udp

Proyecto Sistemas Distribuidos: Manejo de la Información del Análisis de Tráfico en Región Metropolitana

Profesor: Nicolás Hidalgo
Ayudantes: Sofía Belmar, César Muñoz Rivera, J. Tomás Silva, Joaquín Villegas y Marcelo Yáñez

Introducción y Contexto

En la fase final del proyecto, nos centraremos en la presentación de la información procesada de manera gráfica y accesible. Para ello, se utilizará una plataforma de visualización de datos que permitirá a los usuarios interactuar con los resultados y crear gráficos personalizados para métricas clave. Así, se facilitará la toma de decisiones rápidas y eficientes, proporcionando una interfaz intuitiva que favorezca el análisis en tiempo real.

Con el fin de alcanzar este objetivo, se utilizarán todos los módulos previamente desarrollados, como el scraper, la base de datos, la capa de caché, y los procesos de filtrado y análisis de datos. Además, se incorporará un nuevo módulo orientado a la visualización interactiva de los datos, completando el flujo de trabajo y permitiendo la interacción directa con las métricas más relevantes.

Metodología

Este proyecto tiene un carácter semestral y se desarrollará a través de tres entregas o módulos, cada uno con un objetivo específico. Cada entrega debe ser **integrable** con las entregas anteriores y posteriores, garantizando la continuidad y cohesión del sistema. La evaluación de cada entrega se basará en un informe técnico que detalle el proceso de diseño, a excepción de la última entrega, donde se requerirá un vídeo explicativo. Además, se deberá incluir un análisis de las pruebas realizadas para validar el funcionamiento del sistema. Todas las decisiones tomadas durante el desarrollo deben estar debidamente justificadas, y el comportamiento del sistema deberá estar respaldado por datos obtenidos de la evaluación de cada módulo.
Las entregas abordan 3 hitos clave del desarrollo como lo son:

1. Datos: abocados a la recuperación y manejo de datos/eventos.
2. Procesamiento: abocados a la preparación de los datos para su posterior análisis.
3. Visualización: abocado a proveer una vista agregada de métricas relevantes para los tomadores de decisiones.

Para desarrollar y desplegar los diferentes módulos y sus tecnologías se trabajará con Docker[1] como tecnología de virtualización de recursos. No se impone un stack tecnológico específico; la elección de herramientas, lenguajes y metodologías es libre, siempre que se justifiquen las decisiones tomadas de manera apropiada.

[1] https://www.docker.com/

1. Entregable 3: Visualización

El propósito de esta entrega es optimizar la reutilización de los componentes previamente desarrollados en las entregas anteriores y ampliar el sistema mediante la implementación de un nuevo modelo de visualización. Para ello, se construirá una arquitectura basada en Elasticsearch, junto con Kibana, una herramienta de visualización que permite la interacción efectiva con los datos. Asimismo, se deben cumplir los siguientes objetivos:

- Generar métricas y visualizaciones relevantes del scraper, que proporcionen información clave sobre el desempeño y la efectividad del proceso de recolección de datos.
- Crear métricas y visualizaciones para el módulo de caché, permitiendo evaluar la eficiencia en la consulta y almacenamiento de los datos.
- Desarrollar clasificaciones visuales de los incidentes procesados en el módulo de filtrado y análisis de datos, facilitando su análisis y comprensión.
- Diseñar visualizaciones interactivas que ayuden a la Unidad de Control de Tráfico y a los municipios de la Región Metropolitana a tomar decisiones informadas y oportunas sobre la gestión del tráfico.

Arquitectura del Sistema

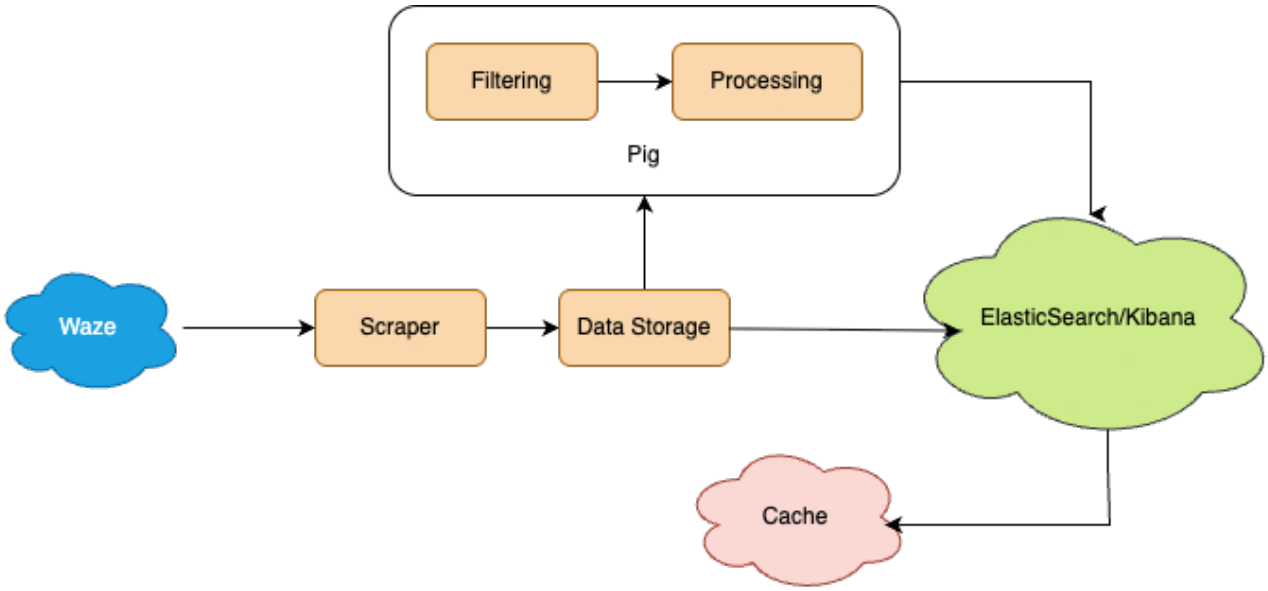

Figura 1: Diagrama de flujo

La Figura 1 muestra los componentes y la organización de estos. El objetivo de los módulos a desarrollar y sus funcionalidades se detalla a continuación:

- **Scraper:** Se deberá emplear el módulo de scraper utilizado en la entrega anterior, el cual automatiza la extracción de datos desde la plataforma Waze, específicamente a través de su mapa en tiempo real (`https://www.waze.com/es-419/live-map/`). Se deberán realizar las modificaciones indicadas en el feedback de cada entrega.
- **Almacenamiento de Datos:** Al igual que en la entrega anterior, se implementará un sistema de almacenamiento para los registros, el cual debe ser capaz de gestionar consultas rápidas y soportar actualizaciones masivas. Este sistema debe garantizar la integridad y disponibilidad de los datos extraídos, permitiendo una rápida accesibilidad para el procesamiento y análisis posterior. Se deberán realizar las modificaciones indicadas en el feedback de cada entrega.
- **Caché:** Se deberá emplear el módulo de caché utilizado en la primera entrega, implementando la política de remoción que mejores resultados haya tenido para los casos planteados. Se deberán realizar las modificaciones indicadas en el feedback de cada entrega.

- **Filtrado y Homogeneización:** Se debe contemplar el módulo de la entrega anterior, el cual se encargaba de la limpieza y homogeneización de los datos. En caso de necesitar más datos, se deberá asegurar que el pipeline esté completo desde la recolección hasta este paso. Se deberán realizar las modificaciones indicadas en el feedback de cada entrega.
- **Procesamiento:** Al igual que en el módulo anterior, se debe reutilizar el módulo de procesamiento para enviar la información al nuevo módulo de visualización. Se deberán realizar las modificaciones indicadas en el feedback de cada entrega.
- **Visualización:** Para este nuevo módulo, se deberá implementar una solución basada en Elasticsearch junto con Kibana, que permita almacenar, consultar y visualizar los datos de manera eficiente. Elasticsearch se encargará de la indexación y búsqueda rápida de grandes volúmenes de datos, mientras que Kibana proporcionará una interfaz visual interactiva que permita la creación de gráficos, paneles y reportes dinámicos. El módulo deberá permitir a los usuarios finales realizar consultas personalizadas sobre los datos procesados, visualizar tendencias en tiempo real y generar informes de manera intuitiva. Además, deberá facilitar la interacción con los datos mediante filtros, selección de rangos temporales y categorización de incidentes, ofreciendo así una herramienta efectiva para la toma de decisiones basadas en información de tráfico.

Requerimientos del Sistema

Diseño Modular

El sistema debe estructurarse bajo una arquitectura modular, permitiendo una clara separación de responsabilidades entre sus componentes. Esta organización facilitará tanto el mantenimiento del sistema como su escalabilidad y evolución. Asimismo, deberá contemplarse la capacidad de integración futura con otros sistemas o servicios externos, lo que permitirá ampliar su funcionalidad sin afectar la arquitectura existente.

Los módulos para esta entrega son:

- **Scraper:** Módulo encargado de extraer eventos adicionales desde la plataforma Waze en caso de ser necesario, con el fin de obtener una cantidad suficiente de eventos que permitan generar métricas significativas y valiosas para el análisis posterior.
- **Almacenamiento:** Módulo destinado a contener y gestionar los datos obtenidos durante el proceso de scraping. Este sistema debe garantizar la integridad y disponibilidad de los datos para su posterior procesamiento y análisis.
- **Cache:** Módulo diseñado para minimizar el tiempo de consultas y procesamiento, mejorando el rendimiento del sistema. Utiliza estrategias de almacenamiento temporal de datos frecuentemente consultados, optimizando el tiempo de respuesta.
- **Filtrado y Homogeneización:** Módulo encargado de la limpieza, unificación y estandarización de los datos (eventos) extraídos. Asegura que los datos estén en un formato consistente y adecuado para su análisis, eliminando duplicados o inconsistencias.
- **Procesador y Analizador de Datos Distribuidos:** Módulo que utiliza Apache Pig para procesar y analizar grandes volúmenes de datos de manera distribuida, permitiendo la ejecución eficiente de operaciones complejas y optimizando el rendimiento de los análisis.
- **Visualización:** Módulo basado en Elasticsearch y Kibana, que permite la visualización interactiva de los datos procesados. Elasticsearch se encarga de la indexación y búsqueda eficiente de los datos, mientras que Kibana proporciona una interfaz gráfica para crear visualizaciones dinámicas e informes interactivos.

Para ello, se deberá implementar uno o varios pipelines completos que garanticen el funcionamiento integral de todos los módulos anteriormente mencionados. Estos pipelines deben ser diseñados para asegurar la correcta integración entre cada uno de los módulos, facilitando la extracción, almacenamiento, procesamiento, análisis y visualización de los datos de manera eficiente y escalable. El pipeline deberá incluir las siguientes etapas:

- Extracción de datos mediante el módulo de **Scraper**.
- Almacenamiento de los datos obtenidos en el sistema de **Almacenamiento**.
- Implementación de la **Cache** para optimizar las consultas y el procesamiento de datos.

- Aplicación de **Filtrado y Homogeneización** para garantizar la calidad y uniformidad de los datos.
- Procesamiento y análisis de los datos utilizando el módulo de **Procesador y Analizador de Datos Distribuidos**.
- Visualización de los resultados utilizando el módulo de **Visualización** basado en Elasticsearch y Kibana.

El objetivo es construir un flujo de trabajo fluido y eficiente que permita una integración continua de los datos desde la recolección hasta su visualización, asegurando que cada módulo cumpla su función de manera autónoma pero interconectada.

Documentación y Buenas Prácticas

Documente detalladamente el código y la funcionalidad implementada, incluyendo una justificación de las decisiones de diseño, la elección de tecnologías y las metodologías utilizadas.

Requisitos de la entrega

- Vídeo explicativo, con una duración mínima de 10 minutos, y máxima de 20 minutos. Este debe proporcionar una descripción detallada de la arquitectura utilizada, incluyendo todo el flujo de datos, desde la recolección hasta la visualización. En el vídeo deben incluirse los siguientes puntos:
  - Descripción de los componentes y funcionalidades implementadas en cada uno de los módulos.
  - Justificación de las decisiones de diseño tomadas, así como las tecnologías y metodologías empleadas en el proyecto.
  - Ejecución de pruebas para demostrar la correcta integración de los módulos y el funcionamiento del sistema en su totalidad.
  - Análisis exhaustivo basado en los resultados obtenidos, con el apoyo de gráficos, tablas comparativas y cualquier otro recurso visual que facilite la interpretación y evaluación de las métricas definidas.
- El código fuente completo del proyecto, alojado en un repositorio público en GitHub o GitLab. El enlace al repositorio deberá incluirse en la sección de comentarios de la plataforma CANVAS.
- Un archivo `Dockerfile` o `docker-compose` que permita la ejecución de los servicios implementados, junto con instrucciones claras en el archivo `README.md` del repositorio.

Reglas y consideraciones de la entrega

- **Fecha de Entrega:** La fecha límite para la entrega de esta tarea es el día 30/06/2025 hasta las 23:59 hrs. Se recomienda gestionar el tiempo de forma efectiva. La entrega es vía Canvas del curso.
- **Integrantes:** La tarea debe ser realizada en grupos de hasta **2** alumnos/as.
- **Formato de Entrega:**
  - Vídeo accesible a través de URL por plataformas como Youtube o Drive.
  - El repositorio debe estar bien documentado y organizado, incluyendo instrucciones para la configuración y ejecución del proyecto.
- **Ética y Autoría:** Se debe respetar el reglamento de la universidad en cuanto a plagio y autoría. Cualquier evidencia de copia o falta de autoría conllevará sanciones según lo establecido.