# Heurísticas para optimización de Aggregation Pipelines en MongoDB

**Grupo 14**

Santiago Goycoechea
Maximiliano Barragán

# Agenda

- Objetivos del proyecto
- Trabajos relacionados
- Heurísticas
- Experimentación
- Conclusiones y trabajo a futuro

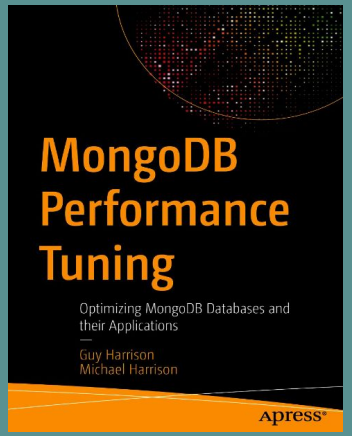
# Objetivos del proyecto

- Relevamiento del estado del arte sobre el tema, búsqueda de heurísticas existentes
- Desarrollo de nuevas heurísticas
- Evaluación en base a experimentos de las heurísticas relevadas y desarrolladas

# Trabajos relacionados

No hay muchos resultados sobre el tema, más allá de la documentación oficial de MongoDB

Nuestro proyecto se basa en el libro ***MongoDB Performance Tuning*** de Guy Harrison y Michael Harrison



# Heurísticas relevadas

- Filtrar de manera temprana y frecuente
- Utilizar índices en los foreignField de los $lookup
- Hacer los $lookup desde la colección más pequeña
- Hacer los $sort que utilizan un índice lo antes posible

# Ideas nuestras

- $unwind vs. $project
- Tipo del campo fecha: string vs. date

# Experimentación

# Data set

**Catálogo de Datos Abiertos de la IdM:**
Viajes realizados en los ómnibus del Sistema de Transporte Metropolitano

Es un conjunto de CSVs que pueden ser interpretados como una BD desnormalizada

| | |
|---|---|
| id_viaje | identifica de forma unica a un viaje dentro del mes. Definiendo por viaje, todos los tramos realizados por el/los pasajeros con un unico pago |
| con_tarjeta | 0 = viajes sin tarjeta, 1= viajes con tarjeta |
| fecha_evento | fecha hora en la cual se registra el tramo del viaje |
| tipo_viaje | codigo del tipo de viaje considerando el tipo de viaje y el grupo de usuario (ej, un estudiante A por defecto realiza un viaje 1 Hora, el cual se considera como un viaje EST. A) |
| descripcion_tipo_viaje | descripcion del tipo de viaje |
| grupo_usuario | codigo del grupo de usuario |
| descripcion_grupo_usuario | descripcion del grupo de usuario (ej Estudiante, Jubilado, Usuario Corriente) |
| grupo_usuario_especifico | codigo del subgrupo dentro del grupo de usuario |
| descripcion_grupo_usuario_espe | descripcion grupo usuario especifico (ej Estudiante A, Jubilado B) |
| ordinal_de_tramo | para viajes con tarjeta, ordinal del tramo dentro del viaje |
| cantidad_pasajeros | cantidad de personas que realizan el tramo |
| codigo_parada_origen | codigo de la parada de ascenso |
| cod_empresa | codigo de empresa a la cual pertenece el omnibus |
| descrip_empresa | descripcion de la empresa transportista |
| linea_codigo | codigo de la linea |
| dsc_linea | nombre publico de la linea |
| sevar_codigo | codigo de la variante |

Columnas de los CSVs

# Bases de datos

Tres tamaños de BD diferentes: ~**2.000.000**, ~**500.000** y ~**50.000** documentos

| Colección | Documentos | Tamaño |
|---|---|---|
| viajes | 2.000.000 | 564.4 MB |
| empresas | 4 | 297 B |
| lineas | 132 | 10.6 KB |
| tipos_viajes | 14 | 1.0 KB |
| grupos_usuarios | 8 | 737 B |
| grupos_usuarios_especificos | 21 | 3.8 KB |

Colecciones de la BD más grande

# Metodología

- Para cada heurística, desarrollar dos consultas equivalentes: una que la aplique y una que no

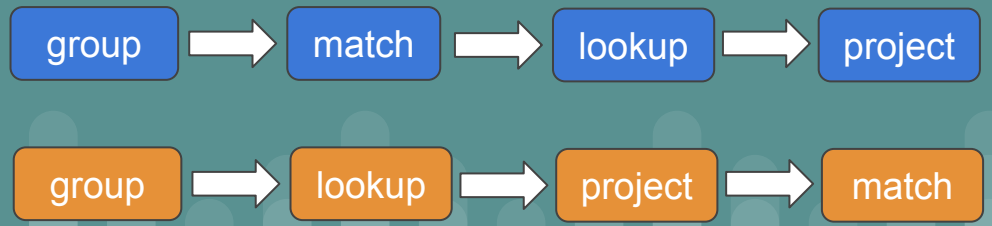
- Ejecutarlas 10 veces en cada una de las 3 bases definidas, midiendo los tiempos de ejecución

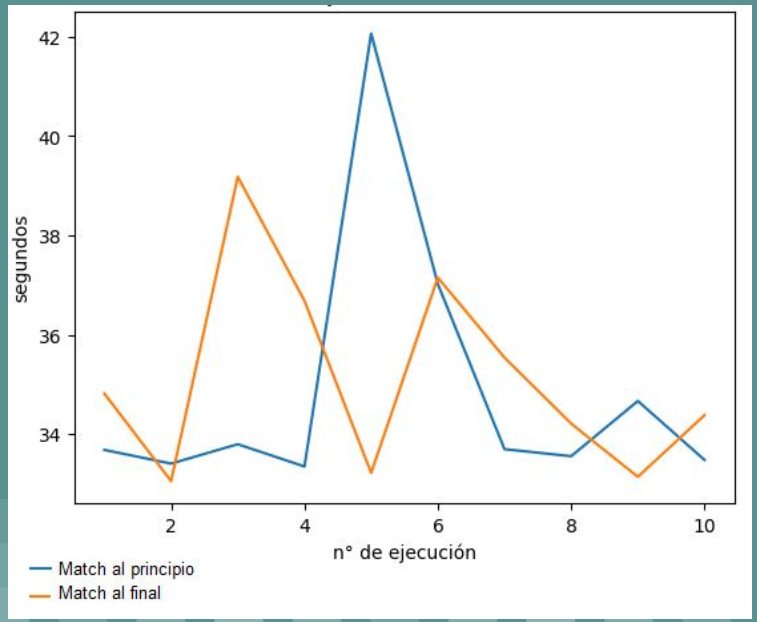
# Filtrar de manera temprana y frecuente

**Consulta**: Viajes con más de un tramo

La evaluamos filtrando con **$match**

Los resultados no fueron concluyentes



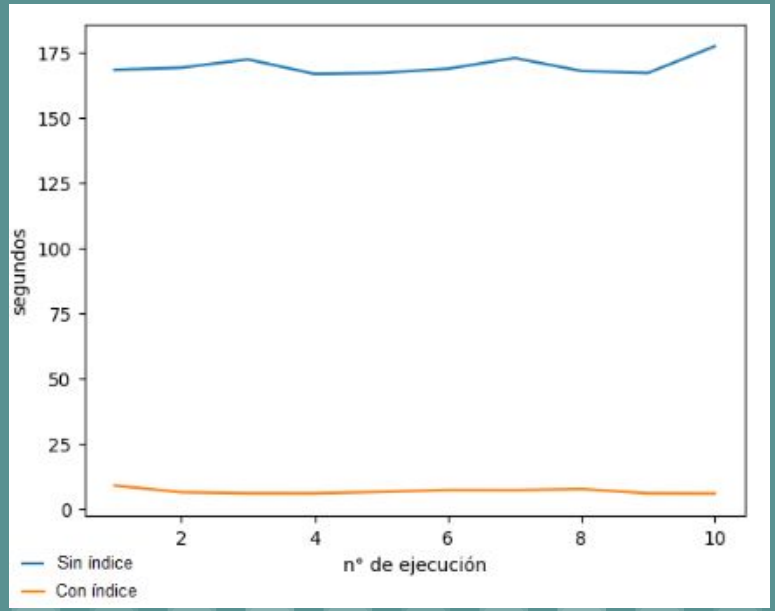



# Utilizar índices en los foreignField de los $lookup

**Consulta**: Líneas con más viajes

En este caso los resultados sí fueron buenos

Sin índice  Con índice

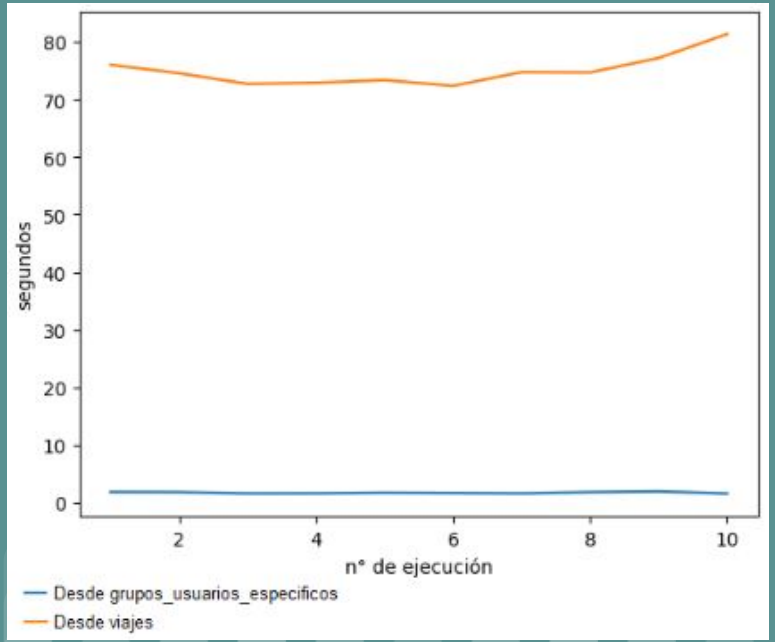



# Hacer los $lookup desde la colección más pequeña

**Consulta**: Paradas donde suben más estudiantes tipo A

Esta heurística también funcionó bien

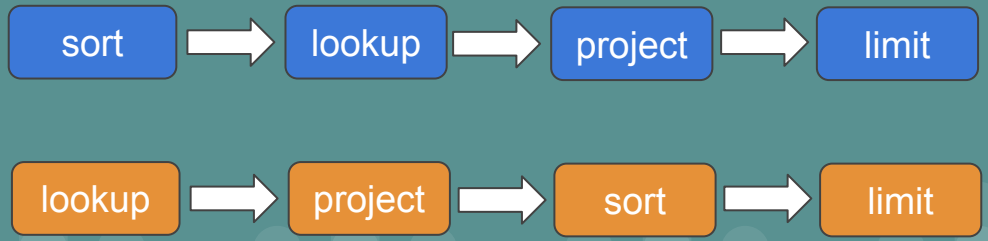
- Desde una colección de 21 de documentos
- Desde una colección de 2.000.000 de documentos

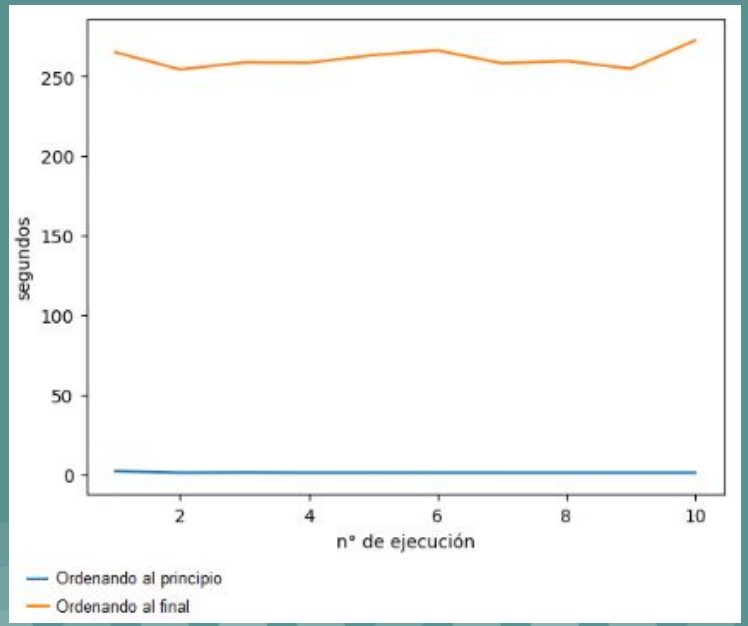


# Hacer los $sort que utilizan un índice lo antes posible

**Consulta**: Últimas líneas con viajes

Esta heurística fue la más efectiva



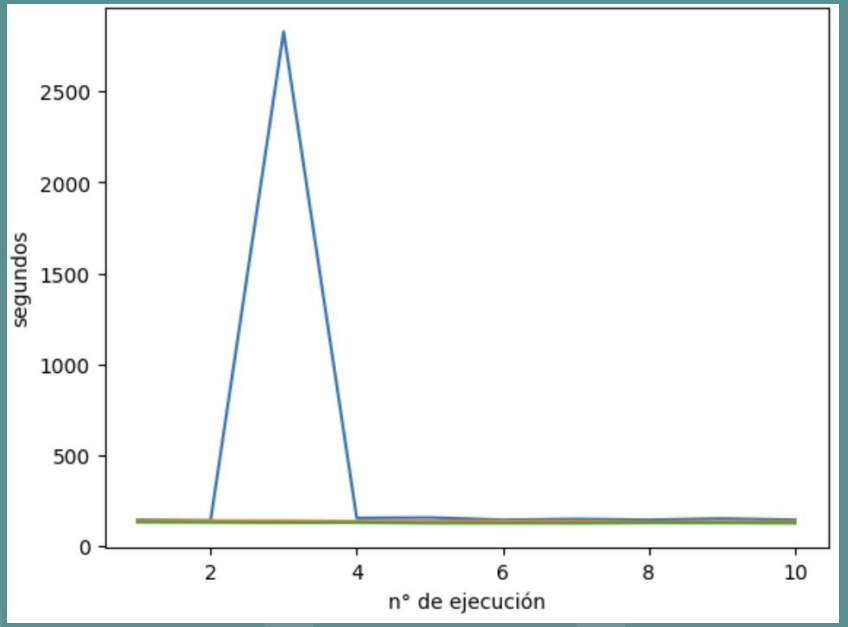
# $unwind vs. $project

**Consulta**: Cantidad total de viajes realizados por cada grupo de usuario

Consideramos dos **$project** distintos

El **$unwind** resultó ser más eficiente, pero no por mucho...





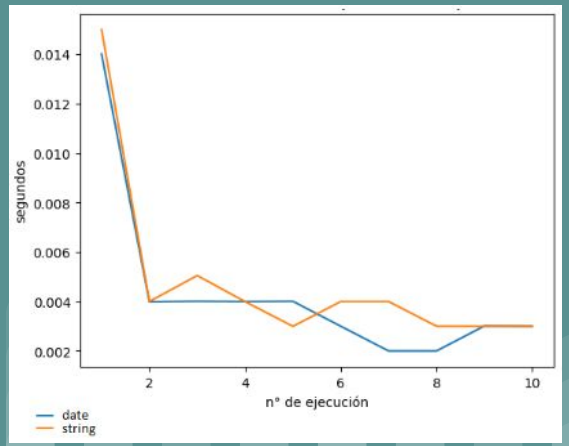
# Tipo del campo “fecha”

**Consulta**: Últimas líneas con viajes

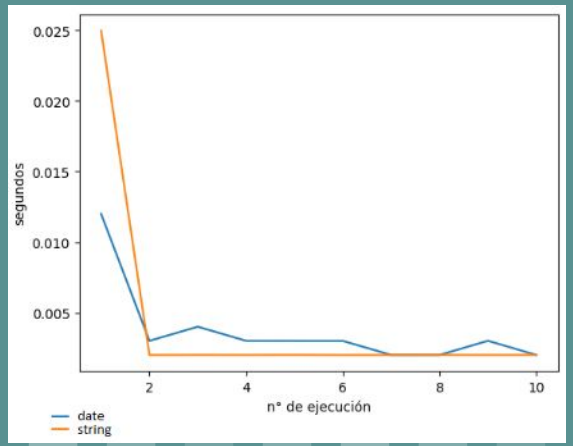
Los resultados son inconclusos en términos de tiempo de ejecución...

Date
String

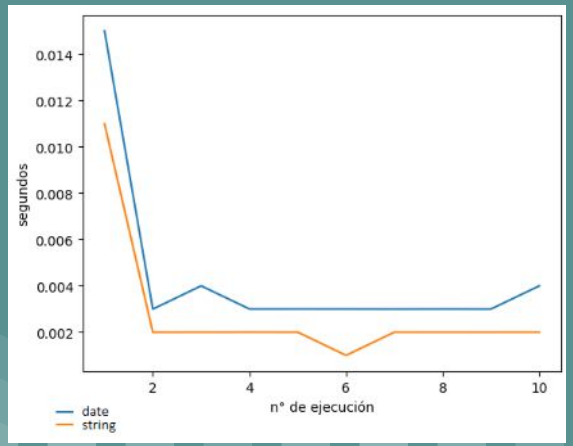
2.000.000 de documentos




500.000 documentos

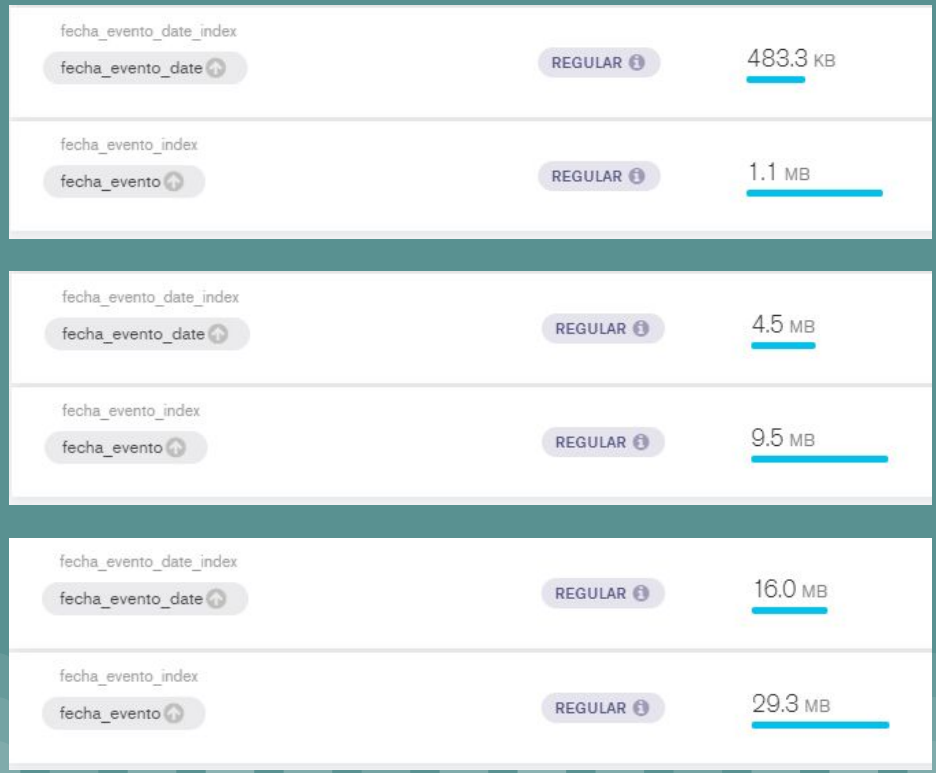



50.000 documentos



# Tipo del campo “fecha”

Pero usar el tipo **Date**...
ocupa menos espacio



# Conclusiones

- “Filtrar de manera temprana y frecuente” con $match en lugar de $limit no supone un gran cambio, por optimizaciones automáticas del manejador

- Las otras tres heurísticas recopiladas del libro mejoran el desempeño de las consultas

# Conclusiones

- En caso de arreglos de un solo elemento, parece preferible usar $unwind por sobre $project
  - + Mejores tiempos
  - + Más sencilla
  - - Intuitiva

- Dependiendo las características de la base y del sistema, nos puede convenir definir las fechas como string o como date

# Trabajo a futuro

- Profundizar más en algunas de las heurísticas mencionadas en el libro que no fueron abordadas en este trabajo

- Realizar más experimentos combinando estas heurísticas para ver de qué manera interactúan entre ellas

- Realizar experimentos con otros pasos del pipeline

# ¡Muchas gracias!

## ¿Preguntas?