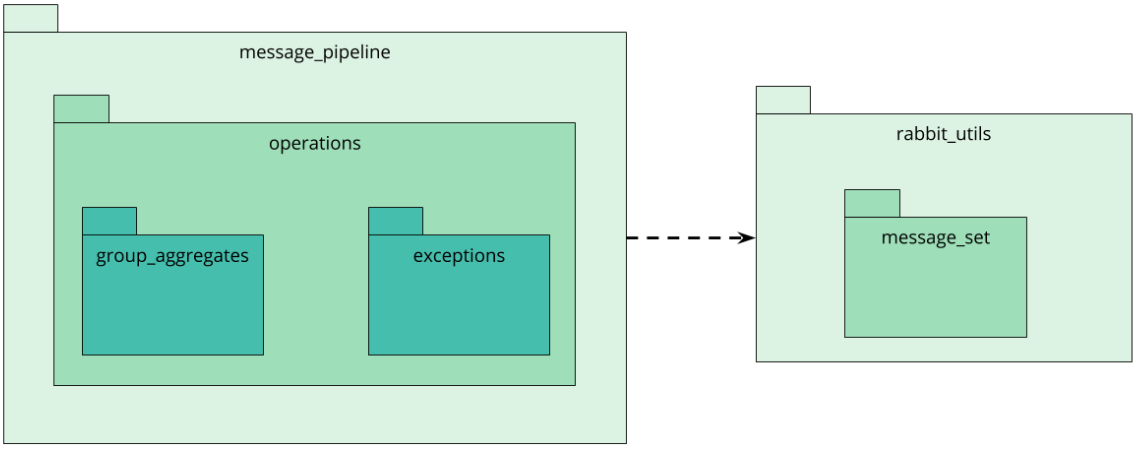
TP2: Yelp Reviews Analysis

75.74 - Sistemas Distribuidos I

Alumno: Gianmarco Cafferata
Padrón: 99423

Diagrama de paquetes

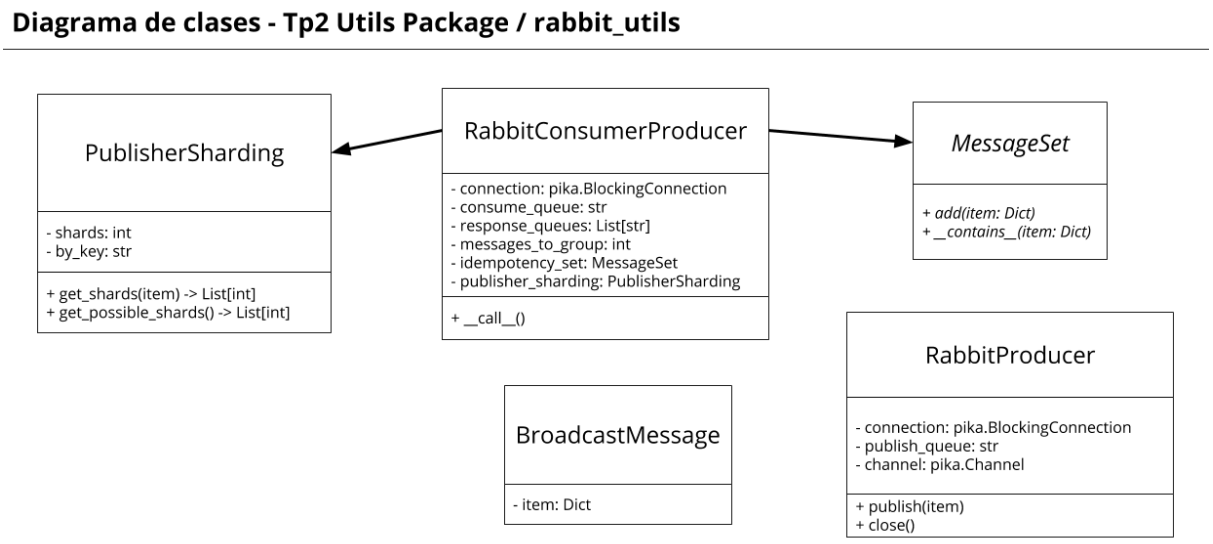
Para este trabajo se programó una pequeña librería que resuelve cálculos y operaciones sobre los mensajes de la forma más general posible. Luego los verdaderos servicios que terminan realizando las operaciones son breves scripts que utilizan estas herramientas y generan las operaciones necesarias por medio de un archivo de configuración y el patrón *factory*.



- Message pipeline: contiene a todo lo responsable de aplicar un pipeline de operaciones a los distintos mensajes en formato diccionario que reciba.
- Rabbit utils: contiene todos los objetos necesarios para enviar y recibir mensajes a través de rabbit.

Diagramas de clases

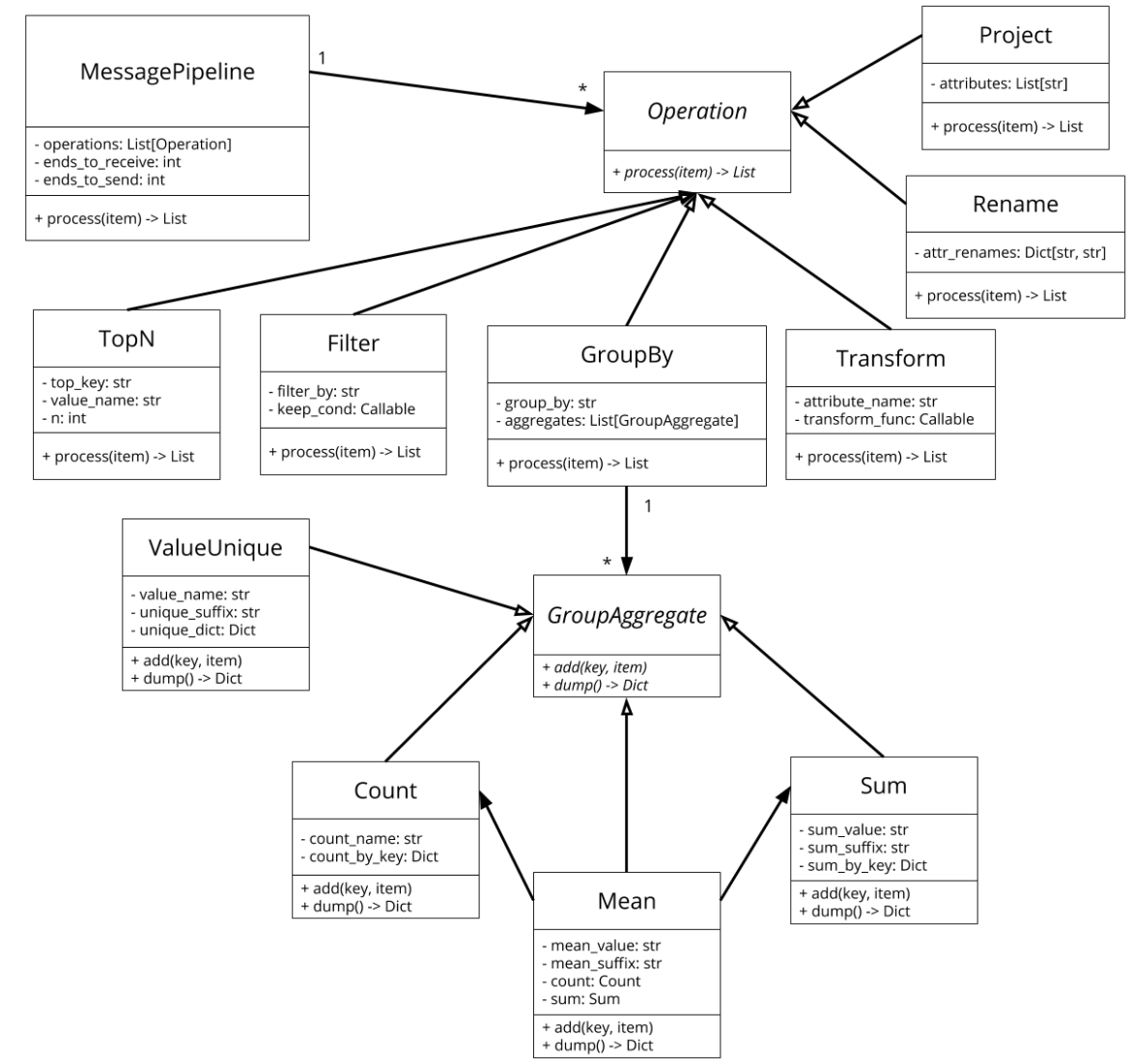
rabbit_utils



- RabbitConsumerProducer: Encargado de consumir de una cola y emitir las respuestas en otra.
- MessageSet: Interfaz que permite saber si un mensaje ya fue procesado y garantizar idempotencia (incluye una implementación con uso de disco y filtros de Bloom que si bien esta testeada finalmente no se utilizó en el trabajo, quería incluir como alcance ese y otros objetos para lograr alguna tolerancia a fallos, pero no llegue con el tiempo).
- RabbitProducer: Encargado de producir en una cola particular
- PublisherSharding: Objeto que se agrega al RabbitConsumerProducer si se desea particionar la salida en shards según alguna clave.
- BroadcastMessage: wrapper de un diccionario utilizado como mensaje especial para indicar que el mensaje debe ser transmitido a todas las colas a las cuales se está produciendo, utilizado para enviar EOFs.

message_pipeline

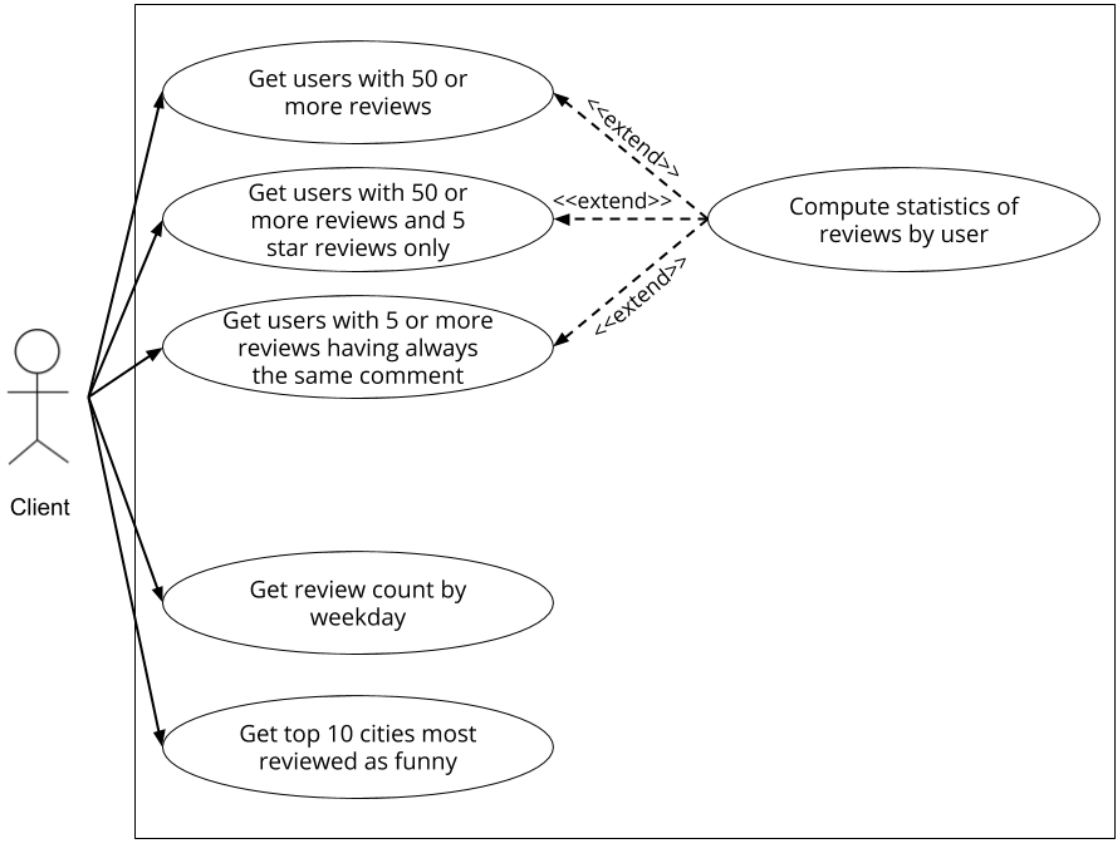
Diagrama de clases - Tp2 Utils Package / message_pipeline



- MessagePipeline: Contiene una lista de operaciones a realizar sobre los items. Su responsabilidad es hacer estas tareas en orden, reconocer cuando la ventana del stream termina y broadcastear la correspondiente señal a sus sucesores.
- Operation: Interfaz de una operación del MessagePipeline.
    - Project: Encargado de quedarse con algunas claves del diccionario.
    - Rename: Encargado de renombrar claves del diccionario.
    - Transform: Aplica alguna función a algún valor del diccionario.
    - GroupBy: Acumula determinados agregados respecto de una clave elegida para todos los mensajes que reciba, finalmente, cuando termina la ventana del stream devuelve el resultado de todos sus agregados (GroupAggregates) y reinicia su estado para recibir otra ventana.
    - Filter: Filtra los ítems según un determinado valor y una función de filtro.
    - TopN: Se queda con las top N claves cuyo valor para algún campo sea del top N más grande.

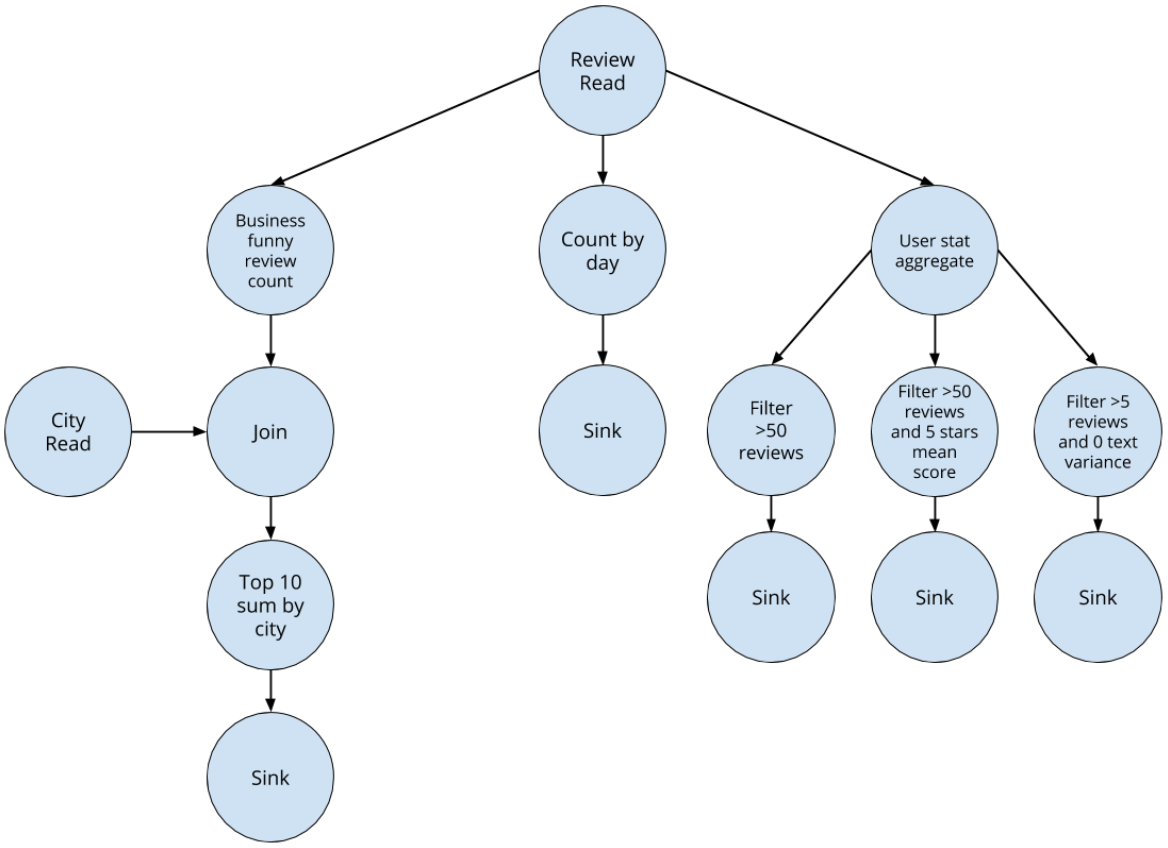
- GroupAggregate: Interfaz que permite calcular un agregado para la operación GroupBy.
    - Count: Cantidad de ocurrencias.
    - Sum: Suma de un valor.
    - Mean: Promedio de un valor.
    - ValueUnique: Si un valor para una determinada clave es siempre el mismo.

Diagrama de casos de uso



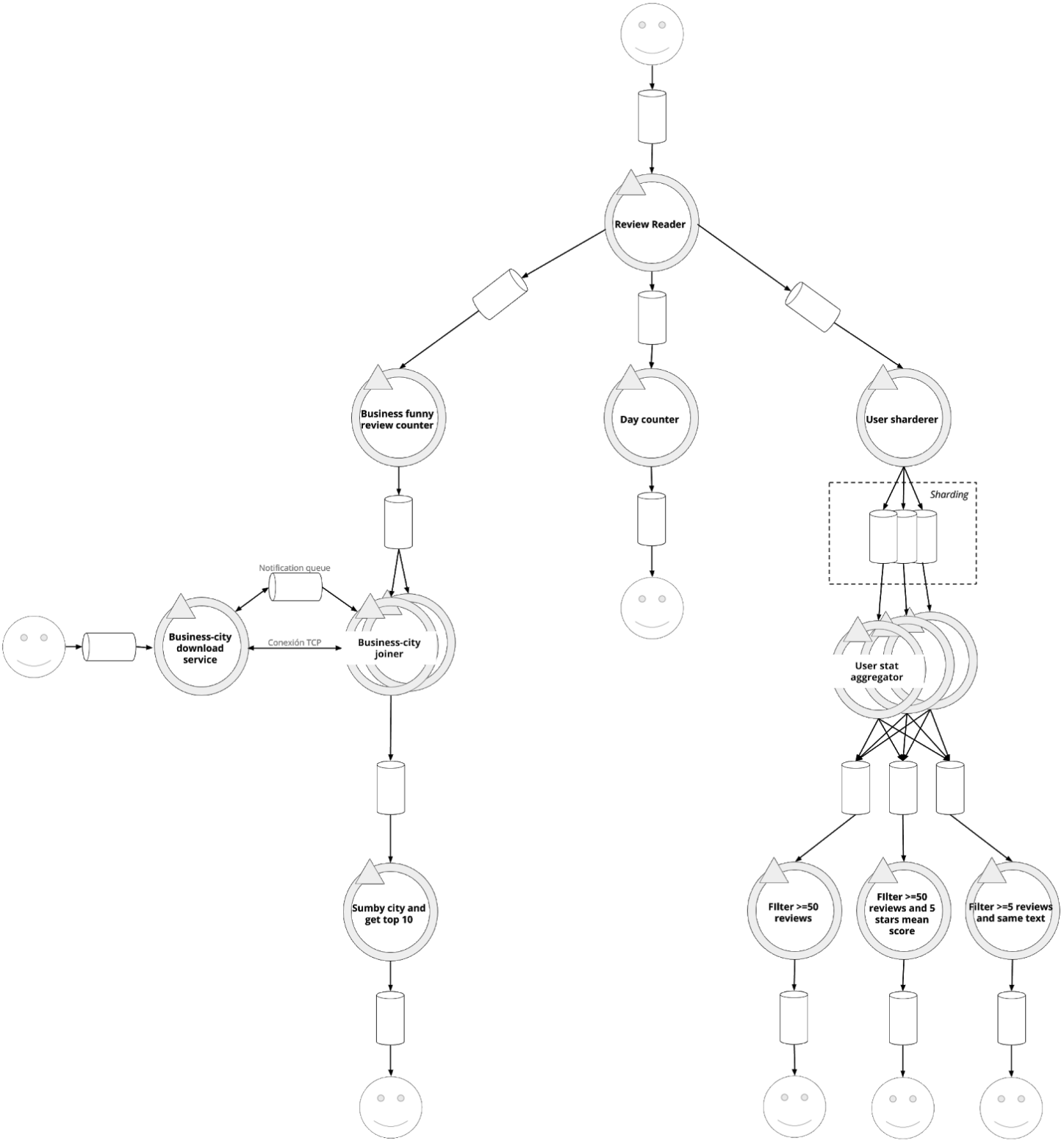
Estos casos de uso se corresponden con las operaciones del DAG.

Grafo dirigido acíclico de operaciones



Podemos ver estas operaciones mapeadas a procesos en el diagrama de robustez.

Diagrama de robustez

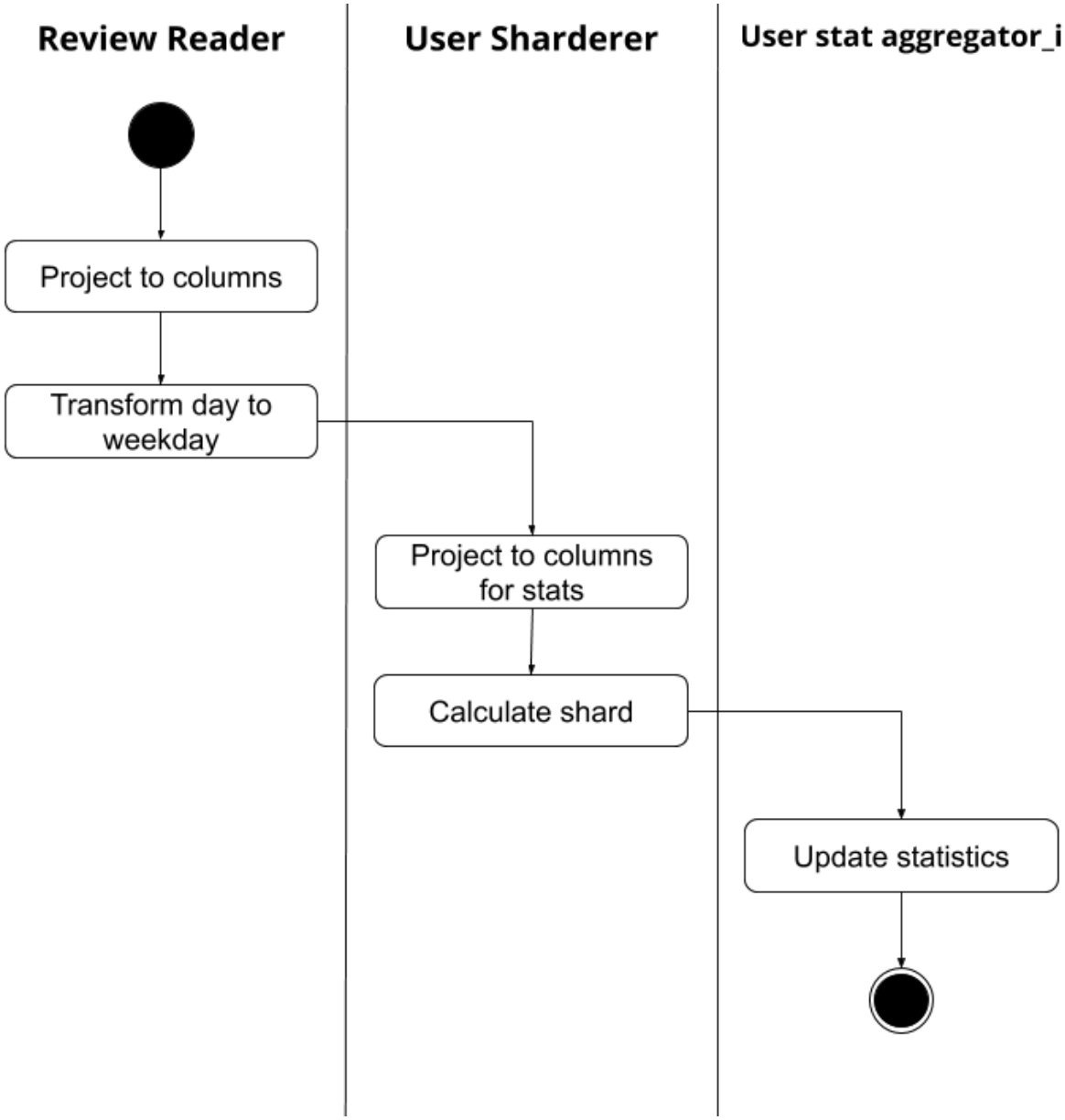


- Review Reader: Lee las reviews, se queda con los valores que se van a utilizar y mapea la fecha a día de la semana.
- User Sharderer: Se queda solo con los valores que se van a utilizar más adelante y shardea por user_id en colas distintas.

- User stat aggregator: Calculan las estadísticas para el shard de usuarios que les corresponde.
- Filter >=50 reviews: Se queda con los usuarios que tuvieran 50 o más reviews.
- Filter >=50 reviews and 5 stars score mean: Se queda con los usuarios que tuvieran 50 o más reviews y siempre puntúan 5 estrellas.
- Day Counter: Cuenta la ocurrencia de cada día de la semana.
- Business funny reviews counter: Cuenta la cantidad de reviews marcadas como ‘funny’ para cada negocio.
- Business city joiner: Espera a ser notificado por el Business-city downloader para estar seguro que le puede pedir por TCP el diccionario que asocia negocios a ciudades. Cuando sabe que puede pedirlo y lo obtiene, realiza el join.
- Business city downloader service: Recibe todas las asociaciones los negocio-ciudad, las guarda en disco, notifica que están listas y hace de servidor para su descarga.
- Sum by city and get top 10: Agrupa por ciudad sumando la cantidad de reviews ‘funny’ y obtiene el top 10.

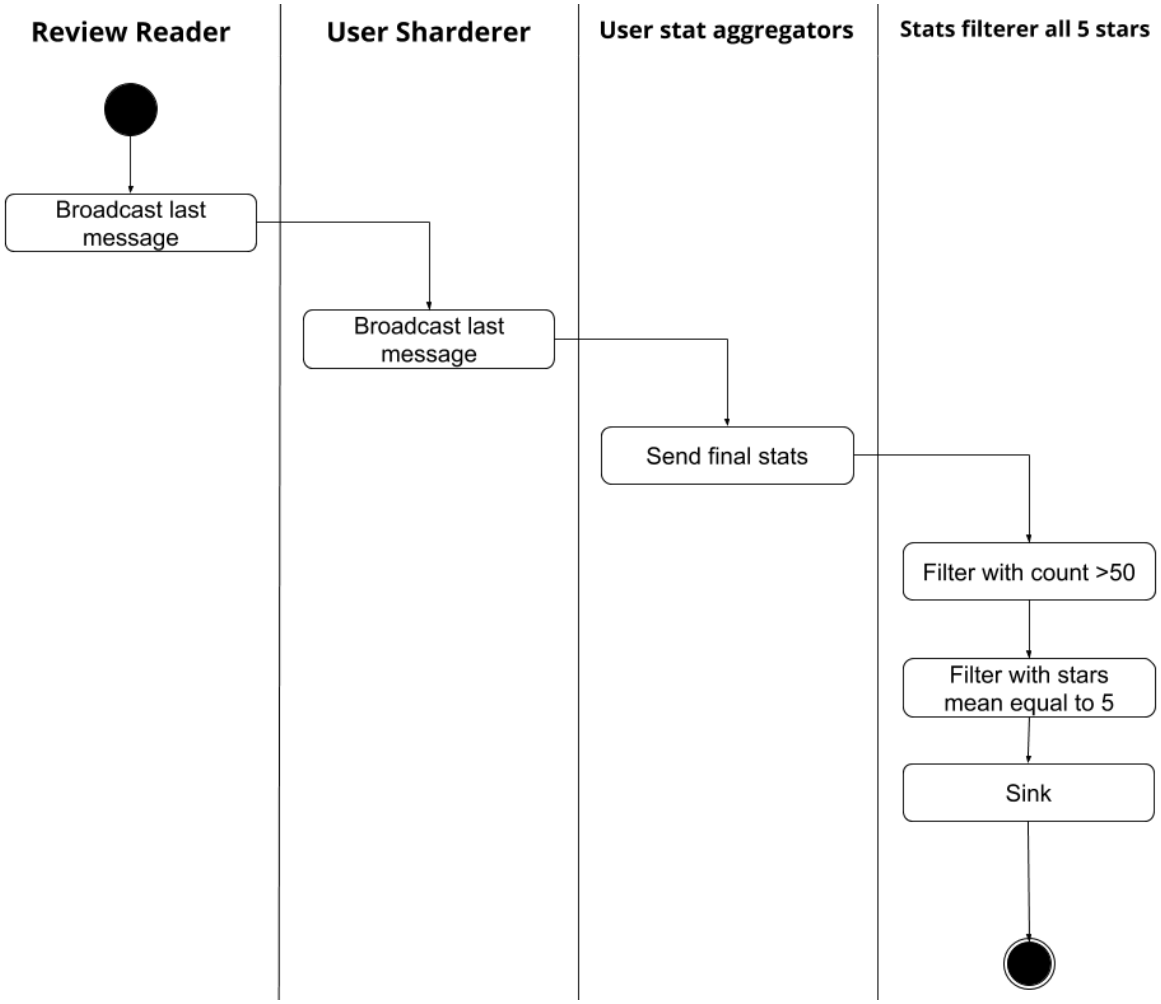
Diagrama de actividad

Lectura de una review para estadísticas de usuario



La actividad comienza cuando se recibe una novedad correspondiente a una nueva review, este flujo es distinto de lo que ocurre cuando se recibe la señal de fin de ventana. En este caso el correspondiente user stat aggregator al shard de ese mensaje actualiza sus estadísticas.

Lectura del fín de ventana para usuarios con 50 reviews o más y siempre 5 estrellas



La actividad comienza con la llegada de un mensaje que indica el fin de la ventana del stream, cada uno de los procesos que no tiene nada que enviar broadcastea la señal a los siguientes, por otro lado, los user stat aggregators deben devolver todas las estadísticas calculadas hasta entonces y reiniciar su estado. Finalmente un proceso se ocupa de filtrar.

Diagrama de secuencia

En el siguiente diagrama de secuencia podemos ver que ocurre desde que se reciben los negocios hasta que se emite el top 10 de ciudades con más reviews ‘funny’.