udp Escuela de Informática y Telecomunicaciones
FACULTAD DE INGENIERÍA

Facultad de Ingeniería
Escuela de Informática y Telecomunicaciones

Tarea 2 - Sistema Distribuidos

Gonzalo Gaete Faúndez
gonzalo.gaete1@mail.udp.cl
Alan Toro Quilaqueo
alan.toro@mail.udp.cl

Índice

1. Introducción **2**

2. Objetivos **2**

3. Limpieza y estandarización de datos **2**

4. Processing **2**

5. Análisis **3**
5.1. Incidentes por comuna 3
5.2. Incidentes por tipo 3
5.3. Incidentes por tipo y comuna 3
5.4. Promedio de rating por comuna 3
5.5. Ranking de comunas 3
5.6. Incidentes por comuna y tipo 3
5.7. Conclusión 4

6. Repositorio **4**

1 | Introducción

Esta entrega se centra en extender un pipeline de procesamiento de datos de tránsito a partir de eventos recolectados desde Waze. El objetivo fue integrar una etapa de limpieza y estandarización de los datos, para luego analizarlos mediante Apache Pig sobre Hadoop. Finalmente, se generaron salidas estructuradas que permiten comprender patrones de incidentes y preparar la base para su visualización futura.

2 | Objetivos

- Agregar a un pipeline ya existente una capa de filtro y estandarización de datos.
- Analizar datos previamente filtrados para su análisis usando Apache Pig y Hadoop.

3 | Limpieza y estandarización de datos

Como se implementan nuevos servicios a un pipeline ya existente, se asume una ejecución correcta de los servicios Scraper y Mongo con datos ya adquiridos. Desde acá se implementa, usando Pandas, distintas transformaciones a los datos tales como registros duplicados y normalización de columnas.

En una primera instancia, también se eliminaron filas con columnas N/A, es decir, con datos faltantes. Pero, en favor de tener un mayor número de entradas se mantienen en el conjunto de datos filas con algunos datos faltantes.

Finalmente, los datos filtrados se persisten en un archivo CSV en la carpeta data/clean.

4 | Processing

Con los datos ya limpios y estandarizados disponibles en `data/clean/incidents_clean.csv`, se realizó un procesamiento distribuido utilizando Apache Pig sobre Hadoop. Este procesamiento tuvo como objetivo aplicar operaciones de análisis exploratorio para generar archivos de salida estructurados que permitan describir los patrones presentes en los incidentes reportados.

- **Conteo por tipo:** los datos fueron agrupados por el campo `type` y se contó la cantidad de incidentes correspondientes a cada tipo. Este resultado se almacenó en `incidents_by_type.csv`.
- **Conteo por comuna:** se agrupó por `comuna` y se contabilizó la cantidad total de incidentes reportados en cada una, generando el archivo `incidents_by_comuna.csv`.
- **Conteo por tipo y comuna:** se realizó un cruce entre tipo de incidente y comuna, obteniendo la cantidad de ocurrencias para cada combinación. Esto permite observar la distribución de tipos de incidentes por zona geográfica. El resultado fue almacenado en `incidents_by_type_and_comuna.csv`.
- **Ranking de comunas:** se ordenó la tabla anterior por cantidad de incidentes en orden descendente, para identificar las comunas con mayor número de reportes. El resultado fue almacenado en `top_comunas.csv`.
- **Promedio de rating por comuna:** se agrupó por comuna y se calculó el promedio del campo `report_rating`, entregando una métrica que refleja la confianza o validación general de los reportes por zona. El resultado se almacenó en `promedio_rating_por_comuna.csv`.
- **Detalle por tipo dentro de cada comuna:** se generó el archivo `incidentes_por_comuna_tipo.csv`, el cual entrega el número de incidentes por cada tipo en cada comuna, facilitando análisis específicos por categoría.

Todas las salidas generadas por Pig fueron almacenadas en la carpeta `data/output/` del repositorio.

5 | Análisis

A continuación, se presentan los principales hallazgos obtenidos a partir de los archivos generados por el procesamiento distribuido. Estos datos permiten identificar patrones y tendencias útiles para la toma de decisiones en materia de gestión del tránsito.

5.1. Incidentes por comuna

El archivo `incidents_by_comuna.csv` muestra la cantidad total de incidentes reportados en cada comuna. Las comunas con mayor concentración de incidentes son:

- **Quilicura**: registra la mayor cantidad de incidentes, lo cual puede asociarse a una alta densidad vehicular o problemas persistentes en su infraestructura vial.
- **Renca** y **Pudahuel**: también presentan cifras elevadas, lo que sugiere zonas de congestión o tránsito complejo.

5.2. Incidentes por tipo

Según `incidents_by_type.csv`, los incidentes se agrupan principalmente en:

- **JAM**: es el tipo más reportado, lo que indica que los atascos representan el problema vial más frecuente.
- **HAZARD** y **ACCIDENT**: presentan también una proporción significativa, lo que señala la importancia de mejorar la seguridad vial.

5.3. Incidentes por tipo y comuna

El archivo `incidents_by_type_and_comuna.csv` permite observar cómo se distribuyen los tipos de incidentes en cada comuna. Por ejemplo:

- **Quilicura**: destaca con una alta cantidad de incidentes tipo `JAM`.
- **Renca**: presenta tanto `JAM` como un número relevante de `HAZARD`, lo cual podría indicar presencia de obstáculos o condiciones peligrosas frecuentes.

5.4. Promedio de rating por comuna

En `promedio_rating_por_comuna.csv`, se calcula el promedio del campo `report_rating` por comuna. Este valor puede interpretarse como una medida de la severidad percibida o la participación ciudadana en los reportes. Comunas con un promedio más alto pueden estar asociadas a incidentes más graves o mejor evaluados por los usuarios.

5.5. Ranking de comunas

El archivo `top_comunas.csv` entrega una lista ordenada de comunas en función del número total de incidentes. Esta información es fundamental para priorizar intervenciones en zonas críticas de la ciudad.

5.6. Incidentes por comuna y tipo

Por último, `incidentes_por_comuna_tipo.csv` entrega una visión combinada que facilita identificar patrones específicos, como comunas con alta concentración de ciertos tipos de incidentes.

5.7. Conclusión

La segunda entrega del proyecto permitió extender un pipeline de procesamiento distribuido, integrando una etapa de limpieza y estandarización de datos provenientes de MongoDB, como también una etapa de procesamiento mediante Apache Pig sobre Hadoop.

El conjunto de transformaciones realizadas permitió extraer información relevante y ordenada, que servirá como base para la etapa de visualización del sistema y para apoyar la toma de decisiones en temas de movilidad urbana.

6 | Repositorio

Repositorio del proyecto:

`https://github.com/wzrdd/SistDist-Tareas`