución a la falta de conocimiento de manejo de desechos, es aquí donde la idea de proyecto cobra sentido dado que se desea brindar una opción cercana a la comunidad quiteña sobre el tema de reciclado, horarios y rutas de recogida de basura a través de un mapa, noticias y videos sobre manejo de residuos y reciclaje, consejos para reutilizar, procesos de reciclado y manejo de desperdicios de COVID-19, un espacio para comentar o sugerir cambios en la aplicación

o problemas con la basura en su barrio. Por último, pero no menos importante, está la sección de información de contacto de recicladores individuales o entidades dedicadas a este campo.

El proyecto aquí propuesto abarca la lógica del sistema web y móvil, se proporcionarán los datos necesarios para el levantamiento de ambos sistemas. Este componente almacena la lógica completa de negocio para acceder y administrar los datos de las aplicaciones, contribuyendo a parte de una solución que se desea otorgar a la población de Quito, hemos llamado a nuestra propuesta TrashInformation.

1.1 Objetivo general

Desarrollar el backend para la administración de información sobre la recolección y manejo de desechos en la ciudad de Quito de las aplicaciones móvil y web.

1.2 Objetivos específicos

1. OE1: Diseñar la infraestructura de la base de datos para las aplicaciones frontend.
2. OE2: Gestionar la información necesaria para alimentar el frontend de las aplicaciones móvil y web.
3. OE3: Trabajar con el frontend para transferir datos de forma eficiente a las aplicaciones web y móvil.
4. OE4: Mantener la infraestructura del backend disponible para las solicitudes que necesite el frontend.

1.3 Alcance

El presente proyecto propone el desarrollo de una aplicación web y móvil que provea información a los quiteños sobre temas relacionados a la recolección, tratamiento adecuado y oportuno de los desechos.

La aplicación web cumplirá la función de administrar los datos que se presentarán al usuario a través de la aplicación móvil. Aquí se realizarán acciones de: agregar, eliminar y modificar las noticias, videos, consejos relacionados al manejo, clasificación de desechos y temas relacionados, y la lista de contactos de personas que hacen reciclaje en las calles. Además, el administrador tendrá la capacidad de revisar las sugerencias de cada

usuario y hacer los cambios respectivos. En esencia la aplicación web se encargará de suministrar información a la aplicación móvil.

La aplicación móvil tendrá por un lado un mapa interactivo que muestre días, horarios y rutas de recolección de basura en Quito por sectores, el cual será tomado con referencia de la Empresa Pública Metropolitana de Aseo de Quito (EMASEO). También se implementará varias secciones informativas sobre el tratamiento, clasificación, consejos, noticias y videos enfocados al manejo y reciclaje de desechos sólidos; junto con información de contactos de personas que realizan reciclaje y recolección de chatarra en las calles. En adición a lo anterior, se implementará un formulario por el cual los usuarios emitan las sugerencias o comentarios que surjan a partir de cualquier apartado de la aplicación móvil o de la información que se está abarcando. De esta manera la propuesta es crear una herramienta que informe de manera precisa sobre el manejo de los desechos, su clasificación, la mejor manera de llevar a cabo el reciclaje, y evidenciar los efectos positivos para la sociedad y el ambiente.

La propuesta del sistema web y móvil en la parte del backend cuenta con una serie de funcionalidades que se evidencian a continuación:

Perfil administrador del frontend:

- Acceso a la base de datos en tiempo real.
- Almacenamiento de datos en forma de JSON y en la nube.
- Solicitar peticiones de escritura y lectura en la base de datos.
- Autenticar perfiles de usuarios.
- Enviar notificaciones de recuperación de contraseña.

1.4 Marco Teórico

En esta sección se hablará sobre conceptos generales de metodología y otros componentes que se emplearan para el desarrollo del presente proyecto, se detallan lo que son las metodologías ágiles y las herramientas consideradas para el desarrollo del backend.

Metodologías

La sociedad de conocimientos requiere de un método que le permita reflejar sus ideales de forma científica, este es el primer paso para la creación de metodologías científicas

que permiten plasmar los procedimientos del conocimiento científicos, las etapas por las que se somete la investigación, las técnicas conexas a dicho método [4]. En resumen, la metodología hace referencia a ese conjunto de aspectos técnicos y operativos que son sustanciales para realizar un estudio. Como en todo estudio la metodología está presente en el desarrollo de software, esta será la base para llevar a cabo el desarrollo de nuevo software. La metodología empleada permitirá tener el camino a las soluciones de las problemáticas. Este método permitirá que el proyecto tenga entregas a tiempo, errores nulos en los sistemas y buenas conclusiones en los procesos de desarrollo [4].

Durante el desarrollo de software se han generado varias metodologías que ayuden a que los proyectos sean un éxito y otorgar un producto de calidad a los clientes. Es indispensable elegir la metodología correcta puesto que de ella dependerán los resultados del producto [5]. Esta deberá ser robusta y que se ajuste a las metas del equipo.

Metodologías tradicionales.

Al inicio el desarrollo de software no se guiaba a métodos, fue la necesidad de mejorar los procesos y alcanzar el objetivo principal que se tuvo que ajustar las metodologías existentes de otras áreas al desarrollo de software. Se eligió, como mejor opción al despliegue de procesos de manera secuencial, lo cual mejoro en su momento la creación de software [5]. Las metodologías tradicionales se centraban en la elaboración de plantillas técnicas, documentación detallada, en la planificación y procesos. Sin embargo, pese al éxito, el desarrollo de software tuvo que enfrentar problemas como la inflexibilidad en los procesos por la formalidad, evaluación de riesgos compleja y alto compromiso del cliente al pedirle que describa el proyecto a gran nivel de detalle [5].

Metodologías ágiles.

Los problemas debían tener una inmediata solución y luego de varias opiniones en contra o a favor de las metodologías tradicionales se generan los nuevos métodos ágiles. Estos serían los nuevos puntos de vista para el desarrollo de software. El retraso de decisiones puntuales y una planificación adaptativa permitió potenciar el desarrollo de software a gran escala [6]. Las metodologías ágiles proponen que la capacidad de respuesta a un cambio es más importante que seguir un plan estricto, esta flexibilidad en los procesos abaratará costos y tendrá en alerta al equipo frente a cualquier cambio. Entre los métodos ágiles más usados se puede destacar la *eXtreme Programming* o Programación extrema (XP), Scrum, Kanban, Lean, entre otras. En la TABLA I, se muestra la comparación entre diferentes metodologías ágiles [6].

TABLA I. Comparación entre metodologías ágiles.

	XP	Kanban	Scrum
Definición	Es un proceso de desarrollo de software centrado en la adaptabilidad, se consideran los cambios de requisitos sobre la marcha. La capacidad de adaptación es mejor y realista que definir los requisitos desde un inicio.	Es un proceso evolutivo e incremental frente al cambio de sistemas para las organizaciones. Lleva a cabo las tareas pendientes y principios importantes.	Proceso liviano para gestionar y controlar el desarrollo de software.
Características	<ul style="list-style-type: none"> - Desarrollo en iteraciones e incremental. - Pruebas unitarias continuas. - Programación por parejas. - Interacción de equipo. 	<ul style="list-style-type: none"> - No requiere configuración y puede ser aplicada sobre flujos de trabajo cualquiera. - Busca e implementa cambios evolutivos. 	<ul style="list-style-type: none"> - Desarrollo en iteraciones e incremental. - Cada iteración termina con una funcionalidad del proyecto. - El equipo se enfoca en la construcción de software de calidad
Ventajas	<ul style="list-style-type: none"> - Apropiado para entornos cambiantes. - La preparación para los cambios reduce costos. - Definir por cada iteración los objetivos 	<ul style="list-style-type: none"> - Su organización tiene una tabla en donde se puede ver todo el flujo de trabajo. - Hace que el cumplimiento sea en tiempos concretos. - Se realiza una distribución visual de tareas en una tabla. 	<ul style="list-style-type: none"> - Definir las características del producto y remueve cualquier obstáculo que entorpezca la tarea del equipo. - Tiene pocas y simples reglas basadas en la inspección continua.

Desventajas	<ul style="list-style-type: none"> - Se debe delimitar el alcance directamente con el cliente. 	<ul style="list-style-type: none"> - Estrictas políticas y muy costoso para equipos grandes. - Tiene un limitado cuando se trata de trabajos extensos por la gran cantidad de tareas que es imposible aplicar esta metodología. 	<ul style="list-style-type: none"> - Funcionamiento propicio con equipos reducidos. - Exhaustiva definición de tareas y sus plazos.
--------------------	---	---	---

Basados en la tabla comparativa se ha decidido usar la metodología ágil Scrum dado los beneficios de trabajo en equipo. Este proyecto se lo ha planteado en grupo por lo que esta metodología permitirá que el equipo de desarrollo encuentre un ámbito propicio para desenvolverse en cada sección del proyecto. Además, se adapta a los cambios de forma inmediata lo cual contribuirá en momentos críticos. En cada iteración o sprint se obtiene un resultado de las funcionalidades planteadas, esto ayuda a que el proyecto sea cumplido en el plazo acordado.

Los sistemas web y móvil se componen de dos secciones: la interfaz de usuario donde el cliente puede interactuar con los datos presentados en pantalla y el funcionamiento del producto que hace referencia al almacenamiento, transformación, envío y recepción de peticiones. Estos dos conceptos son conocidos como frontend y backend.

Backend

El backend se encarga de manejar los datos de la aplicación web y móvil. Contiene toda la lógica para acceder y administrar los datos, la parte que los usuarios no pueden visualizar. El backend también es responsable de manejar las solicitudes y respuestas de la web. Esta parte del desarrollo de aplicaciones funciona en conjunto con el frontend para entregar el producto final. La TABLA II, muestra algunas opciones de herramientas para el desarrollo del backend.

TABLA II Comparación entre herramientas de desarrollo para el backend

	Firestore	Api Restful	Appwrite
Definición	Plataforma en la nube enfocada en el desarrollo de aplicaciones web y móvil.	Desarrollo del backend basado en una API para asegurarse de que el consumo sea desde cualquier sistema o cliente.	Solución para el backend para aplicaciones web o móviles.
Características	<p><i>Realtime Database:</i> Base de datos en tiempo real, no SQL. Almacena los datos en forma de JSON.</p> <p>Autenticación de usuarios: Posee un sistema de autenticación que permite registrar al usuario mediante contraseña y correo.</p> <p>Almacenamiento en la nube: Sistema de almacenamiento de ficheros.</p> <p>Test Lab: Laboratorio que permite testear la aplicación en dispositivos Android virtuales para detección de errores.</p>	<p>Uso de varias librerías o <i>frameworks</i> como PHP, NodeJs o tecnología .NET.</p> <p>Una API REST trabaja con bases de datos NoSQL que permiten gran volumen de información a gran velocidad de procesamiento.</p>	<p>Esta herramienta es de código abierto, con cero dependencias y se integra fácilmente con algunas de las herramientas y lenguajes más populares. <i>Appwrite</i> es un servidor backend que puede ser instalado en cualquier sistema operativo que admita Docker. Esta herramienta viene con un panel que permite la administración de aplicaciones, sencilla y con documentación.</p>

Una vez analizadas y comparado diferentes herramientas de desarrollo, se ha optado por Firestore como plataforma que controle en backend, dado sus características de autenticación de usuarios, capacidad de almacenamiento de datos en la nube, envío de notificaciones al usuario y por el conocimiento previo en otros proyectos. Además, esta opción hace más sencilla la creación de aplicaciones móviles y web ya que se acopla a los objetivos que necesita el frontend en ambos sistemas.

2 METODOLOGÍA

Las metodologías ágiles de desarrollo de software son procesos que ayudan a distribuir los sistemas de software en iteraciones de tiempo determinado. Las metodologías de software buscan facilitar a corto plazo las funcionalidades de un software para que el cliente se sienta satisfecho con cada proceso de desarrollo. Las metodologías son flexibles y permite el trabajo en equipo que ofrece mejoras contantes.

El presente componente usa el tipo de trabajo experimental donde se realizan una serie de pruebas para verificar el funcionamiento del componente backend. Cada prueba efectuada tiene un resultado sea este exitoso o no, esto permitirá que si un error surge se pueda corregir a tiempo. Las pruebas deben ser llevadas a cabo constantemente para comprobar que cada proceso este en óptimo funcionamiento.

2.1 Metodología de Desarrollo

Las metodologías de desarrollo de software ayudan a comprobar las funciones del sistema progresivamente. Estas cuentan con un conjunto de herramientas como son los roles y artefactos que permiten concretar el proyecto en un tiempo corto y estar preparado para algún cambio que pueda surgir en el desarrollo sin afectar la calidad de este [7].

La metodología seleccionada para trabajar entre varias alternativas es Scrum. Esta integra mejoras y fomenta el trabajo en equipo. Los resultados y mejoras en el proyecto dependerán directamente de la colaboración de equipo. En Scrum se clasifican las tareas dependiendo de su impacto, estas son entregadas en un periodo determinado, por esta razón es que Scrum es usada en proyectos que se modifican al paso y que sea necesario alcanzar los resultados rápidamente [8]. Los resultados siempre serán en función de los requerimientos del cliente.

Todos los involucrados pueden ofrecer sus conocimientos y habilidad de planeación del entorno para toma de decisiones al instante de un inconveniente durante el desarrollo. A continuación, se describen los componentes que conforman la metodología ágil Scrum.

Roles

Los roles para el presente proyecto se han planteado como *Product Owner*, *Scrum Master* y Equipo de desarrollo. Estos tendrán una acción constante a lo largo del proyecto para lograr el resultado deseado.

Product Owner

Este es el personaje encargado de guiar al equipo de trabajo con las tareas específicas para cada integrante. Es el interesado en el proyecto y el resultado de este por lo que es el encargado de reunir los requerimientos del producto a desarrollar y de poder rechazarlos. Este rol prioriza las actividades para que el equipo se centre en las tareas según las necesidades del negocio [9]. En la TABLA III, se visualiza la persona encargada de este rol.

Scrum Máster

Esta es la persona encargada del liderazgo del equipo y de enseñar a todos los integrantes como va a ser el proceso Scrum, si es necesario le enseña a cada uno de los que interfieren en el proyecto. Este se asegura de que el equipo siga las reglas y prácticas de Scrum. Este personaje realiza las reuniones diarias para conocer el proceso de las tareas e identificar alguna interferencia que impida el flujo normal del equipo. Además, se encarga de reunirse con el *Product Owner* para compartir los avances y si existe algún cambio poder comunicarlo al Equipo de desarrollo [9]. Así se puede ver en la TABLA III, quien es la persona que toma este rol.

Equipo de desarrollo

Esta parte del equipo está enfocado al desarrollo de cada una de las tareas que proporcionó el *Product Owner*. Estas personas son las encargadas de convertir los requerimientos en iteraciones donde cada uno tendrá un resultado que avance con las funcionalidades del trabajo [9]. Como se observa en la TABLA III, la persona que responsable de la parte del *backend* de la aplicación.

TABLA III: Roles del proyecto

ROLES	NOMBRES
<i>Product Owner</i>	Ing. Alvarez Mayra
<i>Scrum Master</i>	Ing. Alvarez Mayra
<i>Equipo de desarrollo</i>	Cindy Yazán

Recopilación de Requerimientos

Se define como recopilación de requerimientos como la propuesta que el cliente tiene en mente, donde se estipulan las funcionalidades que el producto debe poseer. Aquí se

determina el alcance del proyecto [9]. La TABLA IV, muestra un ejemplo de cómo se ha formulado la recopilación de requerimientos, así toda la información obtenida esta ordenada. Se pueden ver los demás requerimientos en la sección de ANEXO II, sección Recopilación de requerimientos.

TABLA IV: Información para la recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - R	ENUNCIADO DEL ÍTEM
BACKEND	RR001	Como usuario administrador y ciudadano del sistema necesitan transacciones de lectura y escritura para iniciar sesión y cerrar sesión.
	RR002	Como usuario administrador y ciudadano del sistema necesitan transacciones de lectura y escritura para registrarse.

Historias de Usuario

En esta sección se describen las funcionalidades del software, son el producto del trabajo en equipo del cliente y el equipo que evolucionarán durante todo el proyecto. La TABLA V, presenta un ejemplo de una Historia de Usuario, las demás historias se la pueden encontrar en el ANEXO II, sección Historias de usuarios.

TABLA V: Modelo de las Historias de Usuario

Historia de Usuario	
Identificador: HU-01	Usuario: Administrador y Usuario Ciudadano
Nombre Historia: Generar transacciones de lectura, escritura y consultas para el inicio y cierre de sesión de la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Alta	Riesgo en Desarrollo (Alta/Medio/Baja): Alta
Iteración Asignada: 1	
Responsables (es): Cindy Yazán	
Descripción: El usuario administrador en el backend podrá acceder a transacciones de escritura, lectura y consulta en lotes para generar el inicio y cierre de sesión del usuario administrador o usuario ciudadano mediante un correo electrónico y una contraseña	
Observación: El inicio y cierre de sesión será en función del rol administrador o usuario ciudadano, dependiendo de este la interfaz cambiará. Además, el inicio de sesión se lo realizará únicamente mediante un correo y contraseña.	

Product Backlog

Es el llamado inventario donde se guardan las funcionalidades en forma de lista ordenadas por prioridad de desarrollo. Estas funciones o requisitos se irán adquiriendo con el avance de las iteraciones [8]. En la TABLA VI, se muestra un ejemplo de cómo se ha realizado el *Product Backlog*. La tabla completa se puede encontrar en el ANEXO II, sección Product Backlog.

TABLA VI: Modelo del Product Backlog

ID-HU	HISTORIA DE USUARIO	ITERACIÓN	PRIORIDAD
HU-01	Generar transacciones de lectura y escritura para el inicio de sesión en la aplicación.	1	Alta
HU-02	Generar transacciones de lectura y escritura para el inicio de sesión en la aplicación.	1	Alta

Sprint Backlog

El Sprint Backlog es el listado de tareas que el equipo ha elaborado para planificar cada Sprint, las tareas son asignadas con su correspondiente tiempo [8]. Se muestra la TABLA VII, con el modelo de las tareas de cada Sprint planteado. Se puede visualizar a detalle la tabla completa en el ANEXO II, sección Sprint Backlog.

TABLA VII: Modelo del Sprint Backlog

ID-SB	Nombre	Historia de Usuario	Tareas	Tiempo Estimado (horas)
S0	Configuración del ambiente de desarrollo	N/A	<ul style="list-style-type: none">• Creación de base de datos• Creación del proyecto• Definición de usuarios• Diseño Wireframes	40 h

2.1 Diseño interfaces

Luego de identificar los requerimientos y documentarlos mediante el uso de artefactos de la metodología, se realiza el diseño de las interfaces de usuario. En la Fig.1, se puede observar un ejemplo del prototipo, esta es una muestra de los mockups que se realiza por parte del frontend. Cabe mencionar que este documento se centra en el desarrollo del backend del sistema. En el ANEXO II, sección Diseño de prototipos, se puede encontrar los enlaces al prototipado.

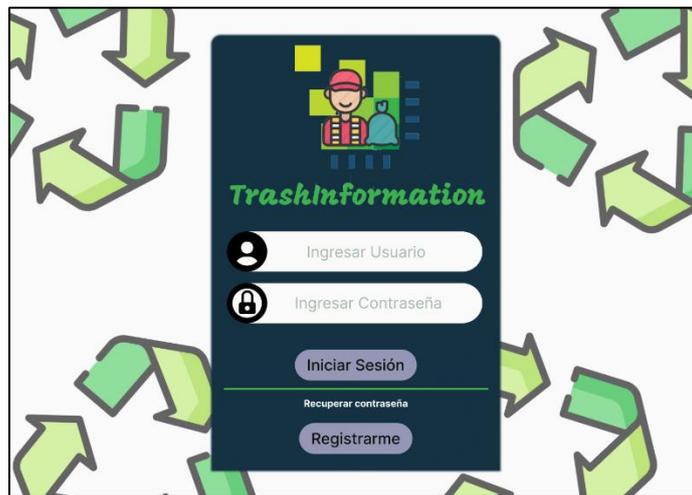


Fig.1: Interfaz de Inicio de sesión del sistema

2.2 Diseño de la arquitectura

El diseño de la arquitectura permitirá al equipo de desarrollo comenzar con el trabajo de cada Sprint. La definición de la arquitectura hará posible la implementación de los módulos y conexiones entre herramientas y bibliotecas. Con esta explicación de la arquitectura se puede generar calidad, correcciones y futuras nuevas funcionalidades en el producto [10]. A continuación, se ha definido el patrón arquitectónico ideal para el desarrollo del proyecto.

Patrón arquitectónico

La arquitectura que guiará al backend será el de Modelo Vista Controlador (MVC), este estilo de arquitectura clasifica los datos de la aplicación, la interfaz de usuario y la lógica del negocio en tres diferentes componentes permitiendo que unos componentes no afecten a otros. Este tipo de arquitectura es utilizada por varias empresas de desarrollo de software [11].

- **Modelo:** define las reglas del negocio, permite llevar un control del acceso a almacenamiento de datos por medio de consultas.
- **Vista:** muestra la información al usuario mediante un apartado visual con la que interactuara el cliente final.
- **Controlador:** es el intermediario entre el Modelo y la Vista, este es el encargado de recibir la información de entrada para tratarla íntegramente para el usuario.

En la Fig.2, se muestra el diseño del patrón arquitectónico MVC el cual genera la infraestructura que tendrá el proyecto, este maneja el backend con el frontend para que el cliente pueda interactuar con la interfaz gráfica.

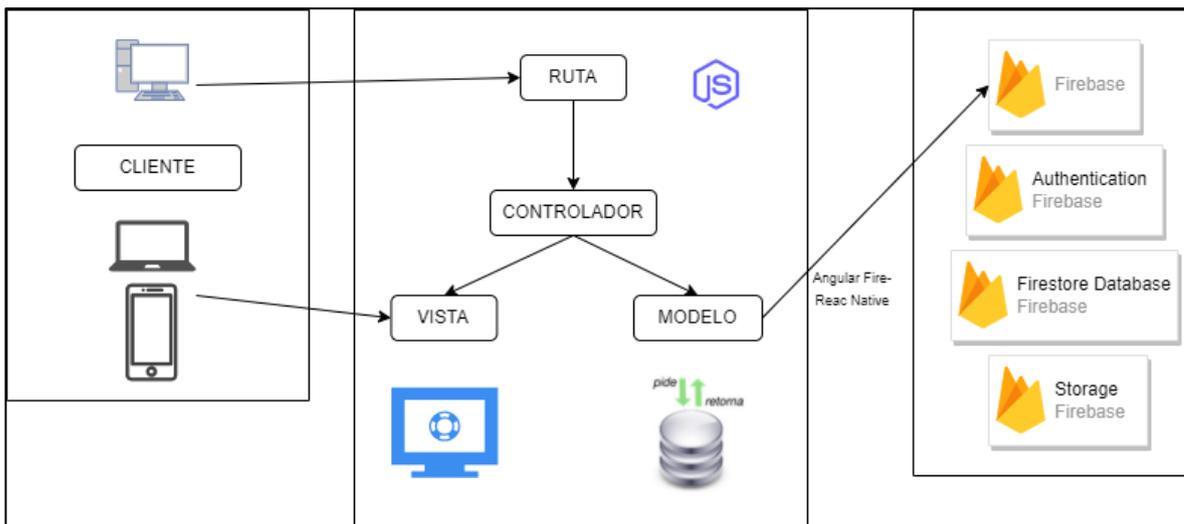


Fig.2: Diseño de la arquitectura del sistema

2.3 Herramientas de desarrollo

El desarrollo de software posee herramientas que ayudan a los desarrolladores a plasmar las funcionalidades físicamente. Estas herramientas apoyan en las diferentes etapas de la creación de software, especialmente en la de codificación. Con ellas se puede modelar, diseñar, integrar y realizar pruebas a los procesos creados [12]. La TABLA VIII, presenta las herramientas se han empleado para el desarrollo del *backend* y las consultas a la base de datos no relacional NoSQL.

TABLA VIII: Herramientas para el desarrollo del proyecto

HERRAMIENTA	JUSTIFICACIÓN
Node.js	Esta herramienta es un entorno de ejecución para JavaScript. Aquí se encuentran los paquetes necesarios para la ejecución de programas.
Firebase	Servicio creado por <i>Google</i> enfocada para el desarrollo rápido de aplicaciones web y móviles. Posee diferentes funciones que permite una mejor forma de controlar el backend de la aplicación.
Cloud Firestore	Base de datos <i>NoSQL</i> en la nube que almacena datos, el manejo de este es flexible. Además, están sincronizados si se usan en varias aplicaciones.
Firebase Authentication	Como su nombre menciona es la identificación de usuarios. Una gran cantidad de aplicaciones necesitan la identidad del usuario para guardar los datos generados por el mismo. <i>Autheticacion</i> usa varios servicios de identificación como son correo electrónico y redes sociales.
Storage	El <i>Storage</i> de <i>Firebase</i> se basa en el almacenamiento de contenido que el usuario dota a la aplicación, pueden ser estos fotos o videos, los mismos que se guardan en la nube de <i>Firebase</i> .

Librerías

En la TABLA IX, se puede evidenciar las bibliotecas o librerías que se han utilizado para la programación del *backend* y su integración con las herramientas mencionadas en la TABLA VIII.

TABLA IX: Librerías para el desarrollo del proyecto

LIBRERÍA	DESCRIPCIÓN
Angular Fire	Biblioteca específica para la conexión entre Angular y Firebase. Esta librería proporciona una buena experiencia en el desarrollo al cumplir con las convenciones del framework Angular
React Native Firebase	Librería usada para la conexión de aplicaciones desarrolladas en React Native con el servicio de Firebase. React Native

	Firestore posee una colección de módulos que conectan con los servicios de Firestore.
--	---

3 RESULTADOS

Esta sección detalla los procesos que se han realizado entorno al producto final y se presentan los resultados de la implementación de las transacciones de escritura, lectura y consultas del *backend* hacia en *frontend*. Así mismo se especifica las pruebas y el despliegue a producción. Cada paso se lo representa a manera de *Sprints*.

3.1 Sprint 0. Configuración del ambiente de desarrollo

El *Sprint 0*, es la base de creación del proyecto, aquí se pueden observar las configuraciones iniciales antes de realizar las tareas de codificación. Se presentan los arreglos para comenzar la aplicación.

En este apartado se encuentran las siguientes tareas de acuerdo con el *Sprint Backlog*.

- Creación del proyecto
- Definición de usuarios
- Creación de base de datos
- Diseño de wireframes

Creación del proyecto.

Se planteo y reformó la idea del proyecto, se discutió con los miembros del equipo como sería el sistema que se crearía.

Definición de usuarios.

Una vez ideado el proyecto, se prosiguió a definir que usuarios estarían interesados en el uso del sistema, a que publico estaría orientado la página web. Ver Fig.3.

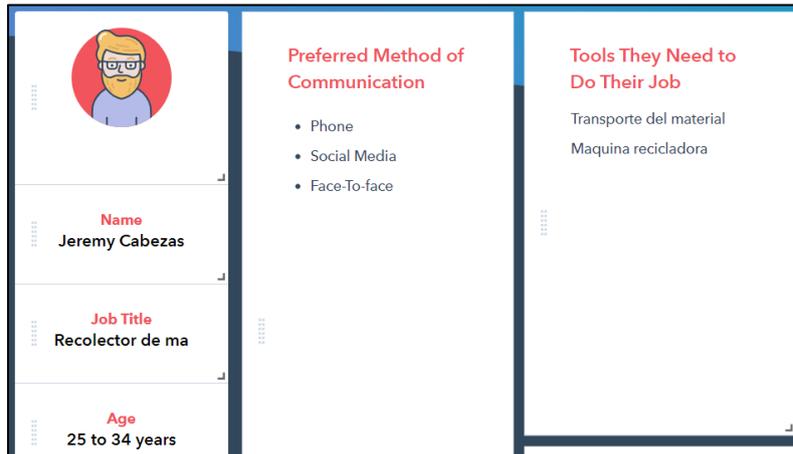


Fig.3: Perfil de usuario

Creación de la base de datos.

La base de datos que se utilizó para el desarrollo del proyecto es no relacional. Se usa la herramienta *Firestore Data Base* para almacenar los datos necesarios y levantar el sistema, así también de datos del usuario. Ver Fig.4. El detalle de colecciones y documentos de la base de datos se encuentra en el ANEXO II, sección Diseño de prototipos.

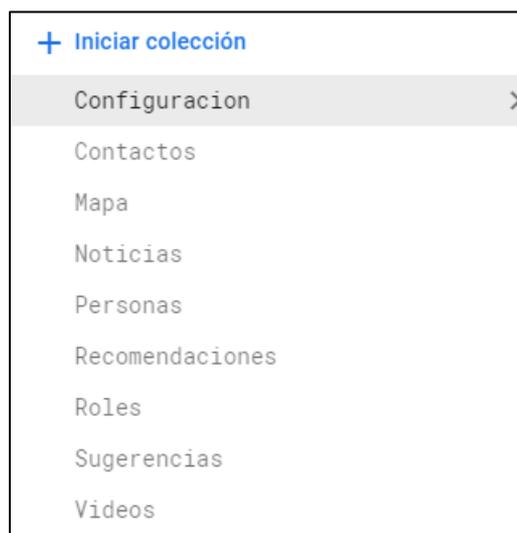


Fig.4: Estructura de la base de datos

Diseño de wireframes.

Se realizó el primer diseño de las páginas que conformarían el sistema, fueron diseños en papel como primer inicio para los futuros diseños. Ver Fig.5.

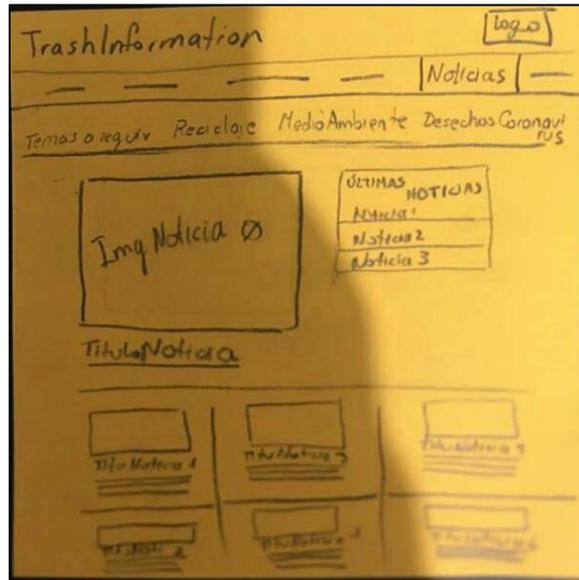


Fig.5: Ejemplo de un wireframe

3.2 *Sprint* 1. Establecimiento de transacciones para el módulo de autenticación.

El siguiente *Sprint* aborda el módulo de autenticación, se generan los servicios para la escritura, lectura y consulta de la sesión de inicio, registro y cierre de sesión. Entre las tareas que se realizaron en esta iteración se encuentran:

- Diseño e implementación de las transacciones de lectura, escritura y consultas para el inicio de sesión y cierre de sesión de administradores y ciudadanos.
- Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de usuarios ciudadanos.
- Validación de los datos del registro de usuario nuevo.
- Chequear que el registro sea exclusivo de un usuario.
- Implementación de las transacciones de lectura, escritura y consulta de recuperación de contraseña.

Diseño e implementación de las transacciones de lectura, escritura y consultas para el inicio de sesión y cierre de sesión de administradores y ciudadanos.

La aplicación tiene como pantalla principal el inicio de sesión para que los usuarios registrados puedan autenticarse tanto en la página web como en la aplicación móvil. Los datos para el inicio de sesión se especifican en *Authentication* de *Firebase*. Para ello, se activó el correo de externos y contraseña de modo que sea el único medio de inicio de

sesión. El servicio para el inicio de sesión se lo puede ver en la Fig.6, aquí se importan las funciones necesarias de *Angular Fire* para crear la conexión a *Firestore*.

```
login({ email, password }: any) {  
  return signInWithEmailAndPassword(this.auth, email, password);  
}
```

Fig.6: Servicio para el inicio de sesión

Este mismo inicio de sesión es para los usuarios administradores que están registrados como administradores en la base de datos de *Firestore*. Estos usuarios no necesitan de un registro de sesión puesto que su rol les permite ingresar a la gestión del sistema (Fig.7).

```
+ Agregar campo  
  
birthdate: "29-02-2000"  
email: "cindyk.20@hotmail.com"  
lastname: "Yazan"  
name: "Cindy"  
profilepic: "https://ui-avatars.com/api/?  
background=0B2460&color=fff&size=600&font-  
size=0.4&name=Cindy+Yazan"  
rol: "admin"
```

Fig.7: Manejo de registros en la base de datos Firestore

Por otro lado, el cierre de sesión se controla con otro servicio a la base de datos de *Firestore* que brinda la herramienta de *Angular Fire*. La librería de *Angular Fire* cuenta con un *signOut* que hace posible el cierre de sesión. La implementación de esta función asíncrona en los servicios se puede ver en la Fig.8.

```
async logout() {  
  try {  
    await signOut(this.auth);  
  } catch (error) {  
    console.log(error);  
  }  
}
```

Fig.8: Transición para el cierre de sesión

Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de usuarios ciudadanos.

El registro de nuevo usuario ciudadano se controla mediante un formulario desde la parte del *frontend* y desde el *backend* se crean nuevas conexiones a la base como servicios que permitan añadir un nuevo usuario y registrarlo en la base. Esta función se especifica en la Fig.9.

```
register(email: any, password: any) {
  return new Promise((resolve, reject) => {
    createUserWithEmailAndPassword(this.auth, email, password)
      .then((userData) => {
        resolve(userData);
      })
      .catch((err) => console.log(reject(err)));
  });
}
```

Fig.9: Servicio para el registro de usuarios nuevos

El proceso siguiente es añadir ese nuevo registro a la colección “Personas” de *Firestore*, se busca y obtienen el correo del usuario como ID primario, se identifica el rol que posee el usuario y se extrae toda la información ingresada por el usuario (Fig.10).

```
//Funciones para manejar usuarios
addUser(user: User, email: any) {
  return setDoc(doc(this.firestore, 'Personas', email), user);
}

getUser(): Observable<User[]> {
  const userRef = collection(this.firestore, 'Personas');
  return collectionData(userRef, { idField: 'email' }) as Observable<User[]>;
}

async getProfile(email: any) {
  const userRef = doc(this.firestore, 'Personas', email)
  const respuesta = await getDoc(userRef)
  console.log('Obteniendo perfil desde servicio: ', respuesta.data())
  return respuesta.data()
}

async searchUserRol(email:any) {
  const docRef = doc(this.firestore, 'Personas', email);
  const docSnap = await getDoc(docRef);
  const responseUser:any = docSnap.data()
  const rol = responseUser['rol']
  return rol
}

async searchUserData(email:any) {
  const docRef = doc(this.firestore, 'Personas', email);
  const docSnap = await getDoc(docRef);
  const responseUser:any = docSnap.data()
  return responseUser
}
```

Fig.10: Métodos para administrar a los usuarios


```

recoverPassword(email: string) {
  return sendPasswordResetEmail(this.auth, email)
    .then(() => {
      // Password reset email sent!
      // ..
    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;

      console.log(errorCode);
      console.log(errorMessage);
    });
}
}

```

Fig.12: Transacción de consulta, lectura y escritura para recuperar contraseña

3.3 *Sprint* 2. Establecimiento de transacciones para el módulo informativo.

Este *Sprint* corresponde a la sección de información donde se puede encontrar todos los servicios de escritura, lectura y consulta de noticias, videos, recomendaciones y sugerencias. Además del mapa donde se puede visualizar las rutas, horarios y días que se recoleta la basura en Quito. Las tareas que se llevaron a cabo en este Sprint son las siguientes:

- Implementación de las coordenadas del mapa para transacciones de consulta y lectura y generar la visualización de rutas, horarios y días de recolección de basura.
- Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de recomendación en el sistema.
- Validación de los datos de la recomendación.
- Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de sugerencias o comentarios de los usuarios en el sistema.
- Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de noticias en el sistema.
- Validación de los datos de las noticias.
- Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de videos en el sistema.

- Validación de los datos de los videos.

Implementación de las coordenadas del mapa para transacciones de consulta y lectura y generar la visualización de rutas, horarios y días de recolección de basura.

La implementación del mapa es directamente de la fuente de EMASEO en donde se visualizan todas las rutas, horarios y días que los camiones de basura recolectan los desechos. Las coordenadas del mapa se proporcionarán como un servicio a la aplicación para que muestre en un dispositivo móvil. El mapa ingresado en la aplicación se puede ver en la siguiente Fig.13.

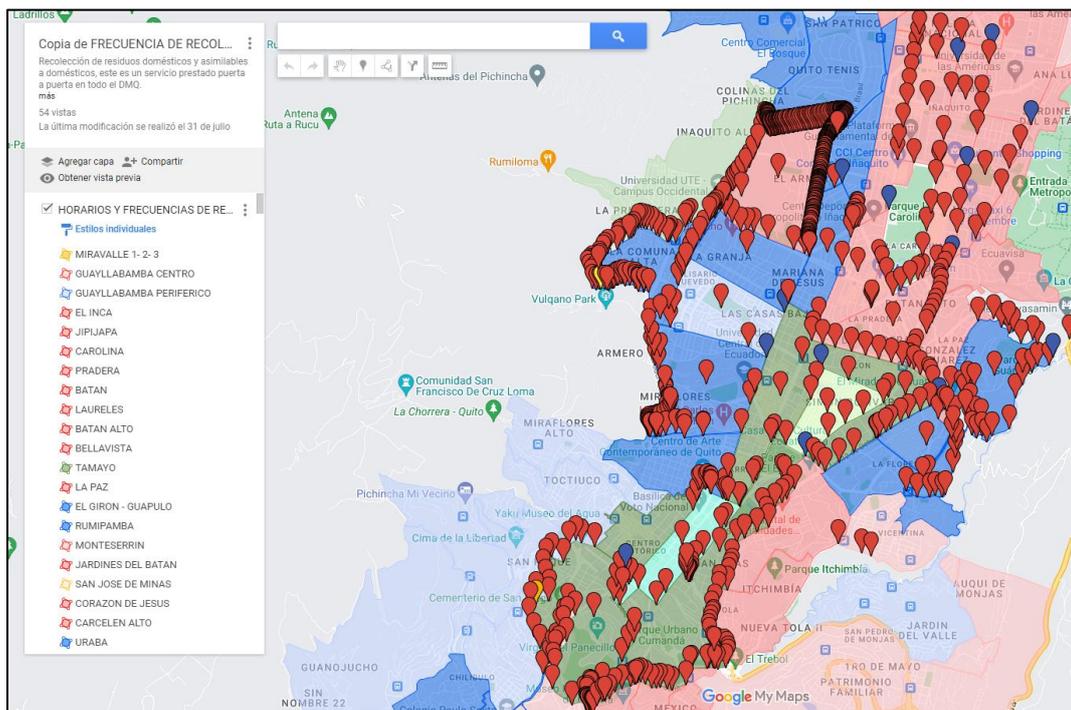


Fig.13: Coordenadas del mapa para transacciones de lectura y consulta

Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de recomendación en el sistema.

El siguiente paso es la implementación de las diferentes secciones informativas, en este caso son las recomendaciones que visualizará el usuario y que serán gestionadas por los administradores. Este es uno de los CRUD que maneja toda la aplicación. Aquí se añaden y se modifican por el administrador. Así también, las recomendaciones serán visualizadas por los usuarios ciudadanos. Se han creado servicios para que se conecten con *Firestore* y realicen las transacciones necesarias para leer, escribir y consultar las

recomendaciones almacenadas. La Fig.14, muestra las funciones que proveerán los servicios desde la aplicación a la base y viceversa.

```
addRecommendation(recomen: Recomendacion, id: string){
  return setDoc(doc(this.firestore, 'Recomendaciones', id), recomen);
}

getRecommendations(): Observable<Recomendacion[]>{
  const recomenRef = collection(this.firestore, 'Recomendaciones')
  return collectionData(recomenRef, {idField: 'id'}) as Observable<Recomendacion[]>
}

async modifiedRecommendation(id: any) {
  const recomenRef = doc(this.firestore, 'Recomendaciones', id)
  const respuesta = await getDoc(recomenRef)
  return respuesta.data()
}
```

Fig.14: Servicios CRUD para la sección de recomendaciones

Validación de los datos de la recomendación.

Las recomendaciones se presentan en forma de formularios que son validados al momento de ingresar o modificar la información. Este módulo de recomendaciones tiene como regla el ingreso de máximo 150 caracteres en el contenido de la recomendación como se evidencia en la Fig.15.

```
export class RecommendationsAdminComponent implements OnInit {
  formRecomen: FormGroup;
  enumRecomen: number = 0;
  recomendaciones: Recomendacion[] = [];
  contentLimitPattern: any = /^[\\s\\S]{0,150}$/;
```

Fig.15: Reglas de validación de campos de recomendaciones

Es obligatorio el ingreso de los campos título, categoría y contenido de la recomendación a modificar o añadir (Fig.16).

```
this.formRecomen = new FormGroup({
  title: new FormControl('', [Validators.required]),
  category: new FormControl('', [Validators.required]),
  content: new FormControl('', [
    Validators.required,
    Validators.pattern(this.contentLimitPattern),
  ]),
});
```

Fig.16: Validación de campos de recomendaciones

Se inicializan los getters para cada campo que requiera una validación, con su respectivo mensaje de error como se indica en las Fig.17 y Fig.18.

```
get title() {
  return this.formRecomen.get('title');
}

get category() {
  return this.formRecomen.get('category');
}

get content() {
  return this.formRecomen.get('content');
}
```

Fig.17: Getters para la validación de recomendaciones

```
getErrorMessageTitle() {
  return this.title.hasError('required')
    ? 'Debe escribir un título para la recomendación'
    : '';
}

getErrorMessageCategory() {
  return this.category.hasError('required')
    ? 'Debe elegir una categoría para la recomendación'
    : '';
}

getErrorMessageContent() {
  if (this.content.hasError('required')) {
    return 'Debe escribir el contenido de su recomendación';
  }
  return this.content.hasError('pattern')
    ? 'Límite máximo de caracteres es 150'
    : '';
}
```

Fig.18: Mensajes informativos para las validaciones de recomendaciones

Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de sugerencias o comentarios de los usuarios en el sistema.

Para el apartado de sugerencias o comentarios ingresados por los usuarios se realiza la adición de sugerencias y la visualización de estas. Estos dos servicios permitirán al usuario del sistema pueda administrar como pueda ver las entradas recientes de comentarios en el sitio realizadas por parte de los clientes. Las gestiones que se lleva a cabo en las sugerencias son las de la Fig.19.

```

addSuggest(sugerencia: Sugerencia, id: string){
  return setDoc(doc(this.firestore, 'Sugerencias', id), sugerencia);
}

getSuggestions(): Observable<Sugerencia[]>{
  const sugestRef = collection(this.firestore, 'Sugerencias')
  return collectionData(sugestRef, {idField: 'id'}) as Observable<Sugerencia[]>
}

```

Fig.19: Transacciones de lectura, escritura y consultas para el registro de sugerencias

Al igual que las recomendaciones, las sugerencias el formulario visto por el cliente tiene campos obligatorios de llenar como es la opinión acerca del tema y el comentario correspondiente (Fig.20).

```

this.formSuggest = new FormGroup({
  id: new FormControl(),
  name: new FormControl(),
  lastName: new FormControl(),
  email: new FormControl(),
  section: new FormControl('', [
    Validators.required ←
  ]),
  comment: new FormControl('', [
    Validators.required ←
  ])
});

```

Fig.20: Validación del formulario de sugerencias

Se colocan los getters y los mensajes de ayuda para que el usuario llene estas secciones y pueda ver el administrador (Fig.21). Las sugerencias pueden ser solo eliminadas por el administrador.

```

get section(){
  return this.formSuggest.get('section');
}

get comment(){
  return this.formSuggest.get('comment');
}

getErrorMessageSection(){
  return this.section.hasError('required') ? 'Debe escribir acerca de que es su opinión' : '';
}

getErrorMessageComment(){
  return this.comment.hasError('required') ? 'Debe escribir el contenido de su opinión' : '';
}

```

Fig.21: Mensajes informativos para la validación de sugerencias

Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de noticias en el sistema.

El apartado de noticias maneja un CRUD, en este se puede ver en la Fig.22, que se pueden añadir, modificar y consultar la colección en donde se almacena toda esta información. Al llamar a la función de añadir una nueva noticia se le pide a la base que ingrese un documento más a la colección "Noticias". Al modificar el contenido de las noticias, es necesario hacer la llamada referenciando al ID principal para que se visualicen solo los datos de la noticia que se va a modificar. Las conexiones a la base de datos de *Firestore* permite que el usuario administrador pueda cambiar, adicionar y eliminar cualquier noticia que se encuentre en la base de datos, así estos cambios serán vistos por el cliente que use el sistema.

```
addNoticia(noticia: Noticia, id: string){
  return setDoc(doc(this.firestore, 'Noticias', id), noticia);
}

getNoticias(): Observable<Noticia[]>{
  const noticiaRef = collection(this.firestore, 'Noticias')
  return collectionData(noticiaRef, {idField: 'id'}) as Observable<Noticia[]>
}

async modifiedNoticia(id: any) {
  const noticiaRef = doc(this.firestore, 'Noticias', id)
  const respuesta = await getDoc(noticiaRef)
  return respuesta.data()
}
```

Fig.22: Servicios CRUD para la sección de noticias

Validación de los datos de las noticias.

Los campos que validar de la sección noticias corresponden al título, la categoría, la descripción, el autor y el URL que es la fuente de donde se ha consultado dicha noticia. Estos datos son requeridos para crear o modificar una noticia como se indica en la Fig.23, el administrador debe obligatoriamente ingresar todos estos datos para que cualquier acción proceda.

```

this.formNoticia = new FormGroup({
  title: new FormControl('', [Validators.required]),
  category: new FormControl('', [Validators.required]),
  description: new FormControl('', [
    Validators.required,
    Validators.pattern(this.contentLimitPattern)
  ]),
  author: new FormControl('', [Validators.required]),
  url: new FormControl('', [Validators.required]),
  id: new FormControl(),
  noticiaPic: new FormControl(''),

```

Fig.23: Validación de campos de noticias

Se ha especificado que la descripción de la noticia se encuentre entre los 380 caracteres. Si se ingresan más caracteres la entrada no será válida y la adición o modificación no se llevará a cabo (Fig.24).

```

export class NewsAdminComponent implements OnInit {
  formNoticia: FormGroup;
  enumNoticias: number = 0;
  noticias: Noticia[] = [];
  verifyNoticias: Noticia[] = [];

  urlNoticia: any = null;
  selectedFile: any = null;

  contentLimitPattern: any = /^[s\S]{0,380}$/;

```

Fig.24: Reglas de validación de campos de noticias

Se procede como en las anteriores secciones a inicializar los getters de la Fig.25, que ayudan a realizar las validaciones de los distintos formularios del sistema. Y finalmente se instancian los mensajes informativos para que el usuario administrador pueda ingresar correctamente los datos (Fig.26).

```

get title() {
  return this.formNoticia.get('title');
}

get category() {
  return this.formNoticia.get('category');
}

get description() {
  return this.formNoticia.get('description');
}

get author() {
  return this.formNoticia.get('author');
}

get url() {
  return this.formNoticia.get('url');
}

```

Fig.25: Getters para la validación de noticias

```

getErrorMessageTitle() {
  return this.title.hasError('required')
    ? 'Debe escribir un título para la noticia'
    : '';
}

getErrorMessageCategory() {
  return this.category.hasError('required')
    ? 'Debe elegir una categoría para la noticia'
    : '';
}

getErrorMessageDescription() {
  if (this.description.hasError('required')) {
    return 'Debe escribir la recomendación de la noticia';
  }
  return this.description.hasError('pattern')
    ? 'Límite máximo de caracteres es 380'
    : '';
}

getErrorMessageAuthor() {
  return this.author.hasError('required')
    ? 'Debe escribir un autor para la noticia'
    : '';
}

getErrorMessageUrl() {
  return this.url.hasError('required')
    ? 'Debe ingresar la fuente bibliográfica donde se encuentra la noticia'
    : '';
}

```

Fig.26: Mensajes informativos para las validaciones de recomendaciones

Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de videos en el sistema.

La división de videos es el último módulo de este *Sprint*. En este se encuentran servicios similares a los de las noticias. La conexión a *Firestore* es para realizar las acciones de añadir, modificar y consultas los datos de la colección "Videos". La Fig.27, muestra las funciones para ingresar un nuevo documento a la colección, además de conseguir un documento específico diferenciándolo por su ID principal para proceder a su modificación.

```
addVideo(video: Video, id: string){
  return setDoc(doc(this.firestore, 'Videos', id), video);
}

getVideos(): Observable<Video[]>{
  const videoRef = collection(this.firestore, 'Videos')
  return collectionData(videoRef, {idField: 'id'}) as Observable<Video[]>
}

async modifiedVideo(id: any) {
  const videoRef = doc(this.firestore, 'Videos', id)
  const respuesta = await getDoc(videoRef)
  return respuesta.data()
}
```

Fig.27: Transacciones de lectura, escritura y consultas para los videos

Validación de los datos de los videos.

La validación de los videos hace referencia a los campos: título, categoría, autor y el URL del video en *YouTube*. Se ve que se requieren que estos valores sean ingresados por el usuario en la Fig.28.

```
this.formVideo = new FormGroup({
  title: new FormControl('', [
    Validators.required,
    Validators.pattern(this.contentLimitPattern),
  ]),
  category: new FormControl('', [Validators.required]),
  prevImg: new FormControl(),
  author: new FormControl('', [Validators.required]),
  url: new FormControl('', [Validators.required]),
  id: new FormControl(),
  urlID: new FormControl(),
});
```

Fig.28: Validación de campos de videos

En el caso del título tiene una validación adicional, aparte de ser obligatoria, este debe estar en el rango de 140 caracteres (Fig.29). Así se reduce las entradas y se puede manejar la vista para el cliente.

```
export class VideosAdminComponent implements OnInit {
  formVideo: FormGroup;
  enumVideos: number = 0;
  videos: Video[] = [];

  contentLimitPattern: any = /^[s\S]{0,140}$/;
```

Fig.29: Reglas de validación de campos de videos

Finaliza la validación con la implementación de los getters de la Fig.30, para proporcionar administrador los mensajes de error y pueda introducir los datos precisos (Fig.31). Los mensajes se los llama desde el *frontend* para que el usuario gestor de la aplicación sea capaz de visualizarlos en la interfaz.

```
get title() {
  return this.formVideo.get('title');
}

get category() {
  return this.formVideo.get('category');
}

get author() {
  return this.formVideo.get('author');
}

get url() {
  return this.formVideo.get('url');
}
```

Fig.30: Getters para la validación de videos

```

getMessageTitle() {
  if (this.title.hasError('required')) {
    return 'Debe escribir un título para el video';
  }
  return this.title.hasError('pattern')
    ? 'Límite máximo de caracteres es 140'
    : '';
}

getMessageCategory() {
  return this.category.hasError('required')
    ? 'Debe elegir una categoría para el video'
    : '';
}

getMessageAuthor() {
  return this.author.hasError('required')
    ? 'Debe escribir un autor para el video'
    : '';
}

getMessageUrl() {
  return this.url.hasError('required')
    ? 'Debe ingresar el enlace de YouTube del video'
    : '';
}

```

Fig.31: Mensajes informativos para las validaciones de videos

3.4 *Sprint* 3. Establecimiento de transacciones para el módulo complementario.

Lo que corresponde al *Sprint* 3 es el apartado de contactos y la configuración del perfil del cliente, se ha propuesto este módulo como un complemento a las funciones del sistema. En este *Sprint* se explica los servicios de escritura, lectura y consulta de contactos, así también de la última sección que corresponde a la edición de perfil de usuario ciudadano. Las tareas que se llevaron a cabo en este *Sprint* son las siguientes:

- Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de contactos de recicladores en el sistema.
- Validación de los datos de los contactos de recicladores.

Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de contactos de recicladores en el sistema.

Este es el módulo complementario al sistema que da un toque único al sistema puesto que aquí se presenta información de personas o entidades que realizan actividades de reciclaje. Esta tarea en la parte de backend llama a un servicio para realizar consultas y

escritura en el *Firestore* y esto se refleje a tiempo real en la interfaz. Los servicios que se pueden ver en la Fig.32, son las transacciones de modificar, leer y añadir los contactos en las colecciones, seguido de los documentos de la base de datos.

```
addContact(contacto: Contacto, id: string){
  return setDoc(doc(this.firestore, 'Contactos', id), contacto);
}

getContacts(): Observable<Contacto[]>{
  const contactRef = collection(this.firestore, 'Contactos')
  return collectionData(contactRef, {idField: 'id'}) as Observable<Contacto[]>
}

async modifiedContact(id: any) {
  const contactRef = doc(this.firestore, 'Contactos', id)
  const respuesta = await getDoc(contactRef)
  return respuesta.data()
}
```

Fig.32: Transacciones de lectura, escritura y consultas para los contactos

Validación de los datos de los contactos de recicladores.

Primero para validar los datos se debe tomar en cuenta que limitaciones va a tener el usuario para introducir los datos. En la Fig.33, se puede observar que el nombre de la persona que proporciona el servicio de reciclaje debe ser únicamente alfabético y sin espacios, esto se maneja con las expresiones regulares que se ve en la Fig.34.

```
this.formContact = new FormGroup({
  id: new FormControl(),
  name: new FormControl('', [
    Validators.required,
    Validators.minLength(3),
    Validators.pattern(this.alfabetWithOutSpacePattern),
  ]),
  lastName: new FormControl('', [
    Validators.required,
    Validators.minLength(3),
    Validators.pattern(this.alfabetWithOutSpacePattern),
  ]),
  address: new FormControl('', [Validators.required]),
  phoneNumber: new FormControl('', [
    Validators.required,
    Validators.pattern(this.phoneNumberPattern),
  ]),
  activity: new FormControl('', [Validators.required]),
});
```

Fig.33: Validación de campos de contactos

```
alfabetWithOutSpacePattern: any = /^(?!.*[0-9])[a-zA-ZÀ-ÿ\u00f1\u00d1]
|+(\s*[a-zA-ZÀ-ÿ\u00f1\u00d1])*[a-zA-ZÀ-ÿ\u00f1\u00d1]+$/;
phoneNumberPattern: any = /^d{10}$/;
```

Fig.34: Reglas de validación de campos para contactos

El siguiente paso para validar los datos que introduce el usuario en el formulario de la sección de contactos es inicializar los getters de las variables (Fig.35), para que se puedan generar los mensajes informativos al usuario en el *frontend*, esto es visible en la Fig.36.

```
get name() {
  return this.formContact.get('name');
}

get lastName() {
  return this.formContact.get('lastName');
}

get address() {
  return this.formContact.get('address');
}

get phoneNumber() {
  return this.formContact.get('phoneNumber');
}

get activity() {
  return this.formContact.get('activity');
}
```

Fig.35: Getters para la validación de contactos

```
getErrorMessageName() {
  if (this.name.hasError('required')) {
    return 'Debe completar el campo';
  }
  return this.name.hasError('pattern')
    ? 'Mínimo 6 caracteres y sin numeros'
    : '';
}

getErrorMessageAddress() {
  return this.address.hasError('required')
    ? 'Debe escribir su dirección'
    : '';
}

getErrorMessagePhoneNumber() {
  if (this.phoneNumber.hasError('required')) {
    return 'Debe escribir su número de contacto';
  }
  return this.phoneNumber.hasError('pattern')
    ? 'Coloque un número telefónico válido de 10 dígitos'
    : '';
}

getErrorMessageActivity() {
  return this.activity.hasError('required')
    ? 'Debe escribir el servicio de reciclaje que oferta'
    : '';
}
```

Fig.36: Mensajes informativos para las validaciones de contactos

3.5 *Sprint* 4. Pruebas y despliegue del sistema.

Finalizada la codificación de cada una de las transacciones de consulta, lectura y escritura de todos los módulos que compete el sistema en el *backend* y siguiendo la cronografía del *Sprint Backlog* se tienen las siguientes tareas:

- Pruebas de seguridad
- Pruebas de compatibilidad
- Pruebas de carga
- Despliegue del sistema en Netlify.

Se ha planteado realizar pruebas no funcionales mencionadas, para visualizar a detalle el óptimo funcionamiento del sistema generado en los anteriores *Sprints*. Cabe mencionar que al término de cada *Sprint* se realizaron pruebas para verificación de requisitos funcionales definidos en los artefactos de la metodología *SCRUM*.

Pruebas de seguridad

Esta es la sección en donde se presentarán las pruebas de seguridad del sistema para garantizar que el software esté libre de vulnerabilidades o que estas sean mínimas, evitar riesgos, amenazas que puedan llegar a la pérdida de datos. Con estas pruebas podemos confiar que los recursos de la app estén protegidos de posibles agentes externos.

La Fig.37, muestra los resultados generales de la prueba realizada. Se observa que los riesgos son mínimos, tres de ellos son bajos. Estos se pueden obviar de las vulnerabilidades del sistema y los riesgos sobrantes son de tipo medio. El primero es un puerto abierto, el puerto 443 usado para peticiones al servidor de páginas web con protocolo *https*, necesario para que la página sea pública. El segundo riesgo es otro puerto abierto, el puerto 80 orientado a páginas web con protocolo *http*. Y la última vulnerabilidad hace referencia al *clickjacking*, técnica maliciosa para revelar información de los usuarios por medio de páginas engañosas. Este se ve como riesgo medio por no contar con políticas de privacidad. Los detalles se pueden verificar en la sección ANEXO II, sección Pruebas de seguridad.

Risks							
TITLE	SCAN TYPE	TARGET	THREAT LEVEL	OPENVAS QOD	STATUS	LAST DETECTED	Filters
> TCP timestamps	OPENVAS	https://trash-information-webapp.netlify.app/login	LOW	80%	OPEN	2 days ago	Accept Risk
> Open TCP Port: 443	NMAP	https://trash-information-webapp.netlify.app/login	MEDIUM		OPEN	3 days ago	Accept Risk
> Open TCP Port: 80	NMAP	https://trash-information-webapp.netlify.app/login	MEDIUM		OPEN	3 days ago	Accept Risk
> Missing Anti-clickjacking Header	OWASP_ZAP	https://trash-information-webapp.netlify.app/login	MEDIUM		OPEN	3 days ago	Accept Risk
> Cross-Domain JavaScript Source File Inclusion	OWASP_ZAP	https://trash-information-webapp.netlify.app/login	LOW		OPEN	3 days ago	Accept Risk
> X-Content-Type-Options Header Missing	OWASP_ZAP	https://trash-information-webapp.netlify.app/login	LOW		OPEN	3 days ago	Accept Risk

Fig.37: Prueba de seguridad en el software *Hosted Scan Security*

Pruebas de compatibilidad

Esta prueba se realiza para establecer la compatibilidad de la aplicación en diferentes dispositivos [13]. Para las pruebas de compatibilidad en la parte del backend se la realiza mediante clientes HTTP como son: *Postman* y *Thunder Client*, estos verifican que los servicios sean presentados correctamente. La TABLA X, presenta los clientes en donde se han ejecutado estas pruebas y se puede verificar los resultados en ANEXO II, sección Pruebas de compatibilidad, de este documento.

TABLA X: Clientes para la prueba de compatibilidad

NOMBRE	VERSIÓN
<i>Postman</i>	V 9.26.0
<i>Thunder Client</i>	V 2.1.0

Las pruebas ejecutadas muestran que los servicios proporcionados por el *backend* no indican fallos en el tiempo de respuesta o al mostrar la información solicitada por el cliente.

Pruebas de carga

Las últimas pruebas ejecutadas son las de carga, para ello, se realiza una petición HTTP con el método GET a diferentes servidores y estos proporcionen una respuesta a la conexión con el sistema, aquí se comprueba la URL de destino, si tiene algún error o algún enlace roto.

En la Fig.38, se puede encontrar las pruebas de carga en diferentes servidores. Los resultados de un servidor a detalle se encuentran en el ANEXO II, sección Pruebas de carga. Cabe destacar que el software que ayudo con este tipo de pruebas es *LoadView* que permite pruebas de esfuerzo con varias conexiones simultaneas.

Realtime Feed						
Last 24 Hours						
Device	Location	Video	Status	Time	Duration	
Sistema TrashInformation Stress Test	New York		OK	11:01:28 AM	341 ms	
Sistema TrashInformation Stress Test	Minneapolis		OK	10:56:24 AM	0.66 s	
Sistema TrashInformation Stress Test	Seattle		OK	10:51:00 AM	245 ms	
Sistema TrashInformation Stress Test	Madrid		OK	10:50:59 AM	398 ms	
Sistema TrashInformation Stress Test	Chicago		OK	10:50:59 AM	279 ms	
Sistema TrashInformation Stress Test	Copenhagen		OK	10:50:59 AM	159 ms	
Sistema TrashInformation Stress Test	Warsaw		OK	10:50:58 AM	313 ms	
Sistema TrashInformation Stress Test	Brisbane		OK	10:50:58 AM	288 ms	
Sistema TrashInformation Stress Test	Sydney		OK	10:50:58 AM	429 ms	
Sistema TrashInformation Stress Test	Singapore		OK	10:50:58 AM	1.34 s	
Sistema TrashInformation Stress Test	Bogota		OK	10:50:58 AM	0.70 s	
Sistema TrashInformation Stress Test	Hong Kong		OK	10:50:58 AM	0.86 s	
Sistema TrashInformation Stress Test	Miami		OK	10:50:58 AM	177 ms	
Sistema TrashInformation Stress Test	Montreal		OK	10:50:58 AM	271 ms	
Sistema TrashInformation Stress Test	Minneapolis		OK	10:50:58 AM	373 ms	
Sistema TrashInformation Stress Test	Frankfurt		OK	10:50:58 AM	73 ms	

Fig.38: Pruebas de carga en *Load View*

Luego de la ejecución de las pruebas de carga, se determina que las peticiones y servicios para el backend cumplen con los requerimientos del cliente.

Despliegue del sistema en Netlify.

El *deploy* del sistema a producción en una terminal requiere el ingreso del comando mostrado en la Fig.39, este encapsula las carpetas necesarias para el despliegue con todas las dependencias que permite generar la aplicación con todas las funcionalidades requeridas.

```
PS C:\Users\cyazan\Documents\BackendAppWebTI> ng build --configuration production --aot
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Names | Raw Size | Estimated Transfer Size
main.6c8d77fd08352203.js | main | 1.59 MB | 333.94 kB
styles.91aa7872370527a3.css | styles | 78.50 kB | 9.25 kB
polyfills.4642cee85afd5469.js | polyfills | 33.01 kB | 10.66 kB
runtime.4f73f64fac128ee0.js | runtime | 1.76 kB | 887 bytes

| Initial Total | 1.70 MB | 354.71 kB

Build at: 2022-08-22T17:00:22.382Z - Hash: 511d98c75d53b2b8 - Time: 7632ms
```

Fig.39: Deploy del proyecto

La carpeta generada del deploy se lo sube a la plataforma de *Netlify*, con un drag and drop se puede obtener el enlace que aloja a la aplicación (Fig.40).



Fig.40: Proyecto desplegado en Netlify

4 CONCLUSIONES

Los resultados de acuerdo con los objetivos y después de realizar el proyecto se presentan en las siguientes conclusiones:

- Se han llevado a cabo los objetivos planteados, con el desarrollo del backend para la aplicación planteada, el diseño y gestión del backend.
- La metodología Scrum proporciono tiempos para que el desarrollo del sistema tenga avances en cada etapa, lo que logro que el producto saliera a producción y con el rendimiento esperado.
- El modelo Vista Controlado permitió el ordenar las secciones del proyecto. Su estructura estaba ligada al framework lo que permitió que el desarrollo fuera fácil y entendible para cualquier persona con fundamentos de programación.
- El diseño y elaboración de la base de datos en Firebase hizo más sencillo el recurrir a los servicios y conectarlos con el frontend. El framework de Angular, en conjunto con la base Firestore resulto rápido al trabajar con el frontend para transferir datos con las aplicaciones.
- El backend desarrollado está listo para trabajar con una aplicación que requiera de servicios CRUD para un usuario administrador y presentar esa misma información en formato informativo a un usuario cliente.
- Las pruebas ejecutadas muestran los aspectos no funcionales de la aplicación y con esto se probó que el sistema está en las condiciones de operar en la web y en una aplicación móvil.

5 RECOMENDACIONES

A continuación, se muestran las recomendaciones que se han podido recolectar durante el progreso del proyecto.

- Es recomendable realizar *backups* de la Base de Datos en determinados periodos de tiempo con el objetivo de salvaguardar los datos en caso de inconvenientes.
- El conocimiento de las colecciones y documentos de la base de datos debe ser clara para la elaboración de cualquier sistema. Se debe estar consciente que el almacenamiento de información es correcto para que cumpla con los requerimientos del cliente.
- El manejo de la documentación del framework, base de datos y demás herramientas utilizadas es de suma importancia. Esto facilitará el desarrollo y búsqueda de soluciones cuando se presenten errores o dificultades.
- La aplicación puede ser escalable al incorporarle las rutas de los camiones recolectores de basura en tiempo real.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Empresa Pública Metropolitana EMASEO EP, «EMASEO EP,» 2022. [En línea]. Available: <http://www.emaseo.gob.ec/un-mejor-quito/>. [Último acceso: 30 Agosto 2022].
- [2] F. Jaramillo, « La basura puede sumir a Quito en otra emergencia,» 9 Junio 2020. [En línea]. Available: <https://www.revistagestion.ec/sociedad-analisis/la-basura-puede-sumir-quito-en-otra-emergencia>. [Último acceso: 22 Agosto 2022].
- [3] Earth911, «iRecycle,» 2022. [En línea]. Available: <https://play.google.com/store/apps/details?id=com.earth911.irecycle&hl=es&gl=US>. [Último acceso: 25 Agosto 2022].
- [4] C. A. Bernal, «METODOLOGÍA DE LA INVESTIGACIÓN,» 2016, pp. 54-56.
- [5] R. Figueroa y C. Solís, «METODOLOGÍAS TRADICIONALES VS METODOLOGÍAS ÁGILES,» 2018.
- [6] J. Garcia, «Metodologías Ágiles en el Desarrollo de Software,» 2018.
- [7] E. G. Maida y J. Pacienza, «Metodologías de desarrollo de software,» de *Metodologías de desarrollo de software. Tesis de Licenciatura en Sistemas y Computación*, Buenos Aires, 2015, pp. 17-20.
- [8] «Metodología Scrum,» de *Gestión de proyectos informáticos. Metodología Scrum*, 2018, pp. 32-40.
- [9] A. Pasini, S. Esponda, M. Boracchia y P. Pesado, «Q-Scrum: una fusión de Scrum y el estándar ISO/IEC,» Octubre 2013. [En línea]. Available: <http://sedici.unlp.edu.ar/handle/10915/32421>. [Último acceso: 8 Junio 2022].
- [10] E. ABELLÁN, «Scrum: qué es y cómo funciona esta metodología,» [En línea]. Available: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>. [Último acceso: 12 2021].
- [11] C. Reynoso, «Introducción a la Arquitectura de Software,» Marzo 2018. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/64699769/Arquitectura_software-with-cover-page-v2.pdf?Expires=1655473410&Signature=W-ZC-gucnc9b5yuDA8N3INO0vFa8dqz8LNqCbx9YwU8zCnmAJrotU2hbtgRfIGzcp2URMB9ab9dWph-s7GFW5f9--x7N68EZpvmzVi5UCDhCTswBDSIN7XORbtI0KvYsYLAqox. [Último acceso: 6 Junio 2022].
- [12] Herramientas de desarrollo con soporte colaborativo en Ingeniería de Software, 2015.
- [13] Ebooks online, «Ebooks online,» 13 04 2021. [En línea]. Available: <https://ebooksonline.es/que-es-una-prueba-de-compatibilidad-prueba-hacia-adelante-y-hacia-atras-ejemplo/#:~:text=La%20prueba%20de%20compatibilidad%20es,de%20red%20o%20>

M%C3%B3vil%20gairis.. [Último acceso: 12 2021].

- [14] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [15] I. Zabala, «Como recopilar requisitos para un proyecto,» 2019. [En línea]. Available: <https://enredandoproyectos.com/recopilar-los-requisitos-de-un-proyecto/#:~:text=al%20estado%20futuro.-,La%20recopilaci%C3%B3n%20de%20Requisitos,proyecto%20para%20cumplir%20los%20objetivos..> [Último acceso: 12 2021].
- [16] Universidad de Alicante, [En línea]. Available: [https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20\(MVC\)%20es,control%20en%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz%20de,los%20mecanismos%20interacci%C3](https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20(MVC)%20es,control%20en%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz%20de,los%20mecanismos%20interacci%C3). [Último acceso: 12 2021].
- [17] okhosting, «Herramientas de Desarrollo de Software,» [En línea]. Available: <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>. [Último acceso: 12 2021].
- [18] J. Rivera, «¿Qué es NodeJS? Tu mejor aliado para crear aplicaciones web,» 08 02 2021. [En línea]. Available: https://www.crehana.com/ec/blog/desarrollo-web/que-es-nodejs/?gclid=EAlaIqobChMI7df-z4bT9QIVqwaICR2RZQ9REAAYASAAEgJLzvD_BwE. [Último acceso: 12 2021].
- [19] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/es/what-is-mongodb>. [Último acceso: 12 2021].
- [20] Capterra, «HEROKU,» [En línea]. Available: <https://www.capterra.ec/software/158191/heroku>. [Último acceso: 12 2021].
- [21] MSN web docs, «Introducción a Express/Node,» [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction. [Último acceso: 12 2021].
- [22] Codigo facilito, «Curso de Nodejs,» [En línea]. Available: <https://codigofacilito.com/articulos/que-es-mongoose>. [Último acceso: 12 2021].
- [23] izertis, [En línea]. Available: <https://www.izertis.com/es/-/blog/enciptacion-de-password-en-nodejs-y-mongodb-bcrypt>. [Último acceso: 12 2021].
- [24] L. M. L. Magaña, «Qué es Json Web Token y cómo funciona,» 2020. [En línea]. Available: <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>. [Último acceso: 12 2021].

7. ANEXOS

En sección se encontrar los anexos que se han realizado como parte del desarrollo del presente proyecto.

- ANEXO I. Resultados de originalidad de *Turnitin*.
- ANEXO II. Manual Técnico
- ANEXO III. Manual de Usuario
- ANEXO IV. Manual de Instalación.

ANEXO I

El siguiente anexo se presenta el certificado del Director de Tesis donde evidencia el resultado de originalidad del software en base a la herramienta *Turnitin*.



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 07 de 09 de 2022

De mi consideración:

Yo, MAYRA ISABEL ALVAREZ JIMÉNEZ, en calidad de Director del Trabajo de Integración Curricular titulado "DESARROLLO DEL BACKEND PARA PROVEER INFORMACIÓN DE LA RECOLECCIÓN Y MANEJO DE DESECHOS EN QUITO" elaborado por la estudiante CINDY KATHALINA YAZÁN CHUQUIMARCA de la carrera en DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

**Ing. Mayra Isabel Alvarez Jiménez
Técnico Docente
ESFOT**

ANEXO II

En este anexo se muestra información que complementa las secciones anteriores. Aquí se pueden observar a detalle la Recopilación de los Requerimientos, Historias de Usuario, Product Backlog, Sprint Backlog y la sección de pruebas que complementa el apartado del Sprint 4.

Recopilación de requerimientos

En la TABLA XI, se visualiza la Recopilación de Requerimientos que se ha obtenido al iniciar el proyecto.

TABLA XI: Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - R	ENUNCIADO DEL ÍTEM
BACKEND	R003	Como usuario administrador del sistema necesita transacciones de lectura, escritura y consulta de las recomendaciones sobre el manejo de desechos. Con la finalidad de: <ul style="list-style-type: none">• Controlar la sección de recomendaciones.
	RR004	Como usuario administrador del sistema necesita transacciones de lectura y consultas de las sugerencias del usuario ciudadano. Con la finalidad de: <ul style="list-style-type: none">• Informarse de los comentarios que el usuario ciudadano proporcione sobre la aplicación móvil o web, además de sugerencias sobre el servicio de recolección de basura.
	RR005	Como usuario administrador del sistema necesita transacciones de lectura, escritura y consulta de las noticias informativas relacionadas a la clasificación de desechos orgánicos, COVID, reciclaje y medio ambiente.

	RR006	Como usuario administrador del sistema necesita transacciones de lectura, escritura y consulta de los videos informativos relacionados a la clasificación de desechos orgánicos, COVID, reciclaje y medio ambiente.
	RR007	Como usuario administrador del sistema necesita transacciones de lectura, escritura y consultas de la información de contacto de las personas o entidades que realizan reciclaje. Con la finalidad de: <ul style="list-style-type: none"> • Presentar al usuario ciudadano los contactos de personas que clasifican la basura.
	RR008	Como usuario administrador del sistema necesita implementar consultas de la información sobre las rutas, días y horarios de los transportes que recolectan la basura en Quito. Con la finalidad: <ul style="list-style-type: none"> • Conocer la información del servicio de recolección de basura.

Historias de usuarios

Finalizada la recopilación de requerimientos, el siguiente paso es elaboración de las Historias de Usuario. A continuación, se visualizan las historias de usuario utilizadas para el desarrollo del sistema desde la TABLA XII hasta la **¡Error! No se encuentra el origen de la referencia..**

TABLA XII: Historia de Usuario #2

Historia de Usuario	
Identificador: HU-02	Usuario: Administrador y Usuario Ciudadano
Nombre Historia: Generar transacciones de lectura, escritura y consultas para el registro en la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Alta	Riesgo en Desarrollo (Alta/Medio/Baja): Alta

Iteración Asignada: 1	
Responsables (es): Cindy Yazán	
Descripción: El usuario administrador y usuario ciudadano en el backend podrá acceder a transacciones de lectura, escritura y consultas en lotes para registrar nuevos usuarios en el sistema. Para el registro se toma en cuenta: <ul style="list-style-type: none"> • Nombre • Apellido • Correo electrónico • Contraseña • Dirección 	
Observación: El registro de sesión es necesario para las secciones de sugerencias y recomendaciones, así se sabrá que usuario está haciendo entradas en el sistema.	

TABLA XIII: Historia de Usuario #3

Historia de Usuario	
Identificador: HU-03	Usuario: Administrador y Usuario Ciudadano
Nombre Historia: Generar transacciones de lectura, escritura y consultas para la recuperación de contraseña.	
Prioridad en Negocio (Alta/Medio/Baja): Alta	Riesgo en Desarrollo (Alta/Medio/Baja): Medio
Iteración Asignada: 1	
Responsables (es): Cindy Yazán	
Descripción: El usuario ciudadano en el backend podrá acceder a transacciones de lectura, escritura y consultas en lotes para recuperar la contraseña de su cuenta ingresando el correo electrónico con el que se registró.	
Observación: Recuperar la contraseña es una de las funciones que viene ligada al inicio, registro de cuenta.	

TABLA XIV: Historia de Usuario #4

Historia de Usuario	
Identificador: HU-04	Usuario: Administrador
Nombre Historia: Implementar las coordenadas del mapa para transacciones de consulta y lectura y generar la visualización de rutas, horarios y días de recolección de basura.	
Prioridad en Negocio (Alta/Medio/Baja): Alta	Riesgo en Desarrollo (Alta/Medio/Baja): Alta
Iteración Asignada: 2	

Responsables (es): Cindy Yazán
Descripción: El usuario administrador en el backend podrá acceder a las coordenadas de las rutas para generar la vista del frontend con las rutas, horario y días que pasa un camión para recolectar la basura urbana.
Observación: La información del mapa es propiciado de la página oficial de EMASEO.

TABLA XV: Historia de Usuario #5

Historia de Usuario	
Identificador: HU-05	Usuario: Administrador
Nombre Historia: Generar transacciones de lectura, escritura y consultas para el registro de recomendaciones en la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Medio	Riesgo en Desarrollo (Alta/Medio/Baja): Medio
Iteración Asignada: 2	
Responsables (es): Cindy Yazán	
Descripción: El usuario administrador en el backend podrá acceder a transacciones y escritura en lotes para generar la vista del frontend de la sección de recomendaciones o manejo de desechos, tanto de la aplicación web como de la móvil. Los campos que maneja esta sección son: <ul style="list-style-type: none"> • Título • Categoría • Contenido • Imagen 	
Observación: El último campo se debe ingresar un URL de la imagen o elegir una de las fotos predeterminadas.	

TABLA XVI: Historia de Usuario #6

Historia de Usuario	
Identificador: HU-06	Usuario: Administrador
Nombre Historia: Generar transacciones de lectura, escritura y consultas para visualizar comentarios o sugerencias en la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Medio	Riesgo en Desarrollo (Alta/Medio/Baja): Medio
Iteración Asignada: 2	
Responsables (es): Cindy Yazán	
Descripción: El usuario administrador en el backend podrá realizar transacciones de lectura para visualizar los comentarios de los usuarios ciudadanos, donde se podrá observar información respecto a: <ul style="list-style-type: none"> • Nombre de usuario 	

<ul style="list-style-type: none"> • Correo • Comentario
<p>Observación: Los comentarios se pueden marcar como revisado y al inicio presentará una previsualización con la opción de ver la sugerencia en detalle.</p>

TABLA XVII: Historia de Usuario #7

Historia de Usuario	
Identificador: HU-07	Usuario: Administrador
Nombre Historia: Generar transacciones de lectura, escritura y consultas para la gestión de noticias en la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Medio	Riesgo en Desarrollo (Alta/Medio/Baja): Medio
Iteración Asignada: 3	
Responsables (es): Cindy Yazán	
<p>Descripción: El usuario administrador en el backend podrá realizar transacciones de lectura y escritura para generar, modificar y eliminar las noticias del sistema web y móvil. En este apartado se podrá observar información respecto a:</p> <ul style="list-style-type: none"> • Título de la noticia • Categoría • Autor • Descripción 	
Observación: El administrador para realizar la gestión de esta sección necesita iniciar sesión.	

TABLA XVIII: Historia de Usuario #8

Historia de Usuario	
Identificador: HU-08	Usuario: Administrador
Nombre Historia: Generar transacciones de lectura, escritura y consultas para la gestión de videos en la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Medio	Riesgo en Desarrollo (Alta/Medio/Baja): Medio
Iteración Asignada: 3	
Responsables (es): Cindy Yazán	
<p>Descripción: El usuario administrador en el backend podrá realizar transacciones de lectura y escritura para generar, modificar y eliminar los videos del sistema web y móvil. En este apartado se podrá observar información respecto a:</p> <ul style="list-style-type: none"> • Título del video • Categoría • Autor • URL 	

Observación: El administrador para realizar la gestión de los videos en los sistemas necesita iniciar sesión.

TABLA XIX: Historia de Usuario #9

Historia de Usuario	
Identificador: HU-09	Usuario: Administrador
Nombre Historia: Generar transacciones de lectura, escritura y consultas para la gestión de contactos de recicladores en la aplicación.	
Prioridad en Negocio (Alta/Medio/Baja): Alta	Riesgo en Desarrollo (Alta/Medio/Baja): Alta
Iteración Asignada: 3	
Responsables (es): Cindy Yazán	
<p>Descripción: El usuario administrador en el backend podrá realizar transacciones de lectura y escritura para generar, modificar y eliminar los contactos de los recicladores del sistema web y móvil. En este apartado se podrá observar información como:</p> <ul style="list-style-type: none"> • Nombre • Dirección • Número de contacto • Actividad 	
Observación: El administrador es el único que podrá realizar la gestión de los contactos en los sistemas para ello necesita iniciar sesión.	

Product Backlog

A continuación, en la TABLA XX, se presenta un listado ordenado de los requerimientos del Sistema, esta lista se ha elaborado mediante las Historias de Usuario. Contiene información relacionada al identificador, el número de iteración y la prioridad asignada.

TABLA XX: Product Backlog

ID-HU	Historia de Usuario	Iteración	Prioridad
HU-01	Generar transacciones de lectura, escritura y consultas para el inicio y cierre de sesión en la aplicación.	1	Alta
HU-02	Generar transacciones de lectura, escritura y consultas para el registro en la aplicación.	1	Alta
HU-03	Generar transacciones de lectura, escritura y consultas para la recuperación de contraseña.	1	Alta

HU-04	Implementar las coordenadas del mapa para transacciones de consulta y lectura y generar la visualización de rutas, horarios y días de recolección de basura.	2	Medio
HU-05	Generar transacciones de lectura, escritura y consultas para el registro de recomendaciones en la aplicación.	2	Medio
HU-06	Generar transacciones de lectura, escritura y consultas para visualizar comentarios o sugerencias en la aplicación.	2	Medio
HU-07	Generar transacciones de lectura, escritura y consultas para la gestión de noticias en la aplicación.	3	Medio
HU-08	Generar transacciones de lectura, escritura y consultas para la gestión de videos en la aplicación.	3	Medio
HU-09	Generar transacciones de lectura, escritura y consultas para la gestión de contactos de recicladores en la aplicación.	3	Alta

Sprint Backlog

En la TABLA XXI, se presentan las tareas que se han ejecutado dentro de cada Sprint para el desarrollo de los sistemas web y móvil, las tareas se han determinado a partir del *Product Backlog*.

PRODUCT

Backlog

TABLA XXI: Sprint Backlog con las actividades identificadas para cada Sprint

ID-SB	Nombre	Historia de Usuario	Tareas	Tiempo Estimado (horas)
S0	Configuración del ambiente de desarrollo	N/A	<ul style="list-style-type: none"> • Creación de base de datos • Creación del proyecto • Definición de usuarios • Diseño Mockups 	40 h
S1	Autenticación	HU-01: Generar transacciones de lectura, escritura y consultas para el inicio y cierre de sesión en la aplicación.	<ul style="list-style-type: none"> • Diseño e implementación de las transacciones de lectura, escritura y consultas para el inicio de sesión y cierre de sesión de administradores y ciudadanos. • Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de usuarios ciudadanos. • Validación de los datos del registro de usuario nuevo. • Chequear que el registro sea exclusivo de un usuario. • Implementación de las transacciones de lectura, escritura y consulta de contraseña. 	40 h
		HU-02: Generar transacciones de lectura, escritura y consultas para el registro en la aplicación.		
		HU-03: Generar transacciones de lectura, escritura y consultas para la recuperación de contraseña.		
		HU-04: Implementar las coordenadas del mapa para transacciones de consulta y lectura y generar la visualización de rutas, horarios y días de recolección de	<ul style="list-style-type: none"> • Implementación de las coordenadas del mapa para transacciones de consulta y lectura y generar la visualización de rutas, horarios y días de recolección de 	

S2	Módulo informativo	basura.	<p>basura.</p> <ul style="list-style-type: none"> • Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de recomendación en el sistema. • Validación de los datos de la recomendación. • Diseño e implementación de las transacciones de lectura, escritura y consultas para el registro de sugerencias o comentarios de los usuarios en el sistema. • Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de noticias en el sistema. • Validación de los datos de las noticias. • Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de videos en el sistema. • Validación de los datos de los videos. 	100 h
		HU-05: Generar transacciones de lectura, escritura y consultas para el registro de recomendaciones en la aplicación.		
		HU-06: Generar transacciones de lectura, escritura y consultas para visualizar comentarios o sugerencias en la aplicación.		
		HU-07: Generar transacciones de lectura, escritura y consultas para la gestión de noticias en la aplicación.		
		HU-08: Generar transacciones de lectura, escritura y consultas para la gestión de videos en la aplicación.		
S3	Módulo complementario	HU-09: Generar transacciones de lectura, escritura y consultas para la gestión de contactos de recicladores en la aplicación.	<ul style="list-style-type: none"> • Diseño e implementación de las transacciones de lectura, escritura y consultas para la visualización de contactos de recicladores en el sistema. • Validación de los datos de los contactos de recicladores. 	20 h

S4	Pruebas y Despliegue	<ul style="list-style-type: none">• Pruebas de seguridad• Pruebas de compatibilidad• Pruebas de carga• Desplegar el sistema en Netlify.	40 h
TOTAL			240 h

Diseño de prototipos

El prototipado usando la herramienta Figma se puede encontrar en los siguientes URL.

Aplicativo Web:

- <https://www.figma.com/file/5TpgYT6AbKtWqxKLqWu3a3/TrashInformation?node-id=4%3A2>

Aplicativo Móvil:

- <https://www.figma.com/file/xQju3AQ6vhnHZ9cA955jIW/TI?node-id=20%3A4>

Esquema de la base de datos

En este apartado se muestran las 9 colecciones necesarias para el desarrollo del sistema y los servicios del backend. Con estas colecciones se puede manejar la información de forma organizada y permitir la consulta adecuada de los datos.

La Fig.41, muestra la configuración para la llamada de todos los documentos de los módulos informativos a la vista del usuario cliente y administrador.

+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Configuracion >	GlobalContactos >	+ Agregar campo
Contactos	GlobalNews	lastitemContact: 10
Mapa	GlobalRecomendation	
Noticias	GlobalSuggestions	
Personas	GlobalVideos	
Recomendaciones		
Roles		
Sugerencias		
Videos		

Fig.41: Colección y documentos de "Configuración"

Desde la Fig.42 hasta la Fig.49 se muestran las colecciones Contactos, Noticias, Personas, Recomendaciones, Roles, Sugerencias y Videos, cada uno con sus respectivos documentos donde se puede ver los campos que almacenan.

+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Configuración	10c >	+ Agregar campo
Contactos >	1c	activity: "Recicladora de papel, cartón, plástico, vidrio, metal, tetrapack"
Mapa	2c	address: "Parque de Carcelen Bajo"
Noticias	3c	id: "10c"
Personas	4c	lastName: "Carucuamba"
Recomendaciones	5c	name: "Gabriela"
Roles	6c	phoneNumber: "0959213616"
Sugerencias	7c	
Videos	8c	
	9c	

Fig.42: Colección y documentos del módulo "Contactos"

+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Configuración	Direcciones >	+ Agregar campo
Contactos		6
Mapa >		label: "Floresta"
Noticias		value: "frta"
Personas		7
Recomendaciones		label: "Jipijapa"
Roles		value: "jppj"
Sugerencias		8
Videos		label: "La Gasca"
		value: "lgsa"
		9
		label: "La Granja"
		value: "lg"

Fig.43: Colección y documentos de la sección "Mapa"

testinsercionjson	Noticias	10n
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Configuración	10n >	+ Agregar campo
Contactos	2n	author: "Empresa Metropolitana EMGIRS"
Mapa	33n	category: "Covid19"
Noticias >	36n	description: "Debido a la presencia del virus Covid19, la recolección de desechos contaminados ha aumentado en los últimos años, por lo que esta cifra de recolección puede incrementar aún más."
Personas	3n	id: "10n"
Recomendaciones	4n	noticiaPic: "https://firebasestorage.googleapis.com/v0/b/trash-information-appweb.appspot.com/o/plantillaimages%2Fccovid19.jpg?alt=media&token=9330e991-2f82-4ba5-a2fe-d0db316998dd"
Roles	5n	title: "EMGIRS-EP procesa 1000 kg/hora de desechos sanitarios en la capital"
Sugerencias	6n	url: "https://www.emgirs.gob.ec/index.php/noticiasep/674-emgirs-ep-procesa-1000-kg-hora-de-desechos-sanitarios-en-la-capital"
Videos	7n	
	9n	

Fig.44: Colección y documentos del módulo "Noticias"

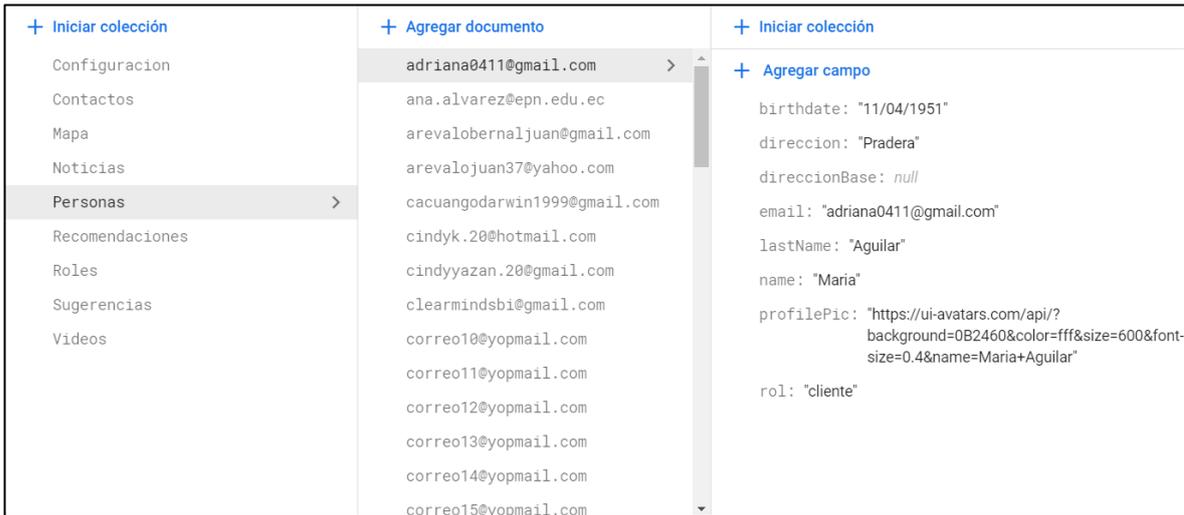


Fig.45: Colección y documentos del módulo "Personas"

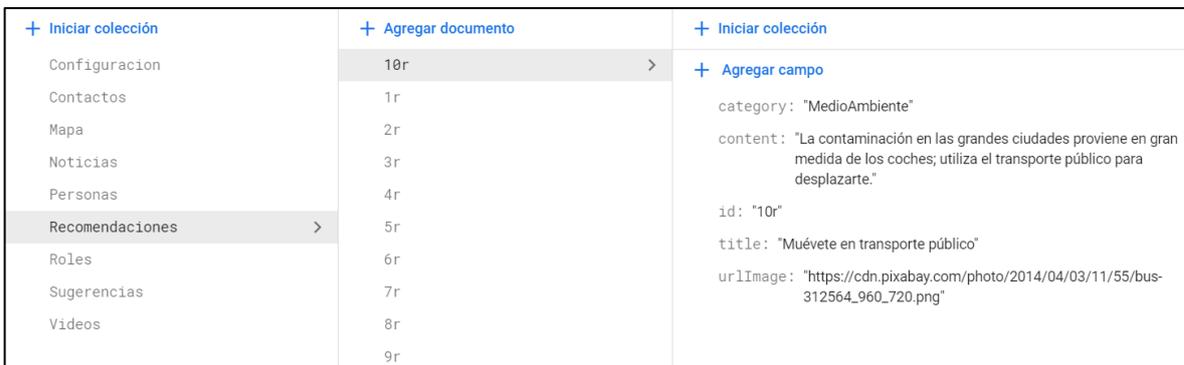


Fig.46: Colección y documentos del módulo "Recomendaciones"

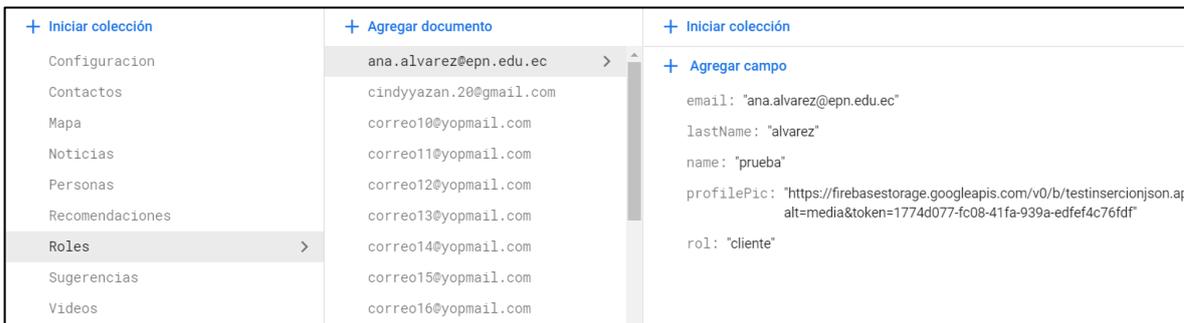


Fig.47: Colección y documentos del módulo "Roles"

+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Configuración	11s	+ Agregar campo
Contactos	12s	comment: "Este contacto debe ser actualizado ya que la persona se cambio de domicilio y esto puede causar confusiones"
Mapa	15s	email: "danielguachamin15@gmail.com"
Noticias	18s	id: "11s"
Personas	19s	lastName: "Guachamin"
Recomendaciones	22s	name: "Daniel"
Roles	23s	section: "Contacto de Abigail Benavides"
Sugerencias	24s	timeStamp: "11/07/2022"
Videos		

Fig.48: Colección y documentos del módulo "Sugerencias"

+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Configuración	10v	+ Agregar campo
Contactos	1v	author: "Teleamazonas Ecuador"
Mapa	2v	category: "Reciclaje"
Noticias	3v	id: "10v"
Personas	4v	prevImg: "https://img.youtube.com/vi/Xb4w6-0_Zp0/maxresdefault.jpg"
Recomendaciones	5v	title: "Una novedosa opción de reciclaje está dando resultados positivos en Quito"
Roles	6v	url: "https://www.youtube.com/watch?v=Xb4w6-0_Zp0"
Sugerencias	7v	urIID: "Xb4w6-0_Zp0"
Videos	8v	
	9v	

Fig.49: Colección y documentos del módulo "Videos"

Pruebas

Lista la etapa de promagramacion del proyecto se ha llevado a cabo la ejecucion de pruebas de seguridad, compatibilidad y de carga para verificar el optimo funcionamineto del backend.

Pruebas de seguridad

La Fig.50, muestra el detalle de la vulnerabilidad del puerto 443 abierto, este usado para servicios https. La Fig.51, identifica otro puerto abierto, este es el 80 para conexiones http, este trabaja en conjunto con el puerto 443. Y la Fig.52, presenta el resumen de como el proyecto presenta una riesgo al no tener políticas de seguridad.

Open TCP Port: 443	NMAP	https://trash-information-webapp.netlify.app/login	MEDIUM	OPEN	3 days ago	Accept Risk
Title	Open Port: 443					
Details	An open port may be an expected configuration. For example, web servers use port 80 to serve websites over http and port 443 to serve websites over https. For a list of commonly used ports see https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers .					
	An unexpected open port could give unintended access to applications, data, and private networks. Open ports can also be dangerous when expected services are out of date and exploited through security vulnerabilities.					

Fig.50: Prueba de seguridad, vulnerabilidad puerto 443

Open TCP Port: 80	NMAP	https://trash-information-webapp.netlify.app/login	MEDIUM	OPEN	3 days ago	Accept Risk
Title	Open Port: 80					
Details	<p>An open port may be an expected configuration. For example, web servers use port 80 to serve websites over http and port 443 to serve websites over https. For a list of commonly used ports see https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.</p> <p>An unexpected open port could give unintended access to applications, data, and private networks. Open ports can also be dangerous when expected services are out of date and exploited through security vulnerabilities.</p>					

Fig.51: Prueba de seguridad, vulnerabilidad puerto 80

Missing Anti-clickjacking Header	OWASP_ZAP	https://trash-information-webapp.netlify.app/login	MEDIUM	OPEN	3 days ago	Accept Risk
Name	Missing Anti-clickjacking Header					
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'Clickjacking' attacks.					
Solution	<p>Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.</p> <p>If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.</p>					
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options					
CWE Id	1021					
WASC Id	15					
Other Information						

Fig.52: Prueba de seguridad, vulnerabilidad de política de seguridad

Pruebas de compatibilidad

Estas pruebas muestran como los clientes HTTP de Postman y Thunder Client realizan las peticiones al servidor, se muestra una respuesta por parte de los servicios de la aplicación. Con Postman se ha usado su aplicación de escritorio para ejecutar las pruebas (Fig.53), por otro lado se ha instalado en Visual Studio Code la extensión de Thunder Client para generar los resultados de la Fig.54.

```
ProyectoGrad / https://trash-information-webapp.netlify.app/login

GET https://trash-information-webapp.netlify.app/login

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (11) Test Results

Pretty Raw Preview Visualize HTML

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>AplicacionWebTrashinformation</title>
7   <base href="/">
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="icon" type="image/x-icon" href="favicon.ico">
10  <link rel="preconnect" href="https://fonts.gstatic.com">
11  <style type="text/css">
12    @font-face {
13      font-family: 'Roboto';
14      font-style: normal;
15      font-weight: 300;
16      font-display: swap;
17      src: url(https://fonts.gstatic.com/s/roboto/v30/KF01CnqEu92Fr1MmSU5fCRc4AMP6lbBP.woff2) format('woff2');
18      unicode-range: U+0460-062F, U+1C80-1C88, U+20B4, U+2DE0-2DFF, U+A640-A69F, U+FE2E-FE2F;
19    }
20
21    @font-face {
22      font-family: 'Roboto';
23      font-style: normal;
24      font-weight: 300;
25      font-display: swap;
26      src: url(https://fonts.gstatic.com/s/roboto/v30/KF01CnqEu92Fr1MmSU5fCBc4AMP6lbBP.woff2) format('woff2');
27      unicode-range: U+0301, U+0400-045F, U+0490-0491, U+04B0-04B1, U+2116;
28    }
29
30    @font-face {
31      font-family: 'Roboto';
32      font-style: normal;
33      font-weight: 300;
34      font-display: swap;
35      src: url(https://fonts.gstatic.com/s/roboto/v30/KF01CnqEu92Fr1MmSU5fCBc4AMP6lbBP.woff2) format('woff2');
36      unicode-range: U+1F00-1FFF;
37    }
38  </style>
</head>
</html>
```

Fig.53: Prueba de compatibilidad con Postman

```

Status: 200 OK Size: 7.36 KB Time: 410 ms
Response Headers 12 Cookies Results Docs {}
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPr4itfJhxaWutUpFmBp4vmVor"
crossorigin="anonymous">
<script src="https://www.youtube.com/iframe_api"></script>
<style>.mat-typography{font:400 14px/20px Roboto,Helvetica Neue,sans-serif;letter-spacing:normal}html,body{height:100%;body{margin:0;font
-family:Inter,Helvetica Neue,sans-serif;font-style:normal;font-weight:500}</style><link
rel="stylesheet" href="styles.91aa7872378527a3.css" media="print"
onload="this.media='all'"><noscript><link rel="stylesheet"
href="styles.91aa7872378527a3.css"></noscript></head>
<body class="mat-typography">
<app-root></app-root>
<script src="runtime.4f73f64fac128ee0.js" type="module"></script><script
src="polyfills.4642cee85afd5469.js" type="module"></script><script
src="main.9acae714ed1179d7.js" type="module"></script>
</body></html>

```

Fig.54: Prueba de compatibilidad con Thunder Client

Pruebas de carga

Estas son las últimas pruebas por las que se ha sometido el sistema. En la Fig.55, muestra la prueba de carga en un servidor de Madrid donde se puede visualizar un estatus de aceptación a los servicios que posee la aplicación, la velocidad de carga es de 398 ms, velocidad considerablemente buena para la carga del sistema. Y en la Fig.56, se puede ver los detalles de la carga del servidor de Madrid a la aplicación, con la velocidad, la velocidad, la duración y el tipo de petición.

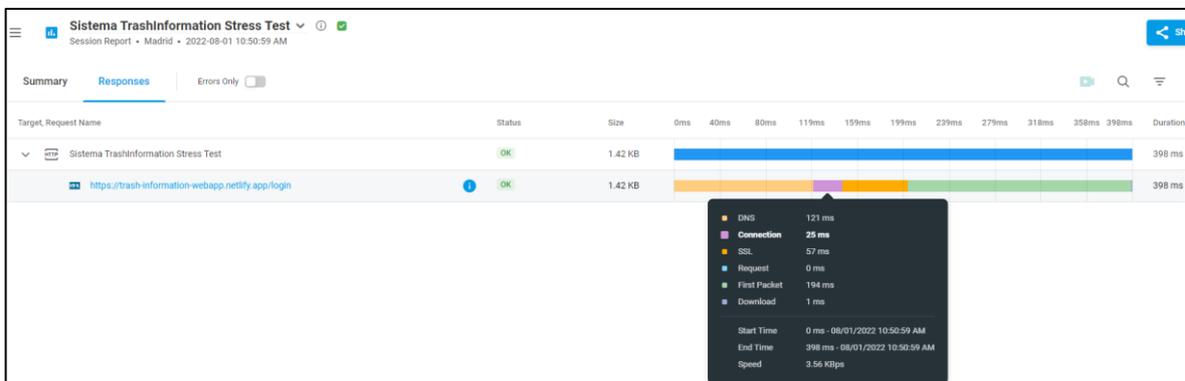


Fig.55: Prueba de carga en un servidor

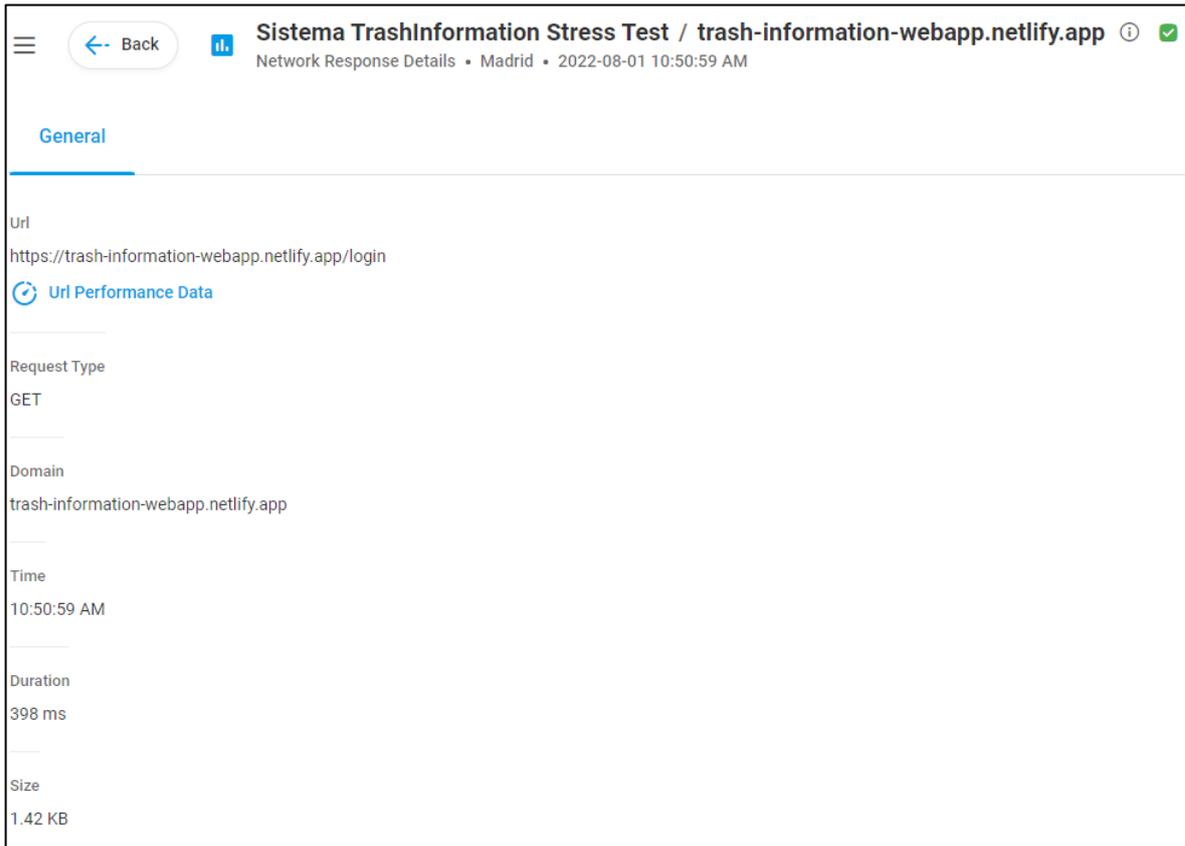


Fig.56: Prueba de carga, detalle de la prueba en el servidor

ANEXO III

En esta sección se encuentra el enlace para acceder al Manual de Usuario:

- https://www.youtube.com/watch?v=xJ3VRup4Zes&ab_channel=JeremyLe%C3%B3n

En este video se puede visualizar el funcionamiento del sistema web y móvil. Este enlace a YouTube se podrá observar rápidamente todas las secciones que tiene el sistema completo.

ANEXO IV

A continuación, se encuentran las credenciales para el acceso al sistema completo, además del repositorio donde se puede obtener el código fuente y el manual de instalación.

Credenciales de acceso

El siguiente enlace lleva al sistema desplegado a producción:

- <https://trash-information-webapp.netlify.app/>

Credenciales administrador:

- Correo del usuario: trashinfoadmin@gmail.com
- Contraseña: admin_2022*

Credenciales usuario ciudadano:

- Correo del usuario: cindyk.20@hotmail.com
- Contraseña: cindy123*

Repositorio del código fuente

En este link se puede visualizar y descargar todo el código fuente del proyecto.

- <https://github.com/Cindyk2052/BackendAppWebTI.git>

Repositorio manual de instalación

A continuación, se presenta el enlace que dirige al manual de instalación de este proyecto.

- https://epnecuador-my.sharepoint.com/:f/g/personal/cindy_yazan_epn_edu_ec/Ehzo3NxFgBpGsbPj5duLkRgBVnlb7tcjRQoCFWUOGcYrvA?e=vDtebz