



UNIVERSIDAD PERUANA LOS ANDES



**Asignatura:
BASE DE DATOS II**

**Tema:
Automatización y mantenimiento**

**Docente:
FERNANDEZ BEJARANO RAUL**

**Estudiante:
BREÑA QUISPE MEGAM**

HUANCAYO-2025

1. Introducción a Azure SQL Database y SQL Managed Instance

1.

- **Definición :**

Azure SQL Database es un servicio de base de datos relacional totalmente administrado en la nube de Microsoft, diseñado para ofrecer alta disponibilidad, escalabilidad automática, rendimiento optimizado y seguridad integrada sin necesidad de administrar infraestructura física. Permite ejecutar cargas de trabajo SQL con mínima configuración y soporta conexiones desde aplicaciones locales o en la nube.



Ventajas

- **Servicio 100% administrado:** Microsoft se encarga de parches, backups automáticos y mantenimiento.
- **Alta disponibilidad garantizada** sin configuraciones adicionales.
- **Escalabilidad automática** según la carga del sistema.
- **Seguridad avanzada integrada** (TDE, firewall, auditorías, alertas).
- **Costo flexible:** pago por uso o capacidad elástica.
- **Optimización automática** (indexación, tuning).

Desventajas

- No tiene la **compatibilidad completa** de SQL Server local.
- Restricciones en ciertas funciones del sistema y configuraciones avanzadas.
- No permite acceso al **sistema operativo** ni a instancias físicas.
- Migrar aplicaciones muy antiguas puede requerir ajustes.

Tipos de Azure SQL Database

- **Single Database**
 - Una base de datos independiente con recursos dedicados.
- **Elastic Pool**
 - Varias bases de datos compartiendo recursos para reducir costos.
- **Serverless**
 - Se pausa automáticamente cuando no se usa, ideal para cargas intermitentes.
- **Hyperscale**
 - Capacidad de almacenamiento de hasta 100 TB con escalado rápido.

Cómo funciona

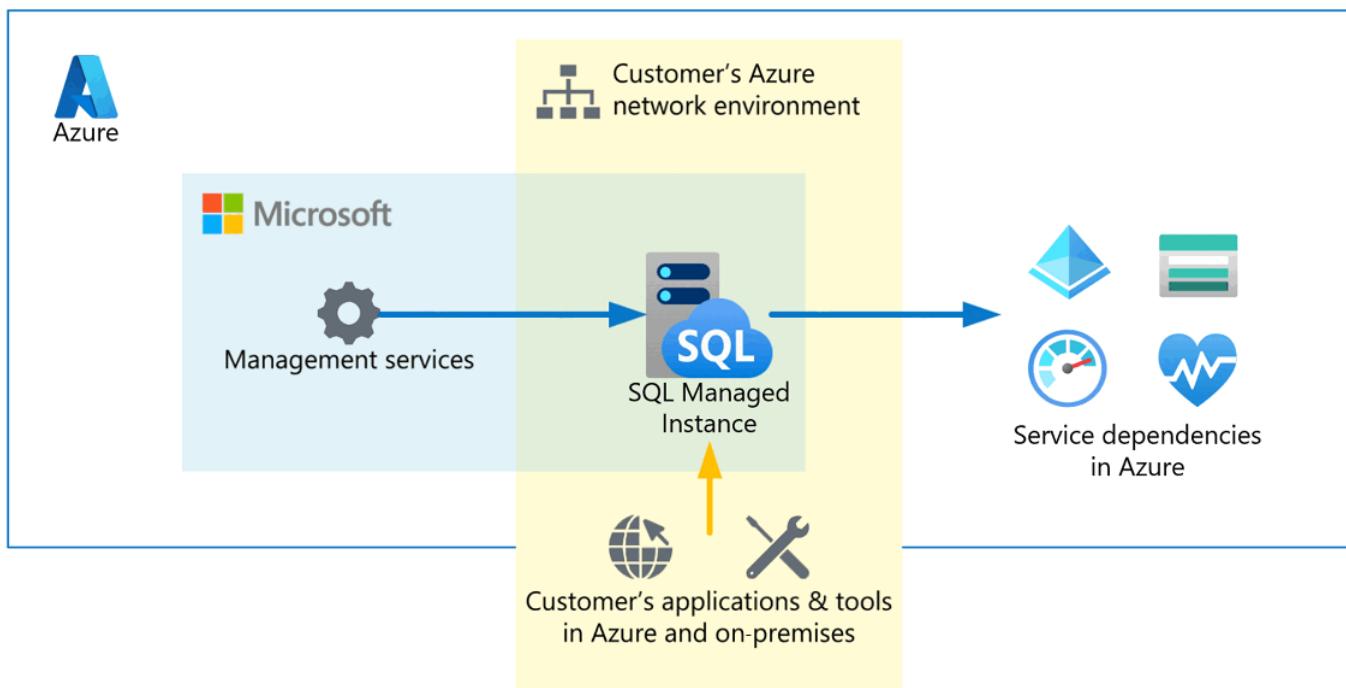
Es un servicio **PaaS (Platform as a Service)**.

Microsoft administra la plataforma, mientras el usuario solo administra sus datos y usuarios. Se despliega en una infraestructura distribuida que replica los datos para garantizar disponibilidad.

2.

- **Definición profesional:**

Azure SQL Managed Instance es una plataforma de base de datos administrada en la nube que ofrece casi el 100% de compatibilidad con SQL Server local, permitiendo migraciones más simples sin modificar aplicaciones. Combina las ventajas de PaaS con el control y características avanzadas de SQL Server, ideal para entornos híbridos y migraciones empresariales.



Ventajas

- Compatibilidad casi total ($\approx 99.9\%$) con SQL Server local.
- Migraciones rápidas sin cambiar código de la aplicación.
- Soporta características avanzadas como:
 - SQL Agent
 - Linked Servers
 - Cross-database queries
 - Jobs programados
- Integración natural con redes privadas (VNet).
- Seguridad empresarial (Always Encrypted, auditorías, firewalls).
- Backups automáticos y alta disponibilidad integrada.

Desventajas

- Es más costoso que Azure SQL Database.
- Escalado más lento que un Database Single.
- Configuración inicial más compleja (uso de redes virtuales).
- No da acceso total al servidor físico ni al SO (como todo PaaS).

Tipos de Managed Instance

Solo tiene **una modalidad**, pero se clasifica según la capacidad:

- **General Purpose** → para cargas normales, económico.
- **Business Critical** → para rendimiento alto y menor latencia.

Cómo funciona

Es un servicio **PaaS avanzado**, similar a ejecutar SQL Server en la nube, pero sin tener que administrar el sistema operativo ni hardware.

Se ejecuta dentro de una **VNet**, lo que permite integrarse con redes locales (VPN o ExpressRoute).

2. Diferencias con SQL Server local (IaaS vs PaaS)

Característica	SQL Server Local (On-Premise)	SQL Server en IaaS (Máquina Virtual en Azure)	SQL Server PaaS (Azure SQL Database / Managed Instance)
Definición	Instalación física en servidores propios dentro de la empresa. Todo se administra manualmente.	SQL Server instalado en una máquina virtual en Azure. La infraestructura es de Microsoft, pero el usuario administra el servidor.	Servicio administrado por Azure. Microsoft se encarga del motor, parches, backups y disponibilidad. Tú solo gestionas la base.
Tipo de servicio	Local (On-Premise)	IaaS (Infraestructura como servicio)	PaaS (Plataforma como servicio)
Administración	Totalmente manual: hardware, SO, SQL Server, parches, seguridad física.	Administras el SO y SQL Server; Azure solo mantiene la infraestructura virtual.	Administración mínima: Azure gestiona parches, backups, HA, seguridad base.
Costo	Alto: compra de hardware, licencias, mantenimiento.	Medio/alto: pagas por VM + licencias.	Flexible y optimizado: pago por consumo.
Escalabilidad	Limitada por el hardware físico.	Escalable, pero requiere reinicios o cambios de VM.	Escalabilidad automática y rápida sin interrupciones.
Alta disponibilidad (HA)	Debe configurarse manualmente (Always On, cluster, etc.)	Usuario debe configurarla en la VM.	Integrada automáticamente por Azure.
Seguridad	Depende totalmente del equipo de TI local.	Depende del usuario en SO + SQL; Azure protege infraestructura.	Incluye cifrado, auditorías, firewalls, alertas, IA de protección.
Mantenimiento	Responsabilidad total del usuario.	Usuario administra SO, SQL y parches.	Azure administra casi todo (sin acceso al SO).
Control del sistema operativo	Total acceso.	Total acceso como cualquier servidor Windows.	No hay acceso al SO; solo al motor SQL.
Compatibilidad	Máxima compatibilidad.	Igual que local (SQL Server completo).	Alta (Managed Instance ≈100%) / Media (Azure SQL Database).
Uso ideal	Empresas con infraestructura propia y control total.	Migraciones rápidas desde local con mínima adaptación.	Sistemas modernos, escalables, en nube o híbridos.

SQL Server Local (On-Premise)

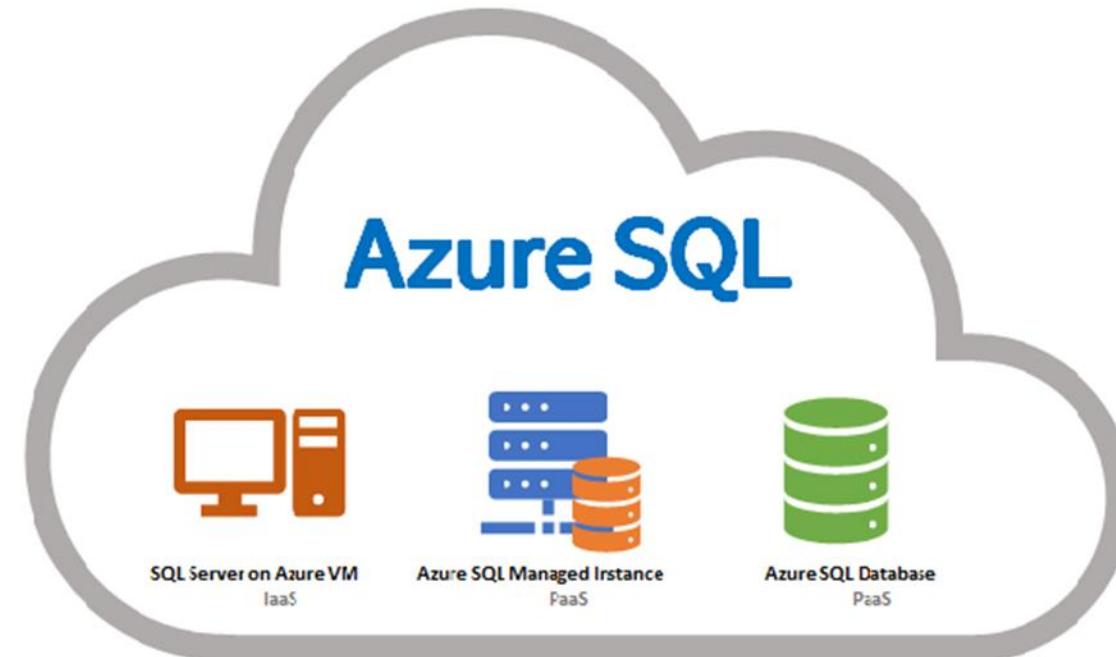
Servidor físico dentro de la empresa. El equipo de TI administra todo: hardware, red, sistema operativo, parches, backups y el motor SQL.

SQL Server en IaaS (Máquina Virtual Azure)

SQL Server instalado en una VM en la nube. Azure administra la infraestructura física, pero el usuario administra el SO y SQL Server como si fuera un servidor local.

SQL Server PaaS (Azure SQL Database / Managed Instance)

Plataforma de base de datos totalmente administrada. Azure gestiona el motor, actualizaciones, alta disponibilidad y seguridad. El usuario solo gestiona datos y usuarios.



3. Migración desde SQL Server local hacia Azure.

Definición :

La migración desde SQL Server local hacia Azure consiste en trasladar bases de datos, instancias o cargas de trabajo desde servidores físicos (on-premise) hacia servicios de nube como Azure SQL Database, Azure SQL Managed Instance o SQL Server en máquinas virtuales (IaaS). Este proceso busca mejorar la escalabilidad, disponibilidad y simplificar la administración, manteniendo la integridad y seguridad de los datos.

Ventajas de migrar a Azure

- Menor carga operativa: menos mantenimiento y parches.
- Escalabilidad inmediata sin comprar hardware.
- Alta disponibilidad (HA) integrada.
- Costos más predecibles (pago por uso).
- Seguridad avanzada y cifrado automático.
- Opciones híbridas para migrar gradualmente.

Desventajas / Retos

- Requiere una evaluación técnica previa para compatibilidad.
- Cambios en la aplicación si se usa Azure SQL Database (menos compatibilidad que Managed Instance).
- Costos mal calculados pueden ser altos si no se optimiza.
- Requiere configuración de redes seguras (VPN, ExpressRoute).
- Posibles tiempos de inactividad (dependen del método).

Tipos de migración

1. Lift & Shift (Rehost)

- Migrar tal cual, sin cambiar la aplicación.
- Se usa SQL Server en una VM (IaaS).
- Rápida, ideal para migraciones urgentes.

2. Migración con mínima modificación (Replatform)

- Se migra hacia Azure SQL Managed Instance.
- Alta compatibilidad con SQL Server local.
- La aplicación casi no requiere cambios.

3. Migración con modernización (Refactor)

- Se migra hacia Azure SQL Database (PaaS).
- Puede requerir cambios.
- Se beneficia de automatización y escalado.

Cómo funciona el proceso de migración (paso a paso)

1. Evaluación (Assessment)

Se analiza:

- Compatibilidad
- Dependencias
- Versiones de SQL
- Tamaño y rendimiento
- Funciones no soportadas

Herramientas:

- **Data Migration Assistant (DMA)**
- **Azure Migrate**
- **Azure Data Studio**

2. Preparación

- Habilitar redes (VPN, ExpressRoute).
- Crear recursos en Azure (Base, MI o VM).
- Ajustar configuraciones de seguridad.

3. Migración de datos

Métodos posibles:

- **Backup/Restore** a Azure
- **Azure Database Migration Service (DMS)**
- **Log Shipping**
- **Replication**
- **BACPAC (export/import)**

4. Pruebas (Testing)

- Verificar integridad de datos.
- Probar rendimiento.
- Validar compatibilidad de la aplicación.

5. Corte y puesta en producción (Cutover)

- Detener cambios en la BD local.
- Sincronización final.
- Redireccionar aplicaciones a Azure.

4. Seguridad, escalabilidad y costos en la nube.

A. Seguridad en la Nube

Definición

La seguridad en Azure SQL consiste en un conjunto de mecanismos automatizados y administrados por Microsoft que garantizan la protección de datos mediante cifrado, control de accesos, monitoreo continuo y cumplimiento de normativas internacionales.

Principales características de seguridad

1. Cifrado

- TDE (Transparent Data Encryption): Cifra la base y los backups automáticamente.
- Cifrado en tránsito (SSL/TLS): Protege la información en comunicación.
- Always Encrypted: Aísla datos sensibles incluso de administradores.

2. Control de acceso

- Autenticación Azure AD.
- Roles y permisos dentro de SQL.
- Firewalls de servidor y base de datos.

3. Monitoreo y protección

- SQL Defender: detección de amenazas con IA.
- Auditorías automáticas.
- Registro de actividad (logs y alerts).

4. Cumplimiento normativo

Azure cumple estándares como:
ISO 27001, SOC, HIPAA, GDPR, etc.

B. Escalabilidad en la Nube

Definición

La escalabilidad en Azure es la capacidad de aumentar o reducir automáticamente los recursos (CPU, memoria, almacenamiento o rendimiento) según la demanda, sin interrumpir los servicios.

Tipos de escalabilidad

1. Escalabilidad vertical (Scale Up/Down)

Aumenta o reduce la potencia del servidor.

Ejemplo: pasar de 4 a 16 vCores.

2. Escalabilidad horizontal (Scale Out)

Distribuye la carga en múltiples réplicas o bases.

Ejemplo: réplicas de lectura para alta concurrencia.

3. Escalabilidad automática

Disponible en:

- Azure SQL Database serverless
- Elastic Pools

Se adapta sin intervención humana.

C. Costos en la Nube

Definición

Los costos en Azure se basan en un modelo flexible de pago por uso, donde se paga solo por los recursos consumidos, eliminando inversiones iniciales en hardware y licencias.

Componentes principales del costo

1. Modelo de cómputo

- **DTU (Database Transaction Unit)**
- **vCore (Virtual Cores)** — más recomendado hoy

2. Almacenamiento

- Por GB usado
- Backups automáticos (retención configurable)

3. Tipo de servicio

- Azure SQL Database (más económico)
- Managed Instance (más completo, mayor costo)
- SQL en VM (IaaS — similar a local pero en la nube)

4. Licenciamiento

- Pago por hora
- Azure Hybrid Benefit (usa licencias existentes)
- Reserva por 1 o 3 años (hasta 55% menos)

Prácticas:

- Implementar una base de datos de prueba en Azure SQL.
- Conectar SSMS local con Azure SQL Database.

-- 1. Aseguramos usar la base de datos correcta

```
USE tiendaInformatica;
```

```
GO
```

-- 2. Limpieza preventiva: Si las tablas ya existen por errores anteriores, las borramos para crearlas limpias

-- (El orden importa: primero borramos la que tiene la llave foránea)

```
IF OBJECT_ID('producto', 'U') IS NOT NULL DROP TABLE producto;
```

```
IF OBJECT_ID('fabricante', 'U') IS NOT NULL DROP TABLE fabricante;
```

```
GO
```

-- 3. Creamos la tabla FABRICANTE

```
CREATE TABLE fabricante (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);
```

```
GO
```

-- 4. Creamos la tabla PRODUCTO

```
CREATE TABLE producto (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    precio FLOAT NOT NULL,
    id_fabricante INT NOT NULL,
    FOREIGN KEY (id_fabricante) REFERENCES fabricante(id)
);
```

```
GO
```

-- 5. Insertamos datos en FABRICANTE

-- Es importante el orden para que los IDs coincidan con los que pide el ejercicio

```
INSERT INTO fabricante (nombre) VALUES ('Asus'); -- id 1
```

```
INSERT INTO fabricante (nombre) VALUES ('Lenovo'); -- id 2
```

```
INSERT INTO fabricante (nombre) VALUES ('Hewlett-Packard'); -- id 3
```

```
INSERT INTO fabricante (nombre) VALUES ('Samsung'); -- id 4
```

```
INSERT INTO fabricante (nombre) VALUES ('Seagate'); -- id 5
```

```
INSERT INTO fabricante (nombre) VALUES ('Crucial'); -- id 6
```

```
INSERT INTO fabricante (nombre) VALUES ('Gigabyte'); -- id 7
```

```
INSERT INTO fabricante (nombre) VALUES ('Huawei'); -- id 8
```

```
INSERT INTO fabricante (nombre) VALUES ('Xiaomi'); -- id 9
```

```
GO
```

-- 6. Insertamos datos en PRODUCTO

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Disco duro SATA3 1TB', 86.99, 5);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Memoria RAM DDR4 8GB', 120, 6);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Disco SSD 1 TB', 150.99, 4);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('GeForce GTX 1050Ti', 185, 7);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('GeForce GTX 1080 Xtreme', 755, 7);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Monitor 24LED Full HD', 202, 1);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Monitor 27LED Full HD', 245.99, 1);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Portátil Yoga 520', 559, 2);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Portátil Ideapad 320', 444, 2);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Impresora HP Deskjet 3720', 59.99, 3);
```

```
INSERT INTO producto (nombre, precio, id_fabricante) VALUES ('Impresora HP LaserJet Pro M26nw', 180, 3);
```

```
GO
```

-- 7. Verificación rápida (Para que veas que todo cargó bien)

```
SELECT * FROM producto;
```

SERVERS

- > tcp:megambq.database.windows.n...
- > megambq.data...

Databases Security

Run Cancel Disconnect Change Database: tiendaInformatica Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

```
1 -- 1. Aseguramos usar la base de datos correcta
2 USE tiendaInformatica;
3 GO
4
5 -- 2. Limpieza preventiva: Si las tablas ya existen por errores anteriores, las borramos para crearlas limpias
6 -- (El orden importa: primero borramos la que tiene la llave foránea)
7 IF OBJECT_ID('producto', 'U') IS NOT NULL DROP TABLE producto;
8 IF OBJECT_ID('fabricante', 'U') IS NOT NULL DROP TABLE fabricante;
9 GO
10
11 -- 3. Creamos la tabla FABRICANTE
12 CREATE TABLE fabricante (
13     id INT IDENTITY(1,1) PRIMARY KEY,
14     nombre VARCHAR(100) NOT NULL
15 );
16 GO
17
18 -- 4. Creamos la tabla PRODUCTO
19 CREATE TABLE producto (
20     id INT IDENTITY(1,1) PRIMARY KEY,
21     nombre VARCHAR(100) NOT NULL,
22     precio FLOAT NOT NULL,
23     id_fabricante INT NOT NULL,
24     FOREIGN KEY (id_fabricante) REFERENCES fabricante(id)
```

Results Messages

	id	nombre	precio	id_fabricante
1	1	Disco duro SATA3 1TB	86.99	5
2	2	Memoria RAM DDR4 8GB	120	6
3	3	Disco SSD 1 TB	150.99	4
4	4	GeForce GTX 1050Ti	185	7
5	5	GeForce GTX 1080 Xtreme	755	7
6	6	Monitor 24 LED Full HD	202	1
7	7	Monitor 27 LED Full HD	245.99	1
8	8	Portátil Yoga 520	559	2
9	9	Portátil Ideapad 320	444	2
10	10	Impresora HP Deskjet 3720	59.99	3
11	11	Impresora HP Laserjet Pro M26nw	180	3



GRACIAS

