

Jonnatan Arias Garcia

Estructura del Curso

- Modulo 1: Introducción NoSQL
 - Definición
 - Historia
 - Conceptos y características
 - Beneficios
- Modulo 2: Definiendo NoSQL, Tipos, Opciones, Casos de uso
 - Tipos de NoSQL
 - Descripción de caso de uso primario
 - Diferencia entre los tipos
- Modulo 3: Elección de data layer para la aplicación
 - Diferencia de database loca, hosted database y database como servicio
 - Examinar factores relevantes y consideraciones para determinar una data layer apropiada.

Modulo 4: Introducción a Cloudant – NoSQL DBaaS

- Lista de beneficios claves de IBM Cloudant, una NoSQL Database como un servicio
- Descripción de la arquitectura Cloudant's
- Características de Cloudant
- Explicación los problemas que Cloudant puede resolver
- Descripción del despliegue Cloudant's

Modulo 5: Empezando con IBM Cloudant:

- Registro en IBM Cloudant
- Vista del dashboard
- Crar una database y add documentos
- Cambiando permisos de database
- Query data
- Replicar muestras de la database
- Trabajando con HTTP API
- Acceso a documentación y soporte

Modulo 1

NoSQL y Data Base as a Service (DBaaS)

NoSQL -> Not Only SQL
(No Relacional)

No SQL se refiere a un tipo de databases que varían ampliamente en estilo y tecnología, pero todas comparten un trato común en que no tienen una naturaleza relacional, ósea que no vienen en filas y columnas estándar.

Un mejor nombre seria No Relacional.

NoSQL Provee nuevas formas de almacenamiento y búsqueda de data.

NoSQL y Data Base as a Service (DBaaS)

La mayoría de DB NoSQL están diseñadas para manejar diferentes problemas de escala.

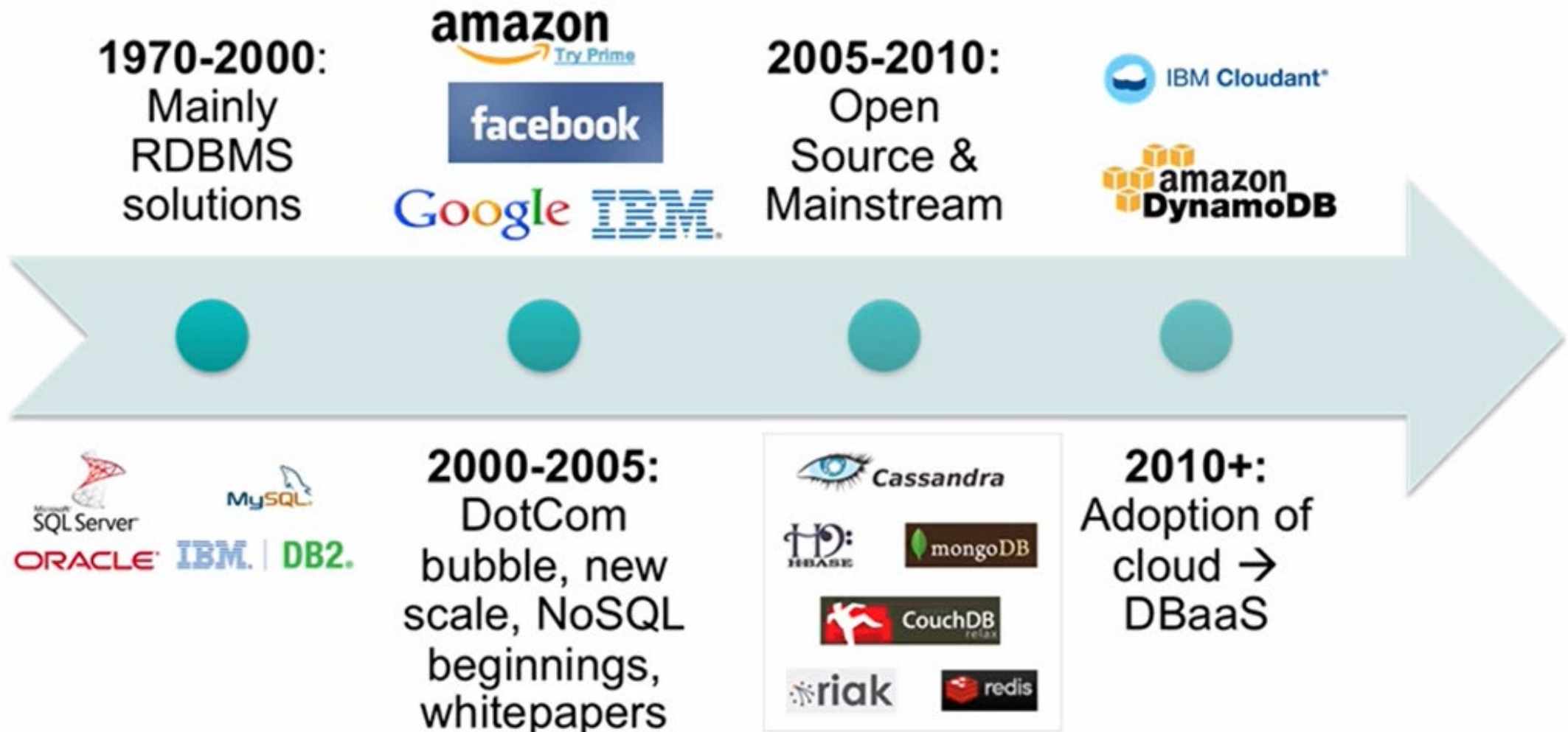
También son mas especializadas para varios casos de uso y pueden ser mucho mas simple desarrollarla.

La llegada de NoSQL y DBaaS

Llego hace 10 Años cuando surgio la necesidad de bases de datos para almacenar datos de una forma diferente y la relacional.

1. Las RDBMS (relacional database) nacieron en los 2000
2. Con la llegada de internet, se necesito servicios para miles y hasta millones, además de como almacenarla y distribuirla en White papers de diferentes empresas como Google.
3. Llegada del open opersource y nuevas comunidades. Algunas DB como apache, mongo, Cassandar, fueron predominantes.
4. Posterior al 2010 las NoSQL se han apalancado los modelos de servicio administrativos o Bases de datos como servicios, para descargar la administración y mantenimiento del usuario final, permitiendo a los desarrolladores centrarse en crear aplicaciones.

La llegada de NoSQL y DBaaS



NoSQL

NoSQL son todas no relacionales, y existe un montón de variedades como:

- Key-Value
- Document
- Big Table/ Column Oriented
- Graph

Que las une como NoSQL?

- Se basan en open source technologies
- Global DB Key → fácil de particionar la data
- Mas casos de uso específicos y mas amigable con el desarrollador que las relacionales
- Esquemas flexibles

Por qué NoSQL?

- Escalabilidad
- Desempeño
- Disponibilidad
- Compatibilidad con la nube
- Costo
- Esquemas flexibles
- Estructuras de datos intuitivas
- Capacidades especializadas

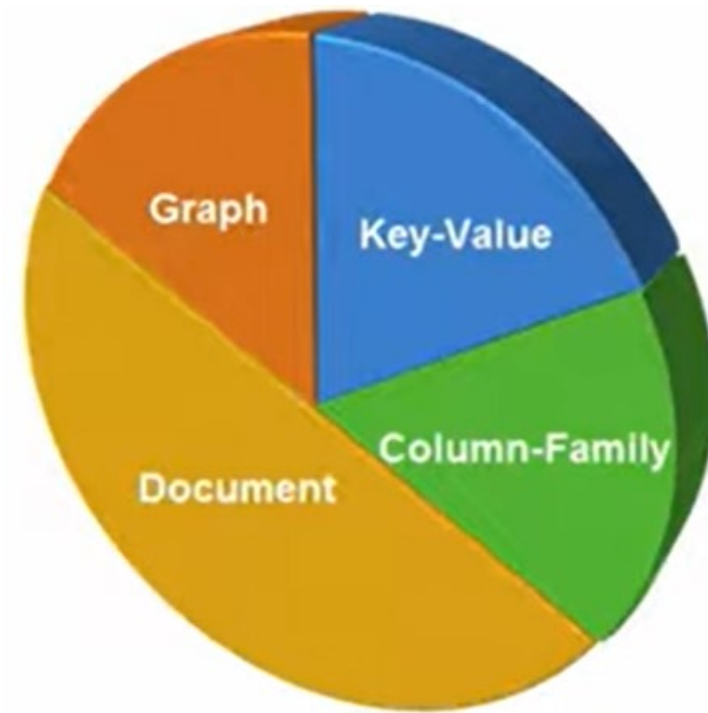
Actividad

- Preguntas

Modulo 2

Tipos De NoSQL

- Key-Value
- Document
- Column-Family
- Graph



Key-Value

- La data se almacena con una única Key en un Hash-Map
- El valor "blob" asociado con una clave no es visible para la base de datos.
- Alto desempeño para operaciones básicas CRUD pero tiene dificultad con búsquedas completas o data interconectada
- Bueno solo para operaciones individuales
- Escalable horizontalmente mediante fragmentación



Key-Value: Casos de uso

Cuándo Usar Key-Value:

- Para una aplicación web y almacenar datos de una sesión
- Uso de perfiles y preferencia
- Carritos de compra en online markets



Cuándo No Usar Key-Value:

- La data esta interconectada (red social, motores de búsqueda...)
- Se requiere consistencia en la información para múltiples operaciones y múltiples keys
- Se necesitan hacer búsquedas en bases complejas de datos

Database Document

- La data es almacenada como un documento (JSON, XML, BSON, CSV,...)
- Documentos son estructuras autodescriptivas y en jerarquía de árbol
- Similar a las Key-Value pero el valor es visible y examinable
- Se usa MapReduce para indexar y analizar
- Usa esquemas de entrada flexible (“Schema-Less”)
- Muy bueno para operaciones simples en el documento
- Horizontalmente escalable a través de fragmentación



Document: Casos de uso

Cuándo Usar Document DB:

- Eventos de logging
- Blogs y aplicativos web
- Operacional y meta data para aplicaciones web/móvil

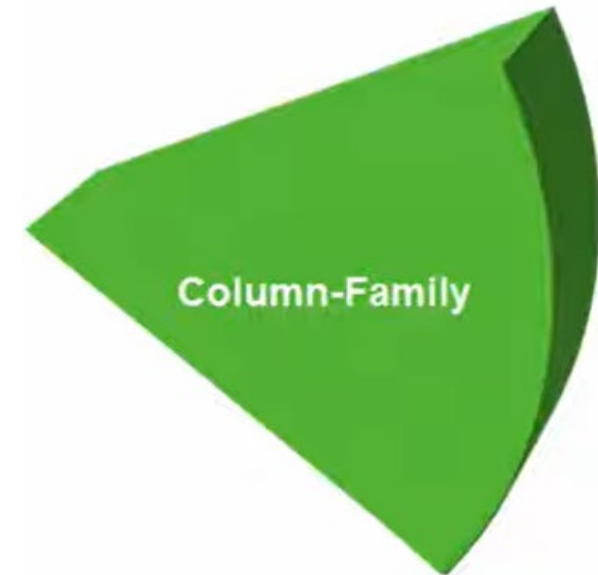
Cuándo No Usar Document DB:

- Se requiere consistencia para operaciones de multiple-documento
- Para data normalizada



Familia de columnas

- La data se almacena en familias de columnas o filas, donde cada fila o columna tiene una key única
- Es similar a las tablas de las bases de datos relacionales
- No es necesario que las filas de la misma familia de columnas contengan las mismas columnas. Las columnas se pueden agregar a cualquiera o a todas las filas de una familia de columnas.
- Representado como a mapa multidimensional
- Bueno para operaciones de filas simples



Familia de columnas: Casos de uso

Cuándo Usar Familia de columnas:

- Eventos de logging
- Blogs
- Contadores
- Caducidad de uso (TTL por columnas)

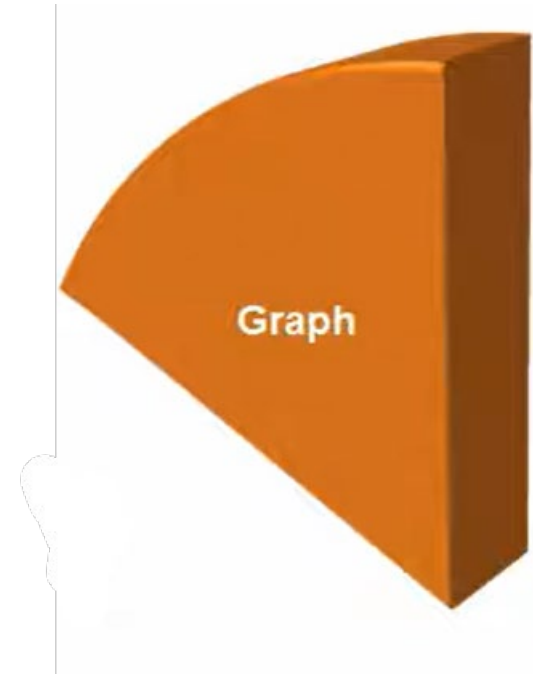


Cuándo No Usar Document DB:

- Lectura y escritura de transacciones ACID (Atomic, Consistency, Isolation, Durability)
- Cuando las columnas son diseñadas para alta volatilidad de datos

Graph DB

- La data se almacena en entidades y relaciones
- Las entidades son nodos y relaciones los bordes
- Relaciones orientadas pero no agregadas como otra NoSQL
- Graph transverso es rápido y eficiente
- No es recomendable fragmentar. Graph transversos con muchos nodos y diferentes maquinas, es difícil y costoso de seguir
- Escala verticalmente
- ACID, similar a RBDMS



Graph DB: Casos de uso

Cuándo Usar Graph DB:

- Data conectada (redes sociales)
- Enrutamiento y aplicaciones espaciales
- Motores de recomendación

Cuándo No Usar Graph DB:

- Se actualizan frecuentemente todos los nodos con nuevos valores y parámetros
- Aplicaciones que necesiten crecer en tamaño



Actividad

- Preguntas
- Carga de datos a database desde script, csv

Modulo 3

Local DB – Hosted DB – DB as a Service

DB: Funcionalidad y desempeño

- Que tipo de DB necesito? Que búsquedas quiero hacer en ella? Cuanto tiempo pretendo esperar la respuesta?

Movil -> Respuestas interactivas

Requiere dataWareHousing (repo unificado con estructura definida) -> BD relacionales

- Que tan grande y cuantas conecciones tendra? Pretendes escalarla horizontalmente?
- Consume mucha RAM? Deberia tener durabilidad el almacenamiento?
- Que requerimiento hay de consistencia y transaccionalidad? (RDB)

Local DB – Hosted DB – DB as a Service

DB: Funcionalidad y desempeño

- Que disponibilidad, replicación de datos y requerimientos de geolocalización necesita?
- Necesita un esquema flexible y de rápido despliegue?
- El modelo de dato puede cambiar en el tiempo?
- Que habilidades debo disponer?
- Se integra fácilmente con la capa de aplicación?
- Quiere se el host yo mismo? Usar La nube o tener un manejo completo?

Preguntas

Modulo 4

IBM Cloudant – una NoSQL DBaaS

Beneficios:

- Fácil de usar pero rico en características
- Usa HTTP API
- Almacenamiento por JSON document
- Esquema Flexible
- Integración con analítica, búsqueda completa de búsqueda, busquedas geo-espacial
- Manejo flexible y monitoreo continuo, además de servicios con expertos de big data
 - Dashboard para manejo
 - Escalamiento up/down
- 24x7x365 Servicio
- Multi atención y clusters dedicados



IBM Cloudant – una NoSQL DBaaS

Retos que intenta resolver para data layer:

- Crecimiento exponencial para usuarios
- Las DB relacionales no son escalables
- Do it yourself
- Administracion sin costos, limitaciones de tiempo o habilidades complejas
- Construir una aplicación desde scratch

IBM Cloudant – una NoSQL DBaaS

Public Cloud, Private Cloud, or connect them together for Hybrid Cloud

Cloudant Managed Guaranteed Performance with Public Cloud DBaaS		Cloudant Local The Power of DBaaS in the privacy of your data centers
Cloudant Dedicated	Cloudant Multi-tenant	
Hosted & Managed by Cloudant	Hosted & Managed by Cloudant	Customer-hosted & managed with <i>Cloudant DevOps tooling</i>
24x7 Premium Support	Community Support	24x7 Premium Support
Use for production deployment , development	Use for development & prototyping	Use for production deployment, development
Monthly, per-node fee	Monthly, metered usage fee	Up-front perpetual license, or monthly, per-node fee
Available @ 30+ SoftLayer, AWS, Rackspace, Azure data centers	Available on SoftLayer, Rackspace, Amazon, Azure	On-premise or cloud platform of choice
Single-tenant clusters	Multi-tenant clusters	Single-tenant clusters

Modulo 5

IBM Cloudant

Add database

- > nombre sin espacios
- > enfocado a JSON document

```
{  
  "_id": "employee123",  
  "department": 23,  
  "name": "Bob Smith",  
  "title": "Developer",  
  "address": {  
    "street": "123 Main St",  
    "city": "Boston",  
    "state": "MA"  
  },  
  "skills": [  
    "Cloudant",  
    "NoSQL",  
    "JSON"  
  ],  
  "active_employee": true  
}
```

```
1 {  
2   "_id": "employee123",  
3   "department": 23,  
4   "name": "Bob Smith",  
5   "title": "Developer",  
6   "address": {  
7     "street": "123 Main St",  
8     "city": "Boston",  
9     "state": "MA"  
10  },  
11  "skills": [  
12    "Cloudant",  
13    "NoSQL",  
14    "JSON"  
15  ],  
16  "active_employee": true  
17 }
```

Podemos compartir los datos con user o por API key

Querying Data- Opciones

CRUD – Document

```
{
  "_id": "badger",
  "min_weight": 7,
  "max_weight": 30,
  "min_length": 0.6,
  "max_length": 0.9,
  "latin_name": "Meles meles",
  "class": "mammal",
  "diet": "omnivore"
}
```

Direct document look
up by `_id`

Primary Index



Exists out-of-the-box;
stored in a b-tree;
Primary key → `doc._id`

Secondary Index (view)



Built using MapReduce;
stored in a b-tree;
Key → user-defined field(s)

Search Index



Built using Lucene FTI;
Any or all fields can be
indexed

GeoSpatial Index



Stored in R* tree;
Lat/Long coordinates
in GeoJSON

Cloudant Query



“Mongo-style” querying;
Built natively in erlang

Manejo de data

GET
PUT
POST
DELETE
COPY



Manejo de data

Insert Data

Read Data

Create Indexes

Make Queries

Monitor the Database

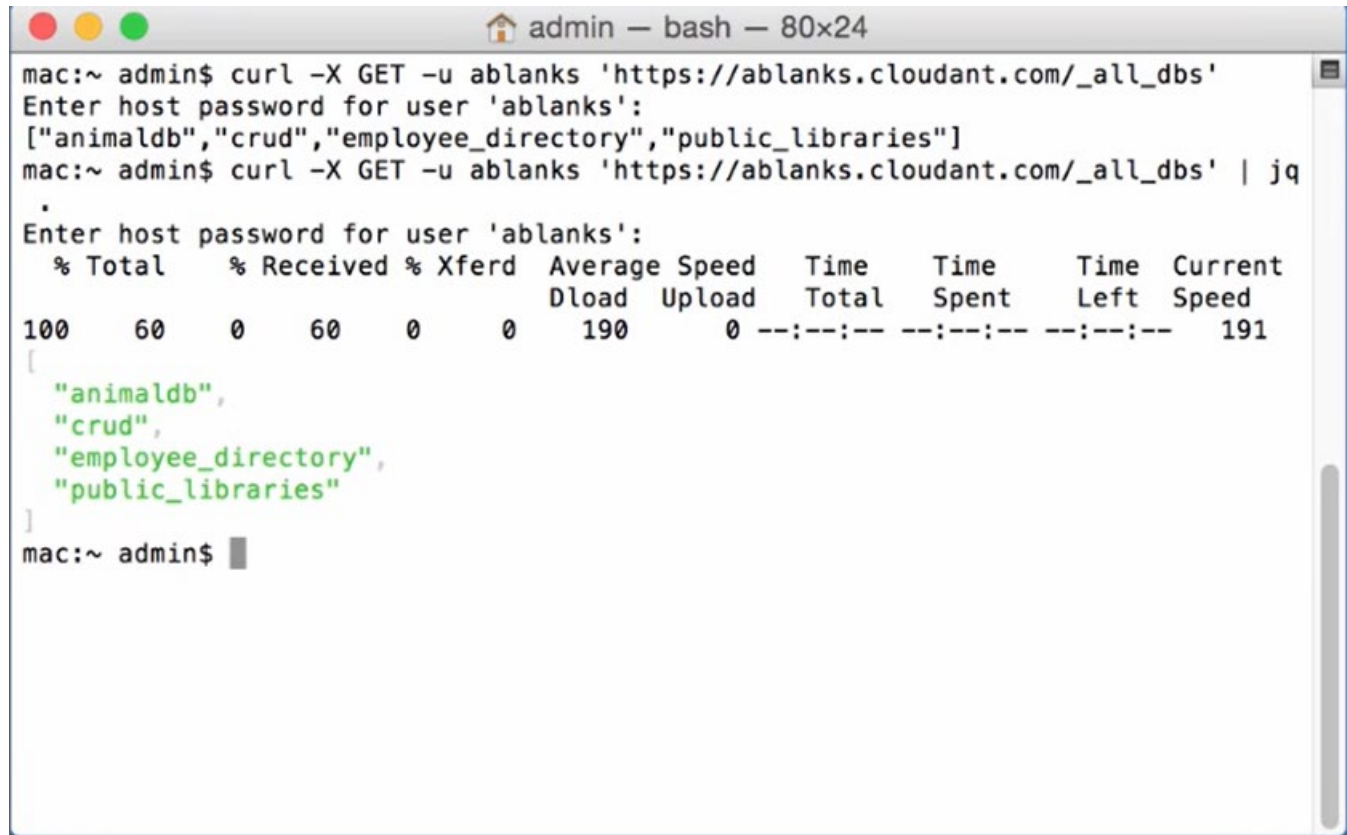
Create Replication Jobs

Create Databases

Manejo de data

HTTP VERB <headers> <data> https://<account>.cloudant.com/<API_endpoint>?<parameters>

```
curl -X GET -u ablanks 'https://ablanks.cloudant.com/_all_dbs'
```

A terminal window titled 'admin — bash — 80x24' showing a curl command and its output. The command is 'curl -X GET -u ablanks 'https://ablanks.cloudant.com/_all_dbs''. The output shows a JSON array of database names: ["animaldb", "crud", "employee_directory", "public_libraries"]. The terminal also shows a progress bar and a jq command being used to format the output.

```
mac:~ admin$ curl -X GET -u ablanks 'https://ablanks.cloudant.com/_all_dbs'
Enter host password for user 'ablanks':
["animaldb","crud","employee_directory","public_libraries"]
mac:~ admin$ curl -X GET -u ablanks 'https://ablanks.cloudant.com/_all_dbs' | jq
.
Enter host password for user 'ablanks':
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent    Left     Speed
100    60      0   60    0     0    190      0  --:--:-- --:--:-- --:--:--   191
[
  "animaldb",
  "crud",
  "employee_directory",
  "public_libraries"
]
mac:~ admin$
```

LAB

- Crea DB
- Hacer Querys
- Accediendo vía Api
- Accediendo vía Browser

<https://priyadogra.com/nosql-and-dbaas-101-cognitive-class-exam-quiz-answers/>

Gracias

- Mas comandos

<https://www.w3schools.com/sql/>