

Base de Datos DynamoDB: Potencia NoSQL en la Nube

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, DynamoDB es serverless: AWS se encarga de todo el backend, mientras tú solo te preocupas de los datos y las consultas.

En esta presentación, exploraremos las características fundamentales de DynamoDB, aprenderemos a gestionar tablas, manipular datos y realizar implementaciones en la nube. También veremos ejemplos prácticos de cómo utilizar esta potente herramienta para construir aplicaciones modernas y escalables.

 **por Kibernetum Capacitación**



Preguntas de Activación de Contenido

- 1) ¿Qué sabes sobre bases de datos NoSQL y en qué se diferencian de las bases de datos relacionales?
- 2) ¿Has trabajado con servicios en la nube como AWS? ¿Qué experiencia tienes?
- 3) ¿Qué crees que significa que una base de datos sea “serverless” y cómo podría beneficiarte en un proyecto?



Características Principales de DynamoDB



Rendimiento en milisegundos

Lectura y escritura con latencia muy baja. Ideal para aplicaciones en tiempo real que requieren respuestas inmediatas.



Escalabilidad automática

Se adapta dinámicamente según la carga de trabajo (escrituras, lecturas, almacenamiento) sin intervención manual.



Alta disponibilidad

Arquitectura multizona y multirregión con respaldo automático que garantiza la continuidad del servicio.



Totalmente gestionado

No necesitas administrar servidores ni aplicar parches, AWS se encarga de toda la infraestructura.

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

E-commerce

Consulta de productos y pedidos en tiempo real. DynamoDB permite gestionar catálogos extensos y procesar transacciones con alta velocidad, ideal para picos de tráfico como Black Friday.

Videojuegos

Almacenamiento de puntuaciones, progreso y perfiles de jugador. La baja latencia garantiza una experiencia fluida incluso con millones de usuarios concurrentes.

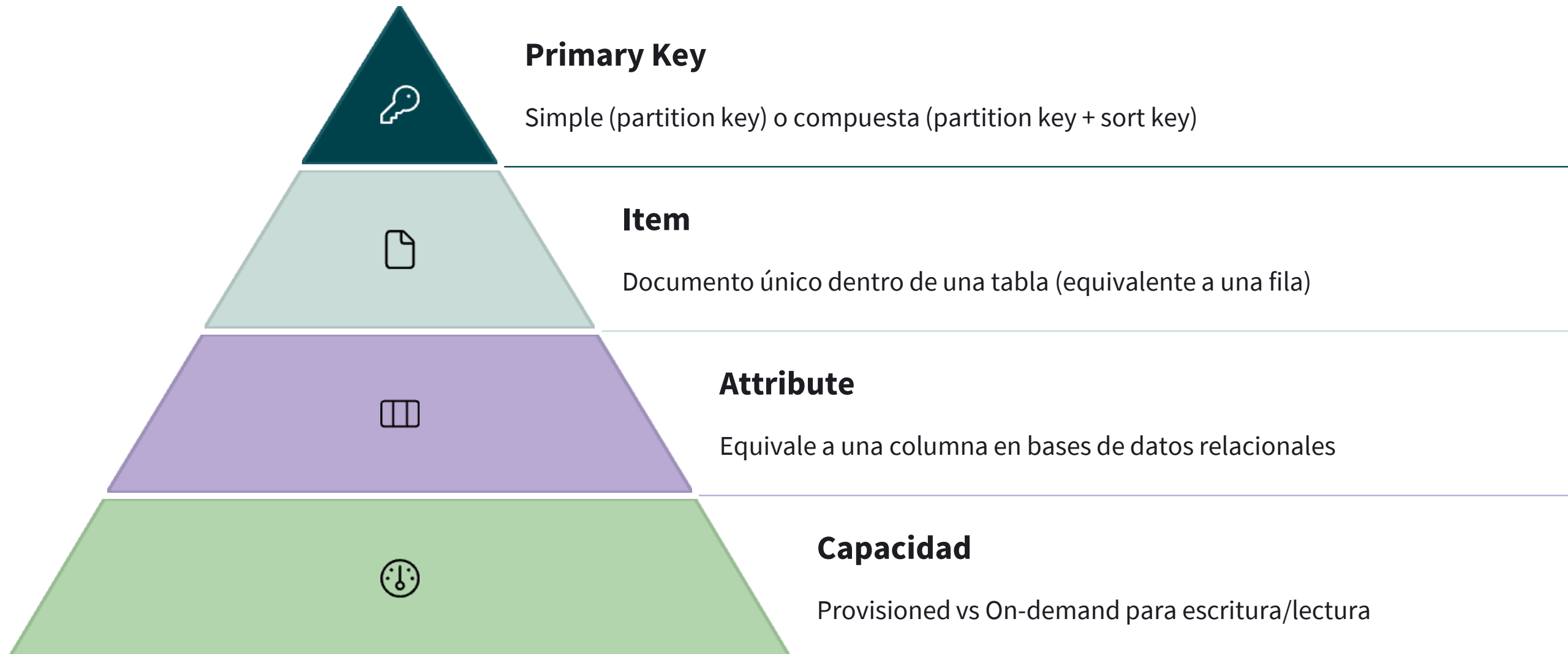
Apps móviles

Sincronización de datos por usuario. Permite mantener la información actualizada entre dispositivos con tiempos de respuesta mínimos.

IoT

Captura de datos de sensores con alta frecuencia. Ideal para procesar y almacenar grandes volúmenes de información generada por dispositivos conectados.

Conceptos Clave de DynamoDB



Estos conceptos fundamentales definen la estructura y funcionamiento de DynamoDB. La elección adecuada de la clave primaria es crucial para el rendimiento, ya que determina cómo se distribuyen los datos y cómo se pueden consultar eficientemente.

Creación de Tablas en DynamoDB

Definir la estructura

Determinar la clave primaria (simple o compuesta) y los atributos principales que contendrá la tabla. Esta decisión es crítica para el rendimiento futuro.

Configurar capacidad

Elegir entre modo bajo demanda (on-demand) o aprovisionado (provisioned) según los patrones de uso esperados y consideraciones de costo.

Crear la tabla

Utilizar la consola de AWS, AWS CLI o SDKs para crear la tabla con los parámetros definidos. El proceso es rápido y la tabla estará disponible en segundos.



Creación de Tablas en DynamoDB

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.

Creación de Tablas en DynamoDB

Consultar tabla

Puedes usar:

- **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"}}'
```

Creación de Tablas en DynamoDB

Eliminar tabla

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



Operaciones Básicas: Crear y Leer



Crear ítem (PutItem)

Inserta un nuevo documento en la tabla especificando todos los atributos necesarios, incluida la clave primaria.



Obtener ítem (GetItem)

Recupera un documento específico utilizando su clave primaria, la forma más eficiente de consulta.



Consultar (Query)

Busca documentos por clave de partición y condiciones opcionales sobre la clave de ordenación.



Escanear (Scan)

Recorre todos los ítems de la tabla, operación más costosa que debe usarse con precaución.

Operaciones Básicas: Actualizar y Eliminar



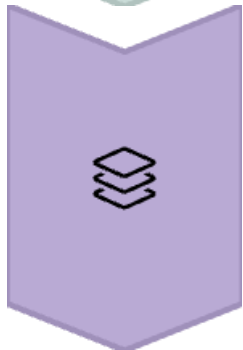
Actualizar ítem (UpdateItem)

Modifica atributos específicos de un documento existente sin necesidad de reemplazar todo el ítem. Permite operaciones atómicas como incrementar valores.



Eliminar ítem (DeleteItem)

Elimina un documento completo de la tabla utilizando su clave primaria. La operación es inmediata e irreversible.



Operaciones por lotes

BatchWriteItem y BatchGetItem permiten realizar múltiples operaciones en una sola llamada, mejorando la eficiencia.



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"}}'
```

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

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"}  
          }  
        },  
      }  
    ]  
  }'
```

Batch Get

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

Implementación en la Nube

Disponibilidad

DynamoDB se ejecuta dentro de AWS y está disponible en todas las regiones globales.

Integración

Se conecta fácilmente con otros servicios AWS como Lambda y API Gateway.



Herramientas

Accesible mediante consola web, AWS CLI y SDKs para diversos lenguajes.

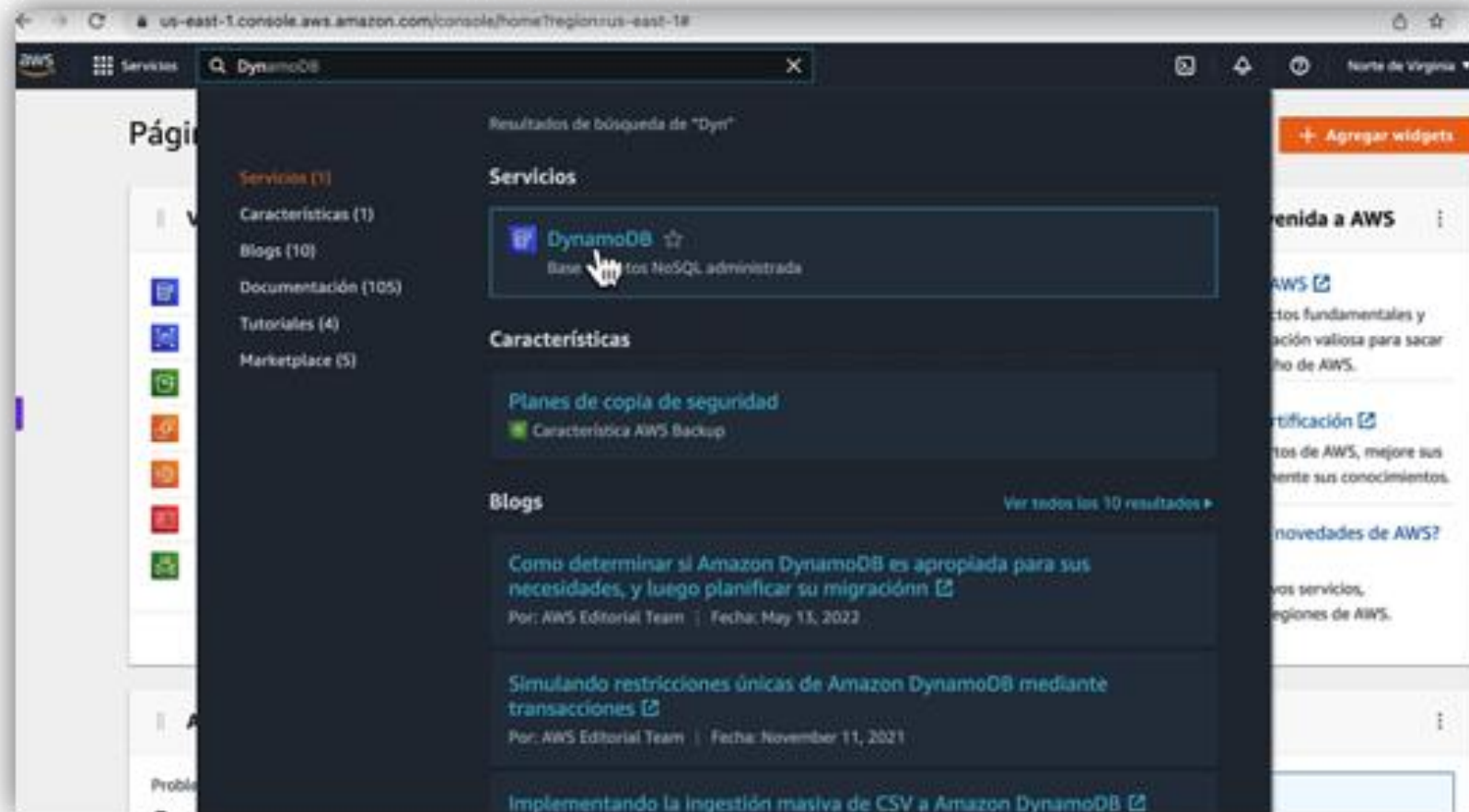
Desarrollo local

DynamoDB Local permite trabajar offline durante el desarrollo.

Ejecutar en AWS (pasos básicos)

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

Buscamos dynamodb



Ejecutar en AWS (pasos básicos)

2. Crear una tabla



Ejecutar en AWS (pasos básicos)

3. 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.

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

Ejecutar en AWS (pasos básicos)

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.

Agregar nueva etiqueta

Puede agregar 50 etiquetas más.

Cancelar

Crear tabla

Creando la tabla DemoTabla. Estará disponible en breve.

DynamoDB > Tablas

Tablas (1) Información

Actualizar

Acciones

Eliminar

Crear tabla

Buscar tablas por su nombre

Cualquier etiqueta de tabla

< 1 >

<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



Seleccionar tabla

Elegir la tabla donde se insertarán los datos



Crear ítem

Hacer clic en "Crear ítem" o "Explorador de elementos"

X_1

Introducir datos

Usar el editor JSON o el formulario para definir atributos



Guardar ítem

Confirmar la creación del nuevo documento

Insertar Datos en DynamoDB

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 Datos en DynamoDB

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"  
  }  
}
```

Insertar Datos en DynamoDB

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

3. Guardar el ítem

1.Haz clic en “Crear ítem” o “Guardar”.

2.¡Listo! Ya has insertado un documento en DynamoDB.

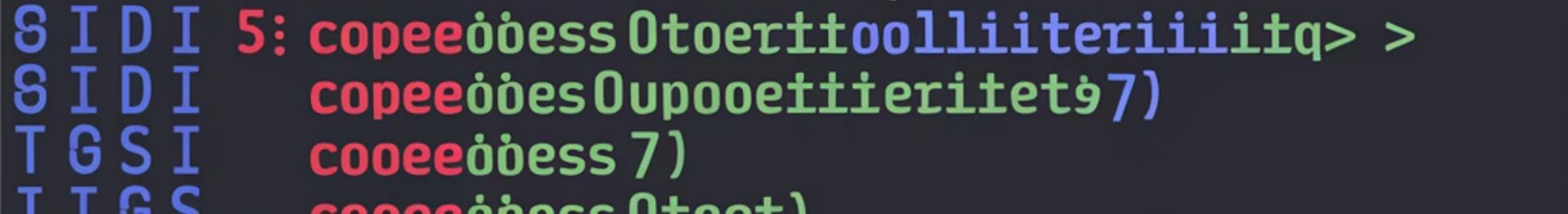
Puedes repetir este proceso para insertar múltiples estudiantes o usar otros datos estructurados según tu modelo.

Consejo extra:



Si quieres consultar o modificar ítems posteriormente, puedes usar:

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



Comandos Principales de 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

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.

1. 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.

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

2. 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"
}
```

Esto crea una tabla con una clave primaria simple.

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

3. Insertar un producto (PutItem)

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

*Esto se inserta en
DynamoDB usando la
operación PutItem.*

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

4. Insertar otro producto

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

Puedes insertar varios productos uno a uno o con BatchWriteItem

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

5. Consultar un producto específico (GetItem)

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

Esto retorna todos los datos del producto con producto_id = P001

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

6. 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:

Esto obtiene todos los productos donde categoría = hogar

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

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

7. Actualizar el stock de un producto (UpdateItem)

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

Actualiza el campo stock a 180 para el producto P002

Actividad Práctica Guiada: Modelado y Operaciones Básicas en DynamoDB

8. Eliminar un producto (DeleteItem)

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

*Elimina el producto con ID
"P001" de la tabla*

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.

Conclusiones



Potencia NoSQL

DynamoDB es una herramienta poderosa para construir aplicaciones modernas y escalables en la nube, con un modelo simple pero eficiente para datos no relacionales.



Arquitectura Serverless

Su gestión completamente administrada por AWS lo convierte en un pilar fundamental para arquitecturas serverless, eliminando la carga operativa.



Rendimiento

Ideal para aplicaciones en tiempo real, móviles, juegos e IoT gracias a su capacidad de escalar sin intervención manual y mantener baja latencia.

Enlace a Material Complementario



Video – Introducción a DynamoDB (en español)

<https://www.youtube.com/watch?v=ybG2Qnucmts>

Este video explica de forma clara qué es DynamoDB, cómo funciona su modelo de datos y cómo realizar operaciones básicas desde la consola de AWS.



Documentación oficial:

https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developer-guide/Introduction.html



Preguntas de Reflexión Final

- 1) ¿Qué criterios consideras importantes al momento de diseñar la clave primaria en DynamoDB?
- 2) ¿Qué ventaja crees que tiene el uso de BatchWrite o BatchGet en comparación con las operaciones simples?
- 3) ¿Qué desafíos ves en la implementación de un sistema completo usando DynamoDB como backend principal?