# Ejercicio de Aplicación #6 – DynamoDB (Gestión de Pedidos)

## Caso de uso: Gestión de pedidos online

Una empresa necesita almacenar información sobre los pedidos de sus clientes, incluyendo el ID del pedido, el cliente, la fecha y los productos comprados. Se utilizará una estructura denormalizada en DynamoDB optimizada para consultas rápidas.

## 1. Diseño de la tabla en DynamoDB

- Nombre de la tabla: Pedidos
- Partition Key (PK): cliente_id
- Sort Key (SK): pedido_id
- Otros atributos: fecha, productos (array), total

Ejemplo de ítem en JSON:

```
{
  "cliente_id": "C001",
  "pedido_id": "P1001",
  "fecha": "2025-07-27",
  "productos": [
    { "nombre": "Notebook", "precio": 700, "cantidad": 1 },
    { "nombre": "Mouse", "precio": 25, "cantidad": 2 }
  ],
  "total": 750
}
```

## 2. Inserción de datos de ejemplo (AWS CLI)

```
aws dynamodb put-item \
  --table-name Pedidos \
  --item '{
    "cliente_id": {"S": "C001"},
    "pedido_id": {"S": "P1001"},
    "fecha": {"S": "2025-07-27"},
    "productos": {
      "L": [
        { "M": { "nombre": {"S": "Notebook"}, "precio": {"N": "700"}, "cantidad": {"N": "1"} } },
        { "M": { "nombre": {"S": "Mouse"}, "precio": {"N": "25"}, "cantidad": {"N": "2"} } }
      ]
    },
    "total": {"N": "750"}
  }'
```

## 3. Consultas

a) Todos los pedidos de un cliente específico:

```
aws dynamodb query \
  --table-name Pedidos \
  --key-condition-expression "cliente_id = :id" \
  --expression-attribute-values '{":id": {"S": "C001"}}'
```

b) Todos los productos de un pedido en particular:

```
aws dynamodb get-item \
  --table-name Pedidos \
  --key '{"cliente_id": {"S": "C001"}, "pedido_id": {"S": "P1001"}}'
```

c) Número total de pedidos por cliente (desde la aplicación):

```
# Python (boto3):
response = dynamodb.query(
    TableName='Pedidos',
    KeyConditionExpression='cliente_id = :id',
    ExpressionAttributeValues={':id': {'S': 'C001'}}
)
print("Total pedidos:", len(response['Items']))
```

## 4. Comparación DynamoDB vs Relacional

| Característica | DynamoDB | Relacional (Ej: MySQL, PostgreSQL) |
|---|---|---|
| Modelo de datos | NoSQL, clave-valor | Tablas normalizadas relacionales |
| Agregaciones | Limitadas (app o Lambda) | Soportadas por SQL (SUM, COUNT) |
| Escalabilidad | Altamente escalable | Escalabilidad vertical |
| Esquema | Flexible | Estructura fija |
| Velocidad | Alta con claves óptimas | Razonable, depende de índices |

## Plus: Tabla de clients

Estructura:

- cliente_id (PK)
- nombre
- email
- fecha_registro

```
aws dynamodb put-item \
  --table-name Clientes \
  --item '{
    "cliente_id": {"S": "C001"},
    "nombre": {"S": "Juan Pérez"},
    "email": {"S": "juan@mail.com"},
    "fecha_registro": {"S": "2025-01-01"}
  }'
```