

# DynamoDB



# Índice

<b>Introducción a las bases de datos NoSQL/ no relacionales.....</b>	<b>2</b>
<b>Diferencias entre BBDD No relacionales y relacionales.....</b>	<b>3</b>
<b>DynamoDB.....</b>	<b>4</b>
Introducción.....	4
Características principales.....	5
Almacenamiento.....	5
Disponibilidad y durabilidad.....	5
Ventajas y desventajas de su uso frente a otros SGBDs.....	7
Casos de uso exitosos.....	9
Guía de instalación y despliegue de la BD que se utiliza en la aplicación a desarrollar.....	10
Requisitos para usar DynamoDB:.....	10
Instalación de DynamoDB(En local).....	10
Uso de DynamoDB(AWS CLI):.....	11
<b>Conclusiones e impresiones.....</b>	<b>12</b>
<b>Bibliografía y referencias.....</b>	<b>13</b>
<b>Anexos.....</b>	<b>14</b>
<b>Glosario.....</b>	<b>15</b>

# Introducción a las bases de datos NoSQL/ no relacionales

Las bases de datos NoSQL o no relacionales son sistemas de almacenamiento de datos diseñados para manejar grandes volúmenes de información de forma distribuida y escalable. A diferencia de las bases de datos relacionales tradicionales no utilizan tablas con esquemas fijos ni relaciones estrictas entre datos lo que permite mayor flexibilidad y mejor rendimiento en aplicaciones modernas como sistemas web móviles o de alta concurrencia.

DynamoDB es una base de datos NoSQL ofrecida por Amazon Web Services que pertenece al modelo clave valor y también soporta estructuras tipo documento. Es un servicio totalmente administrado lo que significa que no requiere configuración ni mantenimiento de servidores y ofrece alta disponibilidad y baja latencia de forma automática. DynamoDB permite escalar fácilmente según la demanda y es especialmente útil para aplicaciones que necesitan respuestas rápidas y consistentes en tiempo real.

# Diferencias entre BBDD No relacionales y relacionales

## 1. Modelo de datos

### Relacionales (SQL)

- Basadas en tablas (filas y columnas).
- Esquema rígido y predefinido.
- Relaciones entre tablas mediante claves primarias y foráneas.

### No relacionales (NoSQL)

- No usan tablas tradicionales.
- Esquema flexible o inexistente.
- Modelos comunes:
  - Documentos (JSON/BSON)
  - Clave–valor
  - Columnas anchas
  - Grafos

## 2. Lenguaje de consulta

### Relacionales

- Usan SQL(Structured Query Language)
- Consultas complejas con JOIN, subconsultas, agregaciones

### No relacionales

- No hay un lenguaje estándar único
- Consultas específicas del motor (Mongo DB Query, CQL, Gremlin, etc)

## 3. Integridad y consistencia

### Relacionales

- cumplen ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)
- Alta integridad de los datos.

### No relacionales

- Suelen seguir BASE (Basically Available, Soft state, Eventually consistent)
- Consistencia eventual en muchos casos

## 4. Escalabilidad

### Relacionales

- Escalado vertical (mejorar hardware).
- Escalado horizontal más complejo.

### No relacionales

- Escalado horizontal nativo (añadir nodos).
- Diseñadas para sistemas distribuidos.

## 5. Rendimiento

## Relacionales

- Muy eficientes en datos estructurados y relaciones complejas.
- El rendimiento puede degradarse con grandes volúmenes distribuidos.

## No relacionales

- Optimizadas para grandes volúmenes de datos y alta concurrencia.
- Mejor rendimiento en lecturas/escrituras masivas.

# DynamoDB

## Introducción

### Características principales

DynamoDB es una base de datos NoSQL de clave-valor y documentos totalmente administrada y sin servidor, pensada para aplicaciones de alto rendimiento y gran escalabilidad. Sus características principales son:

#### **Almacenamiento**

Almacena datos en forma de tablas, en las cuales cada elemento es un conjunto de atributos sin esquema fijo. Admite modelos clave-valor al igual que modelos tipo documento, usado para datos semiestructurados o escritos en lenguaje JSON.

#### **Disponibilidad y durabilidad**

Replica los datos en al menos 3 zonas de disponibilidad de AWS. DynamoDB hace copias de seguridad constantemente para asegurar una recuperación a un momento dado (*point-in-time recovery*).

## Rendimiento y baja latencia

Esta base de datos permite usar DynamoDB Accelerator (DAX), un servicio de caché que reduce la latencia a microsegundos, incluso a gran escala.

## Escalabilidad

Para ajustarse a la carga, tanto en la capacidad de lectura o escritura, DynamoDB funciona en modalidad serverless (sin servidor), se escala de forma automática ya que previamente has definido la capacidad deseada (aprovisionada o bajo demanda)

## Tablas

Permite crear tablas globales con replicación activa-activa entre las diferentes regiones de AWS. Dichas tablas, soportan transacciones ACID entre una o varias tablas.

## Seguridad y operabilidad

Esta base de datos usa Dynamo Streams, una funcionalidad que recoge en tiempo real los cambios en los datos de las tablas (insert, update y delete). Incluye también un cifrado en reposo con controles de IAM, VPC Endpoint (privado) y los propios servicios de logging de AWS (CloudWatch y CloudTrail)

## Ventajas y desventajas de su uso frente a otros SGBDs

### Ventajas

#### 1. Escalabilidad automática:

- No necesitas configurar servidores
- Escala horizontalmente sin límite práctico
- Ideal para picos de tráfico (Black Friday, apps virales)

#### 2. Rendimiento muy alto y estable:

- Latencias de milisegundos de un sólo dígito
- Rendimiento predecible incluso con mucha carga

#### 3. Totalmente gestionado:

- No hay mantenimiento
- Backups automáticos
- Replicación multi-región

#### 4. Alta disponibilidad:

- Replicación automática en múltiples zonas de AWS
- Tolerancia a fallos integrada

#### 5. Flexible (NoSQL):

- No requiere esquema rígido
- Cada ítem puede tener atributos distintos

### Desventajas:

#### 1. No es relacional

- No hay JOIN
- No hay integridad referencial
- Modelar relaciones es más complejo

## 2. Curva de aprendizaje

- El diseño depende de los patrones de acceso
- Mal diseño = mal rendimiento
- Se diseñan los datos según las consultas que se van a hacer, no según la estructura de tablas

## 3. Coste

- Puede ser caro si no se optimiza
- Se paga por lecturas, escrituras y almacenamiento

## 4. Consultas limitadas

- No puedes consultar cualquier campo libremente
- Necesitas índices secundarios (GSI/LSI)

## Casos de uso exitosos

- **Disney Plus:** Utiliza DynamoDB en conjunto con Kinesis para ingerir contenido, metadatos y acciones de clientes(medidas en miles de millones), entre otras utilidades explota las listas, para las listas de observación de los usuarios.
- **Duolingo:** Duolingo guarda en DynamoDB cerca de 31 billones de datos en su servicio web con datos de cerca de 18 millones de usuarios mensuales. Para ello utiliza 24000 unidades de lectura y 3300 de lectura por segundo.
- **AirBnb:** AirBnb usa DynamoDB para escalar sus operaciones a una base de datos global mientras optimiza su procesamiento en tiempo real
- **Amazon:** Como no podía faltar amazon utiliza DynamoDB para obtener respuestas en milisegundos, permitiendo respuestas más rápidas  
Según Mike Thomas, Software Development Manager - Amazon Herd:  
“Si siguieran con la arquitectura antigua, habrían necesitado 1.000 servidores con Oracle para soportar Prime Day. Al usar Amazon DynamoDB, pueden evitar gestionar y escalar tantos servidores, centrarse en desarrollar nuevas funcionalidades y mejorar la experiencia de los clientes.”
- **NFL:** La NFL utiliza DynamoDB para procesar y almacenar sus datos de Next Gen Stats para que puedan ser consumidos por gráficos en pantalla, APIs y herramientas utilizadas por su equipo de medios. Obtén más información en NFL on AWS.

# Guía de instalación y despliegue de la BD que se utiliza en la aplicación a desarrollar

Requisitos para usar DynamoDB:

1. AWS CLI(Amazon Web Service Command Line Interface), el cliente de amazon web service ([Guia oficial de AWS](#))

Para instalarlo en Windows basta con ejecutar el comando

```
msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2-2.0.30.msi
```

Para comprobar que está bien instalado usamos el comando `aws --version`

2. [Java 17](#) o mayor

Instalación de DynamoDB(Descarga por consola)

1. Descargar DynamoDB Local, se puede descargar a través de su [página oficial](#)
2. Al descargar del link nos descargara una carpeta, se puede poner donde más convenga al usuario
3. Accedemos a la carpeta con la powershell o cmd y utilizamos el comando:

Powershell: `java -D"java.library.path=./DynamoDBLocal_lib" -jar`

`DynamoDBLocal.jar`

CMD: `java -Djava.library.path=./DynamoDBLocal_lib -jar DynamoDBLocal.jar`

`-sharedDb`

Por defecto DynamoDB utilizara el puerto 8000, por lo que si está ocupado habrá que modificar el puerto o liberarlo.

4. Configurar las credenciales de DynamoDB con AWS CLI utilizando el comando `aws configure`

## Despliegue(AWS CLI Local):

1. Posicionarse en la dirección donde se ha descargado la carpeta:

```
cd downloads
```

2. Ejecutar el comando:  
`java -Djava.library.path=./DynamoDBLocal_lib -jar`

```
DynamoDBLocal.jar -sharedDb
```

Añadiendo lo siguiente se puede cambiar el puerto en que se conecta la base de datos:  
`-port 9000`

## Despliegue(AWS CLI Servidor)

1. Configurar la nube, en nuestro caso desde el laboratorio de AWS Academy:

- a. Accedemos a **CloudShell**.

- b. Configurar las credenciales de DynamoDB con AWS CLI utilizando el comando **aws configure**

## Despliegue(Docker):

1. Abrir docker y abrir la carpeta donde deseamos que corra el proyecto
2. Crear el fichero docker-compose.yml con el contenido:

```
services:  
  dynamodb-local:  
    command: "-jar DynamoDBLocal.jar -sharedDb -dbPath ./data"  
    image: "amazon/dynamodb-local:latest"  
    container_name: dynamodb-local  
    ports:  
      - "8000:8000"  
    volumes:  
      - "./docker/dynamodb:/home/dynamodblocal/data"  
    working_dir: /home/dynamodblocal
```

3. Ejecutar el comando: **docker compose up**

## Conclusiones e impresiones

## Bibliografía y referencias

- Amazon Web Services. (s. f.). Getting started with the AWS CLI version 2. AWS Documentation.  
<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-version.html>
- Amazon Web Services. (s. f.). Accessing Amazon DynamoDB. AWS Documentation.  
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AccessingDynamoDB.html#Tools.CLI>
- Audiovisual451. (s. f.). Disney utiliza la infraestructura de nube de Amazon Web Services para su despliegue mundial.  
<https://www.audiovisual451.com/disney-utiliza-la-infraestructura-de-nube-de-amazon-web-services-para-su-despliegue-mundial/>
- GeeksforGeeks. (2025, julio 12). *DynamoDB: Understand the benefits with real life use cases.*  
<https://www.geeksforgeeks.org/blogs/dynamodb-understand-the-benefits-with-real-life-use-cases/>

## Anexos

[https://docs.aws.amazon.com/pdfs/amazondynamodb/latest/developerguide/dynamo  
db-dg.pdf](https://docs.aws.amazon.com/pdfs/amazondynamodb/latest/developerguide/dynamodb-dg.pdf)

## Glosario