

Análisis de Datos con MongoDB

Alejandro Mancilla

Sr. Solutions Architect, LATAM

alejandro@mongodb.com

@alxmancilla



Agenda

- ¿Por qué hacer análisis con MongoDB?
- Crear ambiente de MongoDB
- Ingestión de datos en MongoDB (`mongoimport` / `mongorestore`)
- Exploración de datos (MongoDB Compass)
- Análisis de datos con MongoDB (Aggregation Framework)
- Extendiendo análisis con Python (`pymongo`)
- Integración con herramientas de BI (BI Connector)
- Extendiendo análisis con Apache Spark

Repositorio de GitHub: <https://github.com/alxmancilla/DataDayMX>

Análisis de Datos como Arte



La ciencia es conocimiento que entendemos tan bien que podemos enseñarlo a una computadora”.

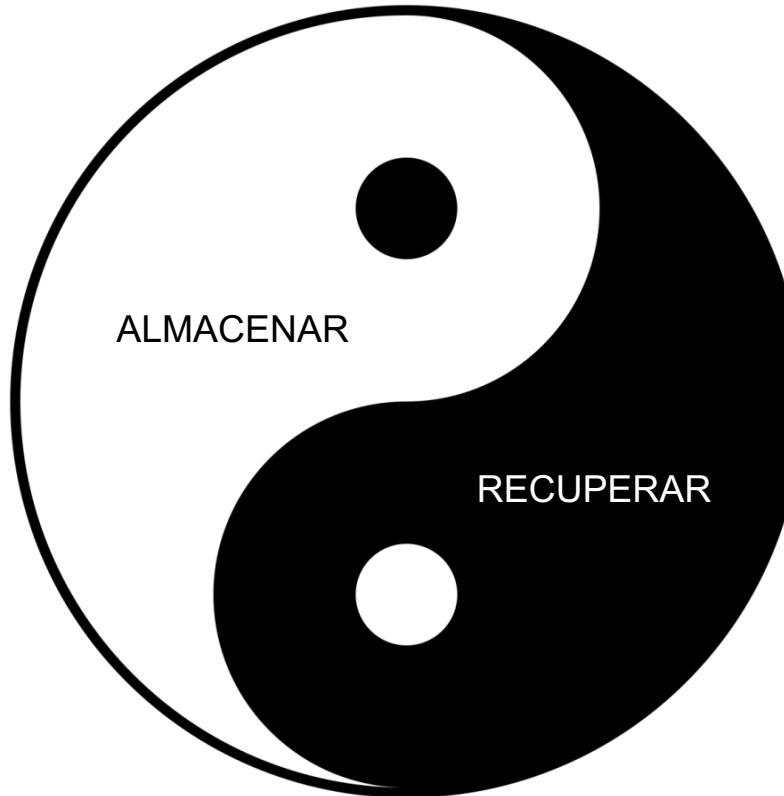
Donald Knuth



¿Por qué hacer análisis con MongoDB?

Aplicaciones & Datos

Consultar & Análisis



`find()` & `aggregate()`

Opciones para Analítica (1/2)

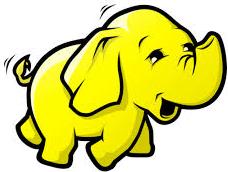


Pre-agregar

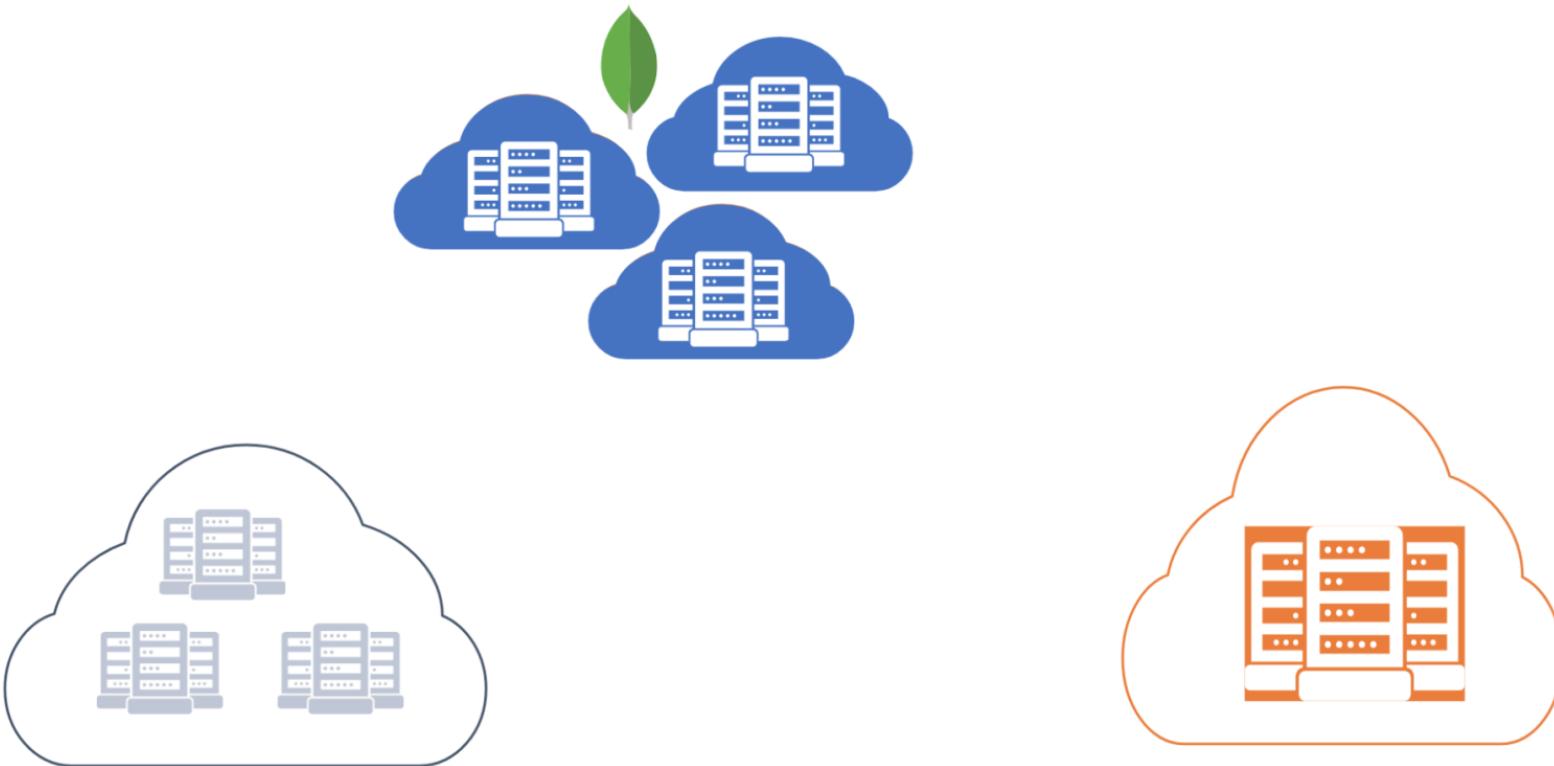


Agregar en otro lugar

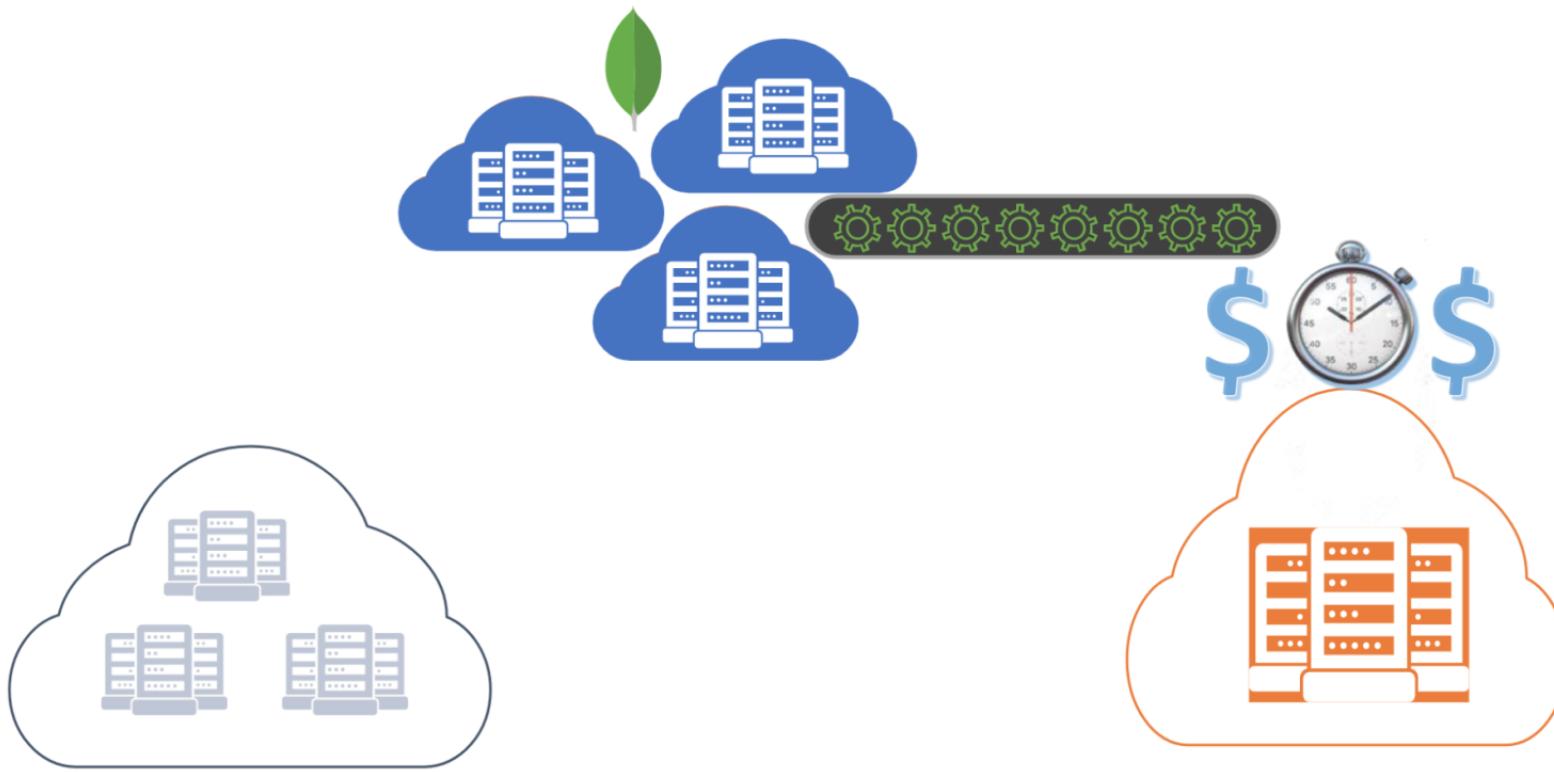
Analizar en otro lugar



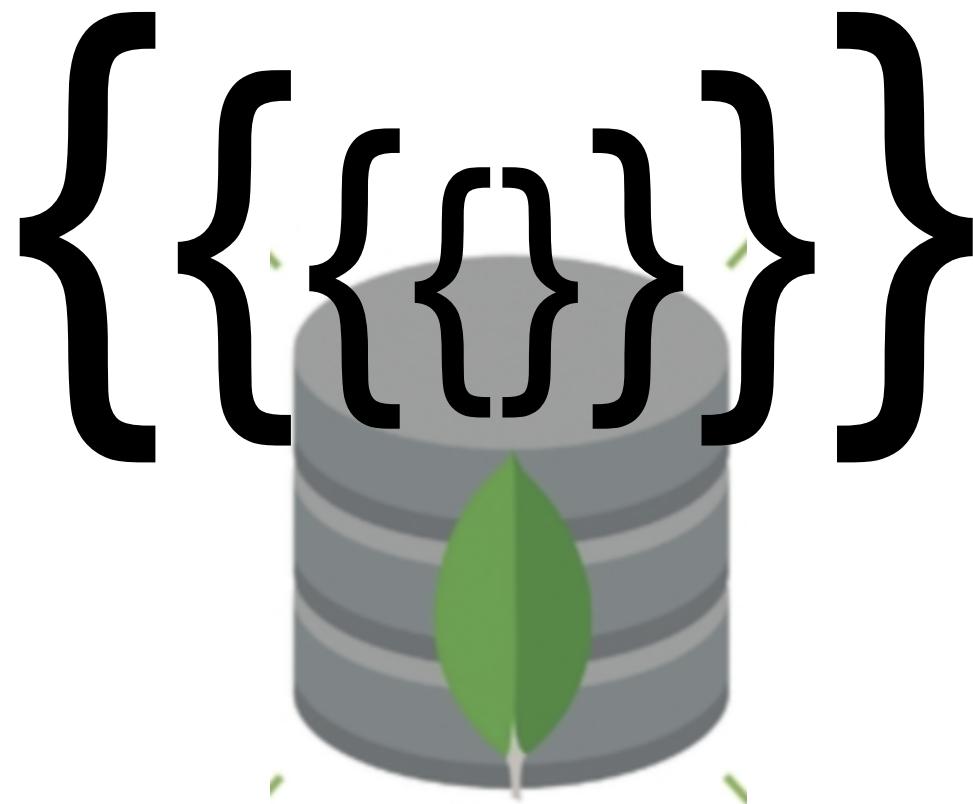
Escenario distribuido



Escenario distribuido



Opciones para Analítica (2/2)



Agregar en MongoDB

MongoDB La Base NoSQL Líder



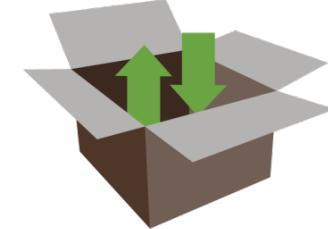
Empresarial

Alto rendimiento

Escalable

```
{  
    nombre: "Juan Perez",  
    pfxs: ["Dr.", "Sr."],  
    direccion: "Reforma 321",  
    telefono: {  
        casa: 1234567890,  
        movil: 5512345689  
    }  
}
```

Modelo de Datos
Documental



Open-Source

El Modelo de Datos de Documento

```
{  
  cliente_id : 12345,  
  nombre : 'Juan',  
  apellido : 'Pérez',  
  direccion : {  
    calle : 'Av. Insurgentes Sur  
321',  
    colonia : 'Roma Norte',  
    ciudad : 'Cuauhtémoc',  
    estado : 'CDMX',  
    cod_postal : '06700'  
  }  
  polizas: [ {  
    num_poliza : 132987,  
    descripcion : 'short term',  
    deducible : 500  
  }, {  
    num_poliza : 147001,  
    descripcion : 'dental',  
    visitas : 7  
  } ]  
}
```

Coincide con los objetos de aplicación

- *Facilita el desarrollo*

Flexible

- *Evoluciona con la aplicación*

Alto rendimiento

- *Diseñado para el patrón de acceso*

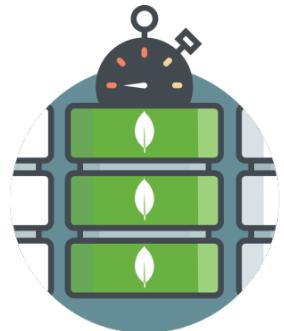
Funcionalidad adicional

Consultas Expresivas	<ul style="list-style-type: none">• Encuentra a alguien con el teléfono # "1-212 ..."• Compruebe si la persona con el número "555 ..." está en la lista de "no llamar"
Geo-espacial	<ul style="list-style-type: none">• Encuentre la mejor oferta para el cliente en las coordenadas geográficas de Insurgentes y Reforma
Búsqueda Textual	<ul style="list-style-type: none">• Encuentra todos los tweets que mencionan la empresa en los últimos 2 días
Agregación	<ul style="list-style-type: none">• Contar y ordenar número de clientes por ciudad
Soporte nativo a JSON Binario (BSON)	<ul style="list-style-type: none">• Agregue un número de teléfono adicional a Mario Robles sin volver a escribir el documento• Seleccione sólo el número de teléfono móvil en la lista• Ordenar por la fecha modificada
Left outer join (\$lookup)	<ul style="list-style-type: none">• Consulta todas las casas de CDMX, consulta sus transacciones y suma la cantidad por persona

{

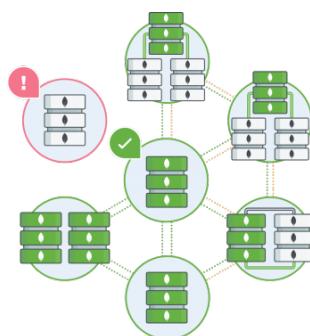
```
cliente_id : 1,  
  nombre : "Mario",  
  apellido : "Robles",  
  ciudad : "CDMX",  
  telefonos: [  
    { numero : "1-212-777-1212",  
      primario : true,  
      tipo : "casa" },  
  
    { numero : "1-212-777-  
      1213",  
      tipo : "cell"  
    }  
  ]  
}
```

Te permite ubicar los datos donde los necesitas



Disponibilidad

plataforma resiliente con replicación y failover



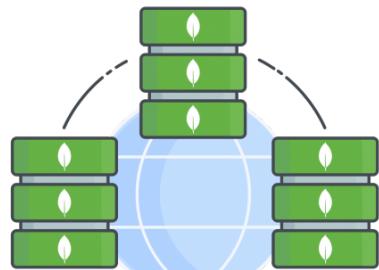
Escalabilidad

Crece horizontalmente con sharding nativo



Aislamiento de Carga de Trabajo

cargas operacionales y analíticas en el mismo cluster



Localidad de datos

Gobierno, calidad de servicio, minimiza latencia

Corre igual en cualquier lugar (on-premises <-> cloud)





Crear ambiente de MongoDB

Clúster gratuito

Creación de perfil y cluster en MongoDB Atlas

<https://cloud.mongodb.com/>

Sign up for MongoDB Atlas
The weight of your ops on our shoulders.

MetLife MTV ebay ADP

Account Profile

Email Address

Password

✓ 8 characters minimum
✓ One letter
✓ One number
✓ One special character

First Name Last Name

Phone Number Company Name

Job Function

Country

I agree to the [terms of service](#)

Already have an account? [Login](#)

Continue

AM_ATLASGROUP (ORGANIZATION) > AM_ATLASGROUP

Clusters

Overview Security

Find a cluster...

You do not have any clusters

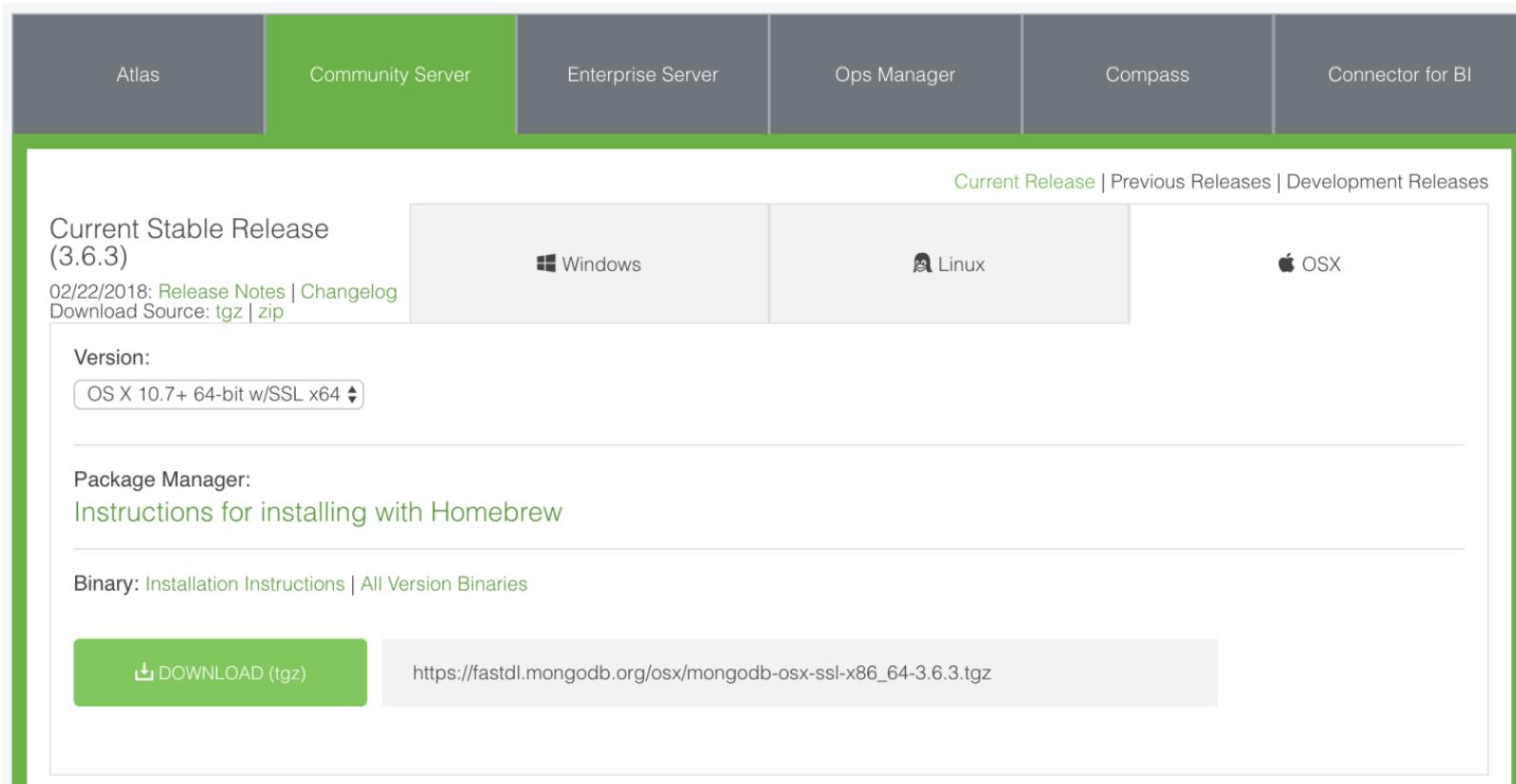
Good thing it's easy to create one

[Build a New Cluster](#)

Need to import existing data into your Atlas cluster? Don't worry, you will be able to after your new cluster is deployed. [Learn how to import data into Atlas](#).

Descargar MongoDB

```
$ curl -O https://fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-3.6.3.tgz
```



The screenshot shows the MongoDB download page for OS X. At the top, there's a navigation bar with tabs for Atlas, Community Server, Enterprise Server, Ops Manager, Compass, and Connector for BI. The Community Server tab is highlighted in green. Below the navigation bar, there's a banner for the 'Current Stable Release (3.6.3)' dated 02/22/2018, with links to 'Release Notes' and 'Changelog'. It also mentions download sources as 'tgz | zip'. To the right of the banner are three icons: Windows, Linux, and OSX. The OSX icon is highlighted. Below the banner, there's a dropdown menu for 'Version' set to 'OS X 10.7+ 64-bit w/SSL x64'. Under the 'Package Manager' section, there's a link to 'Instructions for installing with Homebrew'. In the 'Binary' section, there's a link to 'Installation Instructions' and 'All Version Binaries'. At the bottom, there's a green 'DOWNLOAD (tgz)' button and a text field containing the URL https://fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-3.6.3.tgz.

Descargar repositorio de GitHub

```
$ git clone https://github.com/alxmancilla/DataDayMX
```

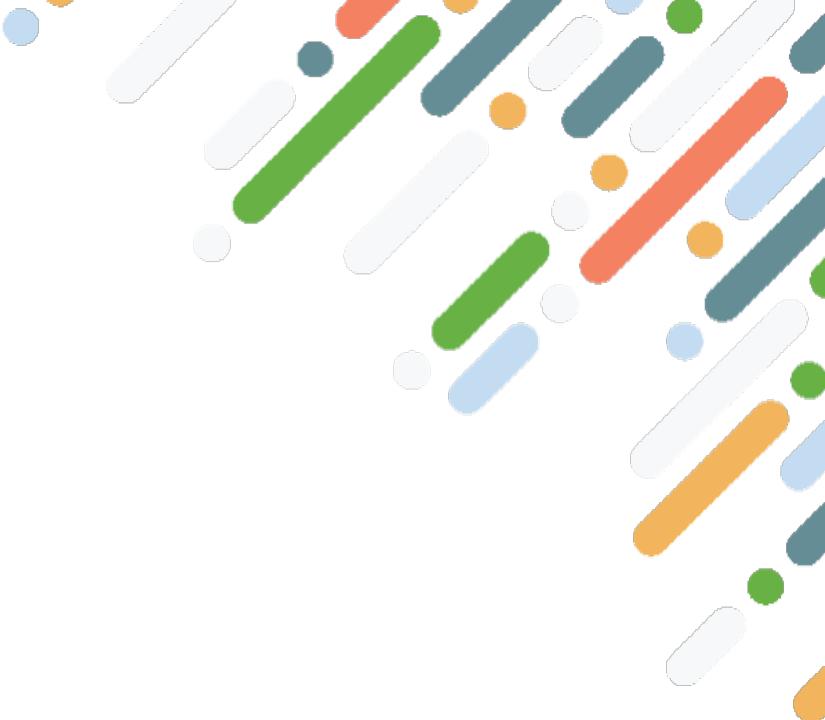
```
Cloning into 'DataDayMX'...
remote: Counting objects: 112, done.
remote: Compressing objects: 100% (111/111), done.
remote: Total 112 (delta 56), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (112/112), 111.88 MiB | 857.00 KiB/s, done.
Resolving deltas: 100% (56/56), done.
```

Ejecutar un nodo stand alone (local)

```
# Crear un nodo stand alone  
  
$ mongod --dbpath rs_pss/m0 \  
  --logpath rs_pss/m0/mongod.log \  
  --port 27017 \  
  --bind_ip 127.0.0.1 \  
  --fork
```

Creación de un replica set (local)

```
# Crear un replica set
mongod --replSet rs_pss --dbpath rs_pss/m0 --port 27017 \
        --logpath rs_pss/m0/mongod.log --bind_ip 127.0.0.1 --fork
# Inicializar un replica set.
'rs.initiate({
    _id: "rs_pss",
    members: [
        { _id: 0, host: "127.0.0.1:27017" },
        { _id: 1, host: "127.0.0.1:27018" },
        { _id: 2, host: "127.0.0.1:27019" } ]});' | mongo
```



Ingestión de datos a MongoDB

Herramientas `mongoimport` / `mongorestore`

Cargando datos usando mongoimport / mongorestore

```
# Se pueden cargar archivos .json/.csv y dumps de MongoDB.  
  
$ mongoimport -d test -c restaurants --drop --file primer-dataset.json  
  
$ mongoimport -c books --drop --type csv --headerline --file ./dataset/books.csv  
  
$ mongoimport -c ratings --drop --type csv --headerline --file ./dataset/ratings.csv  
  
$ mongorestore --dir=dump
```



Conexión al clúster desde el shell en tu laptop

```
# Acceder a localhost en puerto default 27017
$ mongo test
rs_pss:PRIMARY> show dbs
admin      0.000GB
config     0.000GB
local      0.094GB
test       0.220GB
rs_pss:PRIMARY> show collections
books
ratings
restaurants
zips
```

Consultar datos usando findOne () & count ()

```
> db.books.findOne()  
{  
    "_id" : ObjectId("5aa890aa7fe2b08c22b6b246") ,  
    "book_id" : 2 ,  
    "goodreads_book_id" : 3 ,  
    "work_id" : 4640799 ,  
    "books_count" : 491 ,  
    "isbn" : 439554934 ,  
    "authors" : "J.K. Rowling, Mary GrandPr." ,  
    "original_publication_year" : 1997 ,  
    "original_title" : "Harry Potter and the  
Philosopher's Stone" ,  
    "title" : "Harry Potter and the Sorcerer's Stone  
(Harry Potter, #1)" ,  
    "language_code" : "eng" ,  
    "image_url" : "https://images.gr-  
assets.com/books/1474154022m/3.jpg" ,  
    "small_image_url" : "https://images.gr-  
assets.com/books/1474154022s/3.jpg"  
}
```

```
> db.ratings.findOne()  
{  
    "_id" :  
    ObjectId("5aa889807fe2b08c225974e8") ,  
    "user_id" : 2 ,  
    "book_id" : 260 ,  
    "rating" : 5  
}
```

```
> db.books.count()  
10000
```

```
> db.ratings.count()  
5976479
```

Consultar datos usando find()

```
# Filtrar ratings del book_id igual a 1

> db.ratings.find({"book_id": 1})
{ "_id" : ObjectId("5aa889917fe2b08c225cd3eb") ,
"user_id" : 2886, "book_id" : 1, "rating" : 5 }
{ "_id" : ObjectId("5aa889977fe2b08c225dfbdc") ,
"user_id" : 6158, "book_id" : 1, "rating" : 5 }
...
```



Exploración de datos en MongoDB

Usando MongoDB Compass

MongoDB Compass

The screenshot displays three panels of the MongoDB Compass interface, illustrating its features for managing MongoDB databases.

Top Panel: Shows the main dashboard for the "DemoDev" database. It lists 4 DBS and 13 Collections. The "test" database is selected. A table provides an overview of four collections: "book_tags", "books", "ratings", and "restaurants".

Collection	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size
book_tags	999,912	68.0 B	64.8 MB	1	9.4 MB
books	10,000	713.8 B	6.8 MB	1	96.0 KB
ratings	5,976,479	60.0 B	342.0 MB	1	57.5 MB
restaurants	25,359	419.0 B	10.1 MB	1	232.0 KB

Middle Panel: Provides a detailed view of the "books" collection. It shows 10.0k documents, a total size of 6.8MB, and an average size of 714B. The "Schema" tab is active, displaying the document structure:

```
_id: ObjectId("5aa9f17134885bae077b6991")
book_id: 1
goodreads_book_id: 2767052
best_book_id: 2767052
work_id: 2792775
books_count: 272
isbn: 439023483
isbn13: 9780000000000
> authors: Array
    original_publication_year: 2008
    original_title: "The Hunger Games"
    title: "The Hunger Games (The Hunger Games, #1)"
    language_code: "eng"
    > image: Object
    > ratings: Array
        avg_rating: 4.2797070946242215
        ratings_count: 22806
```

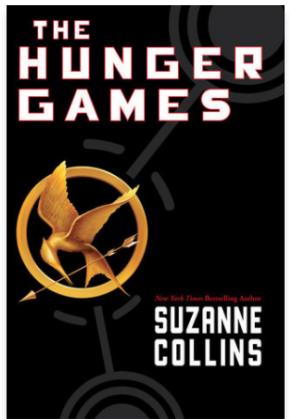
Bottom Panel: Provides a detailed view of the "test.books" collection. It shows 10.0k documents, a total size of 6.8MB, and an average size of 714B. The "Schema" tab is active, showing the same schema as the middle panel. The "Explain Plan" tab is also visible. Below the schema, a query report indicates 3666 documents based on a sample of 1000. The right side of the panel features several data distribution charts for fields like "avg_rating", "original_publication_year", "ratings", and "rated".



Análisis de datos con MongoDB

Usando `db.collection.aggregate()`

Sitio de recomendación de libros



Want to Read

Rate this book
★★★★★

The Hunger Games (The Hunger Games #1)

by Suzanne Collins

★★★★★ 4.34 · Rating details · 5,260,482 Ratings · 158,194 Reviews

Winning will make you famous.

Losing means certain death.

The nation of Panem, formed from America, is a country that consists of 12 poorer districts. It is led by a 13th district against the Capital's destruction and the creation of an as t ...more



Rating details



Want

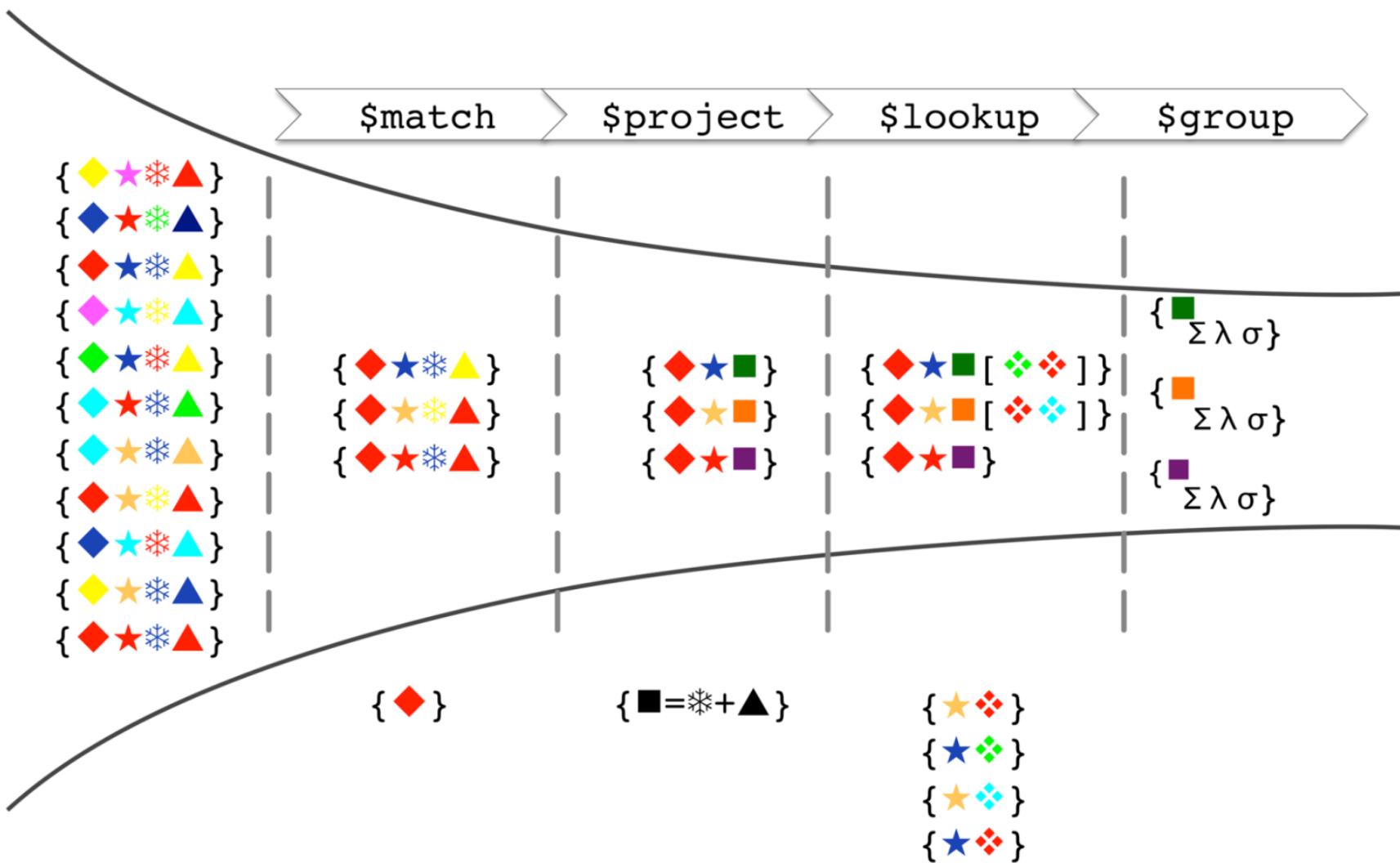
96% of people liked it

All editions: 4.34 average rating, 5260482 ratings, 158194 reviews, added by 6292996 people, 66759 to-reads

This edition: 4.33 average rating, 5084184 ratings, 144034 reviews, added by 6054806 people

Listen

Modelo del Framework de Agregación



Aggregation Framework - \$match & \$group

```
# Filtrar ratings del book_id igual a 1
> db.ratings.aggregate([{$match: {"book_id": 1}}])
{ "_id" : ObjectId("5aa889917fe2b08c225cd3eb") , "user_id" : 2886, "book_id"
: 1, "rating" : 5 }
{ "_id" : ObjectId("5aa889977fe2b08c225dfbdc") , "user_id" : 6158, "book_id"
: 1, "rating" : 5 }
...
# Filtrar y después agrupar por rating y contar total por nivel de rating
> db.ratings.aggregate([{$match: {"book_id": 1}},
                         {$group: {"_id": "$rating", "total": {$sum:1}} }])
{ "_id" : 1, "total" : 231 }
{ "_id" : 3, "total" : 3013 }
...
```

Aggregation Framework - \$sum , \$avg & \$sort

```
# Ratings por libro y su evaluación promedio, ordenados por book_id

> db.ratings.aggregate([{$group: {_id: "$book_id",
                                total: { $sum: 1},
                                promedio: {$avg: "$rating"} } },
                        {$sort: {"_id": 1}} ])
{ "_id" : 1, "total" : 22806, "promedio" : 4.2797070946242215 }
{ "_id" : 2, "total" : 21850, "promedio" : 4.351350114416476 }
{ "_id" : 3, "total" : 16931, "promedio" : 3.214340558738409 }
...
```

Aggregation Framework - \$sum , \$avg & \$sort

```
# Histograma de libros ( mean, std dev, min, max )

> db.ratings.aggregate([{$group: {_id: "$book_id",
                                total: { $sum: 1}}},
                        {$group: {_id: null, total: {$sum: 1},
                                  promedio: {$avg: "$total"},
                                  des_est: {$stdDevPop: "$total" },
                                  min: {$min: "$total"}, 
                                  max: {$max: "$total"} } } ])
```

```
{ "_id" : null, "total" : 10000, "promedio" : 597.6479, "des_est" :
1267.226422122578, "min" : 8, "max" : 22806 }
```

Aggregation Framework - \$sort & \$limit

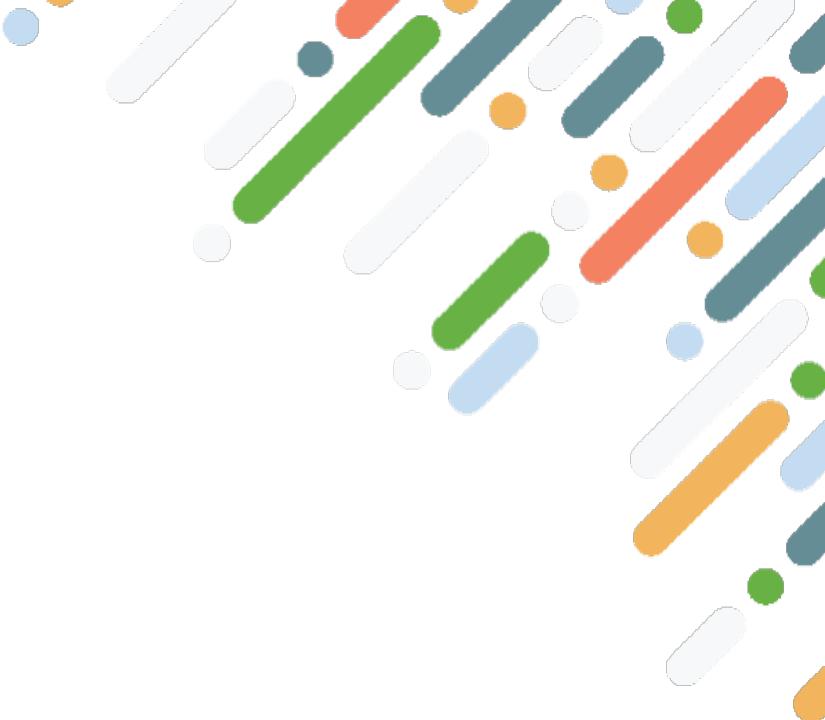
```
# Histograma de libros ( mean, std dev, min, max )

> db.ratings.aggregate([{$group: {_id: "$book_id",
                                total: { $sum: 1}}},
                        {$group: {_id: null, total: {$sum: 1},
                                  promedio: {$avg: "$total"},
                                  des_est: {$stdDevPop: "$total" },
                                  min: {$min: "$total"}, 
                                  max: {$max: "$total"} } } ])
```

```
{ "_id" : null, "total" : 10000, "promedio" : 597.6479, "des_est" :
1267.226422122578, "min" : 8, "max" : 22806 }
```



Demo del Aggregation Framework



Extendiendo análisis con Python

Usando pymongo

Drivers & Frameworks



MEAN Stack



express™

django

Morphia



Instalación de pymongo

```
# Instalar pymongo con pip  
$ python -m pip install pymongo
```

```
# Verificar instalación de pymongo  
$ python -c "import pymongo; print(pymongo.version);"  
3.6.1
```

Ejemplo en Python - Conexión a MongoDB (1/2)

```
import pymongo
# Connection to Mongo DB
try:
    conn=pymongo.MongoClient("mongodb://localhost:27017")
    print "Connected successfully!!!"
    collection = conn.DataDayMX.wkshops
    # dictionary to be added in the database
    record={
        "title": "MongoDB and Python",
        "description": 'MongoDB is no SQL database',
        "tags": [ 'mongodb' , 'database' , 'NoSQL' ] ,
        "viewers": 104}
```

Ejemplo en Python - Conexión a MongoDB (2/2)

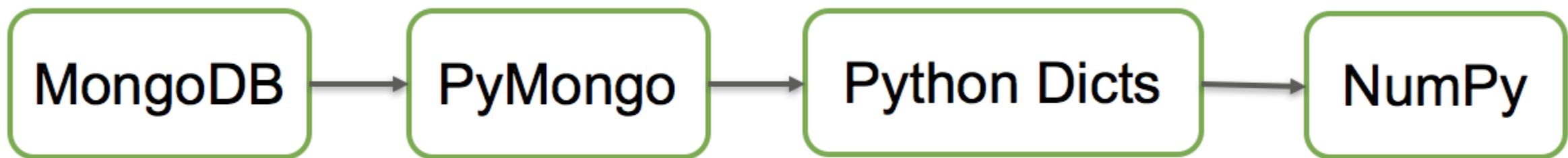
```
# inserting the data in the database
rec = collection.insert(record)
print "Inserted record successfully!!!"

for i in collection.find({"title": "MongoDB and Python"}):
    print(i)
print "Docs returned successfully!!!"

except pymongo.errors.ConnectionFailure, e:
    print "Could not connect to MongoDB: %s" % e
conn.close()
```

Demo de Integración de MongoDB con Python

Integración con Numpy





NumPy

Monary

Scikit -
learn

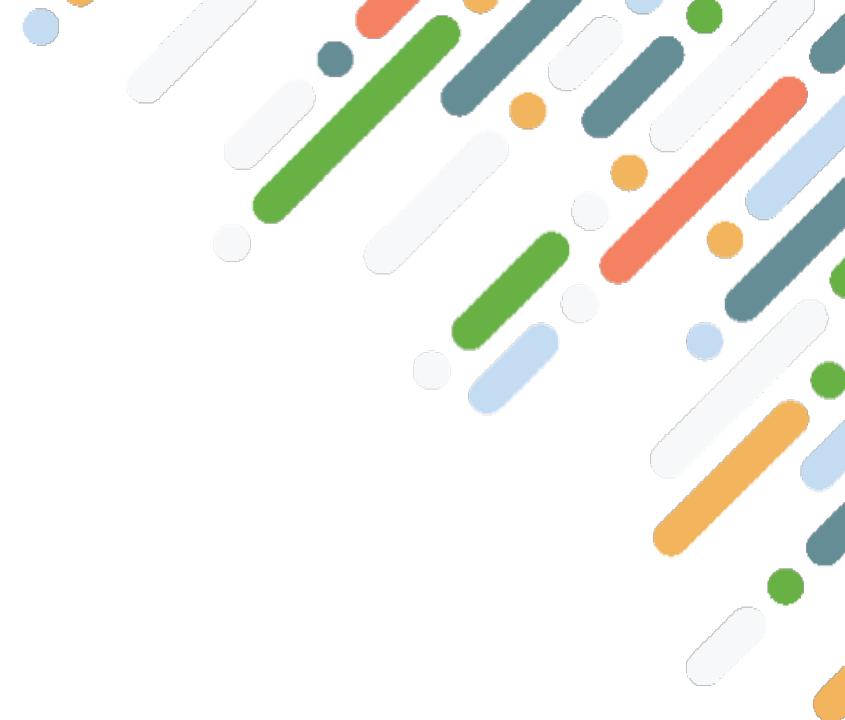
Pandas

Python

Cron
Airflow
Luigi

Matplotlib

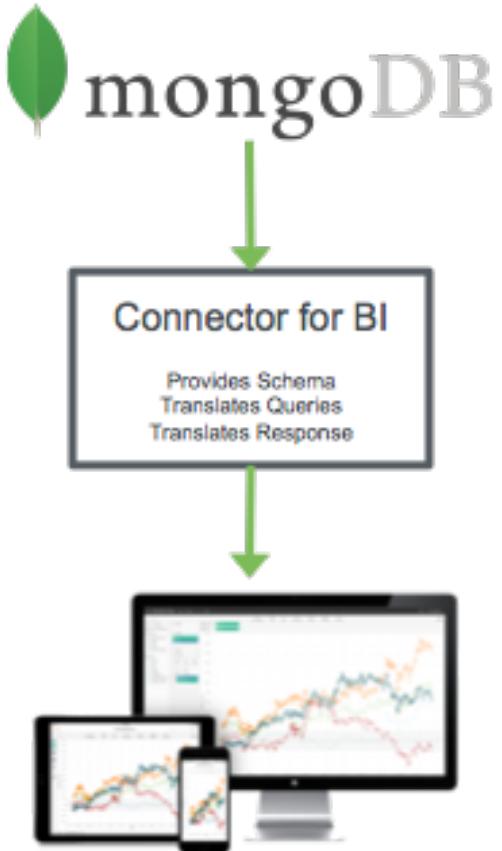
PyTables



Creación de Dashboards

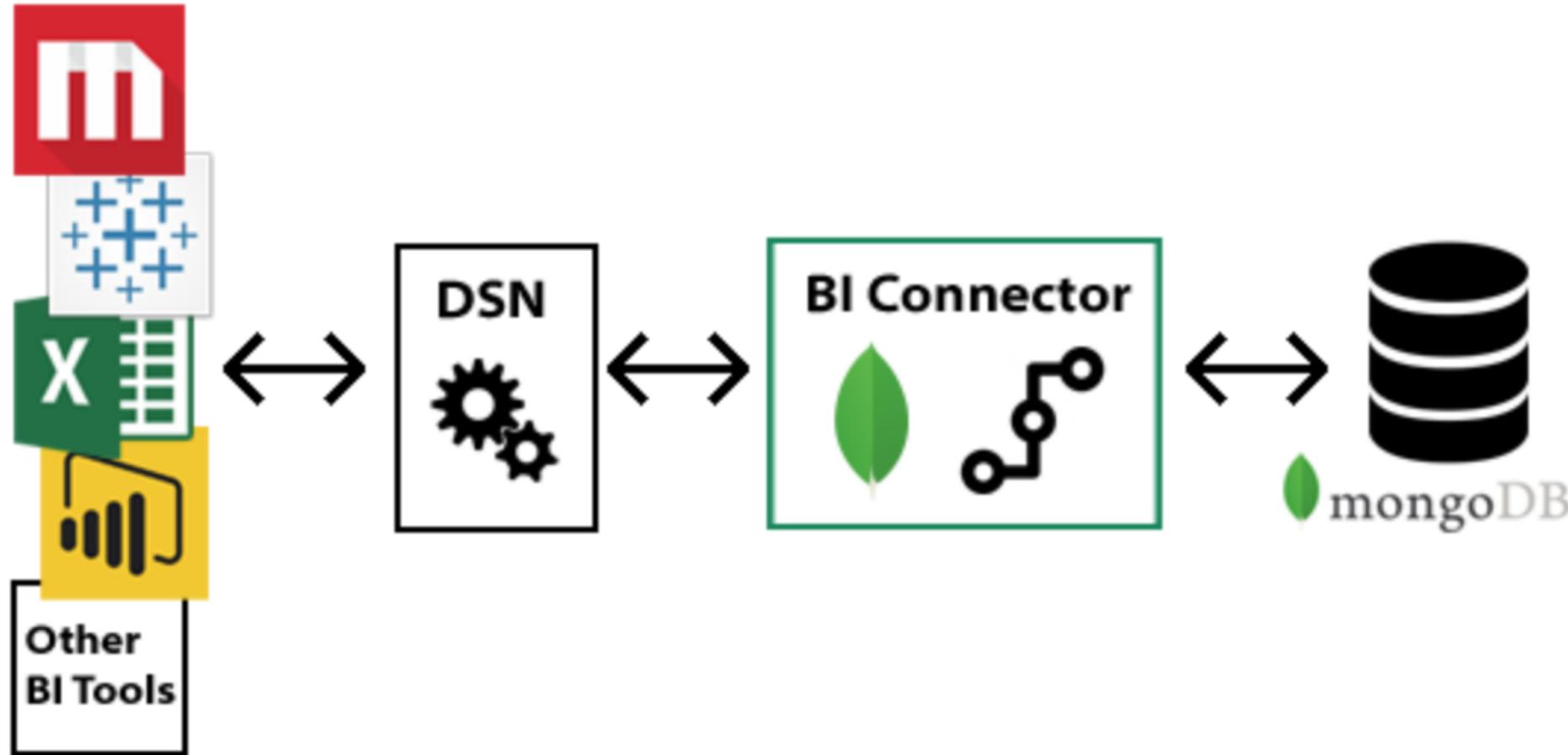
Usando MongoDB BI Connector

MongoDB Conector para BI



- Crea visualizaciones & analítica desde herramientas de BI basadas en SQL
 - Muestreo automático de esquemas
 - Elimina necesidad de ETL
- Rendimiento mejorado en la capa de re-escritura de SQL
 - Mayor procesamiento realizado en la base de datos
- Instalación y autenticación simplificada

Arquitectura del MongoDB Conector para BI



Analítica e Integración con BI



QlikView

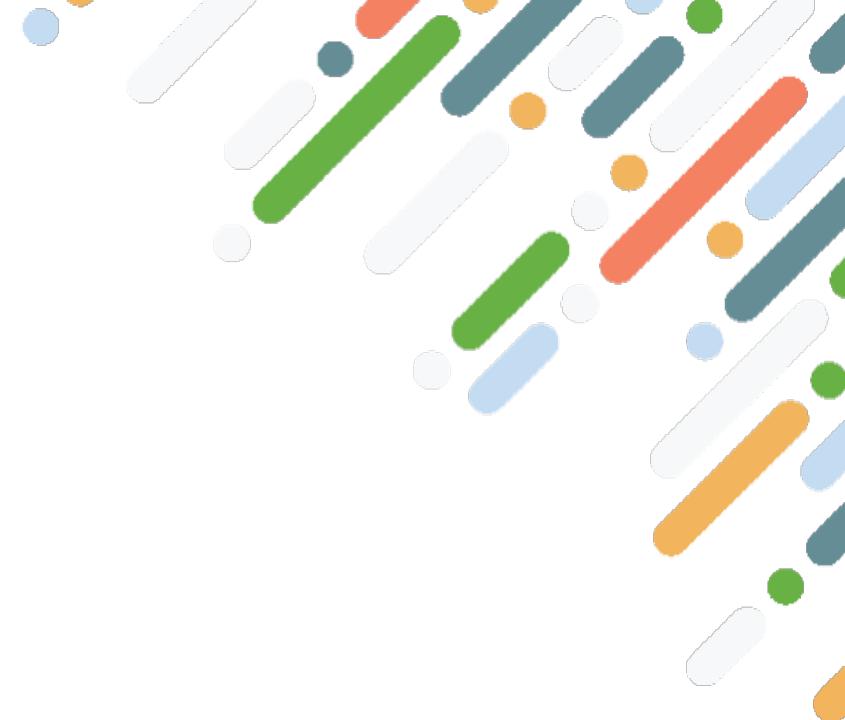


MicroStrategy



talend*

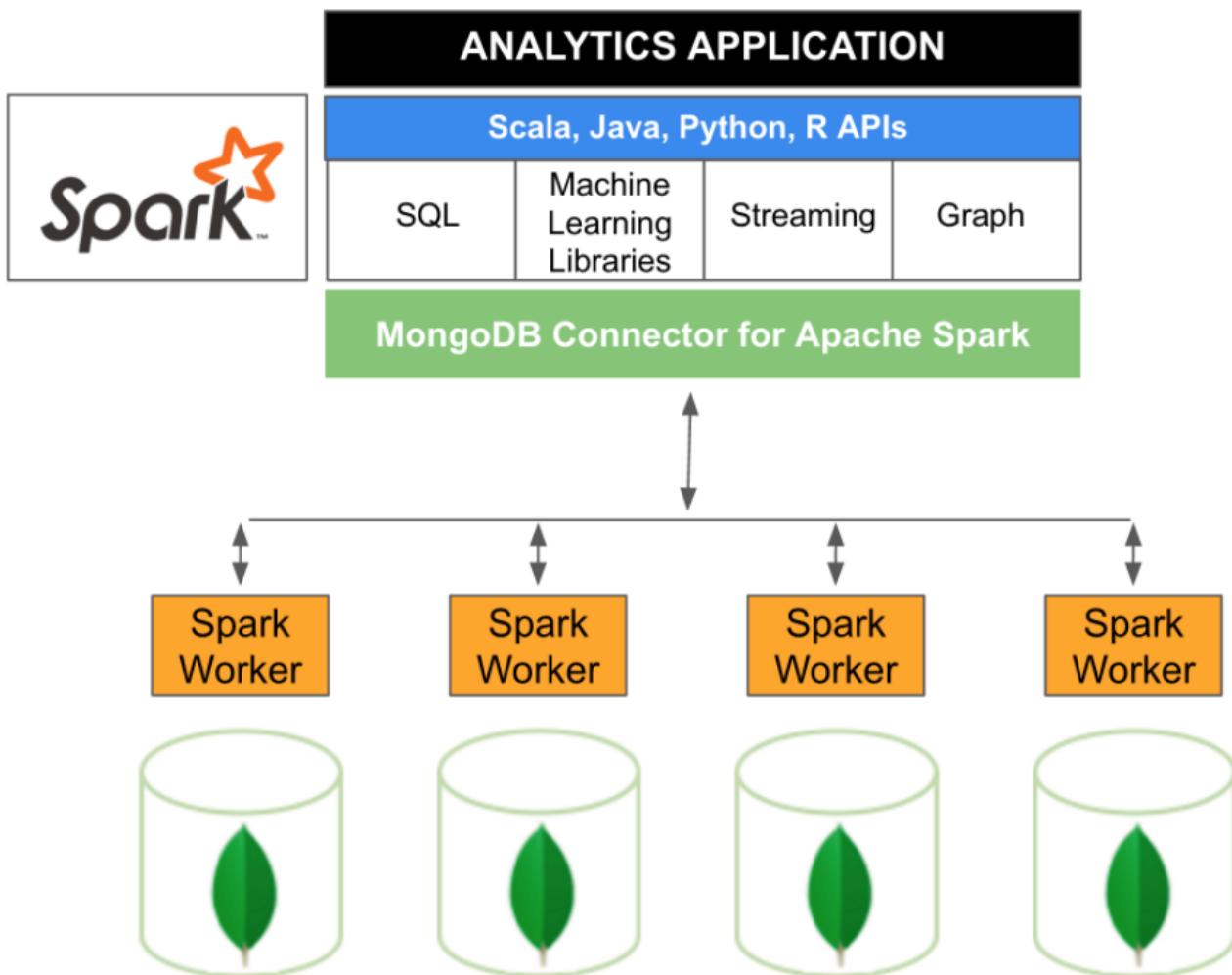
Demo del Conector para BI



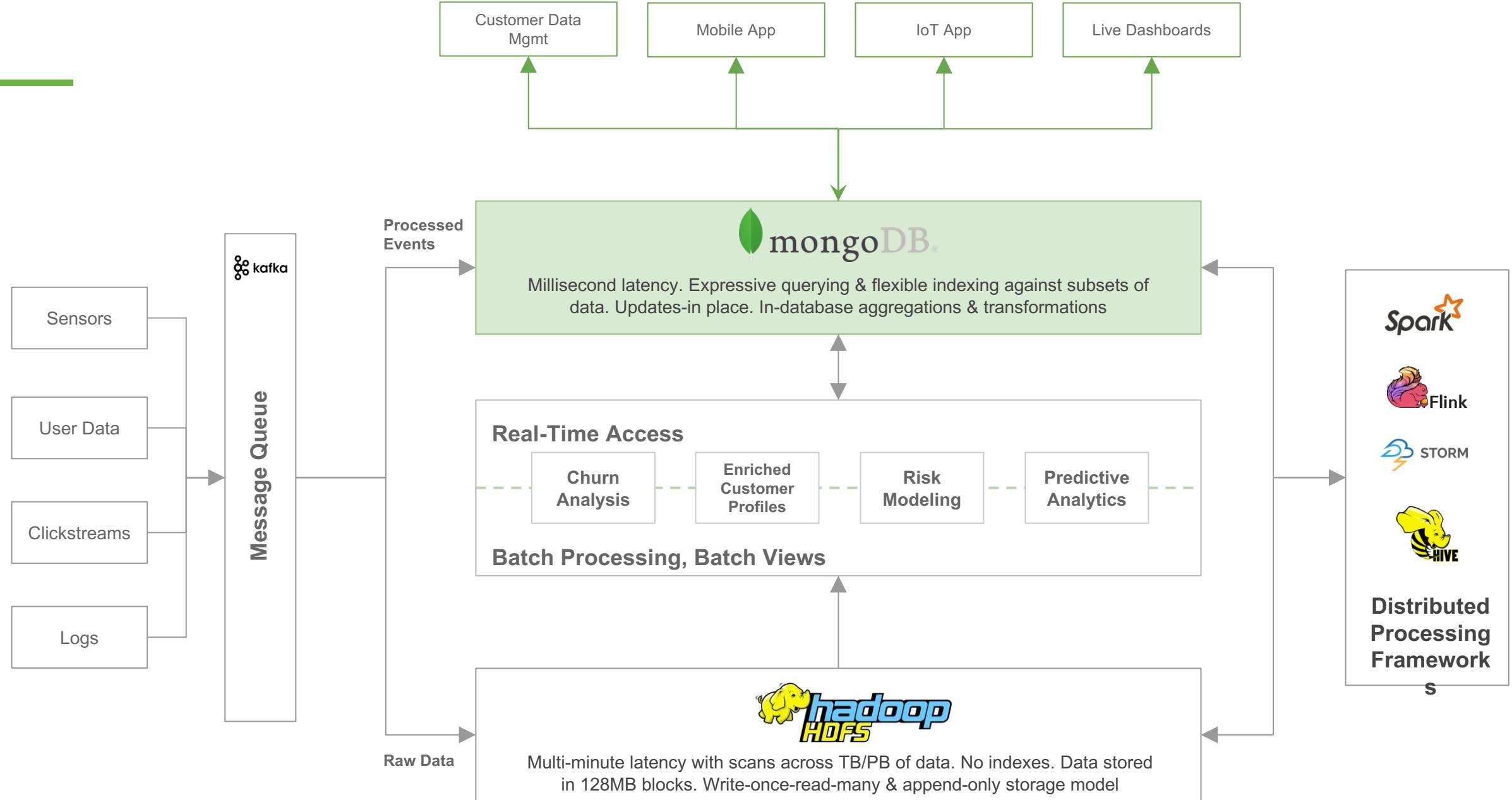
Extendiendo análisis con Apache Spark

Usando MongoDB Connector

MongoDB Conektor para Apache Spark



Data Lake Operacional



MongoDB University

Free • 1-2 Hours • Online

M233: Getting Started with Spark and MongoDB

Learn everything you need to know to ramp up to MongoDB 3.4

<https://university.mongodb.com/courses/M233/>

Courses > M233

Next Session:

Start: 16 May 2017 at 17:00 UTC

End: 15 May 2018 at 17:00 UTC

[View Courseware](#)

What You'll Learn

This course provides an introduction to Spark and helps students get started using the MongoDB Spark connector to build data analytics applications. We provide an overview of the Spark Scala and Java APIs with plenty of sample code and demonstrations.

Prerequisites:

The course does not assume prior knowledge of Spark, but does require an intermediate level of expertise with MongoDB. The suggested prerequisites are a

Q & A



Análisis de Datos con MongoDB

Alejandro Mancilla

Sr. Solutions Architect, LATAM

alejandro@mongodb.com

@alxmancilla

