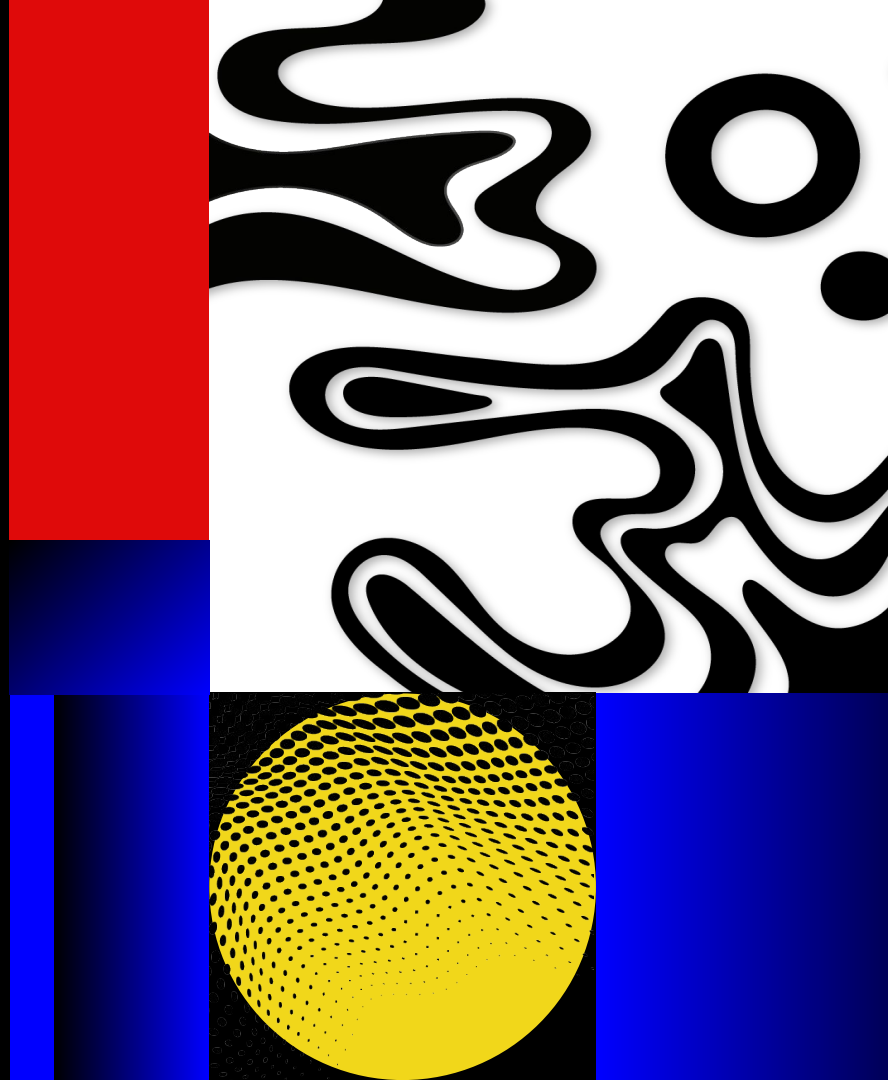
BIG DATA 2022

BECOMING A DATA ROCKSTAR

Katherine García
Andrea Reyes



AGENDA

01 Descripción del proyecto

02 Milestone 1-3

03 Retos y lecciones aprendidas

BECOMING A DATA ROCKSTAR

HOTELES Aa

Una cadena de hoteles busca una solución que permita gestionar toda la información de sus clientes y reservas

- Operaciones: sponsors del proyecto.
- Customer experience: experiencia de usuarios antes, durante y después de la estadía en el hotel.
- Business analytics: preparar información para reportes gerenciales.

OBJETOS DE DATOS

Hotel

Activos, ubicación, etc.

Habitación

Servicios disponibles

Tarifas

Precios base para cada habitación

Ocupación

Disponibilidad de habitaciones.

Cliente

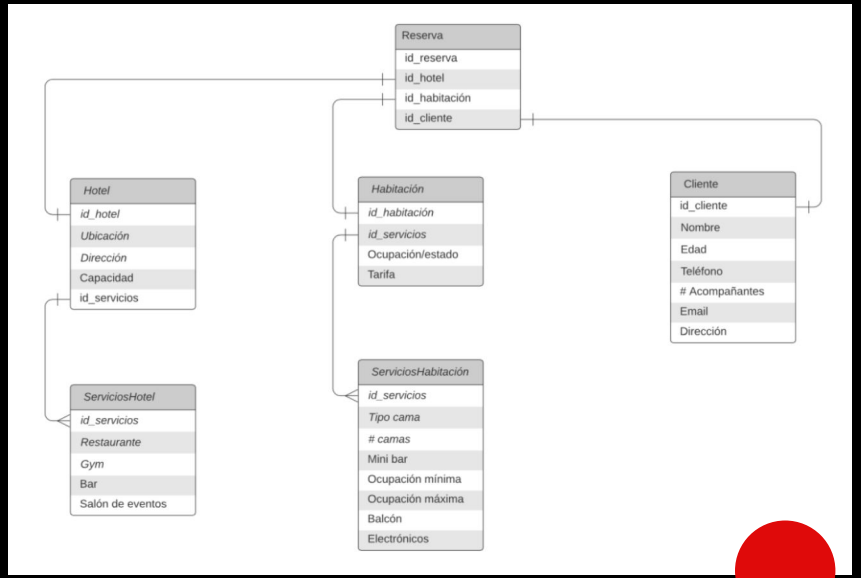
Usuario único con información personal

MILESTONE 1

- Diseñar un modelo entidad relación que se adapte al requerimiento
- Instalar una base de datos
- Crear el modelo
- Cargar datos de prueba

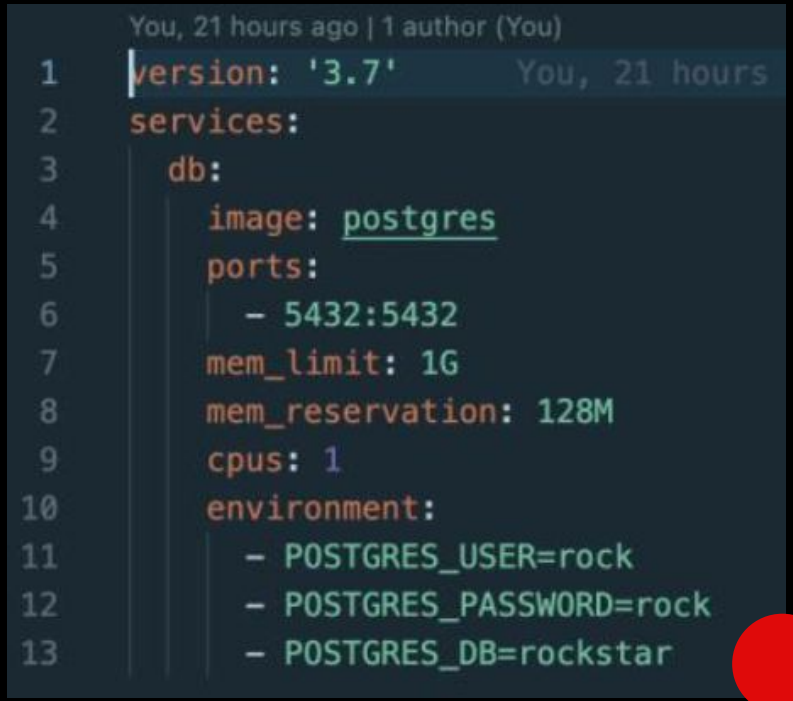
MODELO ENTIDAD RELACIÓN

ERD



BASE DE DATOS

- Docker
- postgrSQL



TABLAS

```sql
CREATE TABLE hotel(
    id_hotel int primary key,
    ubicacion varchar,
    direccion varchar,
    capacidad varchar,
    id_servicios int,
     CONSTRAINT id_servicios
      FOREIGN KEY(id_servicios)
      REFERENCES "serviciosHotel"(id_servicios)
      ON DELETE CASCADE
);

CREATE TABLE habitacion(
    id_habitacion int primary key,
    ocupacion_estado varchar,
    tarifa int,
    id_servicios int,
     CONSTRAINT id_servicios
      FOREIGN KEY(id_servicios)
      REFERENCES "serviciosHabitacion"(id_servicios)
      ON DELETE CASCADE
);
```

```sql
CREATE TABLE reserva(
    id_reserva int primary key,
    id_hotel int,
    id_habitacion int,
    id_cliente int,
     CONSTRAINT id_hotel
      FOREIGN KEY(id_hotel)
      REFERENCES "hotel"(id_hotel)
      ON DELETE CASCADE,

    CONSTRAINT id_habitacion
      FOREIGN KEY(id_habitacion)
      REFERENCES "habitacion"(id_habitacion)
      ON DELETE CASCADE,

    CONSTRAINT id_cliente
      FOREIGN KEY(id_cliente)
      REFERENCES "cliente"(id_cliente)
      ON DELETE CASCADE
);
```

```sql
CREATE TABLE cliente(
    id_cliente int primary key,
    nombre varchar,
    edad int,
    telefono int,
    email varchar,
    direccion varchar,
    cant_acompañantes int,
    id_servicios int
);
```

DATOS DE PRUEBA

WHERE ORDER BY

	id_hotel	ubicacion	direccion	capacidad	id_servicios
1	10001	Guatemala	20 calle 8-43 zona 13	100	11001
2	10002	El Salvador	4ta calle 7-76 zona 8	115	11002
3	10003	Costa Rica	Calle Manuel F. Ayau, zona 10	85	11003
4	10004	Honduras	San Cristobal 18 av 1-83	50	11004
5	10005	Belice	Vista Hermosa III, zona 16	98	11005

WHERE ORDER BY

	id_habitacion	ocupacion_estado	tarifa	id_servicios
1	10001101	Reserva	120	1000110101
2	10001102	Reserva	150	1000110102
3	10001103	Ocupada	200	1000110103
4	10001104	Cancelada	150	1000110104
5	10001105	Reserva	150	1000110105

WHERE ORDER BY

	id_cliente	nombre	edad	telefono	email	direccion	cant_acompañantes
1	12341	Katherine Garcia	20	42186759	Kg@email.com	Guatemala	2
2	12342	Andrea Reyes	21	52030518	Andreareyes@email.com	USA	1
3	12343	Maria Paz	35	56325878	Mpaz@email.com	Guatemala	3
4	12344	Esteban Lopez	66	41076352	lopezEst@email.com	El Salvador	4
5	12345	Monica Sab	25	35667890	monSab@email.com	Panama	1

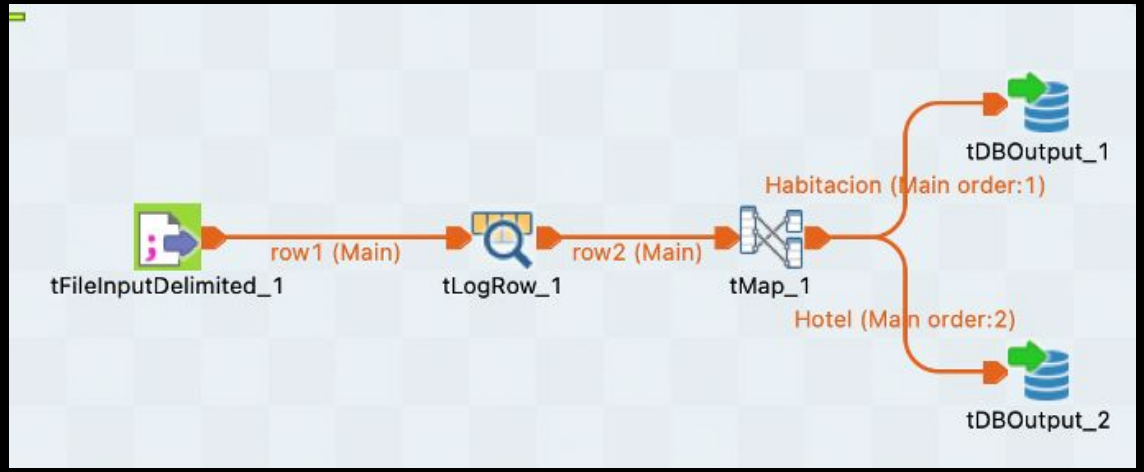
MILESTONE 2

1. Construir un ETL que permita cargar la información de los hoteles desde archivos planos. (CSV) hacia la base de datos relacional.
2. Construir un repositorio en HDFS para los archivos de data analytics.
3. Construir un ETL que tome una captura diaria del estado de reservaciones y clientes y lo cargue en HDFS.
4. Construir una capa en HIVE que permita consultar la información que se ha recibido desde la base relacional.

ETL

Con el archivo proporcionado, se crearon por medio del ETL las dos tablas principales de nuestro modelo.

Las tablas creadas fueron: Habitación y Hotel.



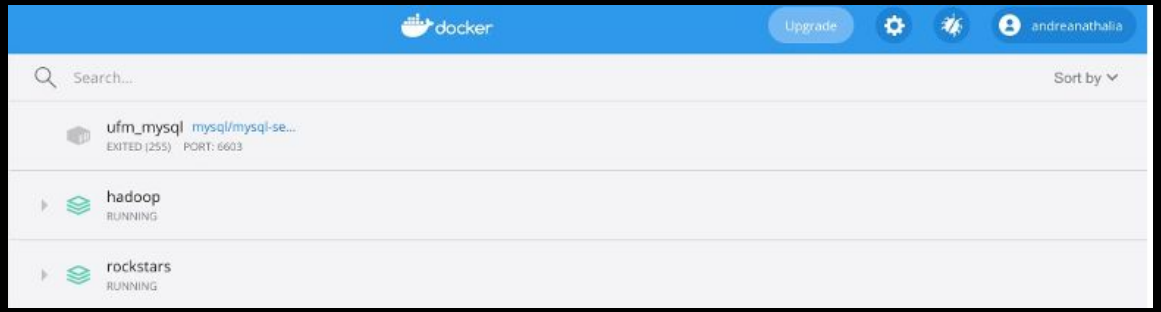
TALEND

HDFS

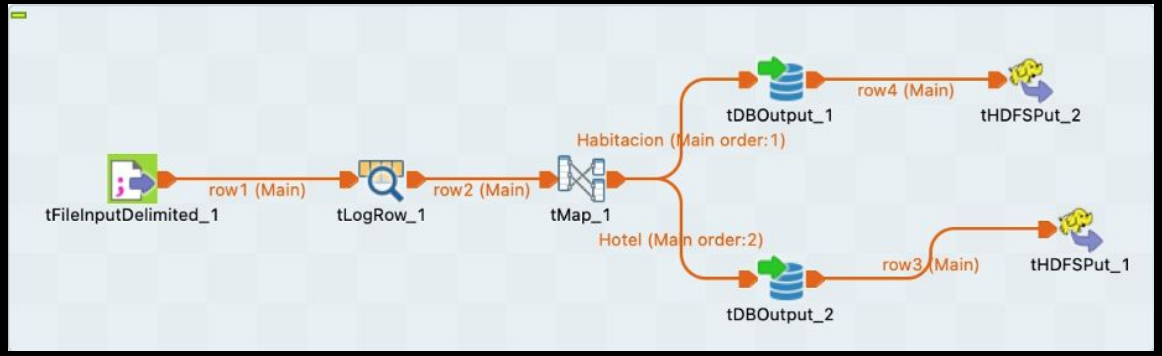
01

Docker

Un Docker Compose que nos proporcionaba las herramientas necesarias para poder construir el repositorio.



CAPTURA DIARIA



CONSULTA HIVE

Tablas que reciben los datos desde hdfs

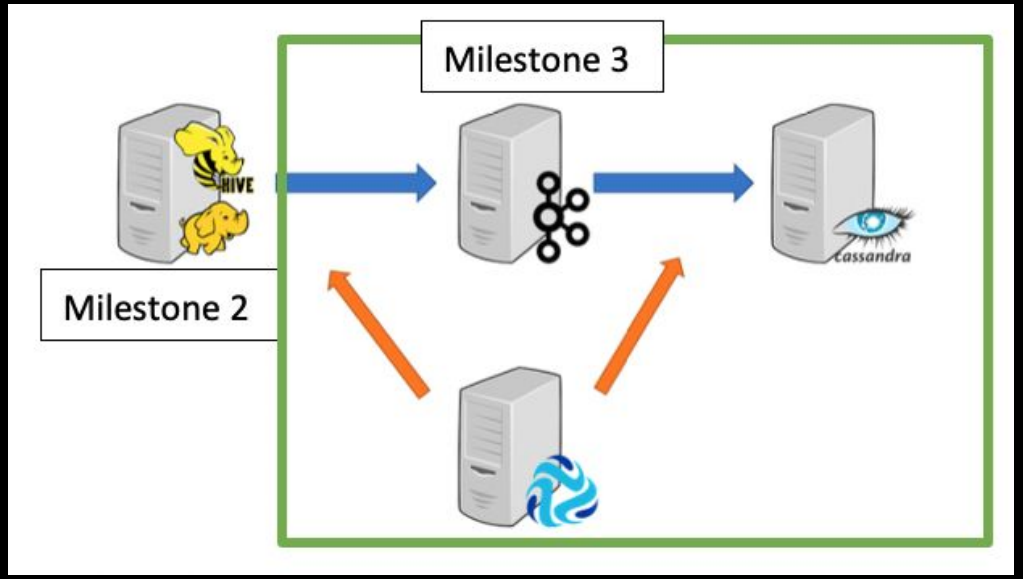
MILESTONE 3

- Construir un modelo en Cassandra que permita replicar el estado actual de las reservaciones para lograr escalabilidad y mejorar tiempos de respuesta.

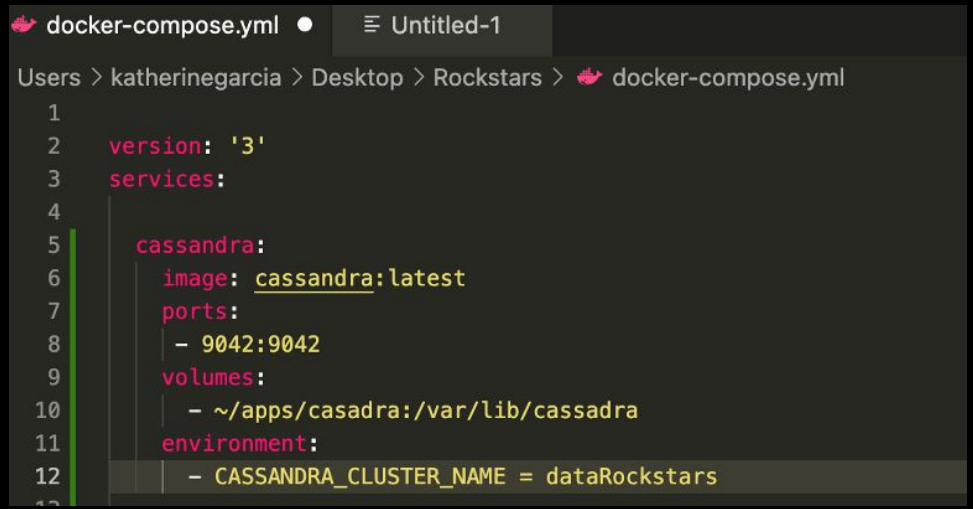
- Construir un tópico de kafka que gestionará todas las actualizaciones que se envían a Cassandra

- Crear un data pipeline en Streamsets que cargue la información de las reservaciones de la base relacional a Kafka, y un Pipeline que tome la información de Kafka y la envíe a Cassandra.

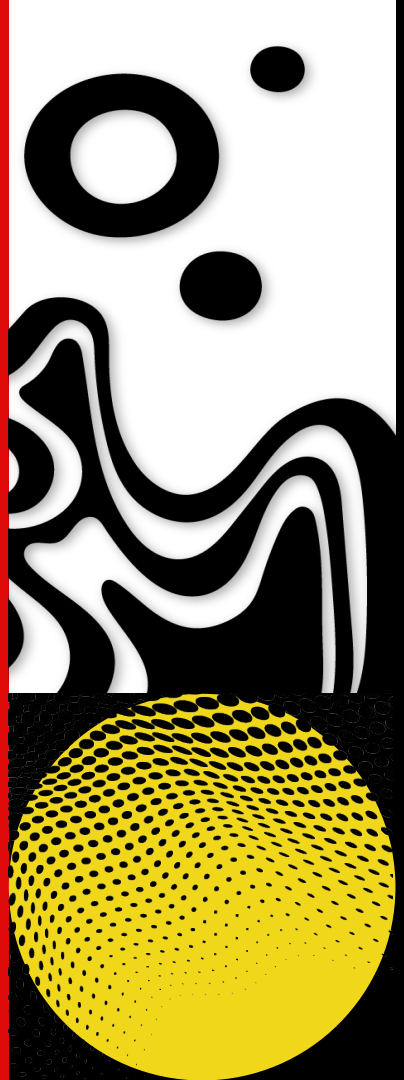
M3



DOCKER



DOCKER



MODELO

Users > katherinegarcia > Desktop > Rockstars > cassandra > cass.cql

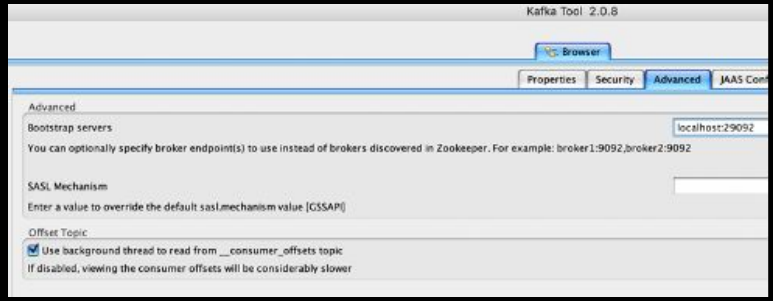
```
1   -- Construir un modelo en Cassandra que permita replicar
2   -- estado actual de las reservaciones
3   -- para lograr escalabilidad y mejorar tiempos de respuesta.
4
5   -- MODELO
6   CREATE KEYSPACE rocky
7       WITH REPLICATION = {
8           'class' : 'SimpleStrategy'
9           'replication_factory' : 1
10  };
11
12  CREATE TABLE reserva(
13      id_reserva int primary key,
14      id_hotel int,
15      id_habitacion int,
16      id_cliente int,
17      fecha_ingreso date,
18      fecha_egreso date,
19  );
20
21  DROP TABLE rocky.reserva
```

KAFKA

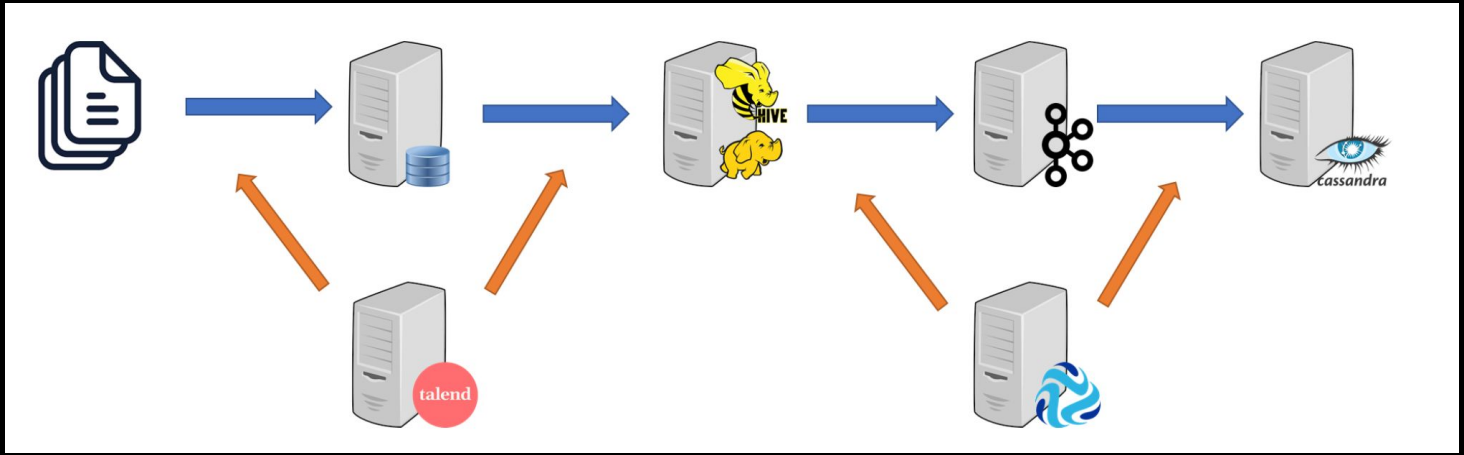
> katherinegarcia > Desktop > Rockstars > docker-cor

```yaml
version: '3'
services:
  zookeeper:
    image: confluentinc/cp-zookeeper:latest
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
    ports:
      - 22181:2181

  kafka:
    image: confluentinc/cp-kafka:latest
    depends_on:
      - zookeeper
    ports:
      - 29092:29092
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
```



MODELO COMPLETO



RETOS

CLOUD

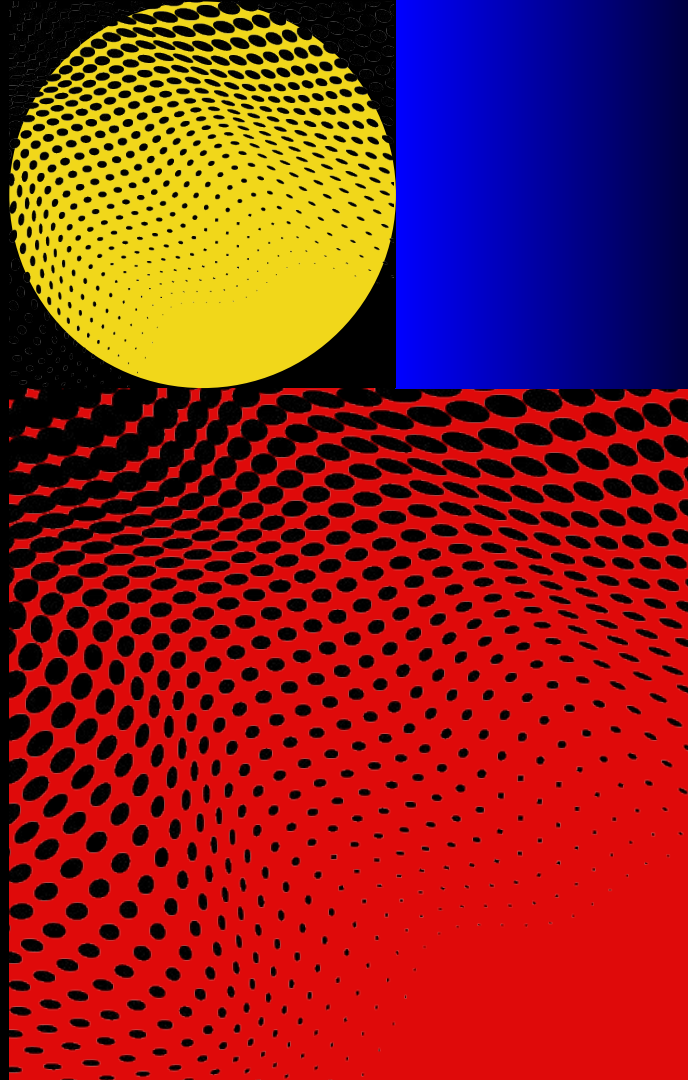
El reto principal fue el tiempo que nos tomó pensar en la sincronización talend-hdfs cloud

DISTRIBUCIÓN

Al trabajar con recursos locales, la distribución de trabajo fue un reto para nosotros como equipo.

Herramientas

En el pasado no habíamos utilizado ninguna de las herramientas necesarias para este proyecto. La curva de aprendizaje y el tiempo fue un reto.



LECCIONES APRENDIDAS

EFECTIVIDAD DE DOCKER

Poder sincronizar todas las fases localmente por medio de docker.

USO DE NUEVAS HERRAMIENTAS

HIVE
HDFS
CASSANDRA
TALEND
DOCKER

GRACIAS

Do you have any questions?

Andrea Reyes
Katherine García