Proyecto DB Películas

PALOMITAS

Pequeño ..... 5.00 €
Mediana ..... 10.00 €
Grande ..... 15.00 €

BEBIDAS

Pequeño ..... 5.00 €
Mediana ..... 10.00 €
Grande ..... 15.00 €

COMBO

Pequeño ..... 5.00 €
Mediana ..... 10.00 €
Grande ..... 15.00 €

Proyecto Base de Datos NoSQL
Adrián Ramos Colón
1º DAM

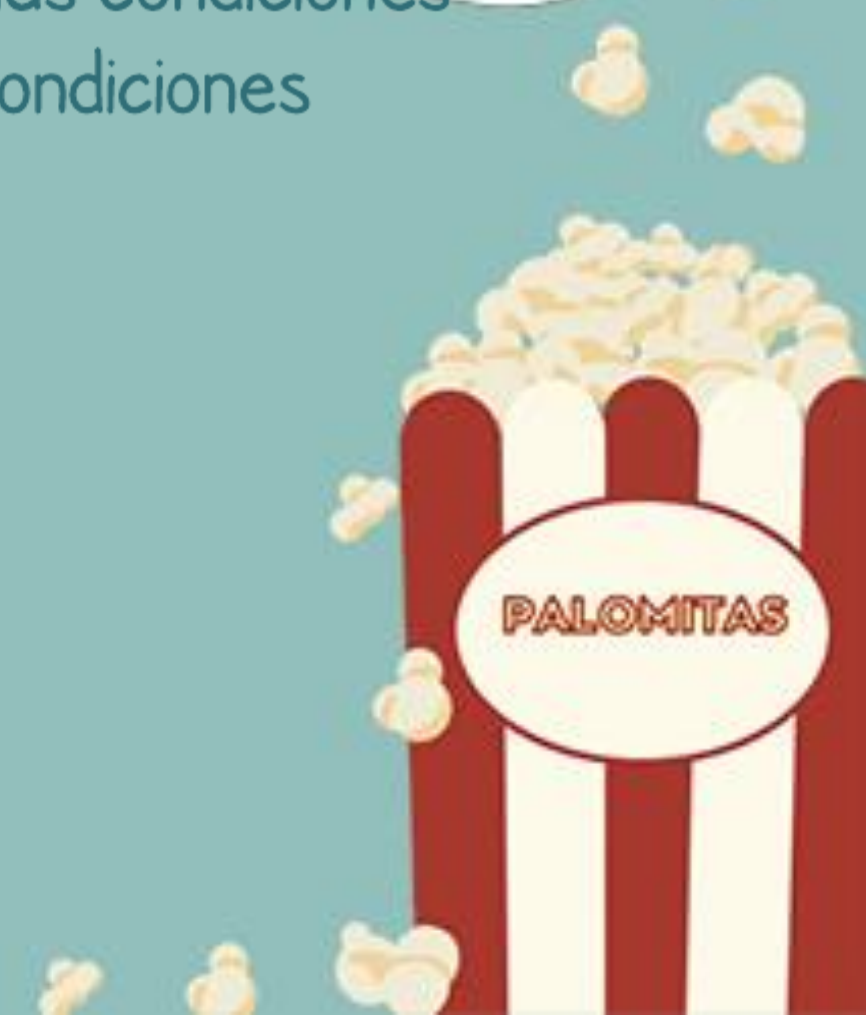
Operadores

Operadores relacionales

- $eq ⇝ Igual
- $ne ⇝ Diferente
- $gt ⇝ Mayor que
- $gte ⇝ Mayor que (estricto)
- $lt ⇝ Menor que (estricto)
- $lte ⇝ Menor que
- $in ⇝ Contiene
- $nin ⇝ No contiene

Operadores relacionales

- $and ⇝ Cumple alguna del las condiciones
- $or ⇝ Cumple todas las condiciones



Aggregate

Aggregate

Las agregaciones procesan varios documentos en pasos correlativos. Estos son los 5 steps más comunes:

- $group: Agrupa documentos y añade cuantificadores
- $match: Filtrado de documentos igual que las queries
- $lookup: Busca en otras colecciones
- $proyect: Deja, o no, visible un set de atributos
- $unwind: Desmonta Arrays

Index

Los índices admiten la ejecución eficiente de consultas. Sin índices, MongoDB debe ralizar un escaneo de la colección completa, es dedir,escanear cada documento en una colección, para seleccionar aquellos documentos que coincidan con la declaración de la consulta

Map-Reduce

Map-Reduce

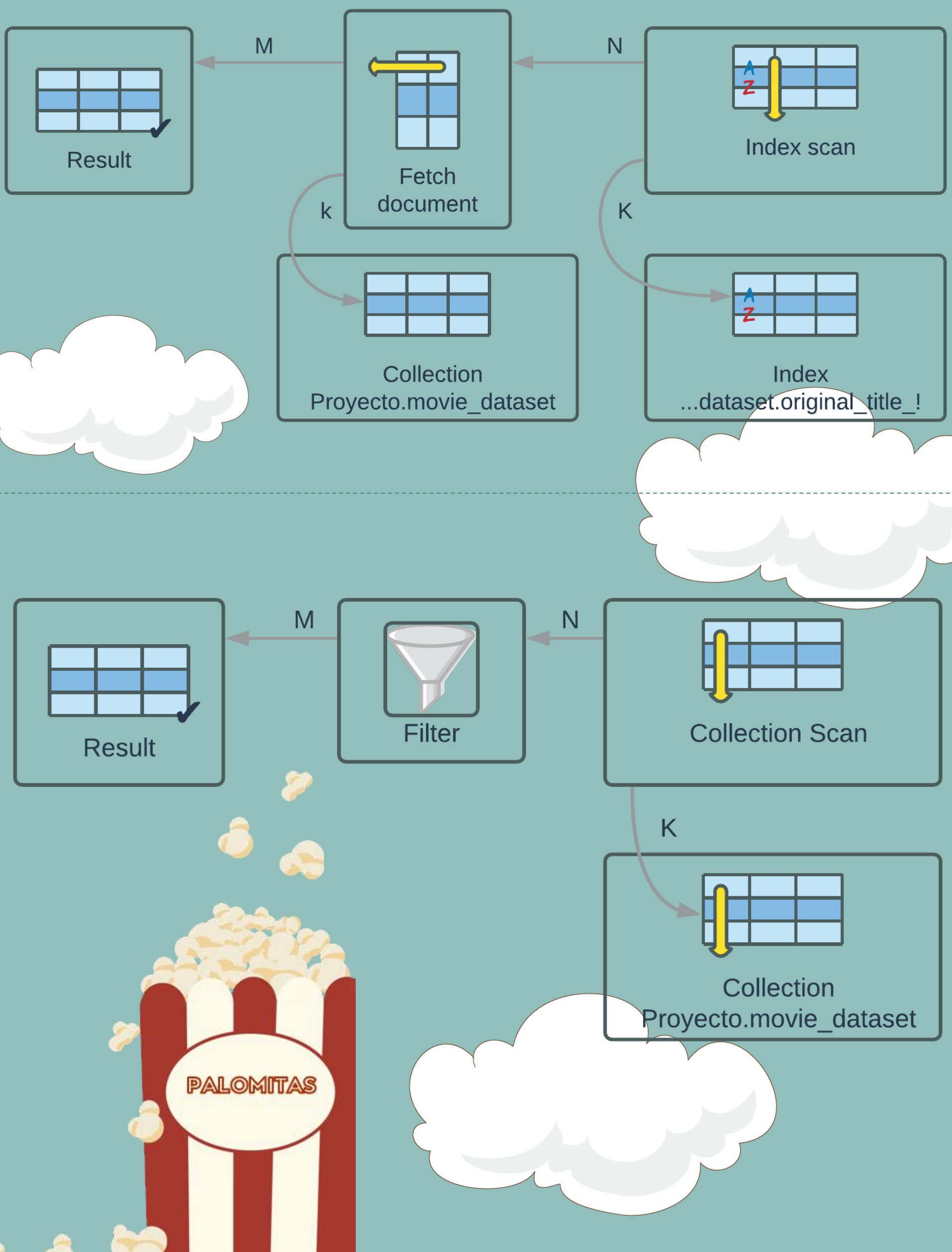
Map-reduce es un paradigma de procesamiento de datos para condensar grandes volúmenes de datos enresultados agregados útiles

- Map Reduce permite realizar acciones sobre datos de forma paralela y altamente escalable
- Es un modelo de progrmacion que consta de dos pasos: Map y Reduce
- Map: La informacion completa se separa en bloques, cada uno con una clave. Se realizan alguna transformacion y se envía cada bloque al paso de Reduce
- Reduce: Un Reduce toma los datos que salen del paso Map (que tengan la misma clave) y los procesa

Las agregaciones son más rapidas porque usan c / c++ (como el copdigo core de mongo) y los map-reduce usan JSON, por lo que tienene que connvertir cada archivo interno BSON a JSON para poder operar con ellos

```
{
    _id: Identificador único MongoDB,
    adult: Adultos si/no,
    belongs to collection: {
        id: Id de colección,
        name:  Nombre colección
    },
    budget: Presupuesto,
    genres:[
        {
            id: Identificador de género,
            name: Nombre del género,
        }
    ],
    id: Identificador unitivo,
    original_language: Abreviatura del idioma,
    original_title: Nombre original,
    popularity: Popularidad,
    production_companies:[
        {
            name: Nombre compañía,
            id: Identificador compañía,
        }
    ],
    revenue: Ingresos,
    runtime: Duración min,
    spoken_languages:[
        {
            iso_639_1: Identificador idiomas,
            name: Nombre idioma,
        }
    ],
    status: Estado, lanzada o no,
    tagline: Marketing,
    title: Titulo,
    vote_average: Promedio de votos,
    vote_count: Número de votos,
    releasedate: Fecha lanzamiento
}
```

Explain


Result
M
Fetch document
N
Index scan
k
K
Collection Proyecto.movie_dataset
Index ...dataset.original_title_!
Result
M
Filter
N
Collection Scan
K
Collection Proyecto.movie_dataset
PALOMITAS

Redundancia

Redundancia

En las bases de datos SQL la consistencia de datos es asegurarse de que un único dato esté una única vez en toda la base de datos; y se suele lograr con el proceso de "Normalización".

En cambio en las bases de datos noSQL como es MongoDB la redundancia esrepetir adrede los datos a conveniencia en varias partes de la BD (datos "de-normalizados").

Ejemplo: almacenamos datos de una reserva de hotel, guardamos todos los datos de una persona en la entidad "Persona". Pero además, guardamos una copia del nombre, teléfono y demás información personal en cada "Reservación" y posiblemente en cada "Factura" de esta persona.

ReplicaSet

ReplicaSet

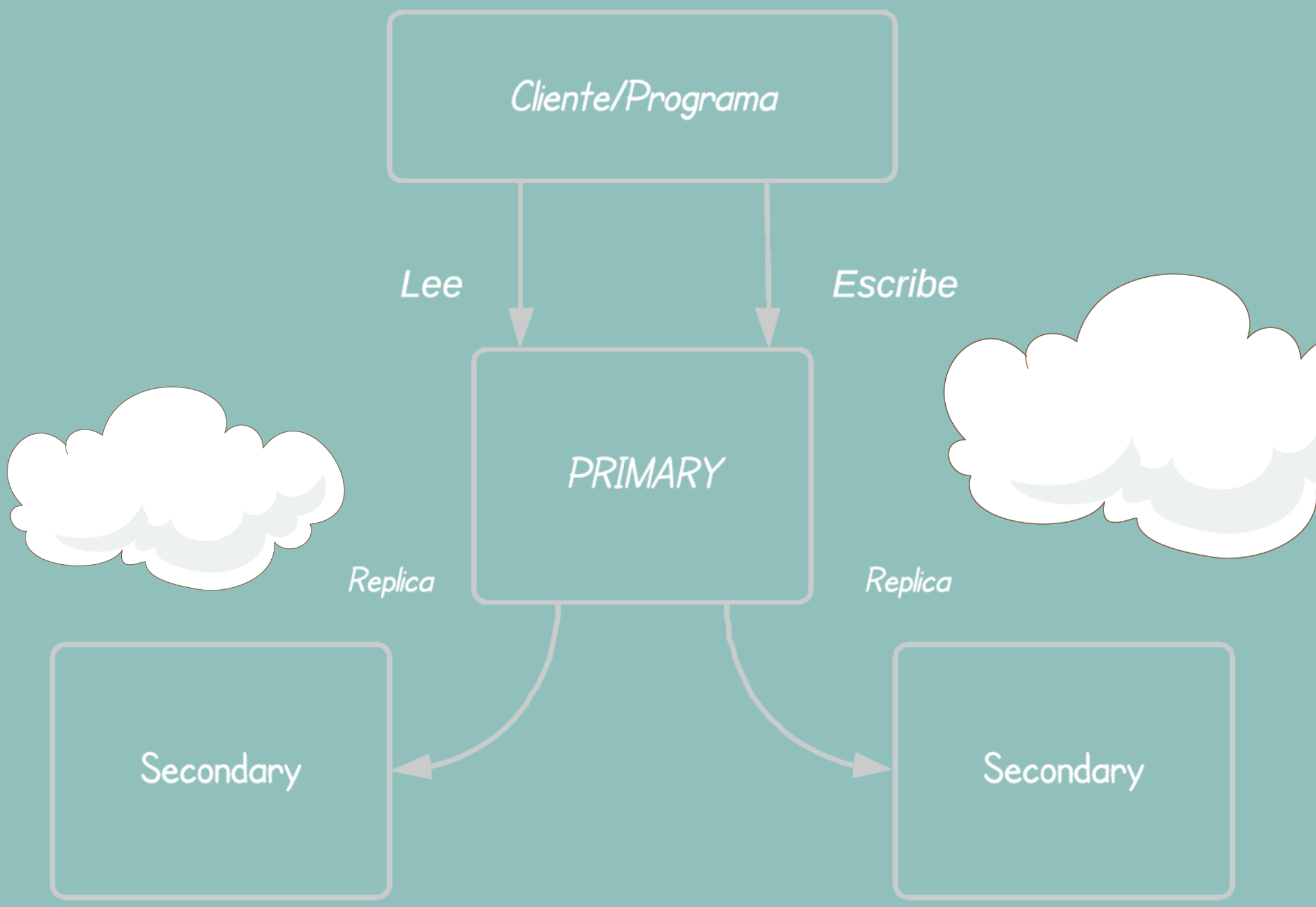
La replicación proporciona redundancia y aumentala disponibilidad de datos.

Con múltiples copias de datos en diferentes servidores de bases de datos, la replicación proporciona un nivel de tolerancia a fallas contra la pérdida de un solo servidor de bases de datos.

Basicamente consiste en la creacion de "n" servidores secundarios, cada uno con los mismos datos dentro de la instancia de mongo. La recomendación es tener cada instancia corriendo en máquinas diferentes para evitar fallas o perdidas.

En un replicaset existen dos roles: Primario y secundario. A primera instancia, los cambios se efectuarán en el replicaset primario y con un lag negligible, se replicará en los replicasets secundarios.

ReplicaSet


Cliente/Programa
Lee
Escribe
PRIMARY
Replica
Replica
Secondary
Secondary


PALOMITAS

Sharding

Sharding

El sharding es un método para la distribución de datos en varias máquinas.MongoDB utiliza el sharding para admitir implementaciones con conjuntos de datos muy grandes y operaciones de alto rendimiento.

- Distribución de los documentos de una colección en diferentes servidores
- *Cada shard puede ser o bien una instancia de mongo o un replicaset*
- *Busqueda de una buena sharding key -> como las funciones de hash para repartir bien los datos*

FIN

PALOMITAS

Pequeño ..... 5.00 €
Mediana ..... 10.00 €
Grande ..... 15.00 €

BEBIDAS

Pequeño ..... 5.00 €
Mediana ..... 10.00 €
Grande ..... 15.00 €

COMBO

Pequeño ..... 5.00 €
Mediana ..... 10.00 €
Grande ..... 15.00 €

1º DAM
Adrián Ramos Colón
Proyecto Base de Datos NoSQL