

Lectura sesión Base de Datos DynamoDB

 por Kibernet Capacitación S.A.

Amazon DynamoDB: Definición y Características

Amazon DynamoDB es un servicio de base de datos NoSQL totalmente gestionado por AWS, diseñado para aplicaciones que requieren baja latencia, alta disponibilidad y escalabilidad automática.

A diferencia de otras bases de datos que requieren gestión manual del servidor, backups, configuración y escalabilidad, DynamoDB es serverless: AWS se encarga de todo el backend, y tú solo te preocupas de los datos y las consultas.

Características de DynamoDB

Característica	Descripción
Rendimiento en milisegundos	Lectura y escritura con latencia muy baja. Ideal para tiempo real.
Escalabilidad automática	Escala según la carga (escrituras, lecturas, almacenamiento).
Alta disponibilidad	Multizona y multirregión con respaldo automático.
Totalmente gestionado	No necesitas administrar servidores ni parches.
Event-driven	Se integra con AWS Lambda, Streams, EventBridge.
Modelo clave-valor/documental	Guarda documentos como JSON, pero orientado a clave-partición.

Ventajas y Desventajas de DynamoDB

Ventajas

- No necesitas administrar infraestructura.
- Extremadamente rápida.
- Se integra fácilmente con otros servicios de AWS.
- Excelente para workloads impredecibles (gaming, IoT, e-commerce).

Desventajas

- No tiene soporte para relaciones complejas (sin joins).
- Las consultas son limitadas al diseño de clave-partición.
- Costos pueden escalar rápidamente si no se configura con cuidado.

Casos de uso comunes de DynamoDB

Caso de uso	Por qué DynamoDB es adecuado
E-commerce	Consulta de productos y pedidos en tiempo real.
Videojuegos	Guardar puntuaciones, progreso, perfiles de jugador.
Apps móviles	Sincronización de datos por usuario.
IoT	Captura de datos de sensores con alta frecuencia.

Online Shopping Post-P
IoT In Ecommerce



Gestión de Tablas en DynamoDB: Conceptos clave

- **Primary Key** (clave primaria): puede ser simple o compuesta.
 1. **Simple**: solo una partition key.
 2. **Compuesta**: partition key + sort key.
- **Item**: Documento único dentro de una tabla (equivalente a una fila).
- **Attribute**: Equivale a una columna.
- **Provisioned vs On-demand**: Opciones de capacidad de escritura/lectura.

Crear, Cargar, Consultar y Eliminar Tablas

Crear tabla

En consola de AWS o vía AWS CLI / SDK. Aquí un ejemplo con AWS CLI:

```
aws dynamodb create-table \  
  --table-name Usuarios \  
  --attribute-definitions AttributeName=id,AttributeType=S \  
  --key-schema AttributeName=id,KeyType=HASH \  
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```

Cargar datos en la tabla

Puedes hacerlo mediante scripts en Python (Boto3), SDK, consola, o con archivos .json en lote.

Crear, Cargar, Consultar y Eliminar Tablas

Consultar tabla

- Scan: recorre todos los ítems (costoso).
- Query: busca por claves específicas (eficiente).

```
aws dynamodb query \  
  --table-name Usuarios \  
  --key-condition-expression "id = :v1" \  
  --expression-attribute-values '{":v1":{"s":"123"}}'
```

Eliminar tabla

```
aws dynamodb delete-table --table-name Usuarios
```


Manipulación de Datos en DynamoDB

- Crear ítem (Create)

```
aws dynamodb put-item \  
  --table-name Usuarios \  
  --item '{  
    "id": {"S": "123"},  
    "nombre": {"S": "Miguel"},  
    "correo": {"S": "miguel@example.com"}  
  }'
```

- Obtener ítem (Get)

```
aws dynamodb get-item \  
  --table-name Usuarios \  
  --key '{"id": {"S": "123"}}'
```

Manipulación de Datos en DynamoDB

- Actualizar ítem (Update)

```
aws dynamodb update-item \  
  --table-name Usuarios \  
  --key '{"id": {"S": "123"}}' \  
  --update-expression "SET correo = :c" \  
  --expression-attribute-values '":c":{"S":"nuevo@email.com"}}'
```

- Eliminar ítem (Delete)

```
aws dynamodb delete-item \  
  --table-name Usuarios \  
  --key '{"id": {"S": "123"}}'
```

Escritura y Lectura por Lotes en DynamoDB

Batch Write

```
aws dynamodb batch-write-item \  
--request-items '{  
  "Usuarios": [  
    {  
      "PutRequest": {  
        "Item": {  
          "id": {"S": "124"},  
          "nombre": {"S": "Laura"}  
        }  
      },  
    },  
    {  
      "PutRequest": {  
        "Item": {  
          "id": {"S": "125"},  
          "nombre": {"S": "Pedro"}  
        }  
      }  
    }  
  ]  
}'
```

Escritura y Lectura por Lotes en DynamoDB

Batch Get

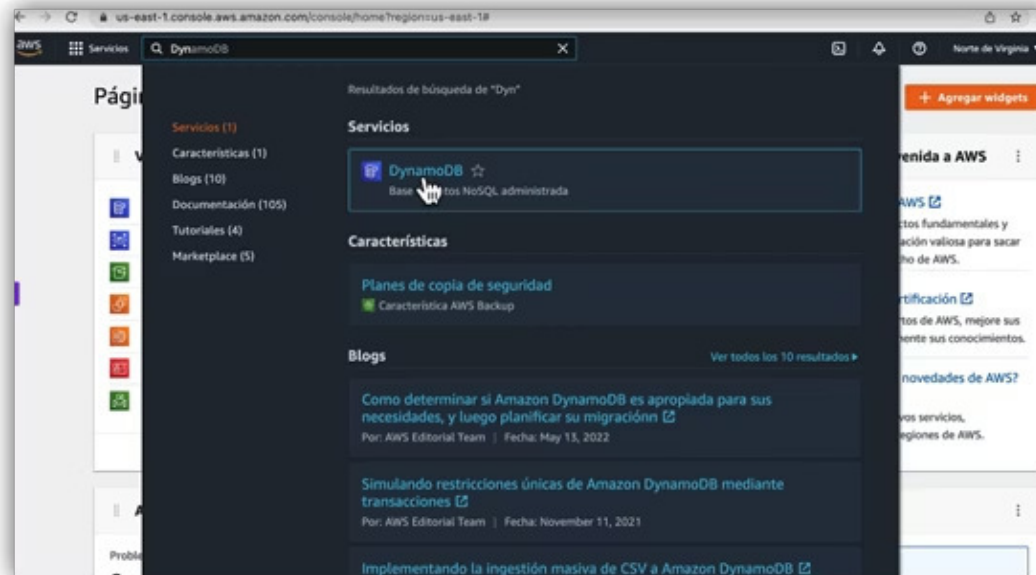
```
aws dynamodb batch-get-item \  
  --request-items '{  
    "Usuarios": {  
      "Keys": [  
        {"id": {"S": "124"}},  
        {"id": {"S": "125"}}  
      ]  
    }  
  }'
```

Implementación en la Nube: ¿Dónde se ejecuta DynamoDB?

- Se ejecuta dentro de AWS, y está disponible en todas las regiones.
- Puedes usar la consola web, AWS CLI, SDKs (Node.js, Python, Java, etc.).
- También puedes trabajar de forma local para desarrollo con DynamoDB Local.

Ejecutar en AWS (pasos básicos)

- Ir a <https://console.aws.amazon.com/dynamodb>
- Buscamos dynamodb



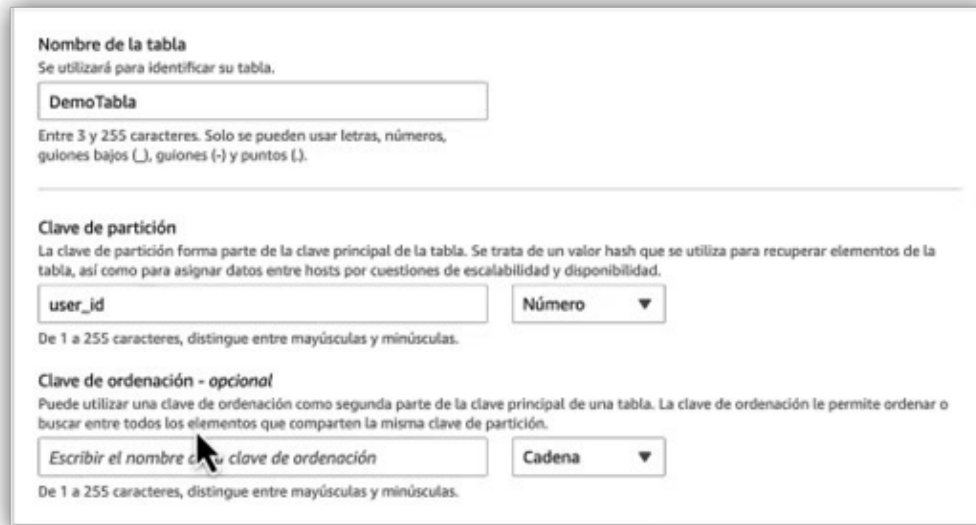
Implementación en la Nube: ¿Dónde se ejecuta DynamoDB?

- Crear una tabla



Implementación en la Nube: ¿Dónde se ejecuta DynamoDB?

- Definir:



Nombre de la tabla
Se utilizará para identificar su tabla.

DemoTabla

Entre 3 y 255 caracteres. Solo se pueden usar letras, números, guiones bajos (_), guiones (-) y puntos (.).

Clave de partición
La clave de partición forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla, así como para asignar datos entre hosts por cuestiones de escalabilidad y disponibilidad.

user_id Número ▼

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Clave de ordenación - opcional
Puede utilizar una clave de ordenación como segunda parte de la clave principal de una tabla. La clave de ordenación le permite ordenar o buscar entre todos los elementos que comparten la misma clave de partición.

Escribir el nombre de la clave de ordenación Cadena ▼

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Implementación en la Nube: ¿Dónde se ejecuta DynamoDB?

- Clave primaria (partition key)
- Clave de ordenamiento opcional (sort key)
- Configurar modo de capacidad (On-Demand o Provisioned)

Configuración de la tabla

☒ **Configuración predeterminada**
La forma más rápida de crear la tabla. Puede modificar estos ajustes ahora o después de que se haya creado la tabla.

☐ **Personalizar configuración**
Utilice estas características avanzadas para que DynamoDB funcione mejor de acuerdo a sus necesidades.

Configuración de la tabla predeterminada

Estos son los ajustes predeterminados de la nueva tabla. Puede cambiar algunos de estos ajustes después de crear la tabla.

Ajuste	Valor	Se puede editar después de la creación
Modo de capacidad	Aprovisionado	Sí
Capacidad de lectura	5 RCU	Sí
Capacidad de escritura	5 WCU	Sí
Auto Scaling	Activado	Sí
Índices secundarios locales	-	No
Índices secundarios globales	-	Sí
Administración de claves de cifrado	Propiedad de Amazon DynamoDB	Sí

Etiquetas

Las etiquetas son pares de claves y valores opcionales que puede asignar a los recursos de AWS. Puede utilizar etiquetas para controlar el acceso a los recursos o realizar un seguimiento de los gastos en AWS.

No hay etiquetas asociadas al recurso.

Puede agregar 50 etiquetas más.

Creando la tabla DemoTabla. Estará disponible en breve.

DynamoDB > Tablas

Tablas (1) Información

Acciones

<input type="checkbox"/>	Nombre	Estado	Clave de partición	Clave de ordenación	Índices	Modo de capacidad de lectura	Modo de capacidad de escritura
<input type="checkbox"/>	DemoTabla	Creando	user_id (N)	-	0	Aprovisionado con Auto Scaling (5)	Aprovisionado

Insertar datos en DynamoDB usando la consola de AWS

Requisitos previos

- Tener una cuenta en AWS (puede ser Free Tier).
- Haber creado una tabla en DynamoDB (si no la tienes, te indico cómo más abajo).

1. Crear una tabla en DynamoDB (si no la tienes)

1. Ve a la consola de AWS: <https://console.aws.amazon.com/>
2. Busca DynamoDB en el buscador de servicios.
3. Haz clic en “Crear tabla”.
4. Completa los siguientes datos:
 - Nombre de la tabla: Estudiantes
 - Clave primaria (Partition key): id (tipo String)
5. Deja las demás configuraciones por defecto (modo de capacidad y cifrado).
6. Haz clic en “Crear tabla”.

Insertar un ítem (registro) en la tabla desde la consola

1. Dentro de la consola de DynamoDB, selecciona la tabla Estudiantes.
2. En el menú lateral izquierdo, haz clic en "Explorador de elementos" (Explore items).
3. Haz clic en "Crear ítem" (Create item).
4. Cambia a la vista "Editor JSON" o mantente en la vista de formulario según prefieras.

Ejemplo de ítem en formato JSON (modo editor JSON):

```
{
  "id": "A001",
  "nombre": "Camila Rojas",
  "edad": 23,
  "carrera": "Ingeniería Informática",
  "activo": true,
  "cursos": ["Python", "DynamoDB", "Big Data"],
  "direccion": {
    "ciudad": "Santiago",
    "pais": "Chile"
  }
}
```

Puedes ajustar los tipos de datos si usas la vista por formulario:

- Strings: nombre, carrera, ciudad, país
- Números: edad
- Booleanos: activo
- Listas: cursos
- Objetos anidados: direccion

Guardar y Consultar Ítems en DynamoDB

1. Haz clic en “Crear ítem” o “Guardar”.
2. ¡Listo! Ya has insertado un documento en DynamoDB.

Puedes repetir este proceso para insertar múltiples estudiantes o usar otros datos estructurados según tu modelo.

Consejo extra:

Si quieres consultar o modificar ítems posteriormente, puedes usar:

- Explorador de elementos para buscarlos por id.
- Acciones rápidas para editar o eliminar.

Tabla resumen de comandos y funciones en DynamoDB

Acción	Comando/Función
Crear tabla	create-table
Insertar ítem	put-item
Leer ítem	get-item
Actualizar ítem	update-item
Eliminar ítem	delete-item
Búsqueda por clave	query
Búsqueda total	scan
Escritura en lote	batch-write-item
Lectura en lote	batch-get-item



Conclusión

DynamoDB es una herramienta poderosa para construir **aplicaciones modernas y escalables** en la nube. Su modelo simple pero eficiente, junto con la gestión completamente administrada por AWS, lo convierten en una opción preferida para aplicaciones en tiempo real, móviles, juegos, IoT y más.

Con su modelo clave-valor/documental, su integración con Lambda, API Gateway y EventBridge, y su capacidad para escalar sin intervención manual, es un verdadero pilar en arquitecturas **serverless**.

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

Objetivo

Guiar al estudiante en el diseño lógico y uso básico de una base de datos NoSQL tipo clave-valor/documento como DynamoDB, mediante ejemplos detallados sobre cómo crear ítems, consultarlos, actualizarlos y eliminarlos usando comandos representativos de AWS.

Paso a Paso Detallado

- Contexto del caso
 - Una startup llamada GreenMarket quiere construir su catálogo de productos ecológicos usando DynamoDB. Este sistema debe permitir:
 - Registrar productos.
 - Consultar productos por categoría.
 - Actualizar el stock de un producto.
 - Eliminar productos descontinuados.
- Crear la tabla Productos
 - En DynamoDB se define una clave primaria para cada ítem. Vamos a usar:
 - Partition Key: producto_id (string)
 - Comando conceptual para crear la tabla:

```
{
  "TableName": "Productos",
  "KeySchema": [
    { "AttributeName": "producto_id", "KeyType": "HASH" }
  ],
  "AttributeDefinitions": [
    { "AttributeName": "producto_id", "AttributeType": "S" }
  ],
  "BillingMode": "PAY_PER_REQUEST"
}
```

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

- Insertar un producto (PutItem)

```
{  
  "producto_id": "P001",  
  "nombre": "Shampoo orgánico",  
  "categoria": "higiene",  
  "precio": 5990,  
  "stock": 100  
}
```

- Insertar otro producto

```
{  
  "producto_id": "P002",  
  "nombre": "Bolsas biodegradables",  
  "categoria": "hogar",  
  "precio": 2990,  
  "stock": 200  
}
```

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

- Consultar un producto específico (GetItem)
- Consultar productos por categoría (Query con índice secundario)
 - DynamoDB no permite filtrar por atributos que no sean claves sin índice.
 - Primero habría que crear un índice secundario global (GSI):
 - GSI Partition Key: categoría
 - Ejemplo conceptual de consulta por categoría:

```
{
  "TableName": "Productos",
  "IndexName": "CategoriaIndex",
  "KeyConditionExpression": "categoria = :cat",
  "ExpressionAttributeValues": {
    ":cat": { "S": "hogar" }
  }
}
```

- Actualizar el stock de un producto (UpdateItem)

```
{
  "Key": {
    "producto_id": "P002"
  },
  "UpdateExpression": "SET stock = :nuevoStock",
  "ExpressionAttributeValues": {
    ":nuevoStock": { "N": "180" }
  },
  "TableName": "Productos"
}
```


Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

- Eliminar un producto (DeleteItem)

```
{
  "Key": {
    "producto_id": "P001"
  },
  "TableName": "Productos"
}
```

Resultado Esperado

- Modelo lógico de tabla DynamoDB comprendido.
- Entendimiento de las operaciones básicas: PutItem, GetItem, Query, UpdateItem, DeleteItem.
- Reconocimiento del uso de índices secundarios para búsquedas no basadas en claves.