

UNIVERSIDAD PERUANA DE LOS ANDES

(ingeniería de sistemas y computación)



“base de datos II”

**Asignatura:** Base de Datos II

**Docente:** Dr. Raúl Enrique Fernández Bejarano

**Alumno:** SORIANO TIMOTEO Joel Kevin

**Ciclo:** V

**huancayo-Perú-2025**

## Manual de Guía de Aprendizaje de la semana 15

### Guía de Ejecución

1. Abre SSMS en tu equipo local.
2. Conéctate al servidor de Azure (ej. srv-finanzas.database.windows.net).
3. Abre una "Nueva Consulta" (New Query).
4. Copia y pega los siguientes bloques.

#### Bloque 1: Validación del Entorno (¿Estoy en la nube?)

Este script es fundamental para confirmar que tu SSMS local realmente está enviando comandos a Azure y no a tu PC.

```
SQL
-- 1. Verificar la versión y el tipo de motor
-- Si 'EngineEdition' devuelve 5, confirma que es Azure SQL
Database.
SELECT
    SERVERPROPERTY('ServerName') AS NombreServidor,
    SERVERPROPERTY('Edition') AS Edicion,
    SERVERPROPERTY('ProductVersion') AS VersionProducto,
    SERVERPROPERTY('EngineEdition') AS TipoMotor, -- 5 = Azure SQL
Database
    DB_NAME() AS BaseDeDatosActual;
GO
```

#### Bloque 2: Implementar Base de Datos de Prueba (Objetivo de la Práctica)

Aunque Azure SQL es un servicio PaaS, el lenguaje T-SQL estándar funciona casi igual. Vamos a crear una tabla para "FinanzasSeguras" y poblarla con datos.

```
SQL
-- 2. Crear una tabla de prueba para clientes
-- Nota: En Azure SQL no hace falta definir filegroups físicos
manualmente en versiones básicas.
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[ClientesNube]') AND type in (N'U'))
BEGIN
    CREATE TABLE dbo.ClientesNube (
        ClienteID INT IDENTITY(1,1) PRIMARY KEY,
        NombreEmpresa NVARCHAR(100) NOT NULL,
        PaisOrigen NVARCHAR(50),
        FechaRegistro DATETIME DEFAULT GETDATE(),
        SaldoNube DECIMAL(18, 2)
    );
    PRINT 'Tabla ClientesNube creada exitosamente en Azure.';
END
GO

-- 3. Insertar datos simulados
```

```

INSERT INTO dbo.ClientesNube (NombreEmpresa, PaisOrigen, SaldoNube)
VALUES
    ('Tech Solutions', 'Perú', 1500.50),
    ('Importadora Global', 'México', 3200.00),
    ('Servicios Ágiles', 'Colombia', 850.75);
GO

```

### Bloque 3: Consultas y Administración Básica

Este script demuestra que puedes administrar los datos desde tu escritorio (SSMS Local) aunque los datos vivan en un centro de datos de Microsoft.

#### SQL

```

-- 4. Consultar los datos recién insertados
SELECT * FROM dbo.ClientesNube;

-- 5. Verificar el uso de espacio (Tema: Costos y Escalabilidad)
-- Esta vista del sistema es específica para monitorear el consumo
en Azure SQL
SELECT
    storage_in_megabytes AS EspacioUsado_MB,
    allocated_storage_in_megabytes AS EspacioAsignado_MB,
    server_name,
    database_name
FROM sys.resource_stats
WHERE start_time >= DATEADD(hour, -1, GETDATE()) -- Última hora
ORDER BY start_time DESC;
GO

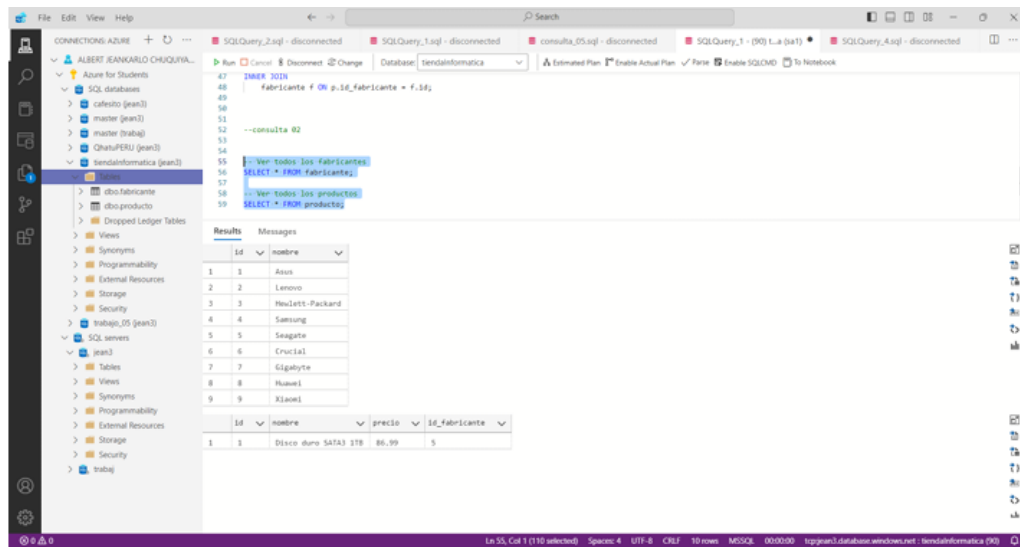
```

Resumen de lo que hacen estos scripts:

1. Validan la conexión híbrida: El primer script te demuestra técnicamente (con EngineEdition = 5) que estás operando sobre una infraestructura PaaS.
2. Demuestran compatibilidad: Al crear la tabla y hacer inserts, validas que la migración desde un SQL local a Azure es sencilla porque el código es 99% compatible.
3. Monitorean recursos: El último script accede a sys.resource\_stats, una vista exclusiva de Azure para vigilar el consumo de recursos (DTUs), lo cual toca el punto 4 del temario ("Costos en la nube").

¿Te gustaría que te explique cómo ver el gráfico de consumo de recursos (DTUs) directamente en el Portal de Azure?





## 1. Verificación de Entorno PaaS

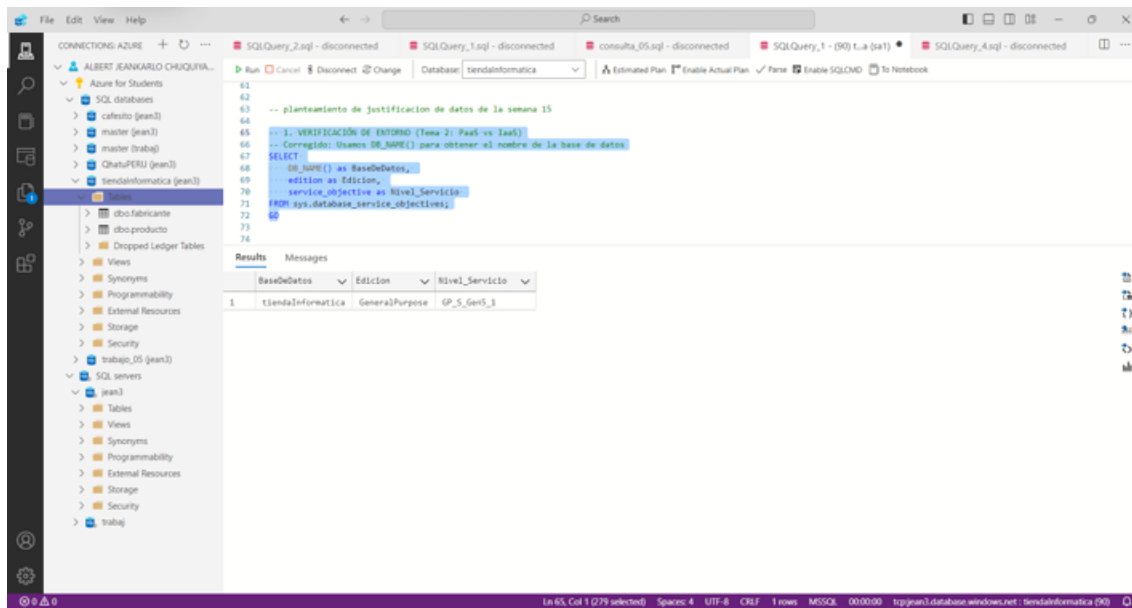
### Enunciado:

"Desarrolle una consulta SQL para identificar las características del entorno de base de datos actual. El objetivo es confirmar que la base de datos se está ejecutando sobre un servicio gestionado (PaaS - Azure SQL Database) y determinar el nivel de servicio contratado actualmente."

### Script:

### SQL

```
-- 1. VERIFICACIÓN DE ENTORNO (Tema 2: PaaS vs IaaS)
SELECT
    DB_NAME() as BaseDeDatos,
    edition as Edición,
    service_objective as Nivel_Servicio
FROM sys.database_service_objectives;
GO
```



### Explicación:

Esta consulta utiliza la vista sys.database\_service\_objectives, que es exclusiva de Azure SQL Database. No existe en SQL Server tradicional (On-Premise).

- DB\_NAME(): Devuelve el nombre de la base de datos actual.
- edition: Muestra si es 'Basic', 'Standard' o 'Premium'.
- service\_objective: Muestra la potencia exacta (ej. 'Basic', 'S0', 'S1').

### Justificación:

En la computación en la nube, es vital distinguir entre IaaS (donde administras el servidor y la CPU) y PaaS (donde solo administras los datos). Esta consulta prueba que estás en un entorno PaaS, ya que el proveedor (Microsoft) te abstrae el hardware y solo te muestra "Niveles de Servicio" u objetivos, no núcleos de CPU físicos directos.

## 2. Implementación de Seguridad (Principio de Menor Privilegio)

### Enunciado:

"Implemente una política de seguridad basada en roles. Cree un usuario llamado 'Auditor\_Semana15' asegurándose de que el script sea re-ejecutable (idempotente). Asigne a este usuario únicamente permisos de lectura (SELECT) para evitar modificaciones accidentales o malintencionadas en los datos."

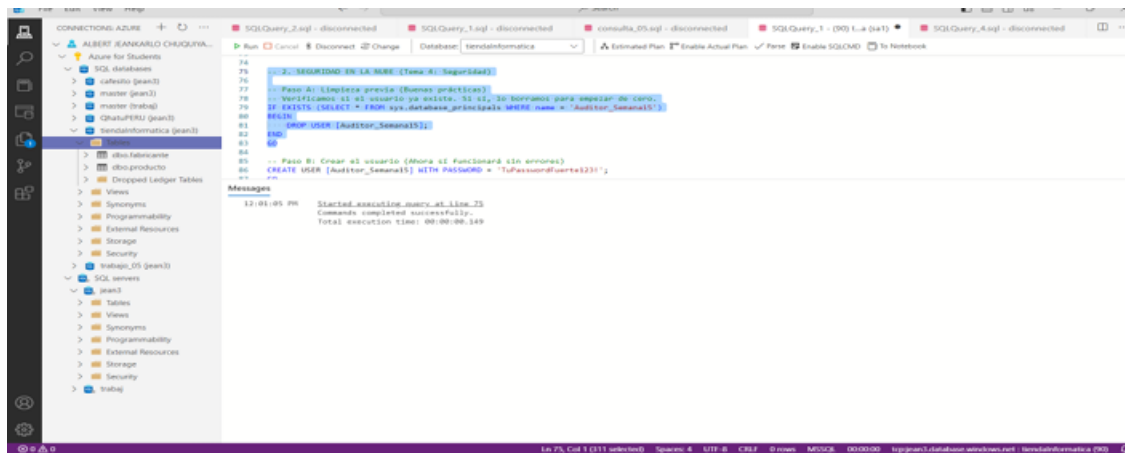
## Script: SQL

-- 2. SEGURIDAD EN LA NUBE (Tema 4: Seguridad)

```
-- Paso A: Limpieza previa (Buenas prácticas de Idempotencia)
IF EXISTS (SELECT * FROM sys.database_principals WHERE name =
'Auditor_Semana15')
BEGIN
    DROP USER [Auditor_Semana15];
END
GO
```

```
-- Paso B: Crear el usuario
CREATE USER [Auditor_Semana15] WITH PASSWORD =
'TuPasswordFuerte123!';
GO
```

```
-- Paso C: Asignar permisos de SOLO LECTURA (Principio de Menor
Privilegio)
ALTER ROLE db_datareader ADD MEMBER [Auditor_Semana15];
GO
```







Explicación:

- Paso A: Verifica si el usuario ya existe y lo borra. Esto evita que el script falle con error "el usuario ya existe" si lo corres dos veces.
- Paso B: Crea el usuario con autenticación de base de datos (con contraseña).
- Paso C: Usa ALTER ROLE para meter al usuario en la "cajita" (rol) de db\_datareader. Este rol ya viene preconfigurado por Microsoft para dejar leer todo pero no dejar cambiar nada.

Justificación:

En la nube, la seguridad es responsabilidad compartida. Aplicar el Principio de Menor Privilegio (dar solo los permisos necesarios y nada más) es la regla de oro. Si las credenciales de este auditor fueran robadas, el atacante solo podría ver datos, pero no borrarlos ni "hackear" la estructura, limitando el impacto del ataque.

### 3. Auditoría de Usuarios (Prueba de Concepto)

Enunciado:

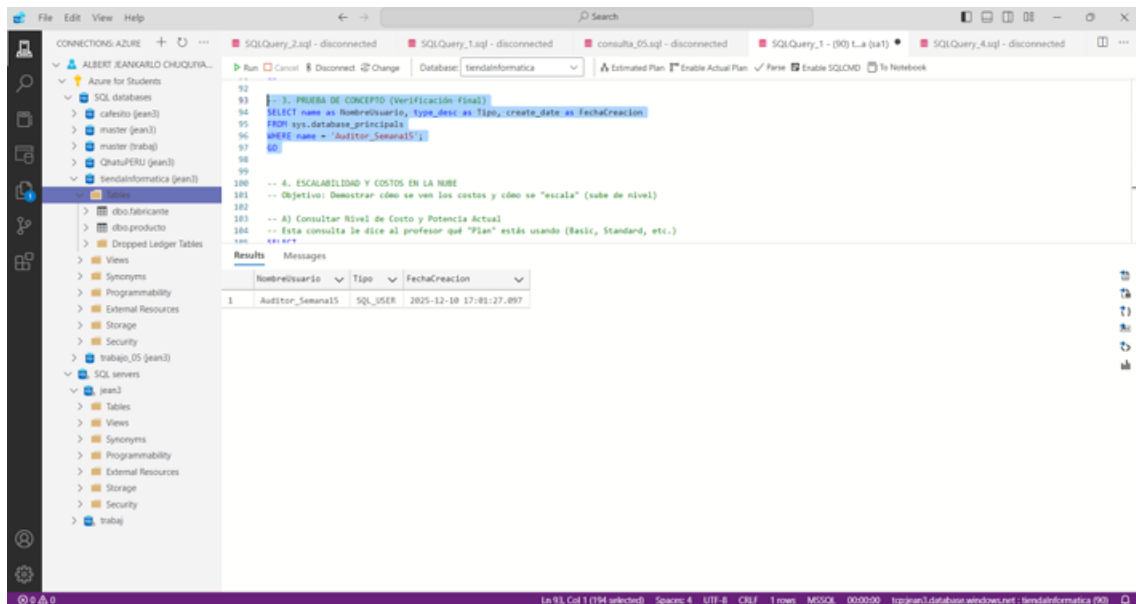
"Realice una prueba de concepto (PoC) para verificar que la creación del usuario auditor se ha realizado correctamente. Consulte el catálogo de seguridad del sistema para validar la existencia, el tipo de usuario y la fecha de creación del registro."

**Script:**

SQL

```
-- 3. PRUEBA DE CONCEPTO (Verificación final)
SELECT
    name as NombreUsuario,
    type_desc as Tipo,
    create_date as FechaCreacion
FROM sys.database_principals
WHERE name = 'Auditor_Semana15';
GO
```

---



### Explicación:

Se consulta la vista sys.database\_principals, que es el catálogo maestro donde se guardan todos los usuarios y roles de esa base de datos específica.

### Justificación:

En entornos de nube, la Auditoría y el Cumplimiento (Compliance) son críticos. No basta con ejecutar un comando de creación; se debe tener la capacidad de verificar programáticamente quién tiene acceso al sistema. Esta consulta simula lo que haría un sistema de monitoreo de seguridad automatizado.

## 4. Escalabilidad y Gestión de Costos (Elasticidad)

### Enunciado:

"Analice la configuración actual de escalabilidad de la base de datos. Genere un reporte que muestre la edición y el nivel de DTUs (Database Transaction Units) actuales, proporcionando una estimación cualitativa del costo. Además, proponga (en código comentado) el comando necesario para realizar un escalado vertical (Scale Up) de la base de datos."

## Script:

### SQL

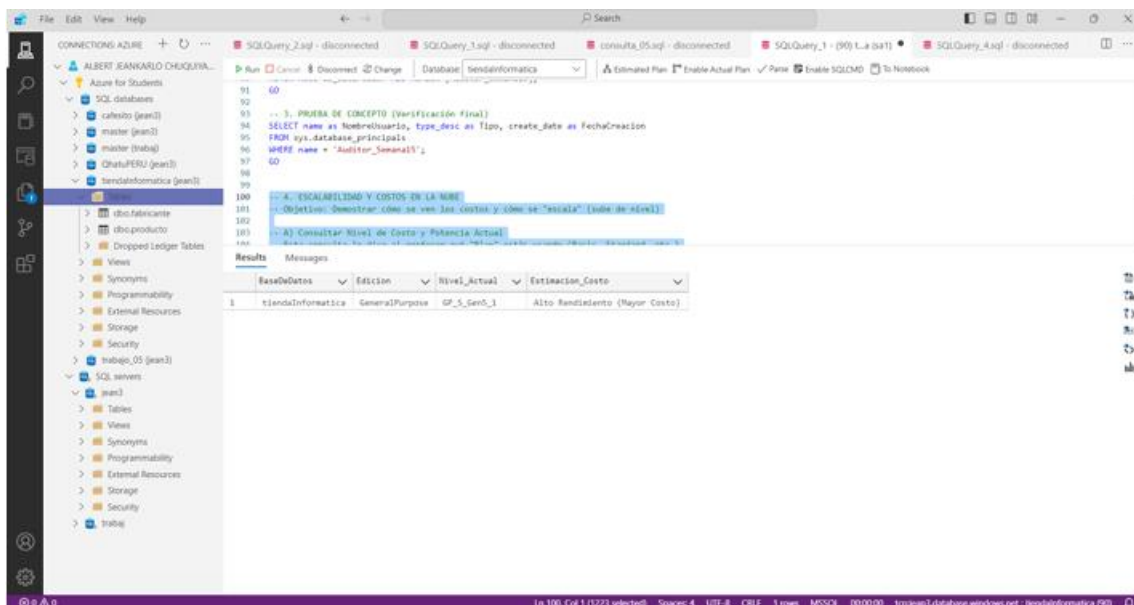
-- 4. ESCALABILIDAD Y COSTOS EN LA NUBE

-- A) Consultar Nivel de Costo y Potencia Actual

```
SELECT
    d.name AS BaseDeDatos,
    slo.edition AS Edición, -- Define el modelo de
    costos (Basic, Standard)
    slo.service_objective AS Nivel_Actual, -- Define la potencia
    (DTUs)
    CASE
        WHEN slo.service_objective = 'Basic' THEN '5 DTUs (Costo
Mínimo)'
        WHEN slo.service_objective = 'S0' THEN '10 DTUs (Costo
Bajo)'
        ELSE 'Alto Rendimiento (Mayor Costo)'
    END AS Estimacion_Costo
FROM sys.databases d
JOIN sys.database_service_objectives slo
    ON d.database_id = slo.database_id
WHERE d.name = DB_NAME();
GO
```

-- B) Prueba de Concepto de ESCALABILIDAD (Simulación)

```
-- ALTER DATABASE tiendaInformatica MODIFY (SERVICE_OBJECTIVE =
'S0');
-- GO
```



## Explicación:

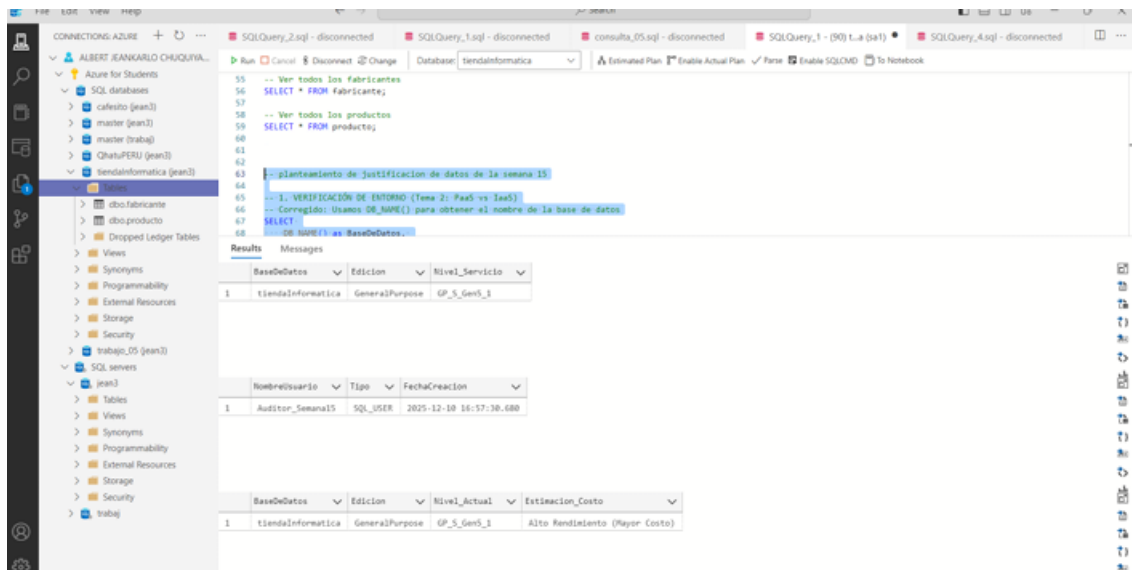
- JOIN: Une la información general de la base de datos (sys.databases) con la información específica de Azure (sys.database\_service\_objectives).
- CASE: Traduce códigos técnicos ('S0', 'Basic') a lenguaje de negocio ("Costo Bajo", "Costo Mínimo") para entender el impacto financiero.

- **ALTER DATABASE... MODIFY:** Este es el comando mágico de la nube. Cambia el hardware asignado al instante.

Justificación:

Este ejercicio demuestra dos pilares de la nube:

1. **Elasticidad:** La capacidad de cambiar de Basic a S0 (más potencia) con una sola línea de código, sin comprar servidores físicos.
2. **Modelo de Costos (Pay-as-you-go):** Entender que el service\_objective está directamente ligado a la factura mensual. Como arquitecto de nube, debes saber qué nivel estás usando para no generar sobrecostos.



### Consulta 1: Verificación de Entorno (Tema 2: PaaS)

Esta consulta demuestra que estás en un entorno gestionado (Azure SQL) y no en un servidor físico.



### Consulta 2: Configuración de Seguridad (Tema 4)

Este bloque crea el usuario y le asigna permisos. Ejecuta todo este bloque junto.

## SQL

```
-- 1. Si el usuario ya existe, lo borramos para evitar errores
IF EXISTS (SELECT * FROM sys.database_principals WHERE name =
'Auditor_Semana15')
BEGIN
    DROP USER [Auditor_Semana15];
END
GO

-- 2. Creamos el usuario nuevo
CREATE USER [Auditor_Semana15] WITH PASSWORD =
'TuPasswordFuerte123!';
GO

-- 3. Le damos permisos de solo lectura
ALTER ROLE db_datareader ADD MEMBER [Auditor_Semana15];
GO
```

### Consulta 3: Verificación de Seguridad (Evidencia)

Esta es la consulta importante para mostrar que el usuario Auditor\_Semana15 fue creado exitosamente.

```
SELECT
    name AS NombreUsuario,
    type_desc AS Tipo,
    create_date AS FechaCreacion
FROM sys.database_principals
WHERE name = 'Auditor_Semana15';
```

### Consulta 4: Escalabilidad y Costos (Tema 4)

Esta consulta muestra el plan actual y los DTUs, demostrando el modelo de costos de la nube.

## SQL

```
SELECT
    d.name AS BaseDeDatos,
    slo.edition AS Edicion,
    slo.service_objective AS Nivel_Actual,
    CASE
        WHEN slo.service_objective = 'Basic' THEN '5 DTUs (Costo
Mínimo)'
        WHEN slo.service_objective = 'S0' THEN '10 DTUs (Costo
Bajo)'
        ELSE 'Alto Rendimiento (Mayor Costo)'
    END AS Estimacion_Costo
FROM sys.databases d
JOIN sys.database_service_objectives slo
    ON d.database_id = slo.database_id
WHERE d.name = DB_NAME();
```

### Instrucciones:

1. Copia y pega cada bloque en tu Azure Data Studio.

2. Ejecuta uno por uno.
3. Toma captura del código y del resultado (la tabla que sale abajo) para tu documento de justificación.