Trabajo Práctico Diseño:
Movies Analysis

[75.74] Sistemas Distribuidos I

Grupo 13

Bianchi Fernandez, Marcos	108921	mbianchif@fi.uba.ar
Gentilini, Franco	108733	fgentilini@fi.uba.ar
Ghosn, Lautaro Gabriel	106998	lghosn@fi.uba.ar

Índice

Escenarios

El proyecto Movies Analysis tiene como objetivo analizar diversas variables relacionadas con un conjunto de datos de películas, utilizando un enfoque basado en sistemas distribuidos. El propósito es obtener los siguientes resultados:

CU1 - Producción Argentina y España (2000s)

Utilizando el dataset de películas el sistema obtiene los títulos y géneros de cada una de las películas estrenadas en la década de los años 2000 con una producción Argentina y Española.

CU2 - Top 5 Inversores en Producciones Nacionales

Utilizando el dataset de películas el sistema obtiene los 5 nombres de países que hayan invertido más dinero en producción tomando en cuenta solamente aquellas películas que hayan sido producidas por un único país.

CU3 - Mejor y Peor Rating – Cine Argentino 2000+

Utilizando los datasets de películas y ratings el sistema obtiene el título y rating de las películas con mayor y menor rating, siendo estas producidas en Argentina a partir del año 2000.

CU4 - Actores Más Frecuentes en Argentina desde 2000

Utilizando los datasets de películas y créditos, el sistema obtiene los nombres de los actores que hayan tenido más apariciones en películas argentinas con una fecha de estreno posterior al año 2000.

CU5 - Tasa Ingreso/Presupuesto: Positivo vs. Negativo

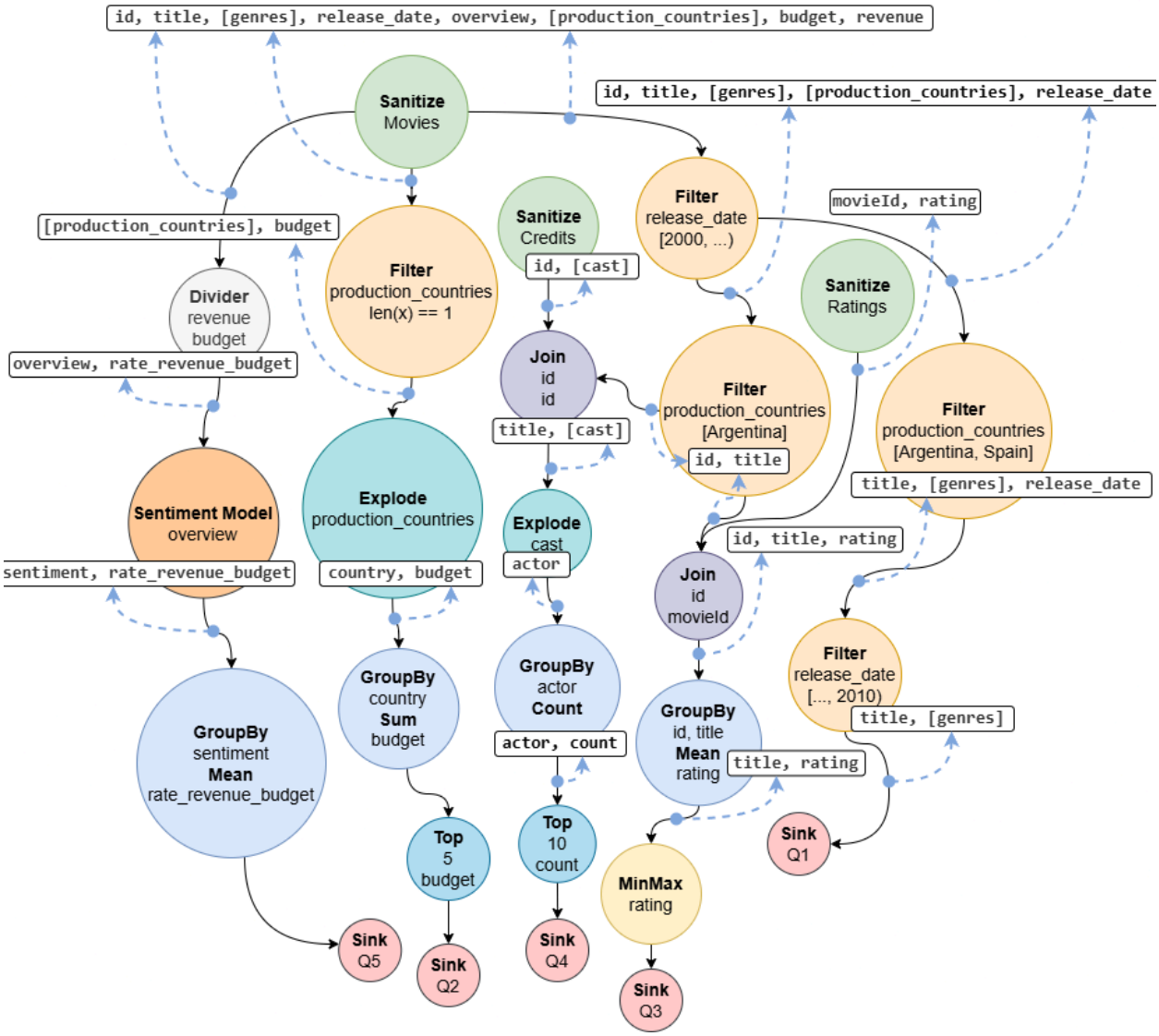
Utilizando el dataset de películas, el sistema obtiene el promedio de la tasa de ingreso/presupuesto por cada sentimiento obtenido de las reseñas de cada película utilizando un modelo de machine learning.

Arquitectura del sistema

Flujo de información en las tareas

A continuación se encuentra el diagrama de digrafo acíclico del sistema, donde se pueden diferenciar los distintos tipos de nodos que interactúan para la correcta resolución de queries pedidas en el enunciado.

Los nodos de color verde son aquellos que producen la información correspondiente al nombre, funcionando como fuentes de información. Los nodos rojos (sinks) son consumidores finales de la información, llegado este punto las queries fueron resueltas y deben ser informadas al cliente.



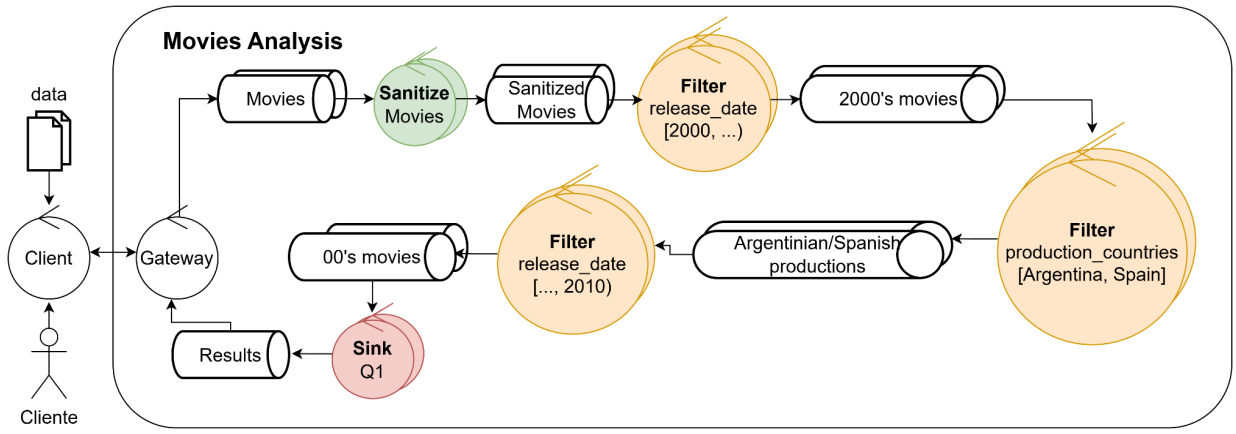
Comunicación entre procesos

El diagrama presentado permite visualizar de forma clara cómo interactúan las distintas entidades del sistema, en conjunto con las colas de mensajes que facilitan la comunicación asincrónica entre componentes.

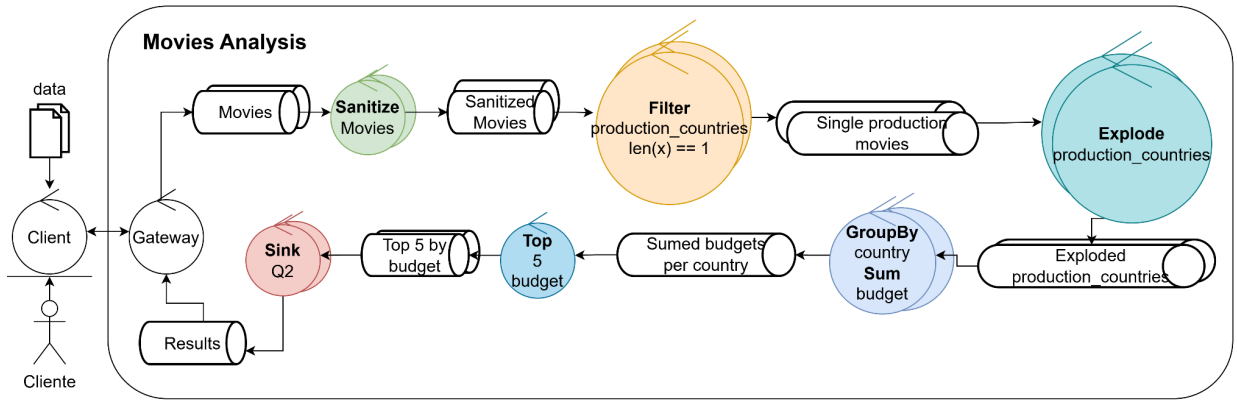
El sistema cuenta con varios tipos de workers, los cuales pueden ejecutarse en múltiples instancias para asegurar escalabilidad y paralelismo. Algunos ejemplos de estos workers son

filtros, explosiones, analizador de sentimientos, joins y agrupamientos. Particularmente para comunicar con las colas en estos dos últimos se hace mediante sharding permitiendo la distribución correcta de trabajo.
Por otro lado, hay otras que de momento creemos deberían correr de forma única y llevar consigo un estado interno para calcular sus resultados, estas instancias son por ejemplo los top-k y el worker minmax que calcula el mínimo y el máximo de una de las columnas.
Con el objetivo de facilitar la comprensión del flujo general del sistema, el diagrama ha sido estructurado en diferentes casos de uso, permitiendo una visualización más segmentada y enfocada de las operaciones involucradas.

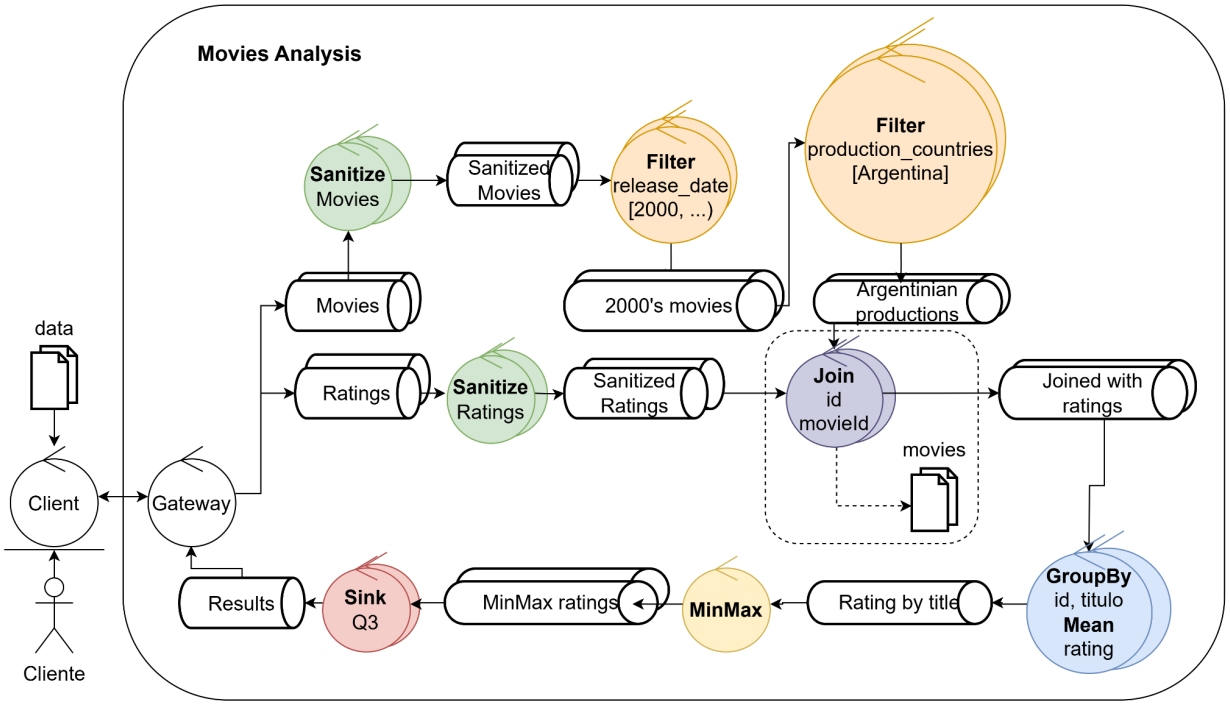
CU1 - Producción Argentina y España (2000s)



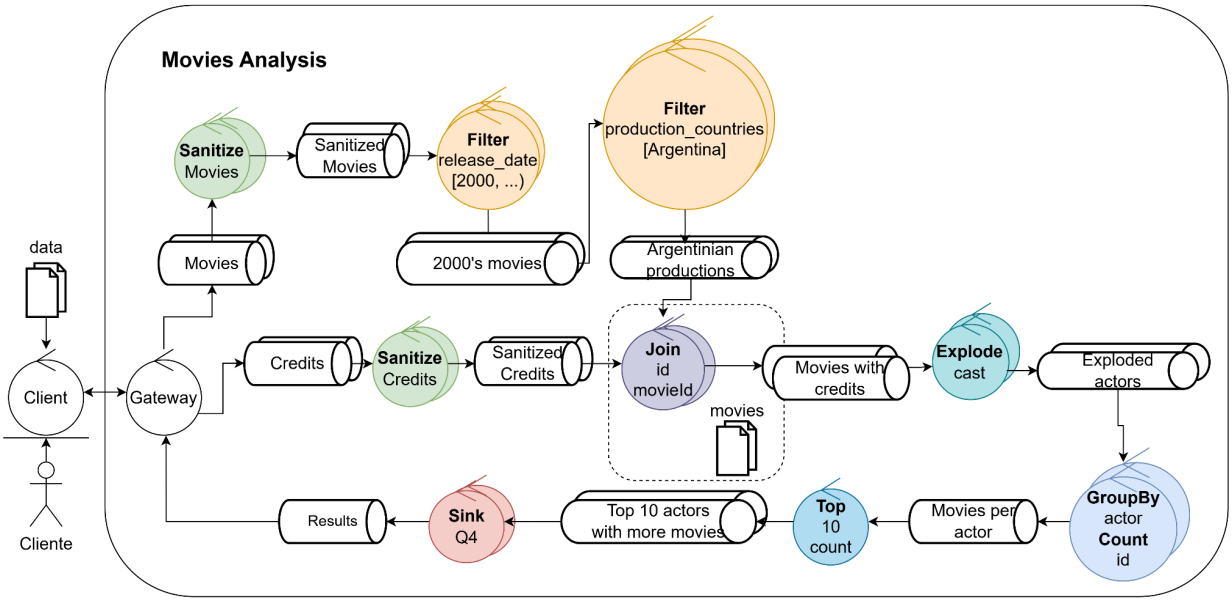
CU2 - Top 5 Inversores en Producciones Nacionales



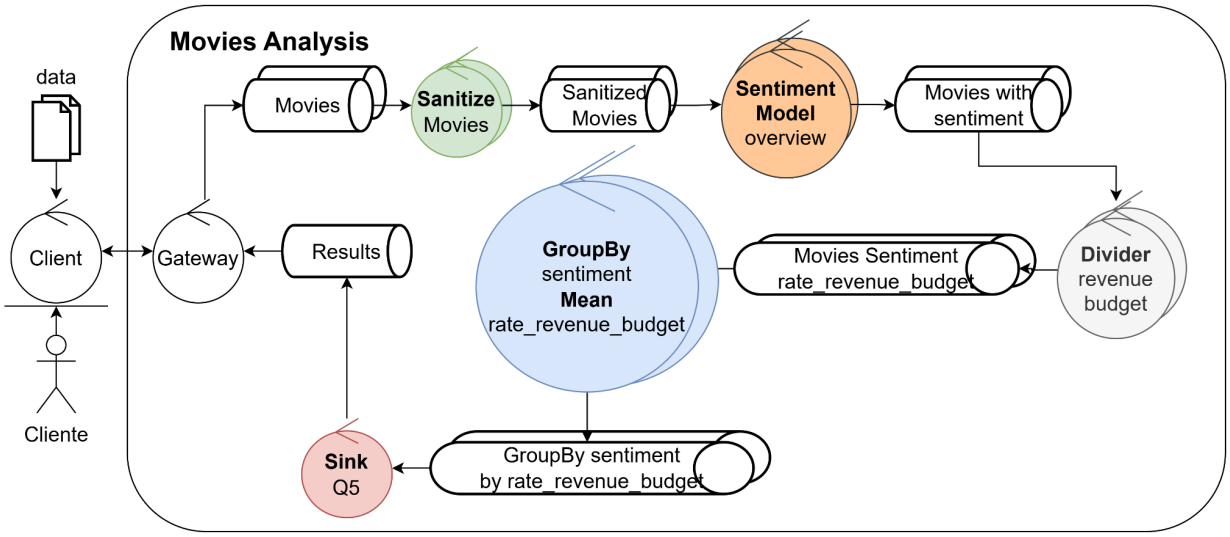
CU3 - Mejor y Peor Rating – Cine Argentino 2000+



CU4 - Actores Más Frecuentes en Argentina desde 2000



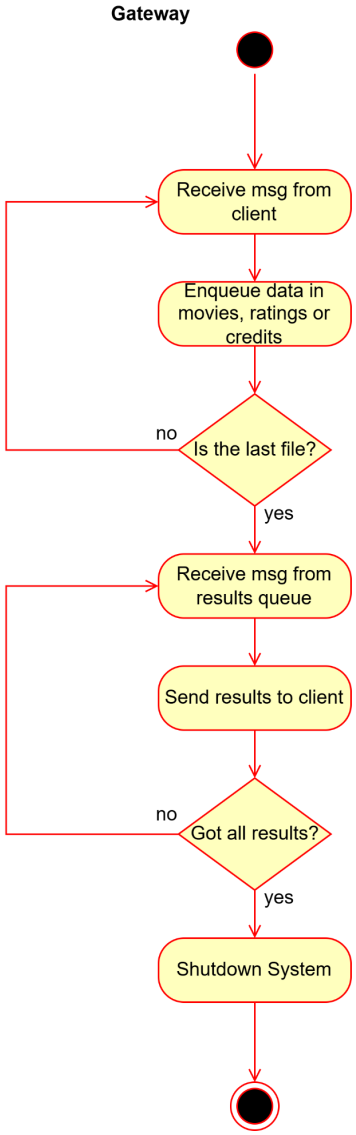
CU5 - Tasa Ingreso/Presupuesto: Positivo vs. Negativo



Casos particulares

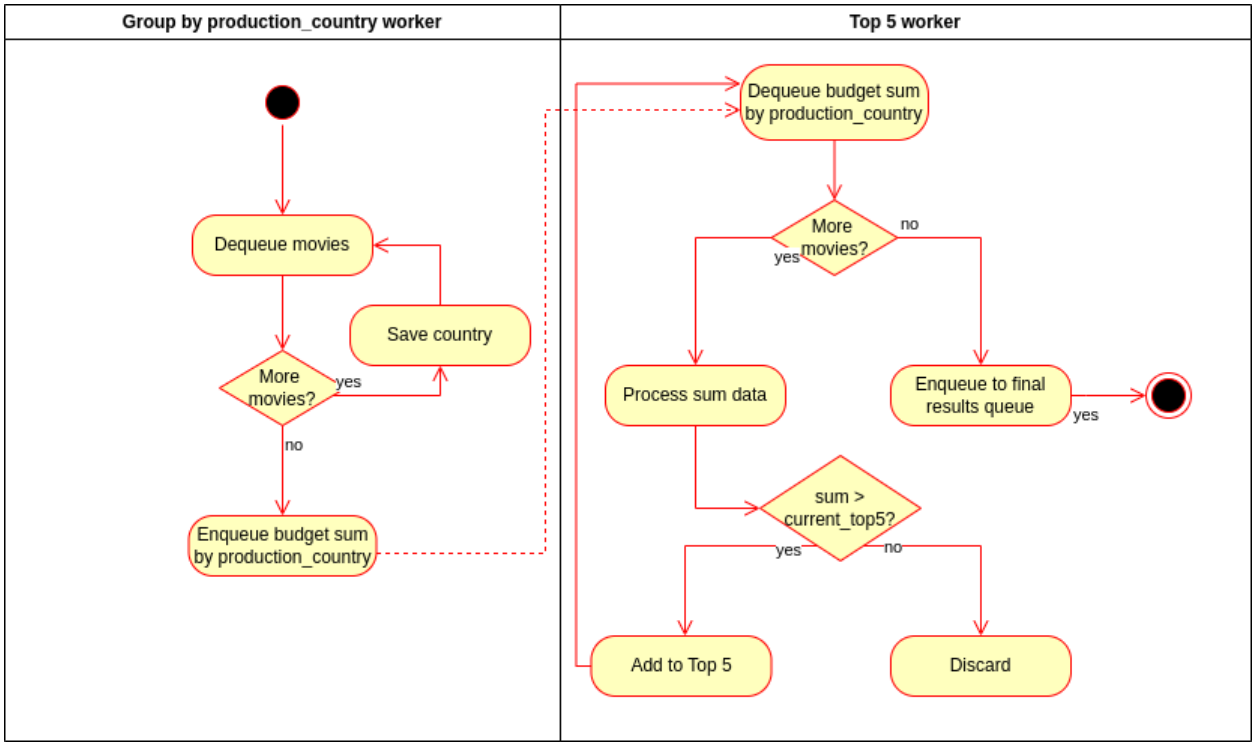
Manejo de mensajes del gateway

El siguiente flujo muestra el comportamiento del gateway, primero espera a recibir todos los datos necesarios del cliente para luego esperar por los resultados de las distintas queries para a medida que llegan ser enviados al cliente. Una vez termina, se hace shutdown al sistema.



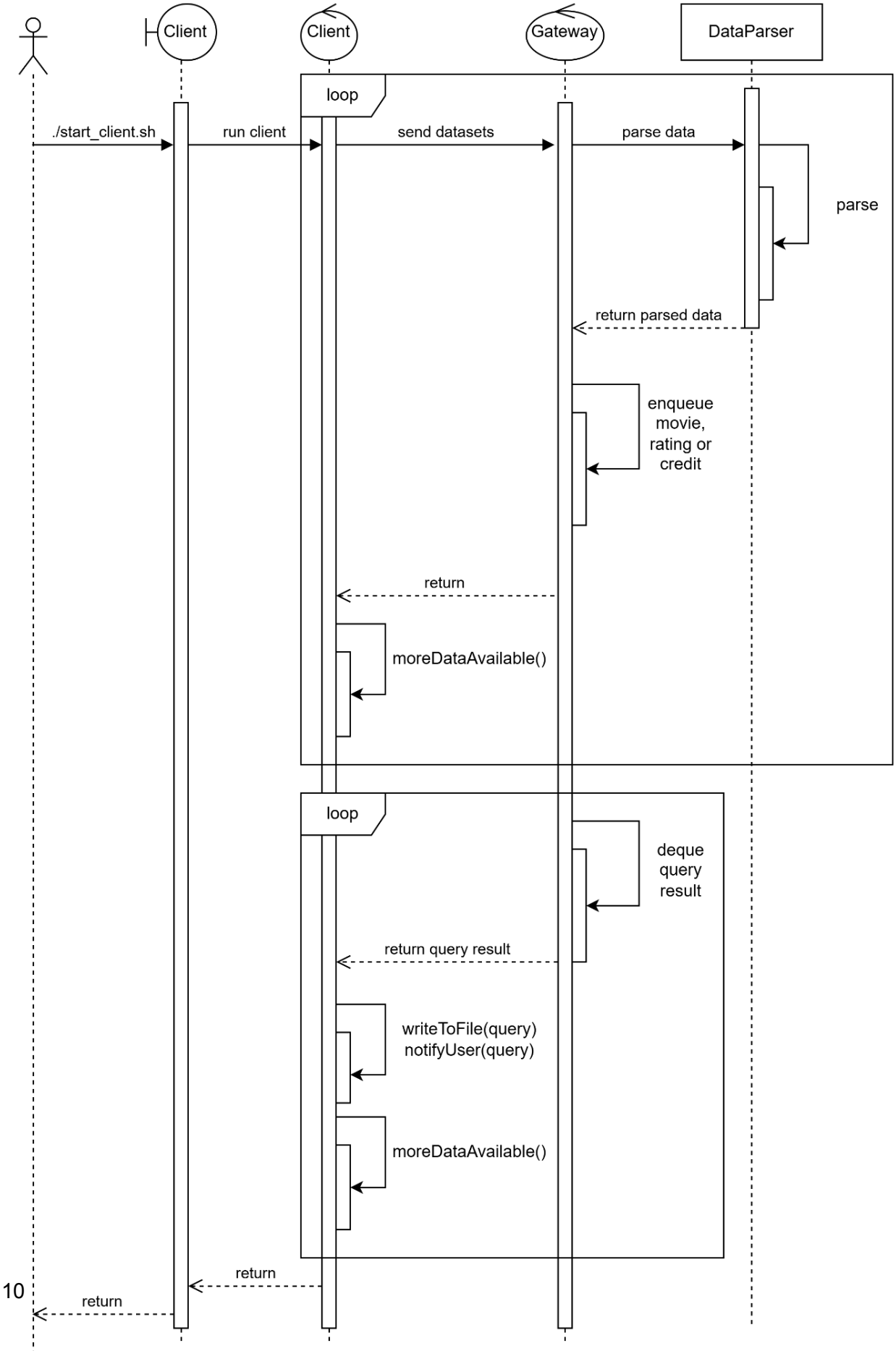
Top 5 de países

En este diagrama se puede observar como es el flujo para obtener el top 5 países con mayor dinero invertido. El Group by production_country worker va desencolando películas y las agrupa por país sumando el dinero invertido, para luego encolarlas en la cola del top 5 worker para que pueda obtener el top de países con mayor dinero invertido.



Interacción entre componentes del sistema

En este diagrama de secuencia se muestra la interacción entre el cliente y el sistema. El cliente inicia el proceso mediante un script que envía la información al gateway y queda a la espera de una respuesta. El gateway, por su parte, se encarga de parsear los datos recibidos y encolarlos para que los workers los procesen. Una vez que ha finalizado la recepción y el encolado de la información, el gateway comienza a desencolar los resultados procesados y los envía en conjunto al cliente.

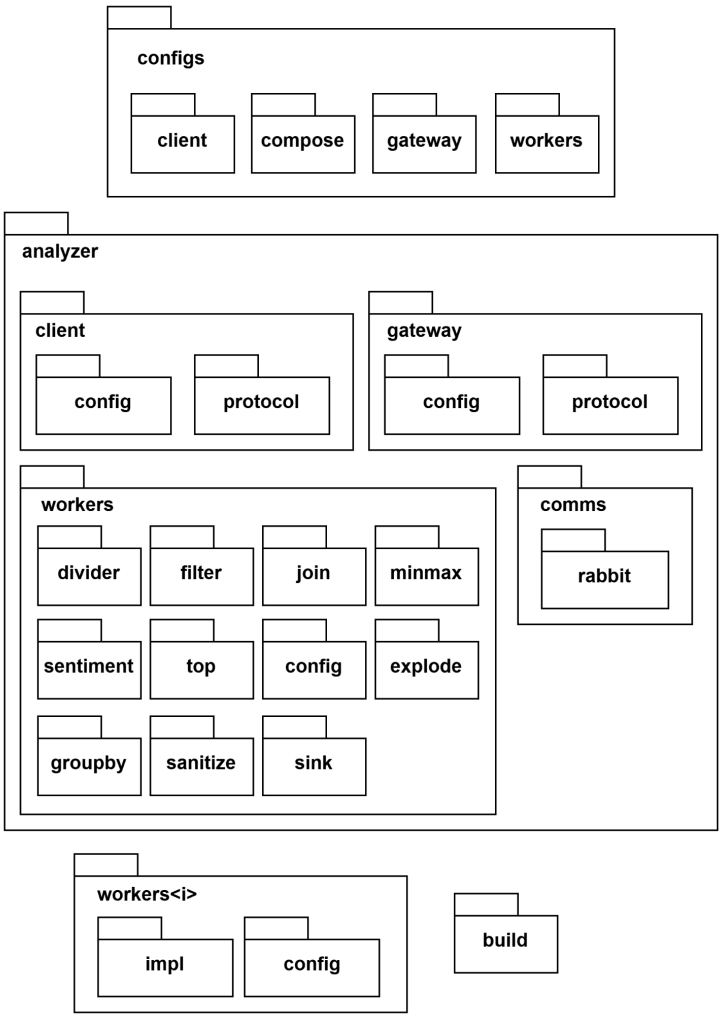

Client
Client
Gateway
DataParser
loop
./start_client.sh
run client
send datasets
parse data
parse
return parsed data
enqueue movie, rating or credit
return
moreDataAvailable()
loop
deque query result
return query result
writeToFile(query)
notifyUser(query)
moreDataAvailable()
return
return

Estructura y organización interna

Organizamos los directorios del proyecto de manera que cada componente tenga responsabilidades claramente separadas. Existe una carpeta general llamada configs, encargada de almacenar las variables de entorno utilizadas por cada worker.

Por su parte, el cliente cuenta con su propia configuración y protocolo para comunicarse con el gateway. El gateway, a su vez, posee su propia configuración, define el protocolo de comunicación con el cliente y contiene una carpeta adicional llamada Rabbit, responsable de manejar la comunicación con los workers.

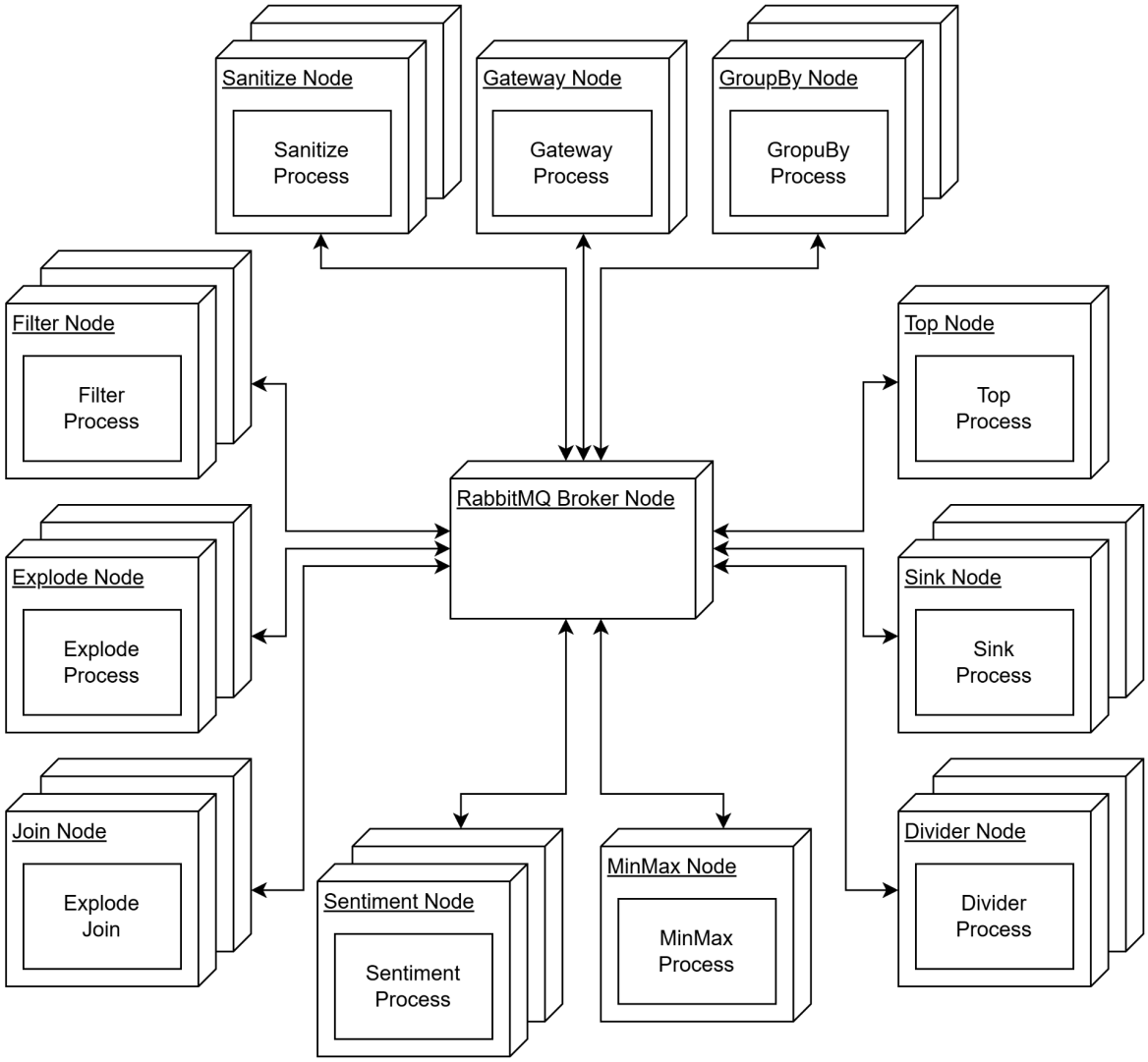
Cada worker comparte una configuración general común y, además, incluye una configuración específica junto con su implementación, representada en el paquete Worker<i>.



Despliegue del sistema

En este diagrama se puede observar la distribución de nodos y procesos dentro de ellos. Decidimos dividir los nodos por funcionalidad, donde hay un encargado por cada tipo de actividad y dentro del él existen procesos que resuelven alguna instancia particular del sistema, por ejemplo dentro de nodo de filtros existe un proceso encargado de filtrar por año en fechas mayor o iguales a 2000.

Todos los nodos realizan su comunicación mediante el nodo central de RabbitMQ que proporciona las colas de mensajes. Para la comunicación entre el cliente y el gateway creímos más cómodo tener una conexión directa.



Distribución de tareas

Tarea	Responsable
Gateway	Todos
Protocolo	Todos
Cliente	Todos
Filter	Ghosn
Explode	Bianchi
Join	Todos
Group By	Bianchi
Top	Gentilini
Sink	Ghosn
Divider	Gentilini
MinMax	Gentilini
Sentiment	Bianchi
Multicliente	Todos