Bases de datos para datos masivos

Contenido de la materia

Tema 1

MÉTODOS DE CAPTURA DE INFORMACIÓN

Origen y calidad de los datos

Evaluación de calidad:
- **Completitud**: grado en el que los valores se encuentran en un conjunto de datos.
- **Credibilidad**: nivel de fiabilidad del organismo que proporciona el conjunto de datos.
- **Consistencia**: grado en el que los datos carecen de contradicciones.
- **Interpretabilidad**: grado en el que los datos deben ser interpretados por una persona.
- Precisión: nivel de exactitud del valor.

Niveles de abstracción:
- **Datos**: conjunto de hechos discretos y objetivos sobre un evento.
- **Información**: datos con significado.
- **Conocimiento**: combinación de información contextualizada, experiencias, valores e intuición.

Organización de los datos

Ficheros planos:
- **CSV** *(comma separated values)*: RFC 4180. Define un registro por línea y separa los campos por comas.
- **JSON**: RFC 7159 y ECMA-404. Describe objetos encapsulados por llaves y listas por corchetes.
- **XML**: descrito en el estándar XML 1.0 de W3C. Permite almacenar información de forma legible utilizando etiquetas.

Bases de datos:
- Conjunto de datos persistentes, utilizados por sistemas de aplicación.
- En el modelo Entidad-Relación (E/R) una entidad es cualquier objeto repensado en la BBDD y un vínculo representa relaciones entre ellos.
- La unidad básica de almacenamiento es el campo, agrupados en registros y estos en ficheros almacenados.

Casos de estudio

Procesamiento de sitio web sobre cursos *online*.

Procesamiento de *logs* de servidor web.

API de acceso a transacciones bancarias.

Almacenamiento de información sobre productos en un fichero CSV.

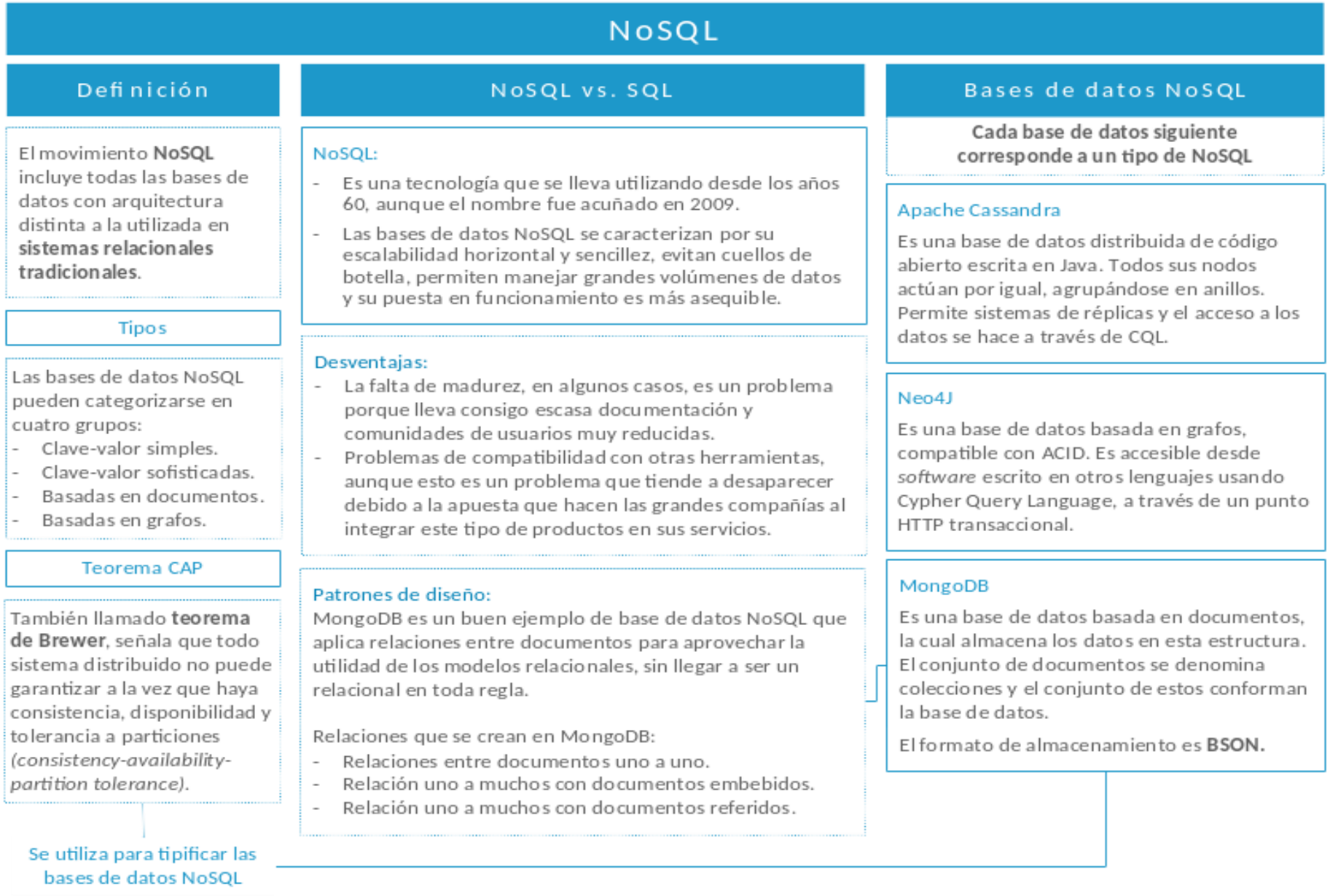
Representación de información geolocalizada en formato JSON.

Almacenamiento de información sobre clientes de una base de datos relacional.

Fuentes de datos:
- Captura manual: encuestas y observaciones.
- Análisis de documentos estructurados: estructurados (HTML) y sin formato (lenguaje natural).
- Salida de aplicaciones: *logs* o bases de datos.
- Sensores: dispositivos de medición.
- Datos de acceso público: gubernamentales y servicios web públicos.

Bases de datos relacionales y SQL:
- Los ficheros almacenados se representan en forma de tablas (relaciones), con columnas (campos) y filas (registros).
- El estándar SQL define un lenguaje para la consulta y modificación de los datos.
- El comando SELECT permite consultar información de tablas.
- Los comandos INSERT, UPDATE y DELETE permiten la inserción, edición y eliminación de registros respectivamente.

Tema 2


NoSQL
Definición
El movimiento **NoSQL** incluye todas las bases de datos con arquitectura distinta a la utilizada en **sistemas relacionales tradicionales**.
Tipos
Las bases de datos NoSQL pueden categorizarse en cuatro grupos:
- Clave-valor simples.
- Clave-valor sofisticadas.
- Basadas en documentos.
- Basadas en grafos.
Teorema CAP
También llamado **teorema de Brewer**, señala que todo sistema distribuido no puede garantizar a la vez que haya consistencia, disponibilidad y tolerancia a particiones *(consistency-availability-partition tolerance)*.
Se utiliza para tipificar las bases de datos NoSQL
NoSQL vs. SQL
NoSQL:
- Es una tecnología que se lleva utilizando desde los años 60, aunque el nombre fue acuñado en 2009.
- Las bases de datos NoSQL se caracterizan por su escalabilidad horizontal y sencillez, evitan cuellos de botella, permiten manejar grandes volúmenes de datos y su puesta en funcionamiento es más asequible.
Desventajas:
- La falta de madurez, en algunos casos, es un problema porque lleva consigo escasa documentación y comunidades de usuarios muy reducidas.
- Problemas de compatibilidad con otras herramientas, aunque esto es un problema que tiende a desaparecer debido a la apuesta que hacen las grandes compañías al integrar este tipo de productos en sus servicios.
Patrones de diseño:
MongoDB es un buen ejemplo de base de datos NoSQL que aplica relaciones entre documentos para aprovechar la utilidad de los modelos relacionales, sin llegar a ser un relacional en toda regla.
Relaciones que se crean en MongoDB:
- Relaciones entre documentos uno a uno.
- Relación uno a muchos con documentos embebidos.
- Relación uno a muchos con documentos referidos.
Bases de datos NoSQL
Cada base de datos siguiente corresponde a un tipo de NoSQL
Apache Cassandra
Es una base de datos distribuida de código abierto escrita en Java. Todos sus nodos actúan por igual, agrupándose en anillos. Permite sistemas de réplicas y el acceso a los datos se hace a través de CQL.
Neo4J
Es una base de datos basada en grafos, compatible con ACID. Es accesible desde *software* escrito en otros lenguajes usando Cypher Query Language, a través de un punto HTTP transaccional.
MongoDB
Es una base de datos basada en documentos, la cual almacena los datos en esta estructura. El conjunto de documentos se denomina colecciones y el conjunto de estos conforman la base de datos.
El formato de almacenamiento es **BSON.**

Tema 3

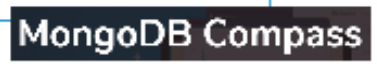
TRATAMIENTO DE DATOS EN MONGODB

Soft ware de apoyo

Existen muchas aplicaciones, tanto internas como externas, que facilitan el uso del MongoDB.

MongoBooster es una herramienta GUI multiplataformas que permite la construcción de consultas fluidas y que posee un gran numero de herramientas muy útiles.

MongoDB Compass es la solución propietaria de este tipo de herramienta.



Flexibilidad del modelo

El esquema de MongoDB es flexible permitiendo modelos de datos totalmente diferentes.

Dos documentos de una colección pueden tener atributos totalmente diferentes.

En la practica, la mayoría de documentos de una colección comparten una estructura similar, aunque esto no es una regla fija.

Inserción de datos

El comando db.< collection>.insert recibe el objeto a añadir en formato JSON.

El comando db .<colecction>.save crea un documento en la *collection* si este no existía previamente.

Lecturas y consultas

El comando principal para la consulta de documentos es db.<collection >.find.

Find recibe dos parámetros: el criterio de búsqueda y la proyección de atributos a mostrar.

El criterio de búsqueda puede incluir operaciones como menor que ($lt) y mayor que ($gt), y en ($in).

Los resultados de una consulta pueden modificarse para ser ordenados, truncados u omitidos.

El comando db.<collection>.findOne permite obtener un solo documento.

Actualización de datos

El comando db.<collection>.save actualiza un documento si ya existe en la colección.

El comando db.<collection>.update permite cambiar atributos específicos de un conjunto de documentos.

El comando update recibe tres parámetros: criterio, acción y opciones.

El comando para eliminar documento es db.<colecction>.remove. Recibe un parámetro con el criterio de búsqueda.

Tema 4

AGREGACIÓN

Concepto

Las funciones de agregación son útiles para agrupar datos y obtener resultados a partir de ellos.

En la base de datos relacionales, este tipo de consultas se realizan con el operador group by (para agrupar) y sum, count, etc., para realizar cualquier operación con sus datos.

Existen dos métodos de agregación en MongoDB: **Map-Reduce** y **Aggregate.**

Existen operaciones especificas de agregación: count, distinct y group.

Map-Reduce

Map-Reduce está formada por dos funciones. Una función *map* que genera los pares clave-valor y otra *reduce*, que opera sobre ellos.

El lenguaje de programación utilizado es JavaScript.

Los parámetros de agregación en la función Map-Reduce se especifican como un objeto JSOM.

La mayor ventaja de la utilización de este método de agregación es la potencia que ofrece, ya que se pueden personalizar totalmente cada una de sus funciones.

Aggregati on Framework

El *framework* de agregación está modelado en varias etapas que transforman los documentos en un resultado agregado.

Las etapas mas usadas son la $mach, encargada de la búsqueda, y $group, la que agrupa y hace los cálculos.

Es obligatorio especificar el campo _id en el nuevo documento, ya que estos son los valores a agrupar.

Además de los operadores de búsqueda, tiene muchos otros que hacen la vida mas fácil al trabajar, *array*, *string*, etc.

Tema 5

GESTIÓN DE MONGODB

Seguridad

Autenticación y listas de acceso: permiten controlar permisos de interacción con bases de datos.

Para restringir conexiones desde una interfaz en específico se debe indicar la dirección IP en el parámetro –Bind_ip

La seguridad del entorno MongoDB puede incrementarse con el uso de un *firewall*.

Respaldo

La herramienta **Mongodump** se utiliza para crear una copia de respaldo de una base de datos en MongoDB.

La herramienta **Montorestore** permite restaurar una copia de respaldo en el sistema MongoDB.

MongoDB brinda capacidades de replicación, para lograr reducirse e incrementar la capacidad.

Rendimiento

Los índices permiten ejecutar consultas de forma eficiente.

Este aumento de rendimiento se ve reflejado en el número de documentos analizados y, por consiguiente, en el tiempo de respuesta.

Sharding

Sharding: método utilizado por MongoDB para distribuir datos en varios servidores y aumentar así su capacidad.

Cada *shard* está encargado de almacenar la información. Cada *shard* es un *replica set*, por lo que brinda alta disponibilidad y consistencia en datos.

Tema 6

DRIVERS DE CONEXIÓN

Conceptos

Conexión

MongoDB puede integrarse con otros lenguajes de programación.

Lenguajes

C
C++
C#
Java
Node.js
Perl
PHP
Python
Motor
Ruby
Scalar
Go
Erlang

Ejemplos de uso

Lenguajes de ejemplo

1 Java

Maven

```
<dependencies> <dependency>
<groupId>org.mongodb</groupId> <artifactId>mongodb-
driver</artifactId> <version>3.4.2</version> </dependency>
</dependencies>
```

2 Neo4J

NPM

```
npm install mongodb –sabe
npm install bson-ext --save
```

3 Python

PyMongo

```
pip install pymongo
```

Tema 7

Apache Cassandra

Definición

Cassandra es un producto de Apache y su sistema de almacenamiento o motor de base de datos es de código abierto, distribuido y descentralizado. Su fin principal es gestionar grandes volúmenes de datos estructurados repartidos por distintos servidores.

Base de datos NoSQL de tipo clave-valor

Cómo aplica el teorema CAP

Conocer los principales componentes de Cassandra y describir el sistema de replicación, la lógica de las operaciones de escritura y las de lectura.

Generalidades

Arquitectura:

- Cassandra está diseñada para trabajar en entornos *big data* sobre múltiples nodos evitando puntos de fallo.
- Dispone de un sistema distribuido de igual a igual en todos sus nodos donde los datos se distribuyen entre todo el clúster.
- Los nodos de un clúster pueden aceptar solicitudes de lectura y escritura, independientemente de dónde se encuentren realmente los datos en dicho clúster.
- Si un nodo falla, las solicitudes de lectura/escritura pasan a ser atendidas por otros nodos de la red.

Modelo de datos:

- Los patrones de acceso a los datos y las consultas de la aplicación determinan la estructura y organización de los datos que luego se utilizan para diseñar las tablas de la base de datos.
- Cada consulta está respaldada por una tabla y, por ello, los datos se duplican en varias tablas en un proceso conocido como desnormalización.
- La duplicación de datos y un alto rendimiento de escritura se utilizan para lograr un alto rendimiento de lectura.

CQL3

***Keyspaces*, tablas y columnas**

Manipulación de datos

- Tipos de datos.
- Contadores.
- Uso de *timestamp*.
- Uso de *date*.
- Definir duraciones.
- Colecciones: *maps*, *set* y *list*.
- Tuplas.
- UDT.
- Operaciones de INSERT, UPDATE, DELETE.

Otras funcionalidades

- TTL.
- WriteTime.
- Índices.

Instalación sobre Linux y Windows

Tema 8

Neo4j

Definición

Neo4j es una base de datos de grafos cuyo lenguaje de consultas es CQL (Cypher Query Language). Neo4j está escrito en Java.

Base de datos NoSQL de tipo grafo

Cómo aplica el teorema CAP

Los principales elementos del modelo de datos Graph DB son:

- Nodos.
- Relaciones.
- Propiedades.

Principales componentes

Nodo:

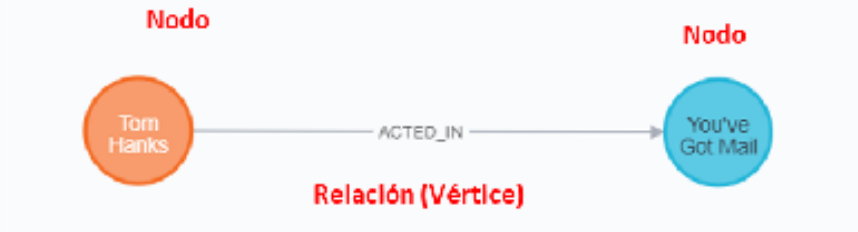
- Las entidades reciben el nombre de nodos y cada uno de ellos es muy similar a una instancia de un objeto (es decir, tienen propiedades).

Relaciones:

- Las relaciones, por su parte, se conocen como vértices y también tienen propiedades, siendo la dirección o sentido el más importante de ellos. Una relación conecta dos nodos y los organizan en estructuras.

Propiedades:

- Nodo y relaciones permiten organizar los datos de tal manera que sea posible encontrar patrones de información existente entre dichos nodos.
- Las propiedades son pares de nombre-valor utilizados para agregar cualidades a los nodos y las relaciones.



CQL (Cypher Query Language)

Esquemas, grafos y Cypher Query Language

Manipulación del grafo

- Tipos de datos y operadores.
- Nodos.
- Relaciones.
- Propiedades.
- Modelado.
- Cláusulas CREATE, MATCH, INSERT, DELETE, REMOVE, WHERE, SKIP, LIMIT.

Caso de uso: fraude en tarjetas de crédito

Tema 9

Redis

Definición

Redis es un popular servidor de bases de datos de múltiples modelos. Redis va más allá de la base de datos NoSQL para proporcionar varias capacidades avanzadas que necesitan las aplicaciones modernas.

Base de datos NoSQL de tipo clave-valor en memoria

Cómo aplica el teorema CAP

Los principales elementos del modelo de datos Redis son:

- Tipos de datos.
- Estructuras.
- Módulos.

Principales conceptos

Base de datos multimodelo:

- Redis proporciona una funcionalidad completa de varios modelos a través de sus módulos. El uso de Redis como una base de datos de múltiples modelos permite una mayor flexibilidad para los desarrolladores de aplicaciones dentro de una organización.

Almacenamiento en memoria:

- Redis mantiene los datos en la memoria para un acceso rápido y persiste los datos en el almacenamiento, así como la replicación de los contenidos en la memoria para escenarios de producción de alta disponibilidad.

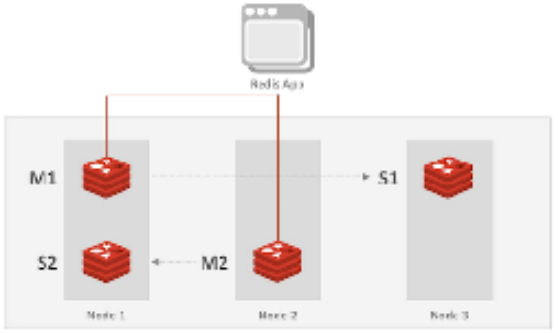
Casos de uso/estructura:

- Cada estructura de datos tiene un caso de uso o escenario diferente para el que se adapta mejor.
- Además de las estructuras de datos, Redis también admite el patrón Publicar/Suscribir (Pub/Sub) y patrones adicionales que hacen que sea adecuada para aplicaciones modernas con uso intensivo de datos.

redislabs
HOME OF REDIS

Redis-cli

Clúster, nodo, tipos de datos, estructuras y módulos



Redis® Cluster

Estructuras de datos

- *Strings.*
- *Lists.*
- *Sets.*
- *Sorted sets.*
- *Hashes.*
- *Bit arrays.*
- *Streams.*
- *HyperLogLogs.*

Módulos

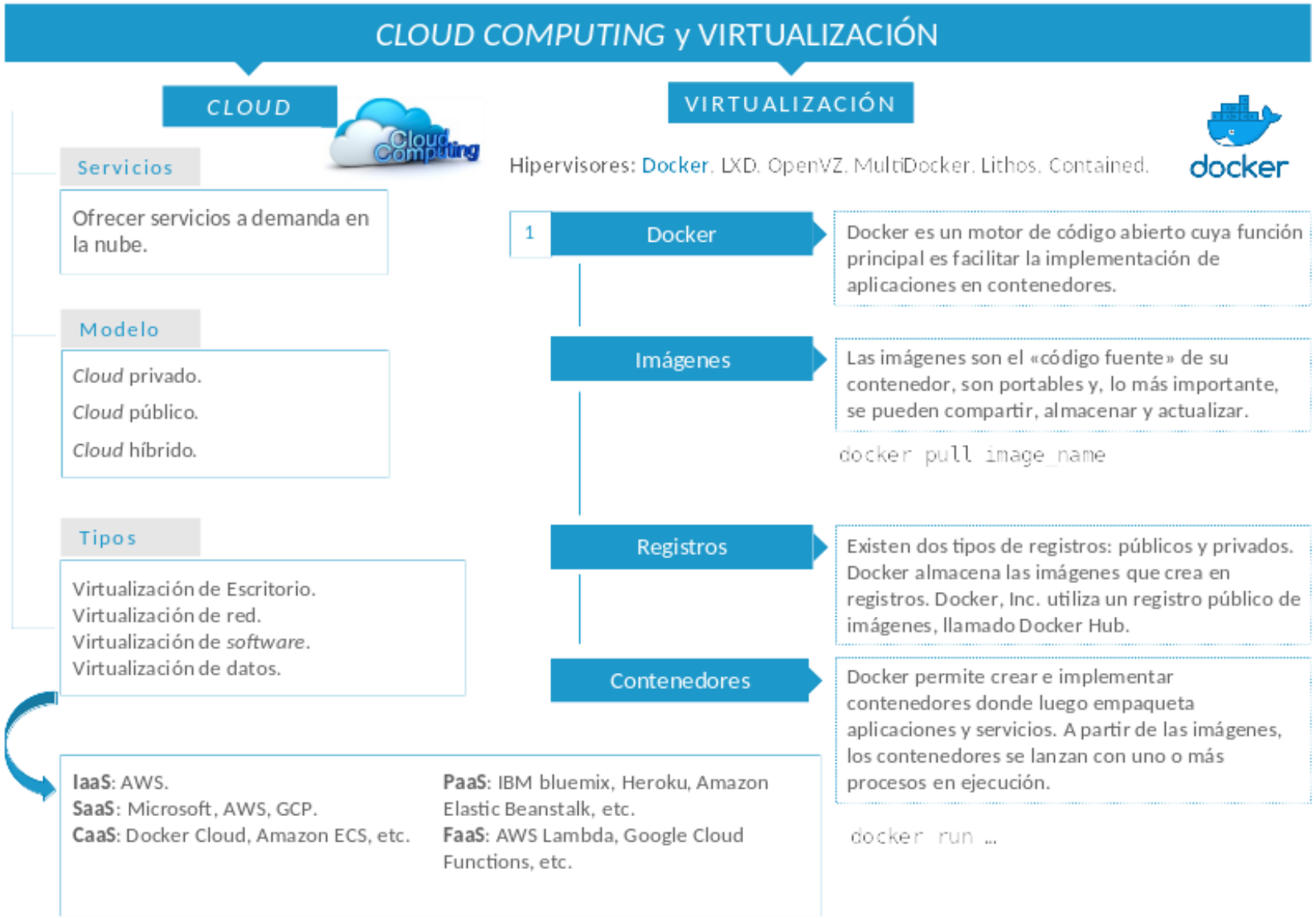
- Redis Graph.
- RedisSearch.
- Redis TimeSeries.
- RedisJSON.

Drivers

Python, Java, PHP, NodeJS, C, C#.

Instalación entorno de test con Docker

Tema 10


CLOUD COMPUTING y VIRTUALIZACIÓN
CLOUD
Servicios
Ofrecer servicios a demanda en la nube.
Modelo
Cloud privado.
Cloud público.
Cloud híbrido.
Tipos
Virtualización de Escritorio.
Virtualización de red.
Virtualización de software.
Virtualización de datos.
IaaS: AWS.
SaaS: Microsoft, AWS, GCP.
CaaS: Docker Cloud, Amazon ECS, etc.
PaaS: IBM bluemix, Heroku, Amazon Elastic Beanstalk, etc.
FaaS: AWS Lambda, Google Cloud Functions, etc.
VIRTUALIZACIÓN
Hipervisores: Docker. LXD. OpenVZ. MultiDocker. Lithos. Contained.
1
Docker
Docker es un motor de código abierto cuya función principal es facilitar la implementación de aplicaciones en contenedores.
Imágenes
Las imágenes son el «código fuente» de su contenedor, son portables y, lo más importante, se pueden compartir, almacenar y actualizar.
docker pull image_name
Registros
Existen dos tipos de registros: públicos y privados. Docker almacena las imágenes que crea en registros. Docker, Inc. utiliza un registro público de imágenes, llamado Docker Hub.
Contenedores
Docker permite crear e implementar contenedores donde luego empaqueta aplicaciones y servicios. A partir de las imágenes, los contenedores se lanzan con uno o más procesos en ejecución.
docker run ...