

## Escritura y recuperación en lote

### Contexto: 🧑

DynamoDB permite insertar y recuperar múltiples ítems en una sola operación mediante *batch writing* y *batch retrieve*. Esta funcionalidad es esencial para aplicaciones de alto volumen.

### Consigna: 📝

Simula la carga y recuperación de múltiples ítems de una tabla de usuarios mediante operaciones por lote.

**Tiempo** ⌚: 30 minutos

### 🔗 Paso a paso:

1. Escribe un comando simulado *batch-write-item* que incluya tres usuarios (con *UsuarioID*, *Nombre*, *Edad*, *Ciudad*).
2. Luego, escribe el comando simulado *batch-get-item* para recuperar dos de esos usuarios.
3. Justifica por qué es útil usar batch en este caso.
4. Opcional: calcula cuántas operaciones individuales se ahorrarían si se usara batch en vez de *put-item* por ítem.

## Ejercicio – Escritura y recuperación en lote con DynamoDB

### Objetivo

Simular operaciones en lote para insertar y recuperar múltiples ítems de una tabla de usuarios, aprovechando *batch-write-item* y *batch-get-item*.

### 1. Comando simulado de escritura *batch-write-item* (3 usuarios)

```
aws dynamodb batch-write-item \
--request-items '{
  "Usuarios": [
    {
      "PutRequest": {
        "Item": {
          "UsuarioID": {"S": "U001"},
          "Nombre": {"S": "Laura"},
          "Edad": {"N": "28"},
          "Ciudad": {"S": "Antofagasta"}
        }
      }
    },
    {
      "PutRequest": {
        "Item": {
          "UsuarioID": {"S": "U002"},
          "Nombre": {"S": "Tomás"},
          "Edad": {"N": "35"},
          "Ciudad": {"S": "Santiago"}
        }
      }
    },
    {
      "PutRequest": {
        "Item": {
          "UsuarioID": {"S": "U003"},
          "Nombre": {"S": "Carlos"},
          "Edad": {"N": "42"},
          "Ciudad": {"S": "Valparaíso"}
        }
      }
    }
  ]
}
```

```

      "Edad": {"N": "34"},
      "Ciudad": {"S": "Santiago"}
    }
  },
  {
    "PutRequest": {
      "Item": {
        "UsuarioID": {"S": "U003"},
        "Nombre": {"S": "Ana"},
        "Edad": {"N": "25"},
        "Ciudad": {"S": "Valparaíso"}
      }
    }
  }
]
}'

```

## 2. Comando simulado de lectura batch-get-item (2 usuarios)

```

aws dynamodb batch-get-item \
--request-items '{
  "Usuarios": {
    "Keys": [
      {"UsuarioID": {"S": "U001"}},
      {"UsuarioID": {"S": "U003"}}
    ]
  }
}'

```

## 3. Justificación del uso de batch

- Reduce tiempo y número de llamadas a la base de datos.
- Mejora eficiencia en latencia y uso de recursos.
- Evita múltiples ejecuciones de put-item y get-item.

## 4. Cálculo de operaciones ahorradas

- Sin batch: 3 operaciones put-item + 2 get-item = 5 operaciones individuales.
- Con batch: 1 batch-write-item + 1 batch-get-item = 2 operaciones.
- Ahorro: 3 operaciones menos (60%).

## Preguntas Clase:

### ¿Por qué usarías batch-write-item en lugar de operaciones individuales?

- **Mayor eficiencia:** Permite insertar hasta 25 ítems en una sola operación, reduciendo el número de llamadas a la API.
- **Menor latencia:** Al agrupar operaciones, el tiempo total de ejecución es menor que realizar múltiples put-item por separado.
- **Reducción de costos:** Menos operaciones implica menor consumo de recursos y, por ende, potenciales ahorros en costos.
- **Ideal para cargas masivas:** Es muy útil cuando necesitas insertar grandes volúmenes de datos rápidamente (por ejemplo, migraciones o sincronización de datos).

### ¿Qué riesgos existen si no planificamos bien el particionamiento de los datos?

- **Hot partitions:** Si muchos ítems comparten una misma clave de partición, se genera una carga desbalanceada que puede afectar el rendimiento y causar cuellos de botella.
- **Desigual distribución del tráfico:** Una mala estrategia de particionamiento puede generar que unas pocas particiones trabajen mucho mientras otras queden inactivas.
- **Límites superados:** Cada partición tiene límites de capacidad (lectura/escritura). Una mala distribución puede hacer que se superen esos límites en ciertas particiones.
- **Pérdida de escalabilidad:** Un diseño deficiente puede impedir que DynamoDB escale automáticamente de manera efectiva.