



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTROTECNIA E INFORMÁTICA
INGENIERÍA EN INFORMÁTICA

DESARROLLO DE UN PANEL DE CONTROL PARA LA VISUALIZACIÓN Y OPTIMIZACIÓN DE LA OPERACIÓN DE RUTAS DE RECOLECCIÓN DE RESIDUOS SÓLIDOS: UN ENFOQUE DE INTELIGENCIA DE NEGOCIO INTEGRADO A SMART WASTE

Zachir Faúndez

zachir.faundez@usm.cl

Gabriel Jara Bulnes
Profesor Guía

Resumen: El presente trabajo aborda la falta de control operativo en flotas municipales de recolección de residuos sólidos, lo que genera cobertura irregular e ineficiencia. Se propone un sistema que incluye una aplicación móvil para conductores, una aplicación ciudadana de seguimiento y un panel administrativo interactivo basado en inteligencia de negocios (BI) y datos GPS. El sistema busca optimizar rutas, reducir costos y mejorar la toma de decisiones mediante métricas clave como tiempo de ruta, tonelaje recolectado y sectores no cubiertos. El desarrollo comenzó con un análisis de requerimientos junto a los operadores, identificando necesidades específicas y alineándolas con las capacidades del sistema. La solución propuesta integra datos en tiempo real y genera reportes históricos utilizando herramientas de BI. Los resultados muestran mejoras en la eficiencia operativa, reducción de tiempos y optimización de recursos, contribuyendo a una gestión más sostenible de residuos sólidos.

Palabras Clave: inteligencia de negocios, recolección de residuos, KPIs operacionales, análisis de datos, toma de decisiones.

1 Introducción

La gestión eficiente de residuos sólidos representa un desafío crítico para muchas municipalidades, tanto por las demandas operativas como por las expectativas ciudadanas. En esta sección, se detalla el contexto del problema en Viña del Mar, se describen las características actuales de la operación de recolección y se justifica la necesidad de una solución tecnológica basada en inteligencia de negocios.

1. Contextualización del Problema

La gestión de residuos sólidos es un desafío crucial para las municipalidades, que deben asegurar la eficiencia en la recolección, el transporte y la disposición final de los residuos, minimizando el impacto ambiental y maximizando la satisfacción ciudadana. El presente trabajo se centra en analizar el caso de la Municipalidad de Viña del Mar, en el contexto del proyecto Smart Waste, donde se busca optimizar la operación de recolección de residuos mediante un enfoque de inteligencia de negocios.

En Viña del Mar, existen dos operadores encargados de la recolección de residuos sólidos: Cosemar, una empresa concesionada que opera con una flota privada, y la propia municipalidad, que gestiona una flota de camiones. La mayor parte de los sectores de la comuna son atendidos por Cosemar, mientras que la flota municipal se encarga de una minoría de los sectores, estos sectores se diferencian en la recolección de cosemar en que aquí se hace a mano a diferencia de los asignados a cosemar que son por medio de camiones de recolección lateral, esto responde a razones históricas (los camiones municipales pasan por el centro de Viña del Mar por ejemplo) o de acceso (los camiones municipales realizaran una ruta a sectores periféricos).



El problema principal de la flota municipal radica en la falta de control sobre las rutas que siguen los camiones. A los conductores se les asigna un sector, pero no una ruta predefinida, lo que implica que la ruta que eligen puede no ser la óptima o, en algunos casos, no cubrir todos los lugares que deberían atenderse. Como consecuencia, se generan áreas sin cobertura adecuada y acumulación de residuos, lo que provoca insatisfacción ciudadana y un aumento en las quejas hacia el servicio de recolección municipal.

La solución propuesta en el proyecto llamada Smart Waste consta de 3 componentes:

1. App Móvil para el Conductor del Camión: Permite a los conductores registrar sus rutas, reportar problemas en tiempo real y optimizar la recolección.
2. App Web Administrativa: Brinda a la municipalidad una herramienta para gestionar rutas, vehículos y conductores, además de un dashboard con métricas clave para la supervisión y mejora del servicio.
3. App Móvil para Ciudadanos: Permite a los ciudadanos conocer la ubicación del camión recolector en tiempo real y mantenerse informados sobre el servicio.

Este sistema de información integrado está diseñado para atender las necesidades de tres públicos específicos: conductores, ciudadanos y administradores. Aunque se distribuye en tres aplicaciones, cada una depende de las otras para cumplir el propósito común de mejorar la eficiencia en la recolección de residuos, reducir costos operativos, aumentar la satisfacción ciudadana y promover la sostenibilidad en la gestión de residuos. De este modo, el sistema trasciende la simple suma de sus partes, operando como una solución unificada y sinérgica, de esta manera se busca mejorar la eficiencia en la recolección de residuos, reducir costos operativos, aumentar la satisfacción ciudadana y promover la sostenibilidad en la gestión de residuos.

Más específicamente, la aplicación captura datos detallados de los camiones, como su ubicación, hora de paso por cada ubicación, velocidad, informes de fallas mecánicas de los propios camiones o de desvíos de la ruta asignada, como también el tonelaje de inicio y final. Y también se recopilan datos generales como la matrícula del camión y la identidad del conductor. Con esta información, se pueden generar análisis de inteligencia de negocios que apoyen la toma de decisiones por parte de los encargados de la flota municipal.

El objetivo de esta tesina es abordar la extracción, almacenamiento y visualización de datos de recolección de residuos para proporcionar un sistema que permita mejorar la operación en la Municipalidad de Viña del Mar, sentando las bases para su futura expansión a otras municipalidades o sectores con necesidades similares.

2. Objetivos

1.2.1 Objetivo General

- Diseñar e implementar un panel de control interactivo y personalizado, basado en datos recopilados por la aplicación Smart Waste, que permita la visualización de la operación de rutas de recolección de residuos sólidos, ayudando a mejorar la toma de decisiones operativas del gestor de dichas rutas, como la optimización de las rutas, la asignación de recursos y la gestión de la flota.

1.2.2 Objetivos Específicos

- Desarrollar un panel de control que centralice y visualice los principales indicadores operativos de la recolección de residuos, tales como tiempos de recorrido, volúmenes recolectados, incidencias y eficiencia por camión.
- Implementar visualizaciones interactivas que permitan el análisis en tiempo real del desempeño de las rutas y su comparación.
- Identificar oportunidades para integrar capacidades predictivas que sugieran ajustes en las rutas según patrones históricos de recolección y condiciones operativas.



Evaluar el impacto del panel de control en la optimización de rutas y la eficiencia de la operación de recolección, utilizando datos disponibles, tanto reales como simulados, para medir la reducción de costos y tiempos operativos.

2 Marco Teórico

1. Antecedentes Generales

La ineficiencia de no tener un sistema para el control de rutas tiene un impacto directo tanto en los tiempos como en los costos operativos de la municipalidad, generando desperdicio de recursos y una mayor carga financiera. La aplicación del control, en cambio, permite reducir significativamente estos costos y mejorar la eficiencia del servicio. Además, la percepción ciudadana se ve afectada: una ruta que no logra recoger toda la basura en un área específica no solo genera molestias en la población, sino que también perjudica la imagen y la confianza en la gestión municipal.^[1]

En este contexto, se analizará como la inteligencia de negocios (BI) aplicada a los datos recogidos por nuestra aplicación puede ayudar a mejorar esta situación. El uso de datos para analizar y optimizar la recolección de residuos sólidos permitirá mejorar aspectos críticos como la eficiencia de las rutas, la distribución de los recursos y la capacidad de respuesta ante imprevistos.

2. Principios de la Inteligencia de Negocios

La inteligencia de negocios se refiere al conjunto de estrategias, tecnologías y herramientas que permiten a las organizaciones recopilar, procesar y analizar grandes volúmenes de datos para tomar decisiones informadas. Los principios fundamentales de la inteligencia de negocios incluyen;

2.2.1 Recolección y Almacenamiento de Datos

Uno de los pilares de BI es la capacidad de recolectar datos de diversas fuentes. Estos datos son almacenados en repositorios centralizados, como data warehouses, donde se consolidan para su posterior análisis.

2.2.2 Transformación de Datos en Información Procesable

La inteligencia de negocios se basa en la transformación de datos crudos en información útil. Esto se logra mediante procesos de limpieza, integración y normalización de datos. Una vez estructurados, los datos pueden ser analizados para obtener indicadores clave de rendimiento para la toma de decisiones.

2.2.3 Visualización y Acceso a la Información

BI permite que los datos se presenten de forma comprensible a través de herramientas de visualización como dashboards interactivos y reportes. Estas visualizaciones, como gráficos, tablas y mapas, facilitan el análisis de patrones, tendencias y áreas de mejora en tiempo real.

2.2.4 Toma de Decisiones Basada en Datos

El objetivo final de la inteligencia de negocios es apoyar a los gestores y operadores en la toma de decisiones basada en datos objetivos.



3. Soluciones Aplicadas a la Industria

Tenemos varios ejemplos como el control de rutas y la inteligencia de negocios fue aplicado en el ámbito de recolección de basura siendo todo esto dentro de un marco de una “Smart City” o ciudad inteligente. El concepto de “Smart City” o ciudad inteligente se refiere a la utilización de tecnologías de la información y la comunicación (TIC) para mejorar la calidad de vida de los ciudadanos, la eficiencia de los servicios públicos y la sostenibilidad urbana. En este contexto, la gestión de residuos sólidos es un área clave donde la tecnología puede generar un impacto significativo. Siendo algunos de los siguientes ejemplos soluciones aplicadas que voy a tomar como casos de estudio.

2.3.1 Ciudad de Barcelona (España), Proyecto de Smart City

Barcelona ha sido pionera en la implementación de tecnologías inteligentes para la gestión urbana, incluida la optimización de la recolección de residuos sólidos. Utilizando sensores colocados en los contenedores de basura, la ciudad puede monitorear el nivel de llenado en tiempo real y ajustar las rutas de los camiones de recolección para atender las zonas más críticas. Esta solución ha logrado reducir los tiempos de recolección y los costos operativos, a la vez que ha mejorado el servicio al evitar desbordamientos de basura en las calles. ^{[2] [3]}

2.3.2 Nueva York (EE.UU.) - DSNY (Department of Sanitation)

Nueva York, a través de su Departamento de Saneamiento, implementó un sistema de BI que analiza datos históricos de recolección y utiliza modelos predictivos para optimizar las rutas de los camiones. Esta plataforma integra datos sobre tráfico, patrones climáticos y densidad de población, permitiendo ajustar las rutas de manera dinámica. Como resultado, la ciudad ha logrado un aumento significativo en la eficiencia operativa y una reducción considerable en los costos asociados al combustible y mantenimiento de vehículos. ^[4]

2.3.3 Singapur - Smart Nation Initiative

Singapur ha utilizado la inteligencia de negocios no solo en la gestión de residuos, sino en otras áreas como el transporte y la gestión de estacionamientos. La plataforma Smart Nation recopila datos de múltiples fuentes y los utiliza para mejorar la eficiencia del transporte público y la movilidad urbana. Los datos recopilados se utilizan para generar paneles de control interactivos que permiten a los tomadores de decisiones ajustar operaciones en tiempo real, mejorando la satisfacción ciudadana y reduciendo los tiempos de espera. ^[5]

4. Estado del Arte

Dentro de la academia, las tecnologías de vanguardia en inteligencia de negocios aplicadas al control de rutas de recolección de residuos sólidos se centran en el monitoreo, la predicción y la toma de decisiones en tiempo real. Sin embargo, además de predecir fallos mecánicos y optimizar rutas, un enfoque innovador podría centrarse en anticipar aumentos en la generación de residuos sólidos, lo cual sigue siendo un área poco explorada en la literatura existente.

El uso de machine learning y big data puede permitir identificar patrones relacionados con festividades, eventos estacionales y cambios demográficos. Por ejemplo, durante festividades como Navidad o Año Nuevo, el volumen de residuos suele incrementarse debido al consumo masivo y el uso de empaques. Algoritmos basados en datos históricos podrían modelar estas tendencias, generando predicciones precisas sobre el aumento esperado de residuos en fechas clave.

Estas capacidades no solo optimizarían la asignación de recursos, sino que también facilitarían una planificación más proactiva de rutas y horarios. Un estudio publicado en Information Systems Research resalta cómo la combinación de datos históricos con modelos predictivos puede dinamizar las operaciones, integrándose en paneles interactivos que ofrecen análisis en tiempo real y comparaciones



históricas bajo diversas condiciones operativas. Aplicar estas ideas a la predicción de generación de residuos abriría nuevas oportunidades para mejorar la eficiencia en contextos urbanos. Smart Waste recolectará datos que dejarían la base para la implementación de un proyecto en esta línea, utilizando la información recopilada como base para desarrollar capacidades predictivas más avanzadas. [\[6\]](#) [\[7\]](#)

3 Metodología

1. Extracción, Transformación y Carga

Los datos serán extraídos desde la base de datos central de la aplicación, la cual está alojada en Firebase. Estos datos incluyen variables como la ubicación geográfica, la hora, el camión y su conductor, y en el futuro también se incluirán reportes asociados a cada camión y ruta. Actualmente, no existen datos históricos de la operación, por lo que se trabajara con nuevos datos adquiridos en una implementación a baja escala de nuestra aplicación

Una vez obtenidos los datos, serán transferidos a un data warehouse alojado en BigQuery. El dashboard para la visualización de estos datos será desarrollado en Looker Studio y se integrará en una página web. La estructura de datos de Firebase, optimizada para la operación en tiempo real de Smart Waste, probablemente difiera de la necesaria para la generación de informes y análisis. Por ello, se ha optado por utilizar BigQuery como data warehouse intermedio. Esto permite transformar y modelar los datos de Firebase para adaptarlos a las necesidades del panel de control en Looker Studio, aprovechando la integración nativa entre ambas herramientas.

2. Desarrollo del Panel de Control

El panel de control, como se mencionó anteriormente, será desarrollado en Looker Studio. Incluirá visualizaciones como gráficos que representen la eficiencia de rutas específicas y generales, junto con estadísticas de los errores más comunes extraídas de los informes operativos. También se presentarán datos sobre el tiempo promedio y total de operación. Todas estas métricas podrán filtrarse por conductor, camión, sector, o por intervalos de tiempo (día, mes, año), lo que permitirá una personalización detallada del análisis operativo.

En el dashboard se podrán comparar conductores, evaluar el tiempo de recolección, y analizar el mantenimiento y la disponibilidad de los vehículos, entre otras métricas.

3. Visualización de Rutas

El panel de control permitirá visualizar las rutas históricas de los camiones mediante la API de Google Maps. Esta funcionalidad ofrecerá diferentes opciones de filtrado para un análisis a medida:

Visualización por período: se podrán visualizar rutas específicas, rutas mensuales o rutas dentro de un período de tiempo definido.

Visualización por conductor o camión: se podrán visualizar las rutas realizadas por un conductor o camión específico.

De esta manera, los gestores podrán analizar el rendimiento histórico de las rutas, identificar patrones y tendencias, y tomar decisiones informadas para optimizar la operación.

4. Validación y Evaluación

Para asegurar la calidad y usabilidad del sistema, se llevará a cabo un proceso de validación que involucre a los operadores de la flota. Este proceso se basará en las siguientes pruebas:

-Pruebas de usabilidad: Se evaluará la facilidad de uso del panel de control y los mapas interactivos, considerando lo intuitiva que fuera ser la interfaz, la claridad de las visualizaciones y la eficiencia en la



navegación. Se realizarán pruebas con los operadores para observar su interacción con el sistema, identificar posibles dificultades y recoger sus sugerencias de mejora.

-Verificación de datos: Se evaluará la coherencia y confiabilidad de los datos históricos, asegurando la integridad de la información utilizada para el análisis.

4 Solución Propuesta

La propuesta de solución está orientada a integrar tecnologías de inteligencia de negocios y sistemas de gestión operativa que permitan mejorar la eficiencia de la recolección de residuos sólidos en Viña del Mar. A continuación, se presenta un diseño que considera los aspectos clave para implementar esta solución de manera efectiva.

1. Diseño

El diseño de la solución aborda tanto los aspectos funcionales como técnicos necesarios para cumplir con los objetivos planteados en esta tesina. Se prioriza la experiencia del usuario, la escalabilidad del sistema y la precisión en la recopilación y visualización de datos, todo ello basado en la integración de herramientas modernas de análisis y visualización. El primer paso es un análisis detallado de los requerimientos para asegurar que la solución propuesta atienda todas las necesidades operativas

4.1.1 Análisis de Requerimientos

Para el diseño y desarrollo del dashboard, se trabajó estrechamente con el encargado de la operación de la flota municipal de Viña del Mar. Durante este proceso, se identificaron las necesidades clave del sistema a partir de un análisis profundo de los procesos de negocio involucrados en la gestión de recolección de residuos sólidos.

El gestor de la flota opera en un entorno dinámico, donde enfrenta desafíos relacionados con la asignación de recursos, la eficiencia operativa y la respuesta a imprevistos, como desvíos de rutas o fallas mecánicas. Además, debe equilibrar las demandas operativas con las expectativas ciudadanas de un servicio eficiente y confiable. Este contexto obliga al gestor a tomar decisiones estratégicas y operativas basadas en datos precisos y en tiempo real.

Entre las decisiones estratégicas, se encuentran la planificación de rutas para maximizar la cobertura y la eficiencia de la recolección, así como la evaluación del desempeño general de los camiones y conductores. A nivel operativo, las decisiones incluyen la redistribución de recursos en caso de contingencias, el monitoreo de indicadores clave de rendimiento (KPIs) y la identificación de áreas críticas que requieren atención inmediata.

Con base en este análisis, se concluyó que el dashboard debe proporcionar herramientas que faciliten tanto la toma de decisiones estratégicas como operativas. Esto llevó a la definición de los siguientes requerimientos:

Métricas clave para la operación:

Tiempo de Ruta: seguimiento preciso del tiempo que toma cada camión en completar su ruta asignada, permitiendo comparar la eficiencia entre las distintas rutas en diversos días.

Tiempo de Camiones Operativos: mostrar el tiempo en que los camiones están efectivamente operativos en la recolección.

Tonelaje de los Camiones: monitoreo del peso total recolectado por cada camión, brindando una visión precisa de la capacidad de carga utilizada frente a la capacidad total de los vehículos.



Requerimientos para la visualización de rutas:

Filtrado por período de tiempo: permitir seleccionar y visualizar rutas de días específicos, semanas, meses o rangos personalizados de tiempo.

Filtrado por vehículo o conductor: facilitar la selección de rutas específicas de un camión o un conductor para análisis detallados.

Mapa interactivo: mostrar las rutas en un mapa interactivo que permita acercar, alejar y desplazarse por el área de interés.

Indicadores visuales: diferenciar claramente las rutas cuando hay más de una ruta en el área de interés.

Datos complementarios: tiempo de inicio y finalización de la ruta, distancia total recorrida y tonelaje recolectado.

Esta estructuración asegura que los KPIs y herramientas ofrecidas en el dashboard respondan directamente a las necesidades y retos identificados en el análisis de los procesos de negocio y contexto del gestor de flota.

4.1.2 Definición de la Arquitectura

La arquitectura de la solución se basa en un flujo de datos diseñado para cumplir con las etapas del ciclo de tratamiento de datos en inteligencia de negocios. Este flujo comienza con la aplicación móvil utilizada por los conductores de los camiones recolectores, que registra en tiempo real datos de ubicación, tiempo, eventos e informes. Estos datos son enviados a una base de datos NoSQL en Firebase, cuya estructura está organizada en colecciones, compuestas por documentos que representan registros únicos, como informes de conductores o puntos de ubicación de los camiones. El siguiente paso en el ciclo de datos es la transferencia a BigQuery, un almacén de datos en la nube que permite consolidar y transformar la información capturada. Esta transferencia se realiza a través de un mecanismo de "changelog", que captura los cambios en la base de datos Firebase y los propaga a BigQuery. Este proceso garantiza que los datos estén disponibles en formato SQL, facilitando su estructuración en vistas optimizadas para el análisis. Las vistas estructuradas creadas en BigQuery organizan los datos para que sean fácilmente accesibles y comprensibles, alineándose con la fase de transformación descrita en el marco teórico.

Finalmente, los datos procesados son visualizados en Looker Studio mediante dashboards interactivos. Aunque los datos lleguen en tiempo real a BigQuery, existe una demora de aproximadamente 15 minutos en la transferencia hacia Looker Studio. Esta última etapa completa el ciclo de BI, donde la información transformada se presenta de manera visual e intuitiva, permitiendo a los gestores tomar decisiones informadas basadas en datos procesados y relevantes.

Este diseño arquitectónico integra cada fase clave del ciclo de BI — recolección, almacenamiento, transformación y visualización — para maximizar la utilidad de los datos y facilitar su aplicación en la optimización operativa.

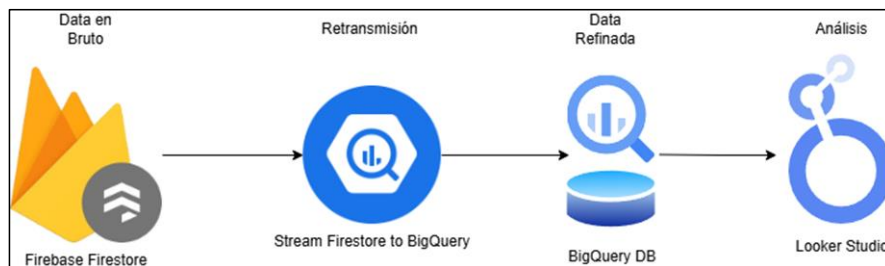


Ilustración 1. Flujo de Datos. (Fuente: Smart Waste, 2024)



4.1.3 Diseño de la Interfaz de Usuario

Las visualizaciones estarán ubicadas en la página de inicio de la sección destinada a los operadores por lo que al entrar será lo primero en verse. En la parte superior, los operadores podrán ver las métricas clave representadas mediante gráficos de barras interactivos, los cuales serán clickeables para permitir el filtrado de los datos según sea necesario. Más abajo, se encontrará el panel de visualización de rutas, donde los usuarios podrán analizar las rutas históricas y obtener información detallada sobre el desempeño operativo.

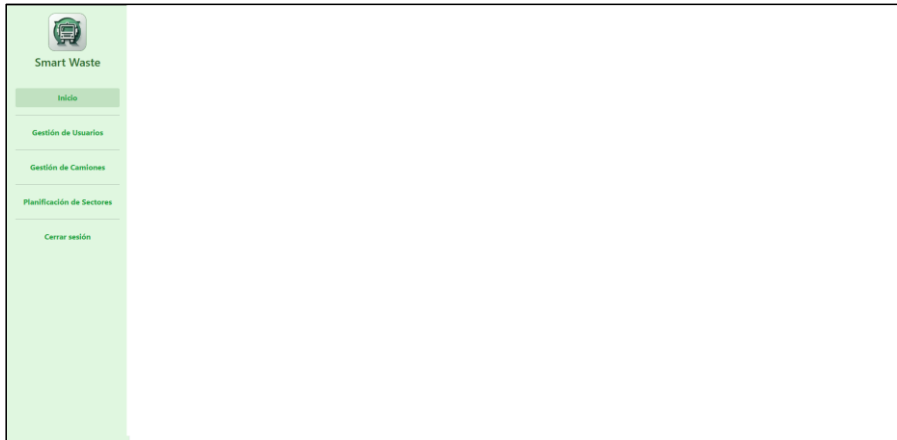


Ilustración 2. Página de Inicio donde estarán las Visualizaciones

2. Desarrollo

4.2.1 Extracción, Transformación y Carga

Extracción

Captura de datos

La captura de los datos ocurre en Firebase por medio de la aplicación de conductores.

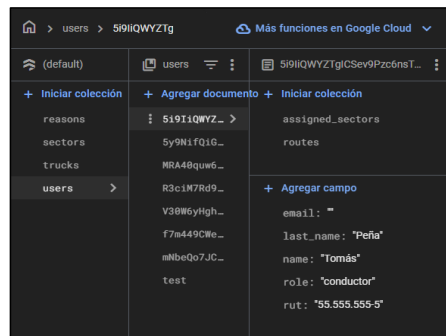


Ilustración 3. Colecciones de la base de datos



Como se puede apreciar en la ilustración 3, existen 4 colecciones las cuales contienen los siguientes datos:

Reasons: donde se guardan las razones de los informes.

Sectors: sectores asignados para la operación.

Trucks: camiones con patente y estado funcional.

Users: colección Principal, en ella se guardan los datos de los usuarios de todo nuestro sistema, en caso de los con rol "conductor", esto también se les guarda el assigned_sector (sector asignado por día y hora) y routes.

Comentado [PG1]: Ojo. Mantener la escritura en tercera persona.

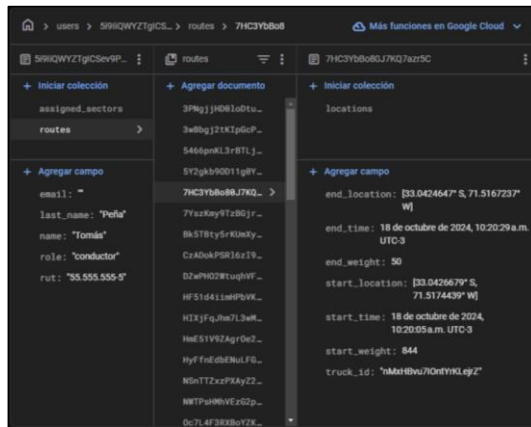


Ilustración 4. Colección Routes

En Routes, se encuentran los datos de las rutas, tanto sus estadísticas de inicio y de final de la ruta, como también qué camión se usó por medio de truck_id.

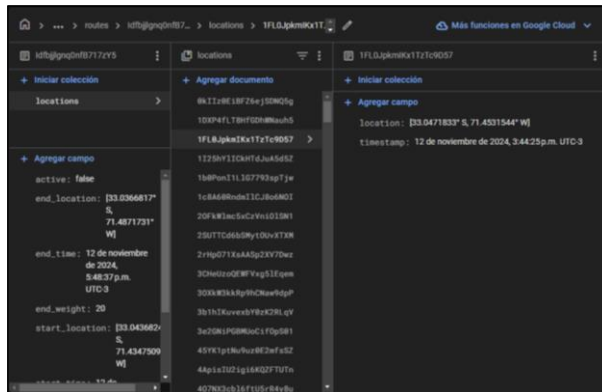


Ilustración 5. Locations



Finalmente tenemos la colección locations que corresponde a cada punto geográfico donde se recolectó datos junto con la hora, estos datos son los que nos permitirá finalmente la visualización de las rutas como también de los datos adjuntos.

Extracción de datos de Firebase a BigQuery

La extracción de datos se realiza mediante la extensión "Stream Firestore to BigQuery". Esta función captura en tiempo real todos los cambios en la base de datos Firestore y los transmite a BigQuery, los cuales generan 2 elementos para cada colección que se escoja; una tabla llamada "changelog", que muestra todas las modificaciones hechas en Firebase (creación, modificación y eliminación); y latest, que es una vista que muestra los datos tal cual están en Firebase. Es en base a estas últimas vistas en las que se va a trabajar; en este caso, para la creación de estos elementos se escogieron las 4 colecciones principales más la colección de locations. [\[8\]](#)

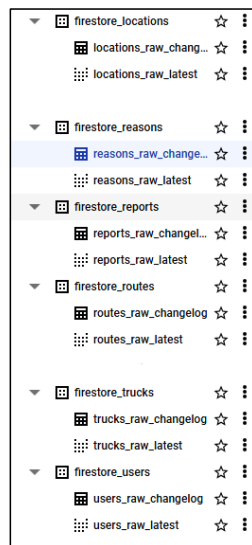


Ilustración 6. Elementos Generados



Transformación

document_name	document_id	timestamp	event_id
<input type="checkbox"/> Value	<input type="checkbox"/> Value	<input type="checkbox"/> Value	<input type="checkbox"/> Value
Count	Count	Count	Count
projects/sm... 1	7yxc2f0y9Tz... 1	2024-11-11 ... 1	d6e416e5-ce... 1
projects/sm... 1	0ZwPH0ZwH... 1	2024-11-11 ... 1	201f2f9d-9e... 1
projects/sm... 1	HFS1d4imH... 1	2024-11-04 ... 1	5129af6a-b7... 1
projects/sm... 1	CzADokPSR... 1	2024-11-06 ... 1	98b7e483-2f... 1
projects/sm... 1	3w8Bg29op... 1	2024-11-12 ... 1	4597a008-0... 1
projects/sm... 1	3PNggHD0o... 1	2024-10-30 ... 1	d53d2576-d... 1
projects/sm... 1	5Y2g69OD... 1	2024-11-13 ... 1	061e8a16-af... 1
projects/sm... 1	BkSTBy5K... 1	2024-11-10 ... 1	4650f033-62... 1
projects/sm... 1	7HC3Yb8oB... 1	2024-10-18 ... 1	7833cc16-1e... 1
projects/sm... 1	S466pnK3r... 1	2024-10-18 ... 1	d851171a-9... 1

operation	data	old_data	path_params
<input type="checkbox"/> Value	<input type="checkbox"/> Value	<input type="checkbox"/> Value	<input type="checkbox"/> Value
Count	Count	Count	Count
UPDATE 68	{active:'fals... 1	NULL 2	{userid:'59... 52
CREATE 2	{active:'fals... 1	{active:'stou... 1	{userid}'77... 11
	{active:'fals... 1	{active:'stou... 1	{userid}'5y... 6
	{active:'fals... 1	{active:'stou... 1	{userid}'tes... 1
	{active:'fals... 1	{star'locati... 1	

Ilustración 7. Datos Crudos routes_raw_latest

Tablas antes del procesado.

Todas las tablas siguen el mismo formato, los datos a usar de estas tablas serían los siguientes:

- document_id: ID Primaria de la tabla
- data: Los campos del documento extraídos de Firebase, estos están en formato JSON.
- path_params: ID secundaria de la tabla

Vista para Métricas

Para la visualización de métricas se realizó la siguiente vista con los siguientes datos:

Field name	Type
route_id	STRING
total_distance_km	FLOAT
start_weight	STRING
end_weight	STRING
start_time	TIMESTAMP
end_time	TIMESTAMP
route_duration_minutes	FLOAT
license_plate	STRING
driver_name	STRING

Ilustración 8. Vista Visualización Métricas

La idea es tener todos los datos disponibles para su muestra de forma fácil. Los datos más importantes serían `total_distance_km`, que corresponde a un campo calculado de la distancia recorrida sumando todos los puntos, `start_time` y `end_time`, que indican la fecha y hora de salida y llegada del camión (cuando empieza y termina su ruta), así como también `start_weight` y `end_weight`, que representan el peso de salida y llegada del camión. Además, se incluye `route_id`, que es el identificador único de cada ruta, y `driver_name`, que muestra el nombre completo del conductor. Otro campo relevante es



license_plate, que corresponde al número de placa del camión utilizado en la ruta. La route_duration_minutes es el tiempo total en minutos que el camión tarda para completar la ruta. Estos 6 valores son los necesarios para la creación de gráficos acordes que ayuden a analizar y optimizar las operaciones de recolección de residuos. Para poder generar esta vista, se usó la siguiente query:

```
1 WITH locations AS (  
2   SELECT  
3     JSON_VALUE(path_params, '$.routeid') AS route_id,  
4     TIMESTAMP(timestamp) AS first_timestamp,  
5     CAST(JSON_VALUE(data, '$.location..latitude') AS FLOAT64) AS latitude,  
6     CAST(JSON_VALUE(data, '$.location..longitude') AS FLOAT64) AS longitude  
7   FROM  
8     `smart-waste-2df41.firestore_locations.locations_raw_latest`  
9 ),  
10  
11 filtered_locations AS (  
12   SELECT  
13     route_id,  
14     first_timestamp,  
15     latitude,  
16     longitude,  
17     ST_DISTANCE(  
18       ST_GEOPOINT(longitude, latitude),  
19       LAG(ST_GEOPOINT(longitude, latitude)) OVER (PARTITION BY route_id ORDER BY first_timestamp)  
20     ) AS distance_to_previous_point  
21   FROM locations  
22 ),  
23  
24 distances AS (  
25   SELECT  
26     route_id,  
27     first_timestamp,  
28     latitude,  
29     longitude,  
30     ST_DISTANCE(  
31       ST_GEOPOINT(longitude, latitude),  
32       LEAD(ST_GEOPOINT(longitude, latitude)) OVER (PARTITION BY route_id ORDER BY first_timestamp)  
33     ) / 1000 AS distance_km  
34   FROM  
35     filtered_locations  
36   WHERE distance_to_previous_point > 80 OR distance_to_previous_point IS NULL  
37 ),  
38  
39 routes AS (  
40   SELECT  
41     document_id AS route_id,  
42     JSON_EXTRACT_SCALAR(data, '$.start_weight') AS start_weight,  
43     JSON_EXTRACT_SCALAR(data, '$.end_weight') AS end_weight,  
44     JSON_EXTRACT_SCALAR(data, '$.start_time..seconds') AS start_time_seconds,  
45     JSON_EXTRACT_SCALAR(data, '$.end_time..seconds') AS end_time_seconds,  
46     JSON_VALUE(data, '$.truck_id') AS truck_id,  
47     JSON_VALUE(path_params, '$.userid') AS user_id  
48   FROM  
49     `smart-waste-2df41.firestore_routes.routes_raw_latest`  
50   WHERE  
51     JSON_VALUE(data, '$.active') = 'false'  
52 ).
```

Consulta 1. Query General para métricas

```
54 trucks AS (  
55   SELECT  
56     document_id AS truck_id,  
57     JSON_VALUE(data, '$.license_plate') AS license_plate  
58   FROM  
59     `smart-waste-2df41.firestore_trucks.trucks_raw_latest`  
60 ),  
61  
62 users AS (  
63   SELECT  
64     document_id AS user_id,  
65     CONCAT(JSON_VALUE(data, '$.name'), ' ', JSON_VALUE(data, '$.last_name')) AS driver_name  
66   FROM  
67     `smart-waste-2df41.firestore_users.users_raw_latest`  
68 )  
69  
70 SELECT  
71   r.route_id,  
72   SUM(d.distance_km) AS total_distance_km,  
73   r.start_weight,  
74   r.end_weight,  
75   TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.start_time_seconds AS INT64)), INTERVAL 3 HOUR) AS start_time,  
76   TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.end_time_seconds AS INT64)), INTERVAL 3 HOUR) AS end_time,  
77   TIMESTAMP_DIFF(  
78     TIMESTAMP_SECONDS(CAST(r.end_time_seconds AS INT64)), INTERVAL 3 HOUR),  
79     TIMESTAMP_SECONDS(CAST(r.start_time_seconds AS INT64)), INTERVAL 3 HOUR),  
80   SECOND  
81   ) / 60 AS route_duration_minutes,  
82   t.license_plate,  
83   u.driver_name  
84 FROM  
85   distances d  
86 JOIN  
87   routes r ON d.route_id = r.route_id  
88 JOIN  
89   trucks t ON r.truck_id = t.truck_id  
90 LEFT JOIN  
91   users u ON r.user_id = u.user_id  
92 WHERE  
93   d.distance_km IS NOT NULL  
94 GROUP BY  
95   r.route_id, r.start_weight, r.end_weight, r.start_time_seconds, r.end_time_seconds, t.license_plate, u.driver_name  
96 ORDER BY  
97   r.route_id;  
98
```

Consulta 2. Query General para métricas



Este primer paso, como se puede apreciar en la Consulta 3 se extrae la información básica de cada punto de ubicación, incluyendo el ID de la ruta, la marca de tiempo, la latitud y la longitud. Esto se hace para posteriormente hacer el cálculo de total_distance_km.

```
WITH locations AS (  
  SELECT  
    JSON_VALUE(path_params, '$.routeid') AS route_id,  
    TIMESTAMP(timestamp) AS first_timestamp,  
    CAST(JSON_VALUE(data, '$.location._latitude') AS FLOAT64) AS latitude,  
    CAST(JSON_VALUE(data, '$.location._longitude') AS FLOAT64) AS longitude  
  FROM  
    `smart-waste-2df41.firestore_locations.locations_raw_latest`  
)
```

Consulta 3. Query Locations

Por otro lado, en la consulta 4 se calcula la distancia entre cada punto y el punto anterior en la misma ruta. Esto permite identificar puntos que están muy cerca uno del otro.

```
filtered_locations AS (  
  SELECT  
    route_id,  
    first_timestamp,  
    latitude,  
    longitude,  
    ST_DISTANCE(  
      ST_GEOGPOINT(longitude, latitude),  
      LAG(ST_GEOGPOINT(longitude, latitude)) OVER (PARTITION BY route_id ORDER BY first_timestamp)  
    ) AS distance_to_previous_point  
  FROM locations  
)
```

Consulta 4. Query Cálculo Puntos

Después, en la consulta 5, se calcula la distancia entre cada punto y el siguiente en la ruta, pero se filtran aquellos puntos que estén a menos de 80 metros del punto anterior. Esto es bueno para distancias en línea recta, pero no es muy efectivo para puntos muy seguidos (como cuando un camión va por calles muy juntas), siendo esto una solución temporal al no estar considerado el cálculo de distancias directamente en la aplicación móvil de conductor. Esto evita la suma de distancias redundantes. Además, se incluye el primer punto de cada ruta, ya que no tiene un punto anterior.

```
distances AS (  
  SELECT  
    route_id,  
    first_timestamp,  
    latitude,  
    longitude,  
    ST_DISTANCE(  
      ST_GEOGPOINT(longitude, latitude),  
      LEAD(ST_GEOGPOINT(longitude, latitude)) OVER (PARTITION BY route_id ORDER BY first_timestamp)  
    ) / 1000 AS distance_km  
  FROM  
    filtered_locations  
  WHERE distance_to_previous_point > 80 OR distance_to_previous_point IS NULL  
)
```

Consulta 5. Query Cálculo Distancia



Además, como se aprecia en la consulta 6, se extrae información relevante de las rutas, como el peso inicial y final del camión, los tiempos de inicio y fin, y los IDs del camión y del conductor. También se obtiene información de los camiones, como la placa, y de los usuarios, como el nombre completo.

```
routes AS (  
  SELECT  
    document_id AS route_id,  
    JSON_EXTRACT_SCALAR(data, '$.start_weight') AS start_weight,  
    JSON_EXTRACT_SCALAR(data, '$.end_weight') AS end_weight,  
    JSON_EXTRACT_SCALAR(data, '$.start_time_seconds') AS start_time_seconds,  
    JSON_EXTRACT_SCALAR(data, '$.end_time_seconds') AS end_time_seconds,  
    JSON_VALUE(data, '$.truck_id') AS truck_id,  
    JSON_VALUE(path_params, '$.userid') AS user_id  
  FROM  
    `smart-waste-2df41.firestore_routes.routes_raw_latest`  
  WHERE  
    JSON_VALUE(data, '$.active') = 'false'  
),  
trucks AS (  
  SELECT  
    document_id AS truck_id,  
    JSON_VALUE(data, '$.license_plate') AS license_plate  
  FROM  
    `smart-waste-2df41.firestore_trucks.trucks_raw_latest`  
),  
users AS (  
  SELECT  
    document_id AS user_id,  
    CONCAT(JSON_VALUE(data, '$.name'), ' ', JSON_VALUE(data, '$.last_name')) AS driver_name  
  FROM  
    `smart-waste-2df41.firestore_users.users_raw_latest`  
)
```

Consulta 6. Query Información Relevante

Finalmente, por medio de la Query mostrada en la consulta 7, se une toda esta información para calcular la distancia total de cada ruta, el peso transportado, la duración, la placa del camión y el nombre del conductor. Los resultados se filtran para mostrar solo las rutas con distancia calculada, se agrupan por información de la ruta y se ordenan por ID de la ruta.

```
SELECT  
  r.route_id,  
  SUM(d.distance_km) AS total_distance_km,  
  r.start_weight,  
  r.end_weight,  
  TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.start_time_seconds AS INT64)), INTERVAL 3 HOUR) AS start_time,  
  TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.end_time_seconds AS INT64)), INTERVAL 3 HOUR) AS end_time,  
  TIMESTAMP_DIFF(  
    TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.end_time_seconds AS INT64)), INTERVAL 3 HOUR),  
    TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.start_time_seconds AS INT64)), INTERVAL 3 HOUR),  
    SECOND  
  ) / 60 AS route_duration_minutes,  
  t.license_plate,  
  u.driver_name  
FROM  
  distances d  
JOIN  
  routes r ON d.route_id = r.route_id  
JOIN  
  trucks t ON r.truck_id = t.truck_id  
LEFT JOIN  
  users u ON r.user_id = u.user_id  
WHERE  
  d.distance_km IS NOT NULL  
GROUP BY  
  r.route_id, r.start_weight, r.end_weight, r.start_time_seconds, r.end_time_seconds, t.license_plate, u.driver_name  
ORDER BY  
  r.route_id;
```

Consulta 7. Query Unión Datos



Vista para Visualización de Rutas

Para la visualización de rutas se ocupan los siguientes datos:

Field name	Type
route_id	STRING
adjusted_timestamp	TIMESTAMP
location_coordinates	STRING
start_weight	STRING
end_weight	STRING
license_plate	STRING
start_time	TIMESTAMP
end_time	TIMESTAMP
route_duration_minutes	FLOAT
report_description	STRING
report_reason	STRING
conductor_name	STRING

Ilustración 9. Atributos Vista Rutas

El propósito de esto es tener todos los datos disponibles para su muestra de forma fácil. Los datos más importantes serían route_id, que es el identificador único de cada ruta; user_id, que es el identificador del usuario asociado con la ruta; y adjusted_timestamp, que es el timestamp ajustado con una corrección de 3 horas para la hora registrada del evento. Además, se incluye latitude y longitude, que representan las coordenadas geográficas de la ubicación de cada punto de la ruta. También se tienen los valores de start_weight y end_weight, que corresponden al peso al inicio y al final de la ruta, respectivamente. start_time y end_time indican la hora de inicio y fin de la ruta, mientras que route_duration_minutes muestra la duración total de la ruta en minutos. Los campos report_description y report_reason permiten agregar detalles sobre los informes relacionados con la ruta, y conductor_name muestra el nombre completo del conductor. Finalmente, license_plate es el número de placa del camión utilizado. Estos 10 valores son fundamentales para la creación de gráficos e informes que permitan analizar y optimizar las operaciones de recolección de residuos.

En la consulta 8 se aprecia cómo se extrae las ubicaciones de los camiones de basura, incluyendo la ruta, el usuario (conductor), la hora y las coordenadas geográficas de routes la información sobre las rutas de los camiones, como el peso inicial y final, la hora de inicio y fin y el camión asignado de reports los reportes o incidentes relacionados con las rutas, incluyendo la ubicación, la descripción y la razón del reporte de trucks la información sobre los camiones de basura, como la placa y el ID del camión y finalmente de users la información de los usuarios, en este caso, los conductores de los camiones, con su nombre e ID.



```
1 WITH Locations AS (  
2 SELECT  
3   JSON_EXTRACT_SCALAR(path_params, '$.routeid') AS route_id,  
4   JSON_EXTRACT_SCALAR(path_params, '$.userid') AS user_id,  
5   TIMESTAMP_SUB(TIMESTAMP, INTERVAL 3 HOUR) AS adjusted_timestamp,  
6   JSON_EXTRACT_SCALAR(data, '$.location_latitude') AS latitude,  
7   JSON_EXTRACT_SCALAR(data, '$.location_longitude') AS longitude  
8 FROM  
9   "smart-waste-2df41.firebaseio.com/locations_raw_latest"  
10 ),  
11  
12 routes AS (  
13 SELECT  
14   document_id AS route_id,  
15   JSON_EXTRACT_SCALAR(data, '$.start_weight') AS start_weight,  
16   JSON_EXTRACT_SCALAR(data, '$.end_weight') AS end_weight,  
17   JSON_EXTRACT_SCALAR(data, '$.start_time_seconds') AS start_time_seconds,  
18   JSON_EXTRACT_SCALAR(data, '$.end_time_seconds') AS end_time_seconds,  
19   JSON_VALUE(data, '$.truck_id') AS truck_id,  
20   JSON_VALUE(data, '$.active') AS is_active  
21 FROM  
22   "smart-waste-2df41.firebaseio.com/routes_raw_latest"  
23 WHERE  
24   JSON_VALUE(data, '$.active') = 'false'  
25 ),  
26  
27 reports AS (  
28 SELECT  
29   JSON_EXTRACT_SCALAR(path_params, '$.routeid') AS route_id,  
30   JSON_EXTRACT_SCALAR(data, '$.location_latitude') AS report_latitude,  
31   JSON_EXTRACT_SCALAR(data, '$.location_longitude') AS report_longitude,  
32   JSON_EXTRACT_SCALAR(data, '$.description') AS report_description,  
33   JSON_EXTRACT_SCALAR(data, '$.reason') AS report_reason  
34 FROM  
35   "smart-waste-2df41.firebaseio.com/reports_raw_latest"  
36 ),  
37  
38 trucks AS (  
39 SELECT  
40   document_id AS truck_id,  
41   JSON_VALUE(data, '$.license_plate') AS license_plate  
42 FROM  
43   "smart-waste-2df41.firebaseio.com/trucks_raw_latest"  
44 ),  
45  
46 users AS (  
47 SELECT  
48   document_id AS user_id,  
49   CONCAT(JSON_VALUE(data, '$.name'), ' ', JSON_VALUE(data, '$.last_name')) AS conductor_name  
50 FROM  
51   "smart-waste-2df41.firebaseio.com/users_raw_latest"  
52 );
```

Consulta 8. Query General Visualización

Mientras que en la Consulta 9 se unen las distintas tablas por medio del join, fuera de los datos estándar se tienen los valores calculados; como, por ejemplo; la duración de la ruta, donde se calcula en minutos utilizando los campos `start_time_seconds` y `end_time_seconds` de la tabla `routes`. Se utiliza la función `TIMESTAMP_DIFF` para calcular la diferencia en segundos entre la hora de inicio y fin de la ruta, y luego se divide el resultado por 60 para obtener la duración en minutos para después tener un ajuste a la hora local restando 3 horas; el ajuste de 3 horas se hace ya que, hasta el momento, la firebase guarda esos valores con esta diferencia.

El otro cálculo realizado es el de asignación de reportes; se utiliza la función `ABS` para calcular la diferencia absoluta entre latitud y longitud de la ubicación del camión y latitud y longitud del reporte. Solo se asignan los reportes que estén a una distancia máxima de 0.0003 grados de latitud y longitud de la ubicación del camión; de esta forma, se le asigna el reporte al punto más cercano, evitando así la duplicación.

```
54 closest_report AS (  
55 WHERE  
56   l.route_id,  
57   l.adjusted_timestamp,  
58   CONCAT(l.latitude, ' ', l.longitude) AS location_coordinates,  
59   r.start_weight,  
60   r.end_weight,  
61   t.license_plate,  
62   TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.start_time_seconds AS INT64)), INTERVAL 3 HOUR) AS start_time,  
63   TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.end_time_seconds AS INT64)), INTERVAL 3 HOUR) AS end_time,  
64   TIMESTAMP_DIFF(  
65     TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.end_time_seconds AS INT64)), INTERVAL 3 HOUR),  
66     TIMESTAMP_SUB(TIMESTAMP_SECONDS(CAST(r.start_time_seconds AS INT64)), INTERVAL 3 HOUR),  
67     SECOND  
68   ) / 60 AS route_duration_minutes,  
69   rr.report_description,  
70   rr.report_reason,  
71   u.conductor_name  
72 FROM  
73   locations l  
74 JOIN  
75   routes r ON l.route_id = r.route_id  
76 LEFT JOIN reports rr ON l.route_id = rr.route_id  
77 AND ABS(CAST(l.latitude AS FLOAT64) - CAST(rr.report_latitude AS FLOAT64)) < 0.0003  
78 AND ABS(CAST(l.longitude AS FLOAT64) - CAST(rr.report_longitude AS FLOAT64)) < 0.0003  
79 LEFT JOIN  
80   trucks t ON r.truck_id = t.truck_id  
81 LEFT JOIN  
82   users u ON l.user_id = u.user_id  
83 )  
84  
85 SELECT  
86 *  
87 FROM  
88 closest_report;
```

Consulta 9. Query General Visualización



Carga

Con las vistas creadas en BigQuery, ahora se conectan los datos para las visualizaciones en Looker Studio, usando las mismas herramientas proporcionadas. En la ilustración número 10, se pueden apreciar las vistas creadas anteriormente, siendo `locations_view` la necesaria para la visualización de las rutas y `routes_view` para la visualización de métrica. Adicionalmente, destaca `locations_view_clean`, la cuál es una vista de `locations` sin todo lo extra, que se usó para calcular la diferencia en tiempo de carga de las dos vistas.

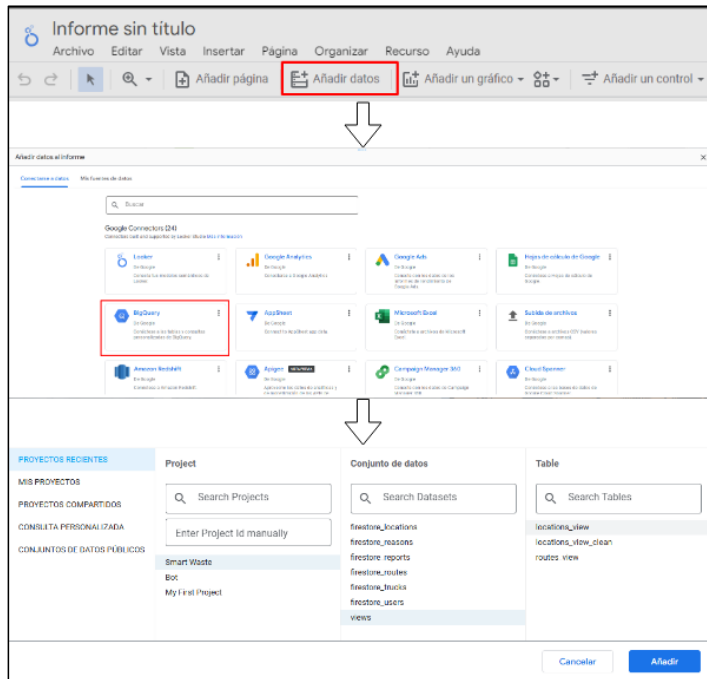


Ilustración 10. Ejemplo de Carga de Datos

4.2.2 Creación de las Visualizaciones

Visualizaciones Métricas

Para la visualización de métricas, se decidió usar los siguientes atributos, los cuales vienen todos desde `routes_view`. De igual forma, dentro de las mismas herramientas de Looker Studio se crearon atributos calculados. Algunos atributos salen de la convención `snake_case`, ya que estos serán visualizados directamente.



DIMENSIONES (14)					
Diferencia Peso	ft	:	123	Número	Ninguna
driver		:	RBC	Texto	Ninguna
end_location		:	RBC	Texto	Ninguna
end_time		:		Fecha y hora	Ninguna
end_weight		:	123	Número	Ninguna
Horas	ft	:	123	Número	Ninguna
Kilometros		:	123	Número	Total
license_plate		:	RBC	Texto	Ninguna
Mes	ft	:		Mes (MM)	Ninguna
Minutos		:	123	Número	Ninguna
routeid		:	RBC	Texto	Ninguna
start_location		:	RBC	Texto	Ninguna
start_time		:		Fecha y hora	Ninguna
start_weight		:	123	Número	Ninguna

Los campos calculados corresponden:

Nombre del campo		ID de campo
<input type="text" value="por ejemplo, Nuevo campo calculado - diferencia_peso"/>		<input type="text" value="ID de campo - calc_ygo69t5vmd"/>
Fórmula (?)		DAR FORMATO A FÓRMULA
1	<input type="text" value="end_weight - start_weight"/>	

Ilustración 12. Cálculo diferencia_peso

Nombre del campo		ID de campo
<input type="text" value="por ejemplo, Nuevo campo calculado - horas"/>		<input type="text" value="ID de campo - calc_u0yrshfpm"/>
Fórmula (?)		DAR FORMATO A FÓRMULA
1	<input type="text" value="minutos /60"/>	
2		

Ilustración 13. Formula Horas

Nombre del campo		ID de campo
<input type="text" value="por ejemplo, Nuevo campo calculado - mes"/>		<input type="text" value="ID de campo - calc_pwuf95gpmd"/>
Fórmula (?)		DAR FORMATO A FÓRMULA
1	<input type="text" value="MONTH(start_time)"/>	
2		
3		

Ilustración 14. Cálculo Mes



Todos estos valores corresponden a cálculos sencillos que al cargar las visualizaciones se vieron necesarios y no se habían transformado directamente en BigQuery, donde se consideró la opción de calcular diferencia_peso durante la fase de ETL, pero finalmente se decidió dejarse para después en caso de necesitar estos valores.



Ilustración 14a, 14b y 14c. Gráficos Creados

Los gráficos diseñados se implementaron como gráficos de barras simples. Esta elección responde a la necesidad de garantizar una representación visual clara y comprensible de los datos, priorizando la facilidad de uso por parte de los operadores. Asimismo, su simplicidad permite un análisis rápido y efectivo de las métricas clave, asegurando que los usuarios puedan extraer información relevante de manera eficiente para la toma de decisiones operativas.

En la Ilustración 16, se aprecian las dimensiones usadas en un gráfico. Todos los gráficos siguen la misma lógica; kilómetros usa como métrica kilómetros y dimensión vehículo/conductor, minutos usa como métrica minutos y dimensión vehículo/conductor, mientras que kilogramos usa como métrica kilogramos y dimensión vehículo/conductor.

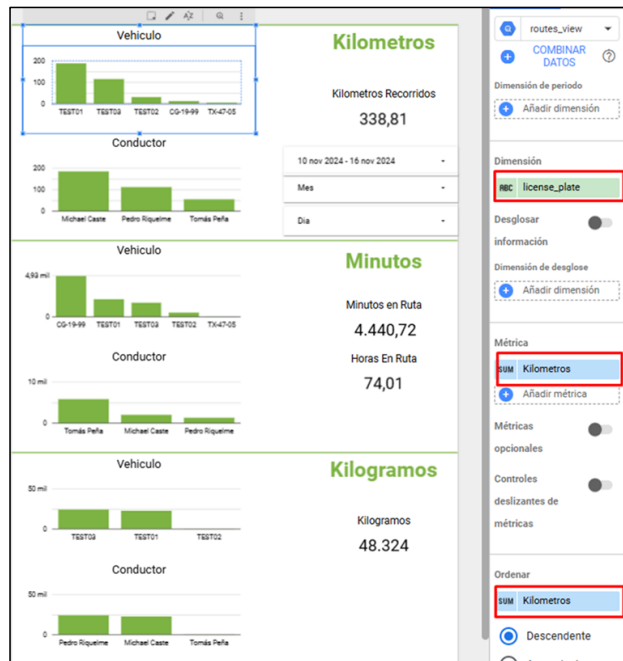


Ilustración 15. Gráficos Creados



Visualizaciones Rutas

A continuación, se presentan los datos obtenidos desde `locations_view`:

DIMENSIONES (18)				
conductor	...	RBC	Texto	▼ Ninguna
description	...	RBC	Texto	▼ Ninguna
Duración Ruta	...	🕒	Minuto	▼ Ninguna
end_location	...	RBC	Texto	▼ Ninguna
end_time	...	🕒	Fecha y hora	▼ Ninguna
Informes	fx	RBC	Texto	▼ Ninguna
license_plate	...	RBC	Texto	▼ Ninguna
location	...	📍	Latitud, longitud	▼ Ninguna
Mes	fx	📅	Mes (MM)	▼ Ninguna
Peso Final	...	123	Número	▼ Ninguna ▼
Peso Inicial	...	123	Número	▼ Ninguna ▼
punto hora	fx	RBC	Texto	▼ Ninguna
reason	...	RBC	Texto	▼ Ninguna
routeid	...	RBC	Texto	▼ Ninguna
Rutas	fx	RBC	Texto	▼ Ninguna
start_location	...	RBC	Texto	▼ Ninguna
start_time	...	🕒	Fecha y hora	▼ Ninguna
timestamp	...	🕒	Fecha y hora	▼ Ninguna

Ilustración 17. Dimensiones Usadas

Fórmula ? DAR FORMATO A FÓRMULA

```
1 CONCAT("Razón: ", reason, " Descripción: ", description )
```

Ilustración 18. Fórmula Informes

Fórmula ? DAR FORMATO A FÓRMULA

```
1 MONTH( start_time )
```

Ilustración 19. Fórmula Mes

Fórmula ? DAR FORMATO A FÓRMULA

```
1 FORMAT_DATETIME("%H:%M", timestamp )
```

Ilustración 20. Fórmula Punto Ruta

Fórmula ?

```
1 CONCAT("Ruta del ", FORMAT_DATETIME("%d-%m-%Y", start_time), " a las ", FORMAT_DATETIME("%H:%M:%S", start_time), " del camión ", license_plate, " del conductor ", conductor )
```

Ilustración 21. Fórmula "Ruta"



Dentro de las dimensiones usadas para la visualización de las rutas, destacan los siguientes; en el campo “informes” (Ilustración 18), el cuál es la unión de la razón y descripción de cada informe; “mes” (Ilustración 19) que es para filtrar, de igual forma que la visualización anterior, solo por mes; “punto hora” (Ilustración 20), la cuál es la hora de cada punto en formato hh:ss para una mejor visualización en el mapa; y finalmente; “rutas” (Ilustración 21), la cuál es una concatenación de todos los datos importantes para una mejor entendimiento de las rutas.

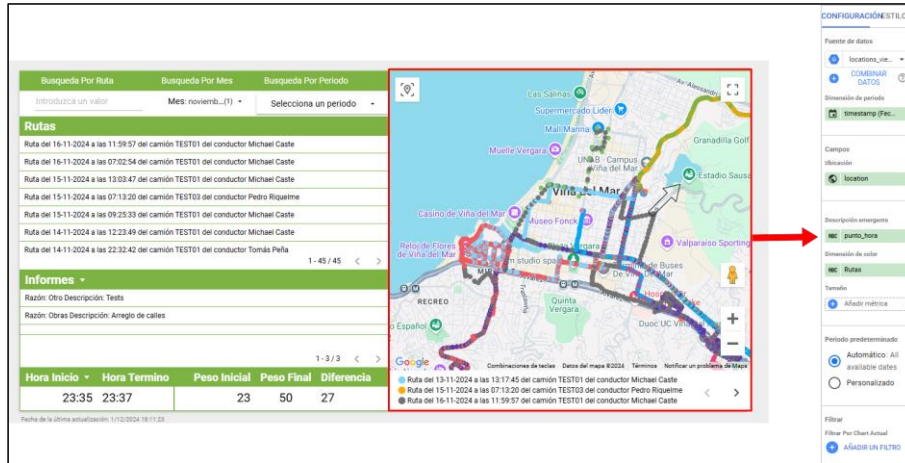


Ilustración 22. Datos Usados Mapa

En la ilustración 22, se aprecian los datos para crear el mapa donde se visualizan las rutas, el cual ocupa la API de Google Maps para mostrar puntos.

En el campo de ubicación se coloca el atributo “location”, los cuales son los puntos obtenidos con nuestra aplicación, mientras que, en descripción emergente, se ocupa el atributo “punto_hora”, la cual tiene como función que, al pasar el cursor por encima, salga la hora del punto.

Para que exista la diferenciación por color de las rutas mostradas, se usa el atributo “rutas”, el cual es el mismo atributo que se usa en el panel para la presentación y la selección de éstas.

4.2.3 Carga de las Visualizaciones en la página

Estas visualizaciones, como lo dicho anteriormente, se insertan directamente en la sección de diseño de la página de inicio de nuestra aplicación, usando la herramienta de inserción proporcionada por Looker Studio.



Ilustración 23. Herramienta Inserción



Este código se copia y se inserta en nuestra página web en la sección de home, como se ve a continuación:

```
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Visualizaciones de Looker Studio</title>
7
8 <style>
9   .iframe-container {
10     margin-bottom: 20px;
11   }
12 </style>
13
14 <body>
15   <div class="iframe-container">
16     <iframe width="1200" height="400" src="https://lookerstudio.google.com/embed/reporting/d1117be7-8bbe-4117-a154-8776b0ec546f/pago/1uHrE" frameborder="0" style="width:100%; height:100%; border:none;" allow="autoplay, fullscreen, geolocation, microphone, mouse, scroll, webcam, videoPlayback" />
17   </div>
18   <div class="iframe-container">
19     <iframe width="1200" height="600" src="https://lookerstudio.google.com/embed/reporting/b4128e9f-82d9-463e-b30c-efe84ba84c1c/pago/NEcFE" frameborder="0" style="width:100%; height:100%; border:none;" allow="autoplay, fullscreen, geolocation, microphone, mouse, scroll, webcam, videoPlayback" />
20   </div>
21 </body>
22 </html>
23
```

Ilustración 25. Inserción en la página WEB

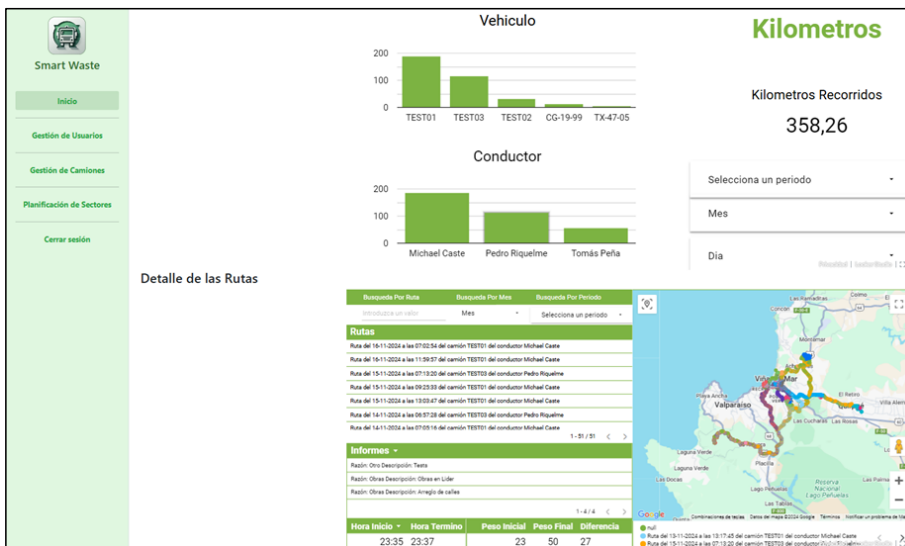


Ilustración 24. Visión de Visualización en Página Web



4.2.4 Uso de las Visualizaciones

Las visualizaciones hechas tienen muchas formas de filtrados, algunos ejemplos de cómo se pueden usar son los siguientes:

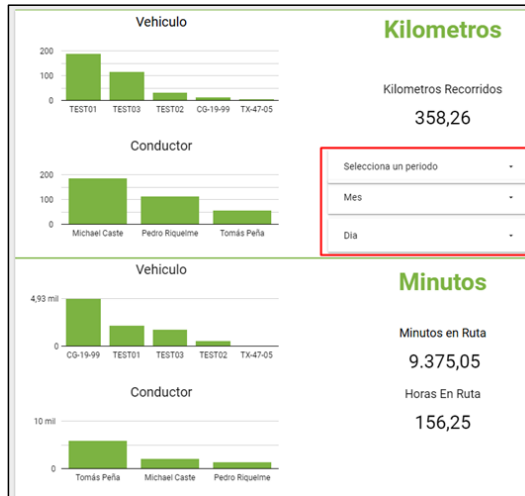


Ilustración 26. Selección por Espacio Temporal

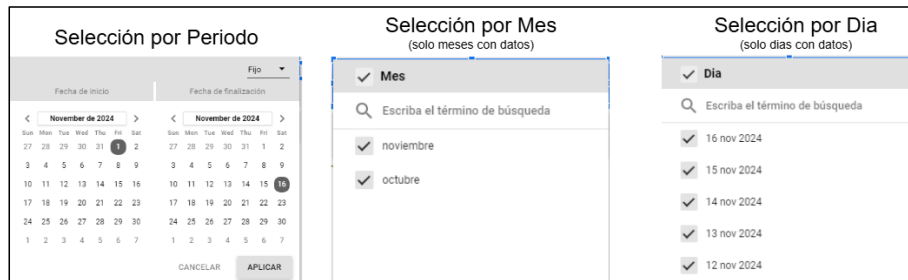


Ilustración 27. Opciones de Filtrado



Ilustración 28. Ejemplo de Filtrado 15 nov

Lo anterior corresponde a las visualizaciones de métricas, mientras que para las visualizaciones de ruta, además del filtrado por mes, periodo y día como se ve en la ilustración 27, se permite un filtrado mas exhaustivo. Un ejemplo de esto sería lo siguiente:

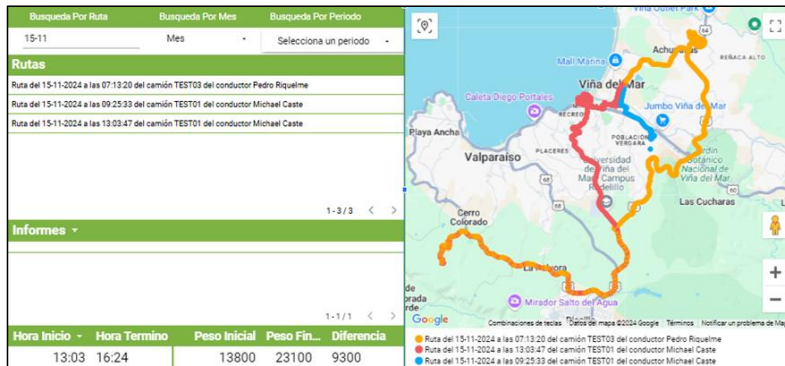


Ilustración 29. Rutas del día 15 nov

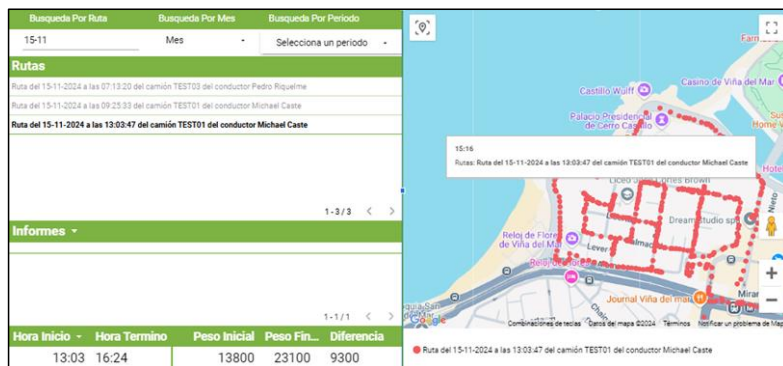


Ilustración 30. Detalle de ruta 15 nov Conductor Michael Caste



Algunas de las cosas que se destacan de la ilustración 30 es que al seleccionar una ruta aparecen datos como la hora de inicio y termino, o los kilogramos de basura recogido de esa ruta, de igual forma al acercarse al mapa se pueden observar los puntos precisos por donde pasó, al igual que poniendo el ratón encima de cualquier punto, la hora por la que pasó por ese punto, como también la ruta correspondiente.

Mientras que el comportamiento de Informes se puede apreciar en la ilustración 31, al igual que con las rutas uno puede seleccionar un informe que aparecerá en el mapa como el punto donde ocurrió, al igual que su información adjunta (Hora y ruta adjunta).

Búsqueda Por Ruta	Búsqueda Por Mes	Búsqueda Por Periodo
Introduzca un valor	Mes	Selecciona un periodo

Rutas

Ruta del 17-10-2024 a las 20:39:49 del camión TX-47-05 del conductor Tomás Peña

Informes

Razón: Obras Descripción: Obras en Líder

Hora Inicio	Hora Termino	Peso Inicial	Peso Fin...	Diferencia
20:39	20:46	0	200	200

Ilustración 29. Ejemplo Visualización Informe

3. Validación

4.3.1 Validación de Ubicaciones e Informes

En esta prueba se comprobó que las ubicaciones mostraran bien las rutas y que los puntos de ubicación registrados formen una trayectoria lógica sin saltos bruscos o inconsistencias. Finalmente, esto resultó en una prueba donde se comparó la ruta generada con la ruta real ejecutada por medio de la aplicación de Google Maps, resultando en la recolección completa de la información. De igual forma, se hizo una prueba de los informes, tal como se puede apreciar en la ilustración 34. Esta prueba resultó en la aprobación de las dos funciones principales del dashboard.



Búsqueda Por Ruta	Búsqueda Por Mes	Búsqueda Por Periodo
Introduzca un valor	Mes	Selecciona un periodo

Rutas

Ruta del 21-10-2024 a las 11:45:26 del camión CG-19-99 del conductor Tomás Peña

Ruta del 26-10-2024 a las 19:59:01 del camión CG-19-99 del conductor Tomás Peña

Ruta del 21-10-2024 a las 11:45:26 del camión CG-19-99 del conductor Tomás Peña

Ruta del 17-10-2024 a las 20:39:49 del camión TX-47-05 del conductor Tomás Peña

1-52 / 52 < >

Informes

Razón: Otro Descripción: Villa en el camino

1-2 / 2 < >

Hora Inicio	Hora Termino	Peso Inicial	Peso Final	Diferencia
11:45	15:56	128	569	441

Ilustración 30. Ruta Guardada por APP

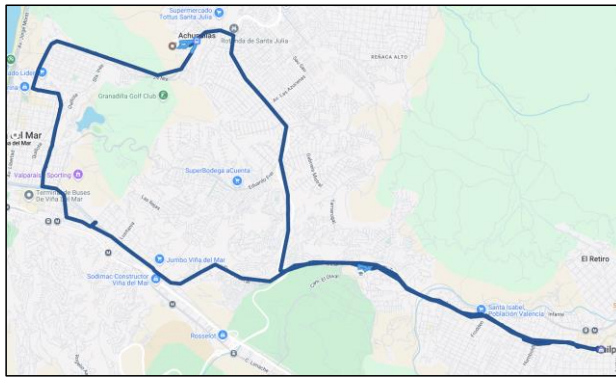


Ilustración 31. Ruta Guardada por Google Maps

Búsqueda Por Ruta	Búsqueda Por Mes	Búsqueda Por Periodo
Introduzca un valor	Mes	Selecciona un periodo

Rutas

Ruta del 21-10-2024 a las 11:45:26 del camión CG-19-99 del conductor Tomás Peña

1-1 / 1 < >

Informes

Razón: Otro Descripción: Villa en el camino

1-2 / 2 < >

Hora Inicio	Hora Termino	Peso Inicial	Peso Final	Diferencia
11:45	15:56	128	569	441

Ilustración 32. Puntos Cercanos Informe



4.3.2 Validación Métricas

En esta prueba se probó la validez de las métricas calculadas. Mientras que todas las métricas responden bien, ya que son datos completamente ingresados y validados (hora de inicio, hora término, basura recolectada, minutos en ruta...), hay un campo que es completamente calculado, el cual, recordando, se calcula por medio de BigQuery, que corresponde al de la consulta 5. En esta prueba se hará una validación de esto.

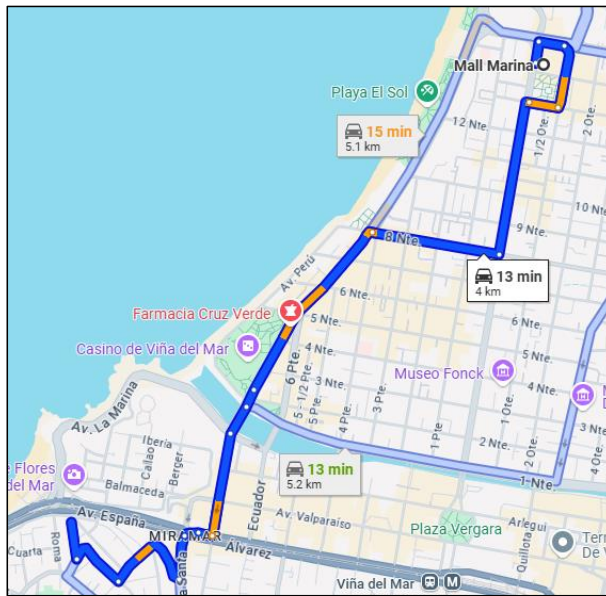


Ilustración 33. Distancia Google Maps

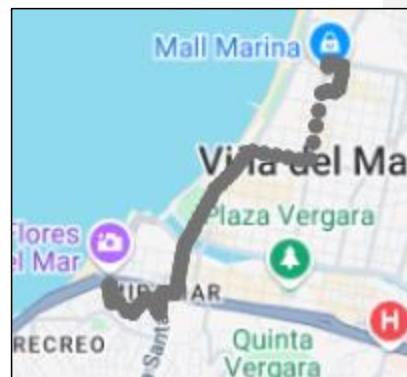
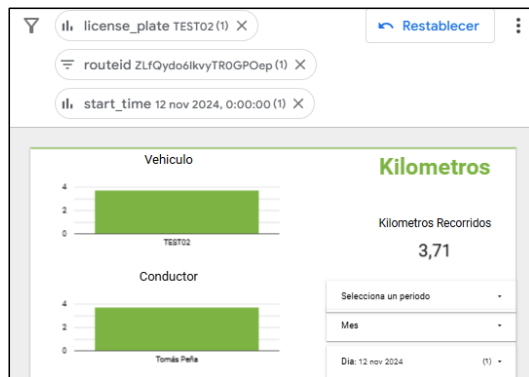


Ilustración 34a y 34b. Distancia Aplicación



Como se había nombrado anteriormente, nuestra aplicación logra recopilar bien las distancias entre rutas relativamente rectas, como se puede apreciar en la ilustración 34, donde Google Maps marca 4km mientras que nuestra aplicación ilustración 34 marca 3,7km. Mientras que podemos comparar esta ruta de ejemplo, las rutas que tienen muchas intersecciones o muchos puntos entre calles muy cercanas no podemos, por lo que para comprobar la veracidad de la información recolectada se necesitaría realizar una ruta como la de la ilustración 35 mientras se toma la distancia con una app externa, en este caso se investigó la utilización de una app llamada **Mi Ruta (My Track)**^[9] para este tipo de pruebas como también la comprensión de cómo funciona la mencionada aplicación para su implementación de cálculo dentro de nuestra solución.

4.3.3 Validación con Operadores

Se realizó una prueba del sistema con uno de los operadores clave del proyecto durante el día 3 de octubre (Rodrigo Pasten, jefe del Departamento de Aseo de la Municipalidad de Viña del Mar), con el objetivo de validar la funcionalidad, la usabilidad y la capacidad del panel de control para satisfacer las necesidades operativas identificadas. Durante la sesión, el operador interactuó con las diversas herramientas del sistema, incluyendo la consulta de rutas, el análisis de métricas clave y la visualización de trayectorias en el mapa interactivo. A lo largo de la prueba, el operador encontró que el sistema cumplía plenamente con sus expectativas, destacando la claridad y precisión de los datos presentados, así como la facilidad de uso de las funcionalidades disponibles.

El panel de control fue descrito como intuitivo y eficiente, facilitando el acceso rápido a información relevante para la gestión diaria de las rutas de recolección. Además, el operador expresó su satisfacción con la capacidad del sistema para filtrar datos por conductor, camión y período de tiempo, lo que le permitió realizar análisis detallados y personalizados sin complicaciones.



5 Conclusión

El desarrollo de Smart Waste ha demostrado ser una solución efectiva para enfrentar las problemáticas relacionadas con la gestión de rutas de recolección de residuos sólidos en el contexto municipal. Las herramientas de inteligencia de negocios integradas en el sistema han permitido centralizar y visualizar métricas clave, facilitando la toma de decisiones informadas por los gestores y operadores.

Desde el punto de vista metodológico, la elección de una arquitectura basada en Firebase y BigQuery demostró ser adecuada para garantizar la integración de datos en tiempo real y una visualización efectiva mediante Looker Studio. Sin embargo, se identificaron limitaciones específicas en el cálculo de distancias utilizando Google Maps API, particularmente en zonas urbanas densas con calles estrechas o intersecciones frecuentes, donde la precisión puede disminuir.

En cuanto a las capacidades predictivas futuras, Smart Waste tiene el potencial de evolucionar mediante la integración de modelos avanzados como análisis de series temporales, modelos de regresión o aprendizaje automático. Estas herramientas podrían considerar variables como datos históricos de generación de residuos, eventos locales (festividades, ferias), condiciones climáticas y patrones de movilidad urbana. Esto permitiría anticipar incrementos en la generación de desechos, optimizando la asignación de recursos y ajustando rutas con mayor precisión.

A lo largo del desarrollo del proyecto, se aprendió la importancia de involucrar a los operadores clave desde el inicio, lo que no solo facilitó la validación del sistema, sino que también ayudó a alinear la solución con las necesidades reales del servicio. Esto se reflejó en una adopción más fluida y en la obtención de retroalimentación valiosa para futuras iteraciones.

En última instancia, Smart Waste no solo optimiza la gestión de residuos sólidos, sino que también establece un modelo replicable y escalable para otras municipalidades y contextos urbanos. Este proyecto abre nuevas oportunidades para integrar tecnologías avanzadas en la gestión de servicios municipales, impulsando un cambio significativo hacia la sostenibilidad y la eficiencia operativa en ciudades de todo el mundo.



6 Referencias

1. Romualdo, M. G. A., Elena, R. P. M., & Alejandra, M. J. V. (s. f.). Percepción ciudadana del manejo de residuos sólidos municipales: El caso Riviera Nayarit. https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-39252013000300004
2. How Smart City Barcelona Brought the Internet of Things to Life. (s. f.). Data-Smart City Solutions. <https://datasmart.hks.harvard.edu/news/article/how-smart-city-barcelona-brought-the-internet-of-things-to-life-789>
3. Tambe, R. (2023, 31 octubre). Smart Waste Management: IoT Solutions for Sustainable Cities. Smart Sight Innovations. <https://www.smartsight.in/industry-insights/smart-waste-management-iot-solutions-for-sustainable-cities/>
4. Rerouting New Yorks waste collection network toward optimization. (s. f.). <https://www.arcadis.com/en-us/projects/north-america/united-states/rerouting-new-yorks-waste-collection-network-toward-optimization>
5. Gautam, S. (2020, 2 noviembre). How Singapore is Excavating a New Road to A Smart Future. Get My Parking. <https://blog.getmyparking.com/2020/10/14/how-singapore-is-excavating-a-new-road-to-a-smart-future/> 4
6. Zhang, J., Chen, F., Cui, Z., Guo, Y., & Zhu, Y. (2021). Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit. IEEE Transactions on Intelligent Transportation Systems, 22(11), 7004-7014. <https://doi.org/10.1109/tits.2020.3000761>
7. Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit. (2021, 1 noviembre). IEEE Journals & Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/document/9136910>
8. Stream Firestore to BigQuery | Firebase Extensions Hub. (s. f.). <https://extensions.dev/extensions/firebase/firestore-bigquery-export>
9. Mi Ruta - Aplicaciones en Google Play. (s. f.). <https://play.google.com/store/apps/details?id=com.zihua.android.mytracks&hl=es>