



Facultad de Ingeniería
Universidad de Buenos Aires

Sistemas Distribuidos I (75.74)

Flights Optimizer

TP Escalabilidad: Middleware y Coordinación de Procesos

Documento de arquitectura

Alumnos

- Guido Bergman (104030)
- Luis Waldman (79279)

Índice

Alcance del sistema.....	3
Arquitectura de software.....	3
Casos de uso.....	3
Vista Física.....	4
Diagrama de Robustez.....	4
Diagrama de Despliegue.....	6
Vista Desarrollo.....	7
Diagrama de paquetes.....	7
Vista Procesos.....	8
DAG.....	8
Diagramas de Actividad.....	8
Diagrama de secuencia.....	14
Vista Lógica.....	14
Diagrama de clases.....	14
Vista Protocolos.....	17
Protocolos de comunicación entre el cliente y el servidor.....	17
Vuelos.....	17
Aeropuertos.....	17
Resultados.....	18
Resultados escalas.....	18
Resultados filtro distancia.....	18
Resultados filtro distancia.....	18
Resultados vuelos rápidos.....	19
Resultados estadísticas precios.....	19
Protocolos de comunicación entre el handler del cliente y los filtros.....	19
Envío de vuelos al filtro de escalas.....	20
Envío de aeropuertos al filtro de distancias.....	20
Envío de vuelos al filtro de distancias.....	20
Envío de vuelos al filtro de vuelos rápidos.....	21
Envío de vuelos al filtro de estadísticas de precios.....	21
Resultados.....	21
Protocolo de comunicación entre el filtro de estadísticas de precios y el calculador de promedios.....	21
Listado de tareas a ejecutar y división entre integrantes.....	23

Alcance del sistema

El sistema *flights optimizer* procesa registros de vuelos de avión y a partir de ellos permite conocer:

- Los vuelos con 3 escalas o más
- Los vuelos cuya distancia total es mayor a cuatro veces la distancia directa entre el origen y el destino
- Los 2 vuelos más rápidos por trayecto, es decir origen y destino del vuelo, considerando solo los vuelos con 3 escalas o más
- El precio promedio y máximo para cada trayecto, considerando solamente los vuelos cuyo precio esté por encima del promedio de todos los precios

Arquitectura de software

El sistema estará desplegado en un entorno multi-computadora, en cual habrá un nodo para el cliente, mientras que el servidor se separará en:

- Cliente handler: es quien se encargará de la comunicación con el cliente
- Filtro escalas: filtra los vuelos de más de 3 escalas
- Buscador vuelos rápidos: busca los 2 vuelos más rápidos para cada trayecto
- Buscador de vuelos con distancias largas: busca los vuelos cuya distancia total es mayor a cuatro veces la distancia entre los puntos de origen y destino
- Calculador estadísticas vuelos costosos: para los vuelos cuyo precio esté por encima del promedio de todos los precios, permitirá conocer su precio promedio y máximo para cada trayecto

Casos de uso

En el siguiente diagrama se muestran los distintos casos de uso, que corresponden a los distintos tipos de resultados que puede recibir el cliente por parte del servidor.

Diagrama de casos de uso

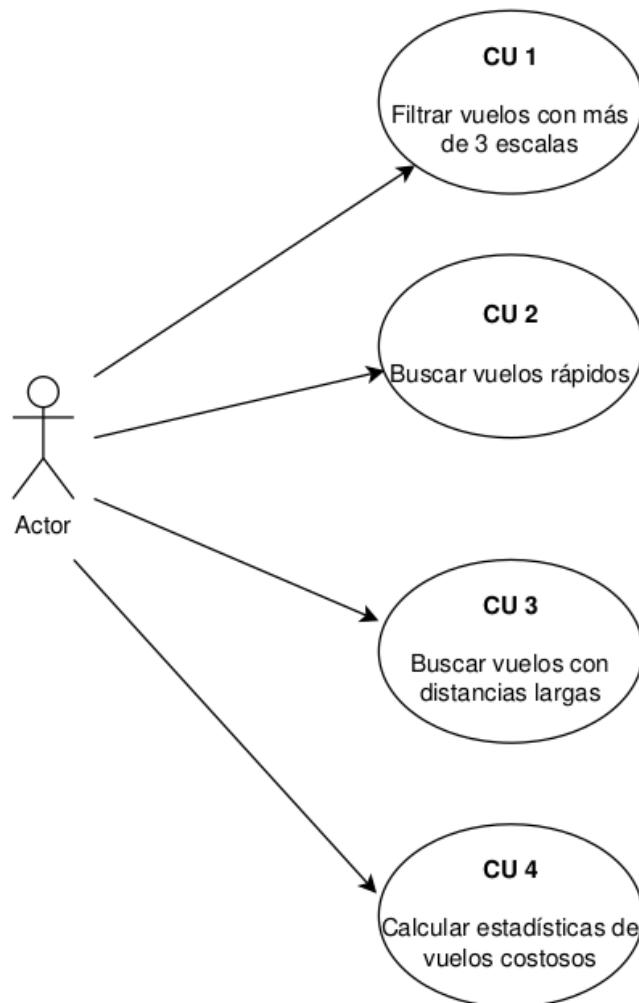


Fig 1. Diagrama de casos de uso

Vista Física

Diagrama de Robustez

En la figura 2 se pueden ver los distintos *controllers* y *entities* que habrá el sistema, así como las colas que se usarán para la comunicación.

Documento de robustez

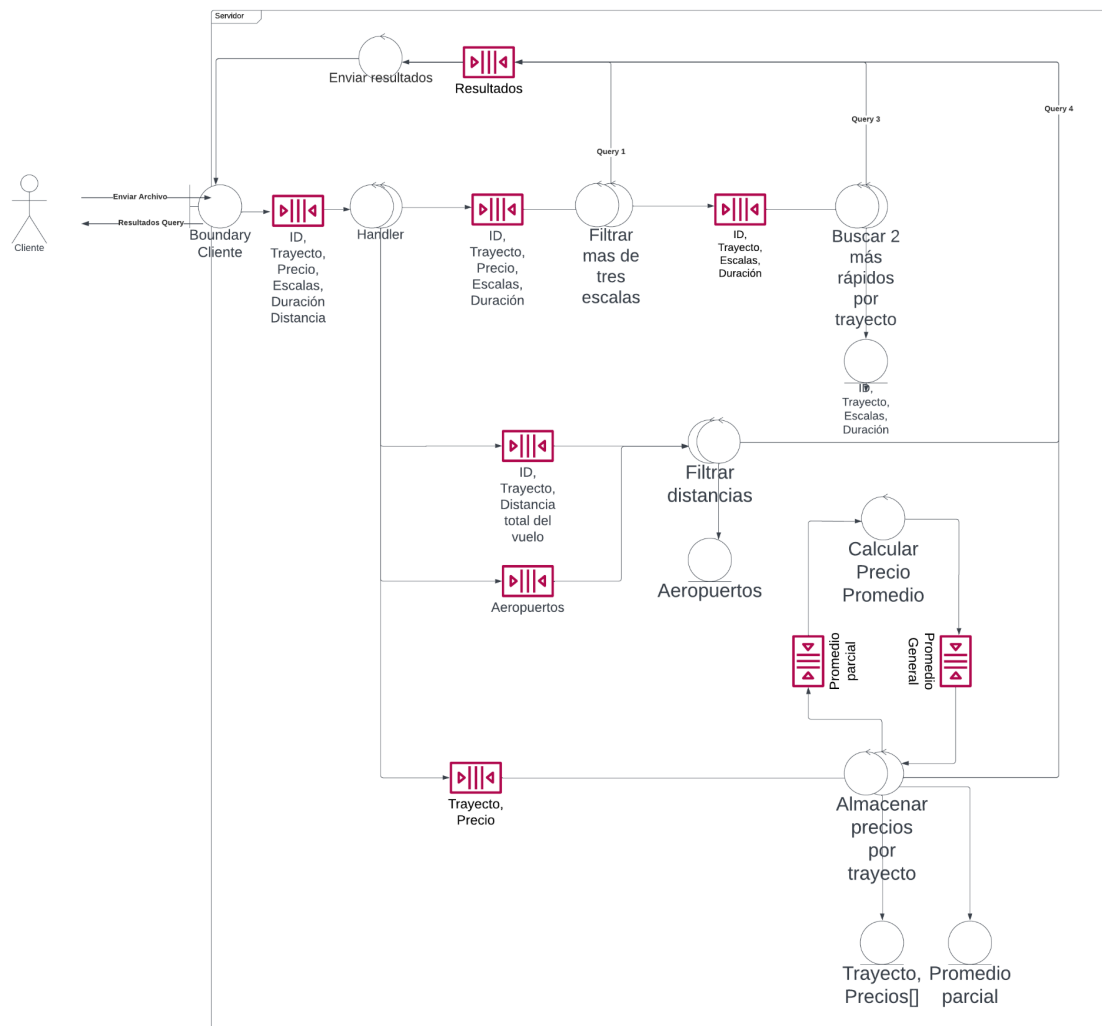


Fig 2. Diagrama de robustez

Diagrama de Despliegue

En la figura 3 se puede observar cómo estarán desplegados los *controllers* en los distintos nodos.

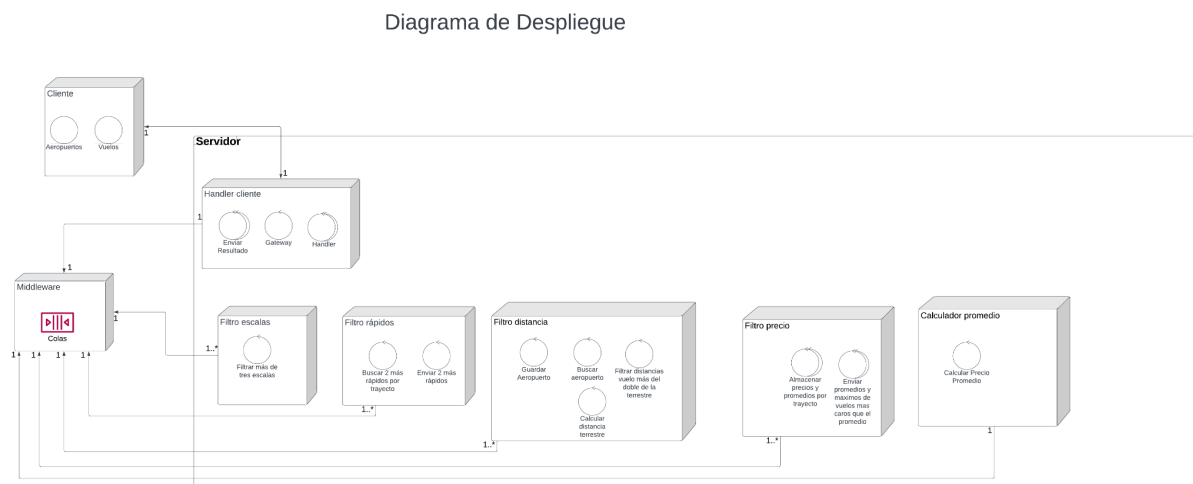


Fig 3. Diagrama de despliegue

Vista Desarrollo

Diagrama de paquetes

En la figura 4 se muestran los paquetes en los cuales estará separado el código.

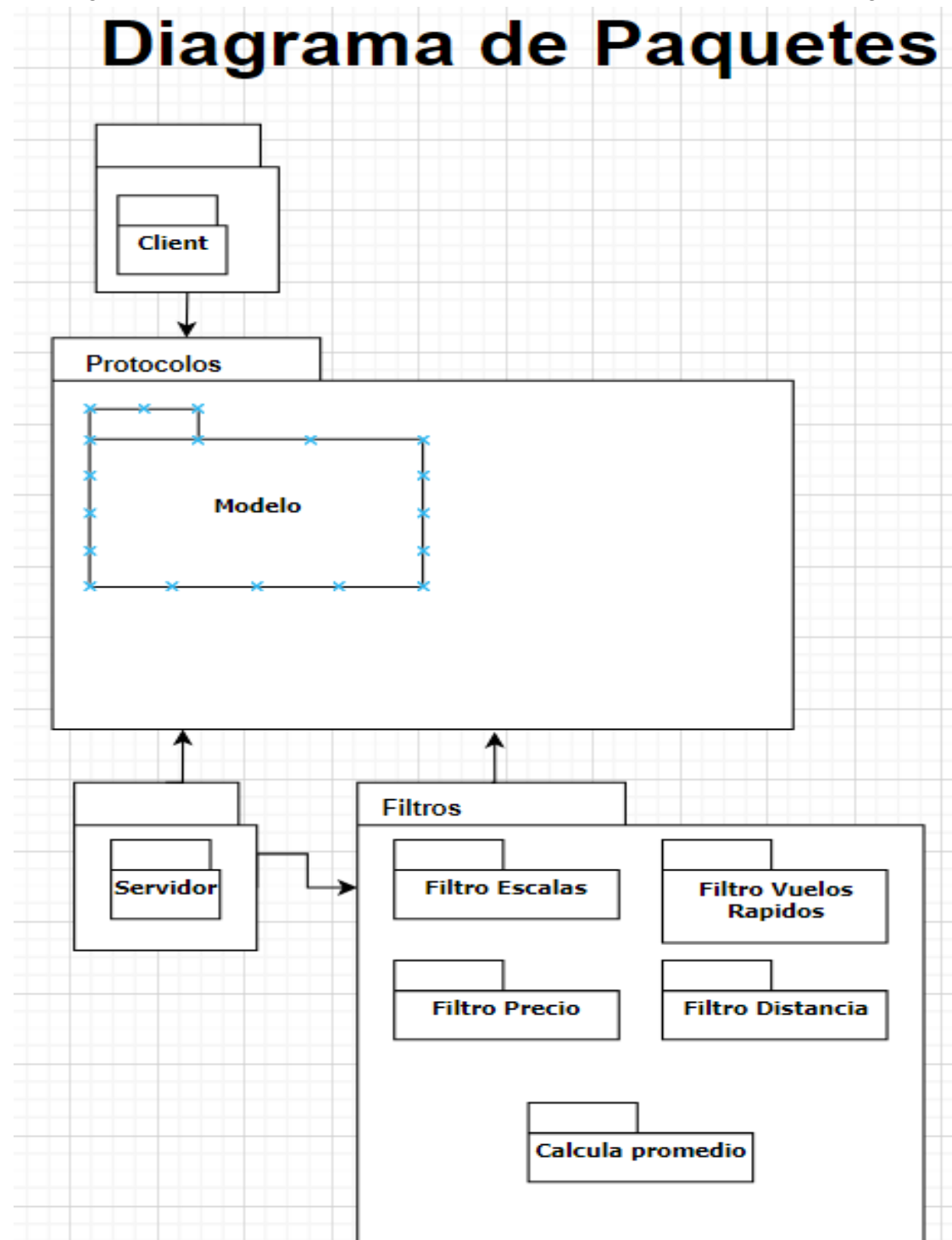


Fig 4. Diagrama de paquetes

Vista Procesos

DAG

En el siguiente DAG se podrán ver las distintas tareas que se realizan para obtener los resultados finales.

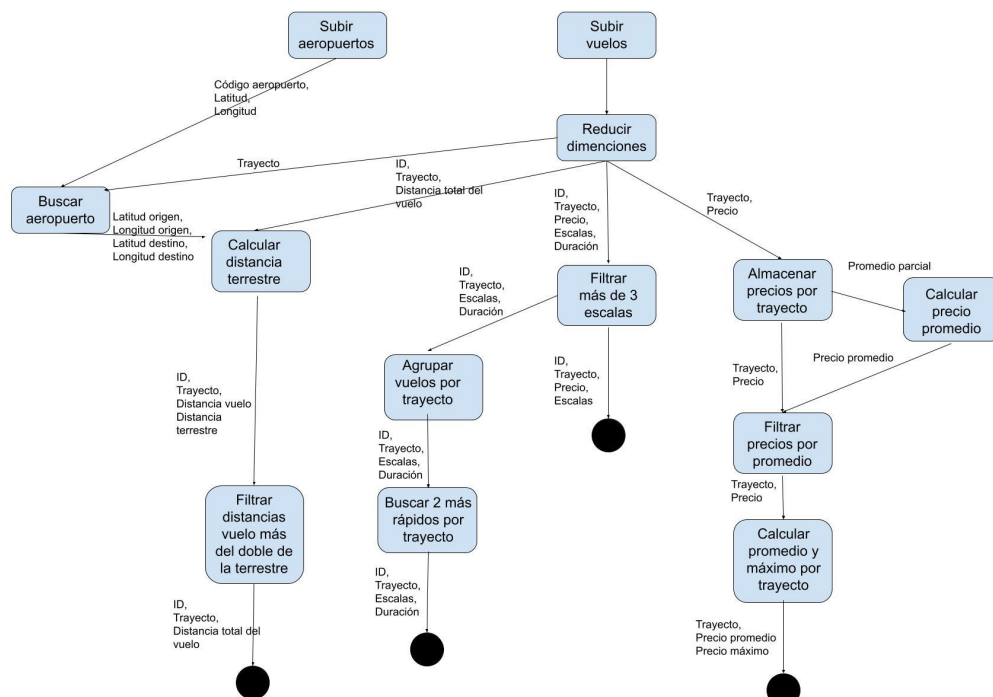


Fig 5. DAG

Diagramas de Actividad

En la figura 6 se muestra el flujo de trabajo que seguirá el sistema para devolverle al cliente los resultados solicitados.

Diagrama Actividades (general)

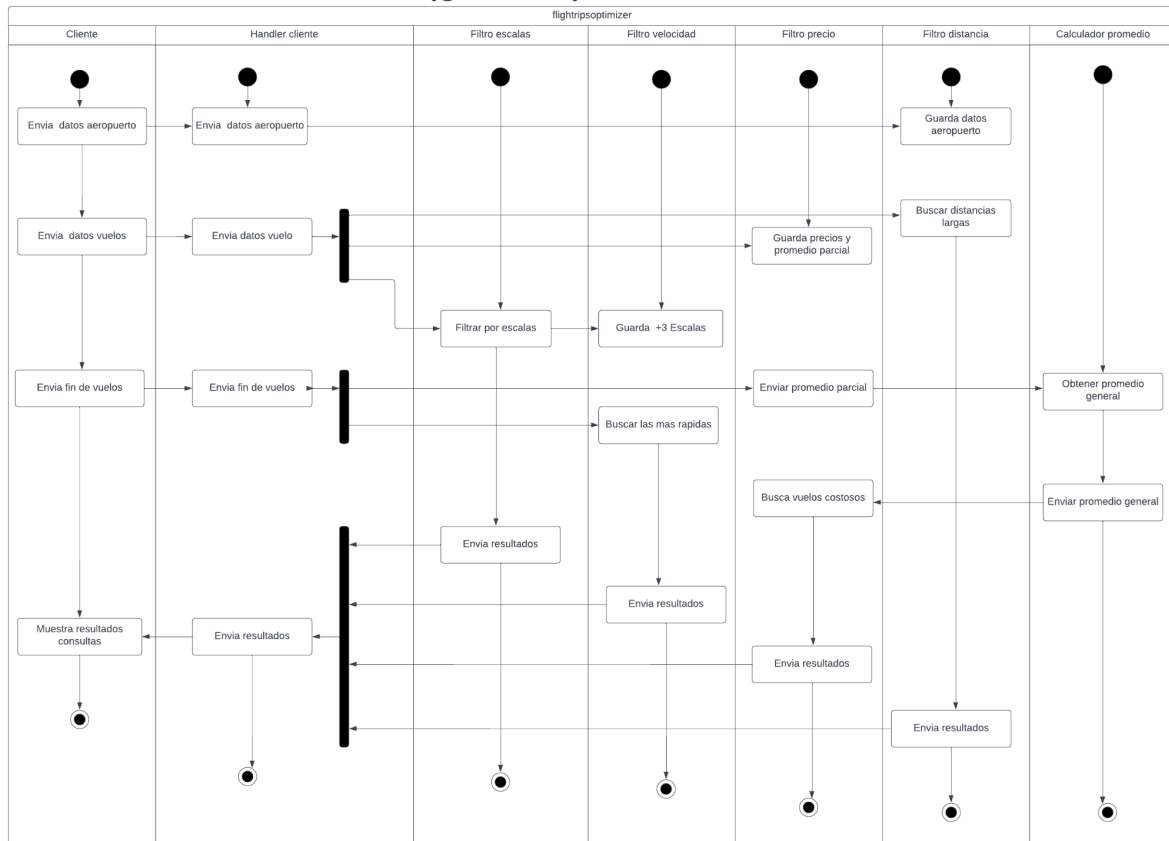


Fig 6. Diagrama de actividades general

En la figura 7, se pueden ver las distintas actividades que realizan el filtro de escalas y el de velocidad, tanto para procesar un vuelo como para procesar el fin de vuelos.

Como se puede ver en la figura, los filtros de escala envían los vuelos con 3 o más escalas a los filtros de velocidad. Para esto, se utiliza una estrategia de *sharding*. Para implementarla, se utiliza una cola con tópicos. Para determinar el tópico correspondiente a un vuelo se calcula un *hash* a partir del trayecto del vuelo. Esto permite que todos los vuelos de un mismo trayecto los reciba un mismo filtro de velocidad.

Al recibir el fin de vuelos, cada uno de los filtros de escalas enviará un mensaje de fin de vuelos a cada uno de los filtros de velocidades. Esto lo hace enviando el mensaje a cada uno de los tópicos correspondientes a un filtro de velocidad.

También se puede observar, que el filtro de velocidad no envía los resultados al recibir el primer mensaje de fin de vuelos, sino que espera a haber recibido los mensajes de fin de vuelos de todos los filtros de escalas. Esto permite que haya varios filtros de escala enviando vuelos y que solo se empiecen a enviar los resultados cuando se hayan terminado de recibir todos los vuelos.

Diagrama Actividades (escalas y velocidades)

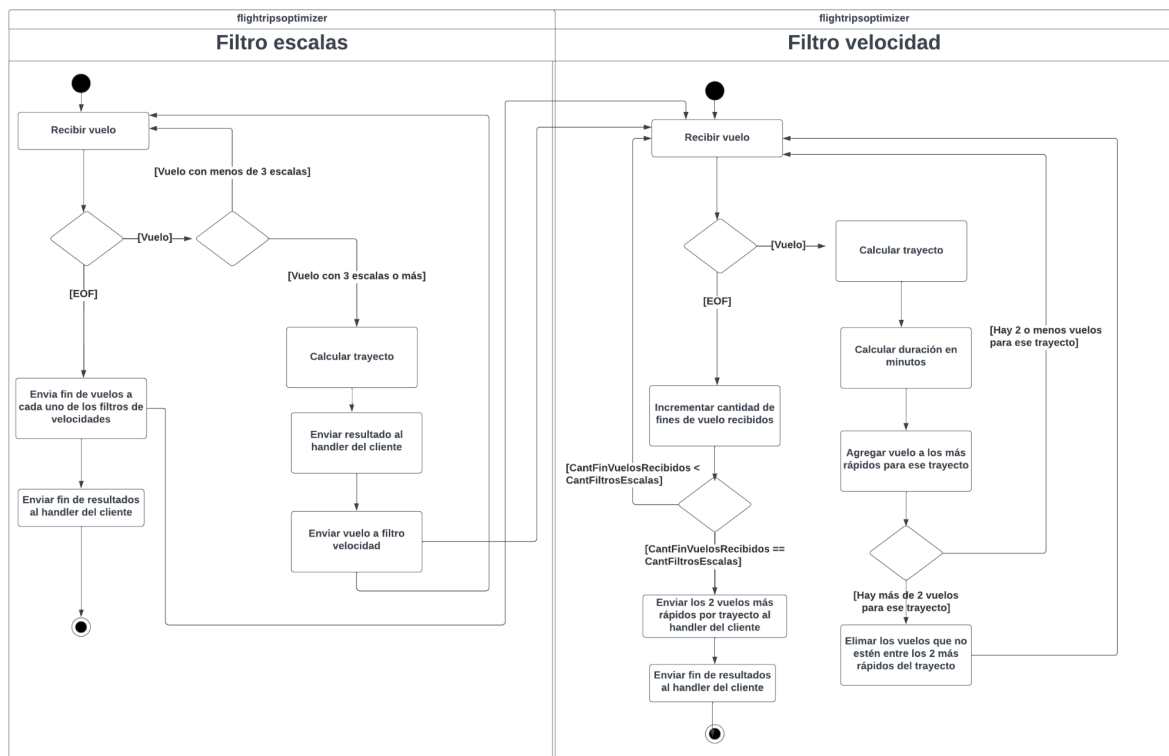


Fig 7. Diagrama de actividades filtros escalas y velocidad

En la figura 8 se pueden ver las distintas actividades que realizan el filtro de distancia y el handler del cliente, para recibir los aeropuertos, procesar los vuelos y el fin de vuelos. En el caso de los aeropuertos, dado que todos los filtros de distancias deben recibir todos los aeropuertos, se utiliza un exchange de tipo 'fanout'.

Diagrama Actividades (distancia)

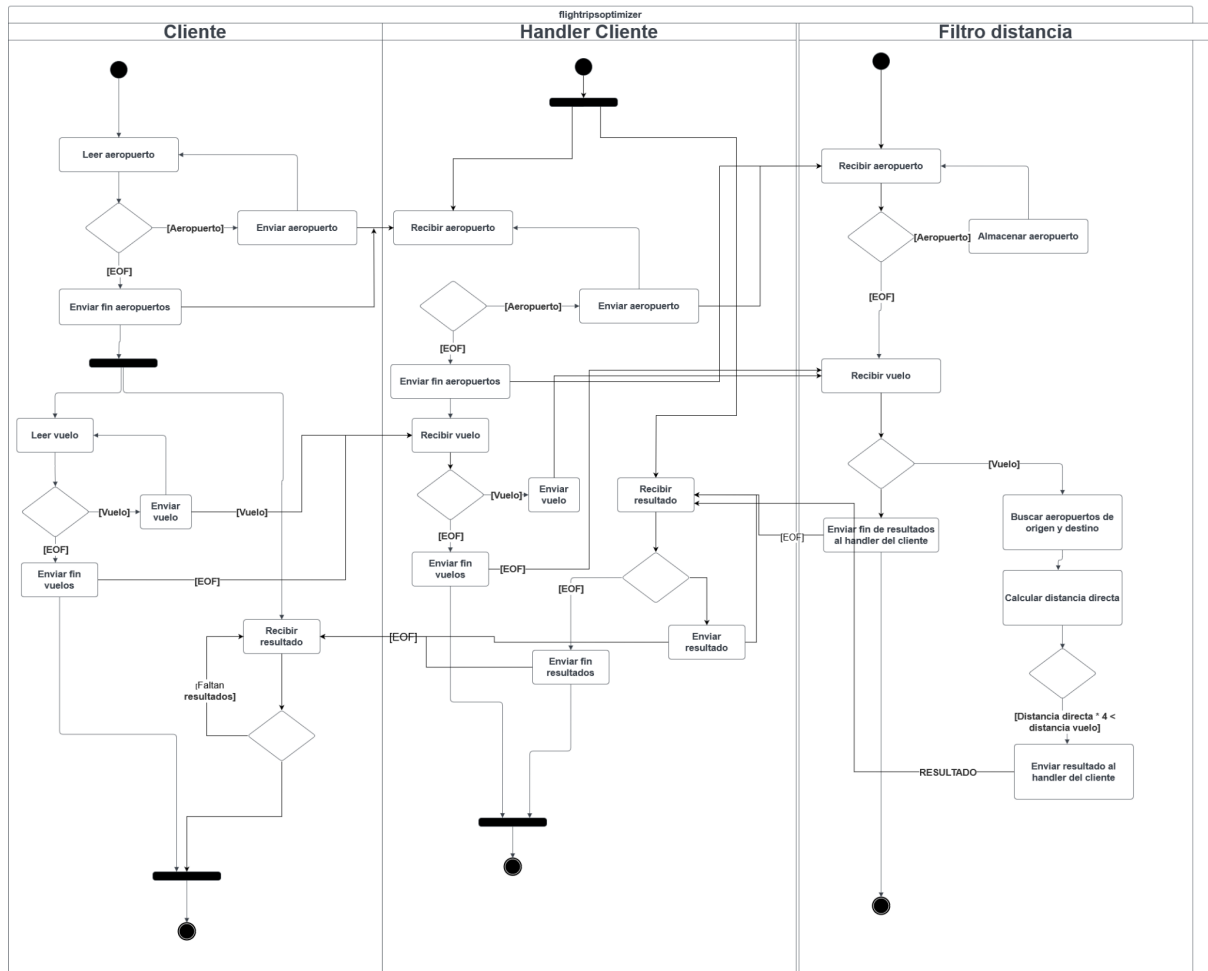


Fig 8. Diagrama de actividades filtro distancia

En la figura 9 se pueden ver las actividades del filtro de precios y el calculador de promedios. Cabe destacar que el filtro de precios utiliza la misma estrategia de *sharding* que el filtro de vuelos rápidos, es decir colas por tópicos basadas en un *hash* del trayecto.

Diagrama Actividades (precio)

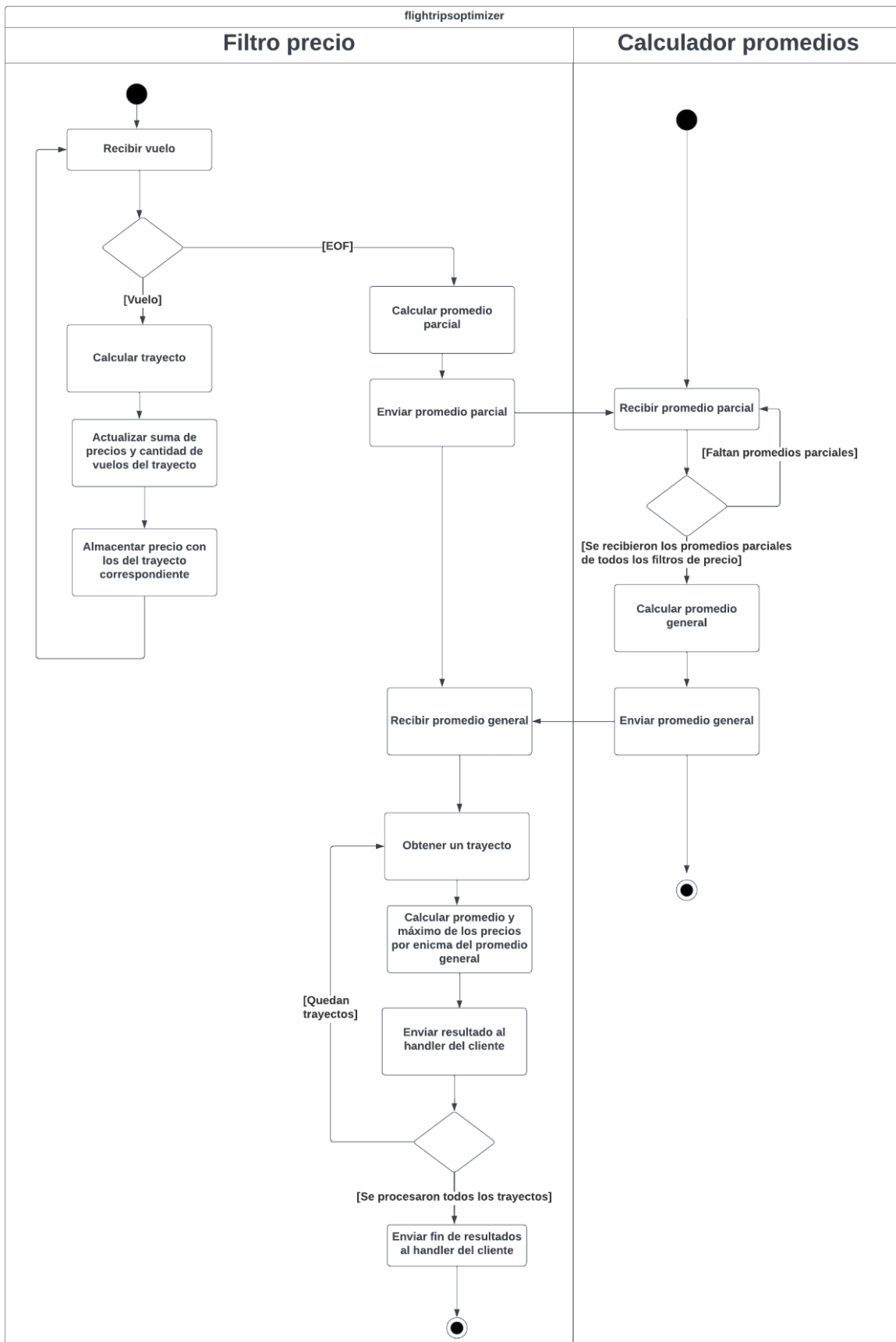


Fig 9. Diagrama de actividades filtro precio

Diagrama de secuencia

En el siguiente diagrama se muestra la secuencia que sigue el filtro de precios para calcular las estadísticas de los precios

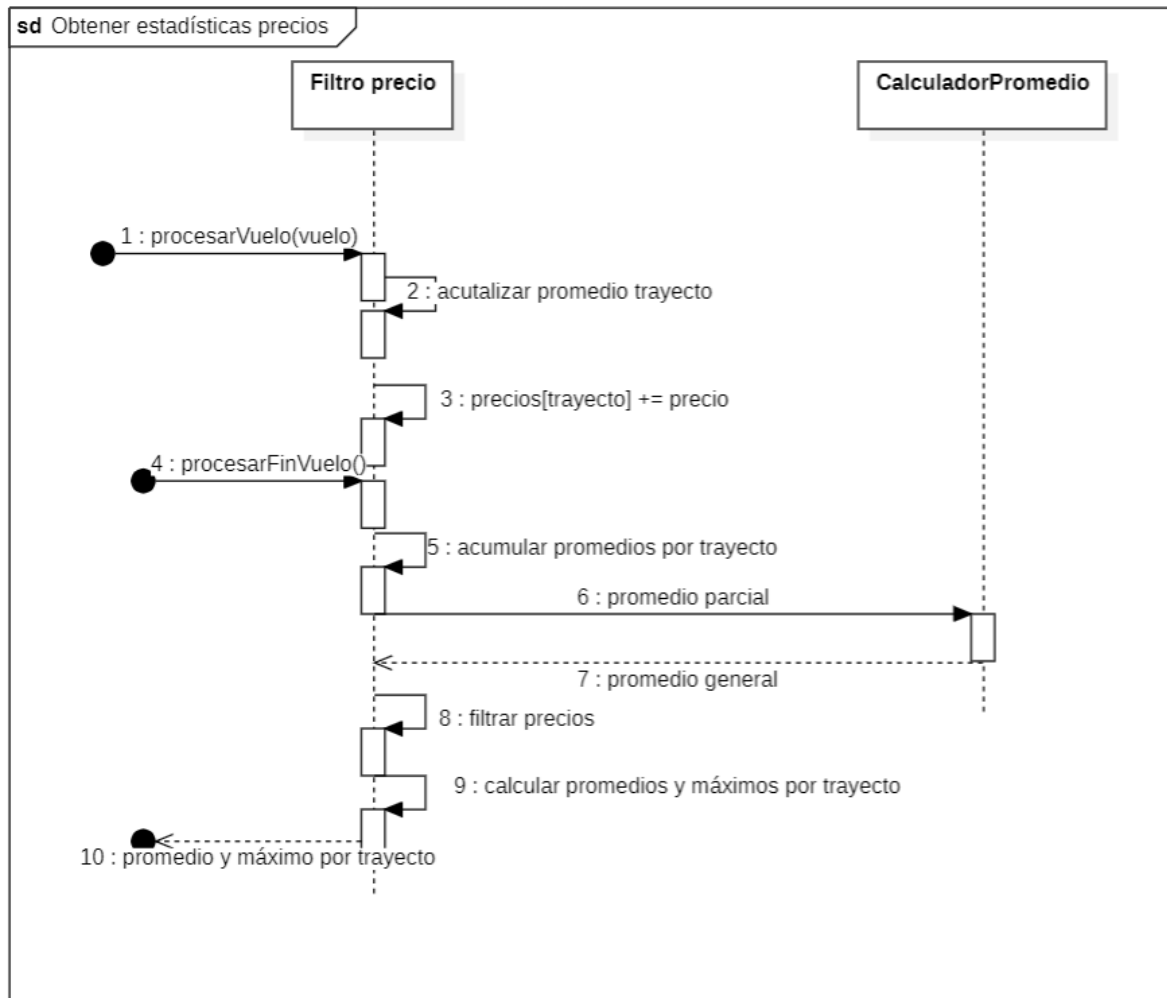


Fig 10. Diagrama de secuencia - Calcular estadísticas precios

Vista Lógica

Diagrama de clases

En la figura 11, se puede observar las clases del cliente, así como los protocolos que usará para comunicarse con el servidor.

Diagrama de clases -
Cliente

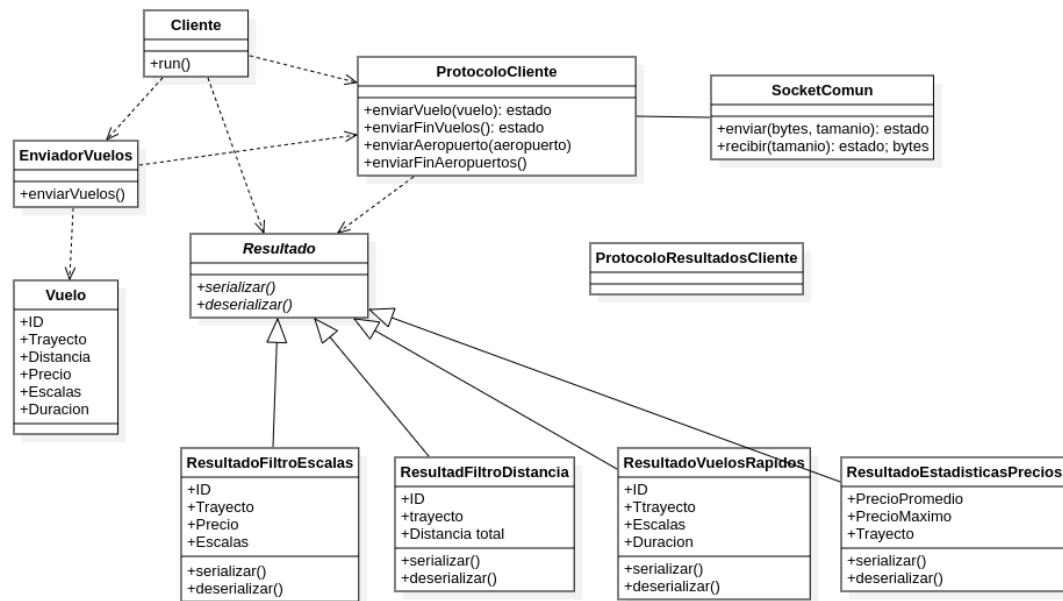


Fig 11. Diagrama de clases - Cliente

En la siguiente figura se muestran las clases en el servidor para el caso de uso correspondiente al filtro de los vuelos con más de 3 escalas. Los protocolos que se pueden observar en el diagrama corresponden al protocolo que usa para comunicarse con el cliente, ProtocoloCliente, y algunos de los protocolos que usarán los distintos nodos del servidor para comunicarse entre sí, que son ProtocoloResultados, ProtocoloFiltroEscalas y ProtocoloVuelosRapidos. Estos últimos se comunicaran usando un Message Oriented Middleware de colas.

Diagrama de clases -
Servidor (filtro de vuelos
con más de 3 escalas)

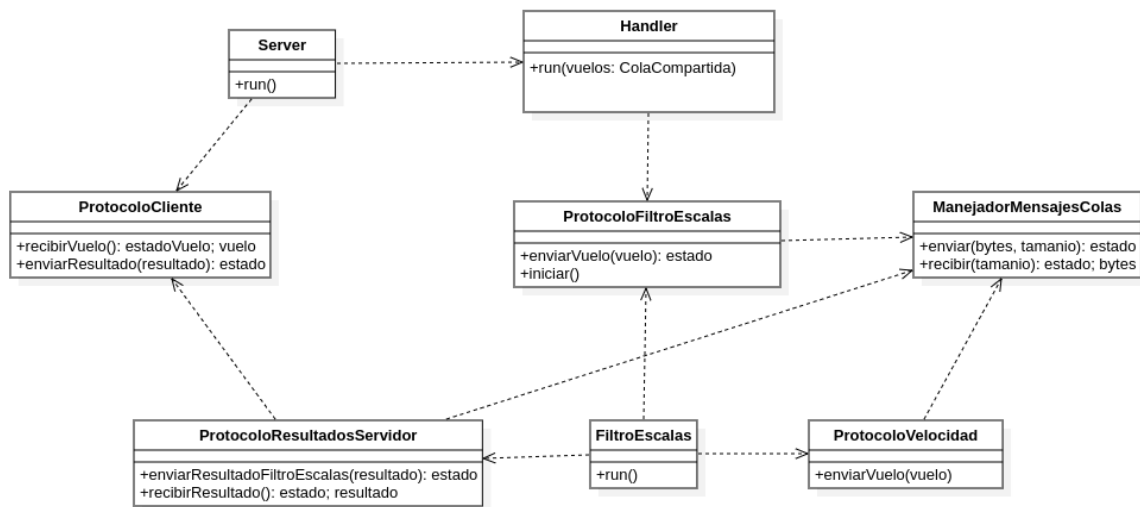


Fig 12. Diagrama de clases - Servidor (filtro de vuelos con más de 3 escalas)

Vista Protocolos

En esta sección se explican los distintos protocolos desarrollados para la comunicación entre nodos. Todos ellos son protocolos binarios.

Protocolos de comunicación entre el cliente y el servidor

Vuelos

Los mensajes para enviar un batch de vuelos inician con:

- Identificador de mensaje para batch de vuelos: un byte con un carácter que será la letra 'V'
- Tamaño del batch: *unsigned short* de 2 bytes

Luego se envía, para cada vuelo, su información en el siguiente formato:

- ID vuelo: *string* 32 bytes con *encoding* UTF-8
- Origen: *string* 3 bytes con *encoding* UTF-8
- Destino: *string* 3 bytes con *encoding* UTF-8
- Precio: *float* de 4 bytes
- Longitud string escalas: *unsigned short* de 2 bytes
- Escalas: *string* de 50 bytes. Para las escalas se usa el *encoding* UTF-8 y luego se agrega un *padding* de ceros hasta completar los 50 bytes
- Duración: *string* de 8 bytes. Se utiliza UTF-8 para el *encoding* del string de la duración, por ejemplo 'PT11H54M', y se completa con ceros al final, en caso de que la duración tenga menos de 8 bytes.
- Distancia: *short* de 2 bytes. Se completa con -1 en caso de que el vuelo no tenga distancia

El fin del envío de vuelos se indica enviando un mensaje con el identificador del fin de vuelos, que es un byte con la letra 'F'.

Aeropuertos

El formato de los mensajes para enviar los aeropuertos consiste en:

- Identificador de mensaje aeropuerto: un byte con un carácter que será la letra 'A'
- Tamaño del batch: *unsigned short* de 2 bytes
- ID aeropuerto: *string* 3 bytes con *encoding* UTF-8
- Latitud: *float* de 4 bytes
- Longitud: *float* de 4 bytes

Para indicar el fin del envío de aeropuertos, se envía un mensaje con el identificador del fin de aeropuertos, que consiste en un byte con la letra 'F'.

Resultados

Resultados escalas

Los mensajes para enviar los vuelos con 3 o más escalas consisten en:

- Identificador de resultado escalas: es un byte con la letra 'E' (mayúscula)
- Id vuelo: *string* 32 bytes con *encoding* UTF-8
- Trayecto: *string* 7 bytes con *encoding* UTF-8, que contiene el Origen y el Destino del vuelo. El formato es '<origen>-<destino>', por ejemplo 'BUE-MAD'.
- Precio: *float* de 4 bytes
- Longitud string escalas: *unsigned short* de 2 bytes
- Escalas: *string* de 50 bytes. Para las escalas se usa el *encoding* UTF-8 y luego se agrega un *padding* de ceros hasta completar los 50 bytes

El fin del envío de estos resultados se indica enviando al cliente un mensaje con byte con la letra 'e' (en minúscula).

Resultados filtro distancia

El formato de los mensajes los vuelos cuya distancia total sea mayor a cuatro veces la distancia directa entre puntos origen-destino es:

- Identificador de resultado distancia: es un byte con la letra 'D' (mayúscula)
- Id vuelo: *string* 32 bytes con *encoding* UTF-8
- Trayecto: *string* 7 bytes con *encoding* UTF-8, que contiene el Origen y el Destino del vuelo. El formato es '<origen>-<destino>', por ejemplo 'BUE-MAD'.
- Distancia: *unsigned short* de 2 bytes.

El fin del envío de estos resultados se indica enviando al cliente un mensaje con byte con la letra 'd' (en minúscula).

Resultados filtro distancia

El formato de los mensajes los vuelos cuya distancia total sea mayor a cuatro veces la distancia directa entre puntos origen-destino es:

- Identificador de resultado distancia: es un byte con la letra 'D' (mayúscula)
- Id vuelo: *string* 32 bytes con *encoding* UTF-8
- Trayecto: *string* 7 bytes con *encoding* UTF-8, que contiene el Origen y el Destino del vuelo. El formato es '<origen>-<destino>', por ejemplo 'BUE-MAD'.

- Distancia: *unsigned short* de 2 bytes.

El fin del envío de estos resultados se indica enviando al cliente un mensaje con byte con la letra 'd' (en minúscula).

Resultados vuelos rápidos

Los mensajes para enviar los 2 vuelos más rápidos por trayecto, entre todos los vuelos de 3 escalas o más son:

- Identificador de resultado vuelos rápidos: es un byte con la letra 'R' (mayúscula)
- Id vuelo: *string* 32 bytes con *encoding* UTF-8
- Trayecto: *string* 7 bytes con *encoding* UTF-8, que contiene el Origen y el Destino del vuelo. El formato es '<origen>-<destino>', por ejemplo 'BUE-MAD'.
- Longitud string escalas: *unsigned short* de 2 bytes
- Escalas: *string* de 50 bytes. Para las escalas se usa el *encoding* UTF-8 y luego se agrega un *padding* de ceros hasta completar los 50 bytes
- Duración: *string* de 8 bytes. Se utiliza UTF-8 para el *encoding* del string de la duración. Se completa con ceros al final, en caso de que la duración tenga menos de 8 bytes.

El fin del envío de estos resultados se indica enviando al cliente un mensaje con byte con la letra 'r' (en minúscula).

Resultados estadísticas precios

Los mensajes para para enviar el precio promedio y máximo por trayecto de los vuelos con precio mayor a la media general de precios consisten en:

- Identificador de resultado precios: es un byte con la letra 'P' (mayúscula)
- Trayecto: *string* 7 bytes con *encoding* UTF-8, que contiene el Origen y el Destino del vuelo. El formato es '<origen>-<destino>'..
- Precio promedio: *float* de 4 bytes
- Precio máximo: *float* de 4 bytes

El fin del envío de estos resultados se indica enviando al cliente un mensaje con byte con la letra 'p' (en minúscula).

Protocolos de comunicación entre el handler del cliente y los filtros

Envío de vuelos al filtro de escalas

Los mensajes que se usan para enviar los vuelos al filtro de escalas contienen:

- Identificador de mensaje para vuelo: un byte con un carácter que será la letra 'V'
- Tamaño del batch: *unsigned short* de 2 bytes
- ID vuelo: *string* 32 bytes con *encoding* UTF-8
- Origen: *string* 3 bytes con *encoding* UTF-8
- Destino: *string* 3 bytes con *encoding* UTF-8
- Escalas: *string* de 50 bytes. Para las escalas se usa el *encoding* UTF-8 y luego se agrega un *padding* de ceros hasta completar los 50 bytes
- Duración: *string* de 8 bytes. Se utiliza UTF-8 para el *encoding* del string de la duración, y se completa con ceros al final, en caso de que la duración tenga menos de 8 bytes.

El fin del envío de los vuelos al filtro de escalas se indica con un mensaje con byte con la letra 'F'.

Envío de aeropuertos al filtro de distancias

Los mensajes que se usan para enviar los aeropuertos, y el fin de los mismos, al filtro de distancias tienen el mismo formato que los que usa el cliente para enviarlos al servidor.

Envío de vuelos al filtro de distancias

Los mensajes que se usan para enviar los vuelos al filtro de distancias contienen:

- Identificador de mensaje para vuelo: un byte con un carácter que será la letra 'V'
- Tamaño del batch: *unsigned short* de 2 bytes
- ID vuelo: *string* 32 bytes con *encoding* UTF-8
- Origen: *string* 3 bytes con *encoding* UTF-8
- Destino: *string* 3 bytes con *encoding* UTF-8
- Distancia: *short* de 2 bytes. Se completa con -1 en caso de que el vuelo no tenga distancia

El fin del envío de los vuelos al filtro se indica con un mensaje con byte con la letra 'F'.

Envío de vuelos al filtro de vuelos rápidos

Los mensajes que se usan para enviar los vuelos al filtro de vuelos rápidos contienen:

- Identificador de mensaje para vuelo: un byte con un carácter que será la letra 'V'
- Tamaño del batch: *unsigned short* de 2 bytes
- ID vuelo: *string* 32 bytes con *encoding* UTF-8
- Origen: *string* 3 bytes con *encoding* UTF-8
- Destino: *string* 3 bytes con *encoding* UTF-8
- Duración: *string* de 8 bytes. Se utiliza UTF-8 para el *encoding* del string de la duración, por ejemplo 'PT11H54M', y se completa con ceros al final, en caso de que la duración tenga menos de 8 bytes.

El fin del envío de los vuelos al filtro de vuelos rápidos se indica con un mensaje con byte con la letra 'F'.

Envío de vuelos al filtro de estadísticas de precios

Los mensajes que se usan para enviar los vuelos al filtro de estadísticas de precios:

- Identificador de mensaje para vuelo: un byte con un carácter que será la letra 'V'
- Tamaño del batch: *unsigned short* de 2 bytes
- ID vuelo: *string* 32 bytes con *encoding* UTF-8
- Origen: *string* 3 bytes con *encoding* UTF-8
- Destino: *string* 3 bytes con *encoding* UTF-8
- Precio: *float* de 4 bytes

El fin del envío de los vuelos al filtro se indica con un mensaje con byte con la letra 'F'.

Resultados

Los mensajes para enviar los resultados, y el fin de los mismos, entre el servidor y los filtros tienen el mismo formato que los que se usan para enviarlos del servidor al cliente.

Protocolo de comunicación entre el filtro de estadísticas de precios y el calculador de promedios

Cada uno de los filtros de estadísticas de precios, enviará su promedio parcial en un mensaje con el siguiente formato:

- Promedio parcial: *float* de 4 bytes

- Cantidad de vuelos: *int* de 4 bytes

Una vez que hayan recibido todos los promedios parciales, el calculador de promedios le enviará, a cada uno de los filtros de precios, el promedio general en un *float* de 4 bytes

Listado de tareas a ejecutar y división entre integrantes

Las tareas a ejecutar para la configuración del sistema son:

Nombre tarea	Integrante que la desarrollará
Desarrollo Middleware	Guido Bergman
Configuración Docker y Docker-Compose	Guido Bergman
Implementación protocolo	Guido Bergman y Luis Waldman

La división de las tareas para las otras partes del sistema será:

Parte del sistema	Integrante que lo implementará
Cliente	Guido Bergman
Hander Cliente	Guido Bergman
Buscador de vuelos con distancias largas	Guido Bergman
Filtro escalas	Luis Waldman
Buscador vuelos rápidos	Luis Waldman
Calculador estadísticas vuelos costosos	Luis Waldman