

*Examen Final*

---

**Administración de Base de Datos**

Página | 1

Apellidos y Nombres: PARIONA VILLANUEVA Jennifer Código: R01107D

Ciclo: V Salón: \_\_\_\_\_

**Enunciado 01:**

De acuerdo con la **base de datos** desarrollada en **Microsoft SQL Server**, responda las siguientes preguntas:

- 1) Explique que problema soluciona su base de datos  
La base de datos centraliza y organiza la información clave de la empresa (personal, inventario y ventas), eliminando redundancias y mejorando la eficiencia operativa. Esto permite:
  - Automatizar procesos manuales.
  - Reducir errores asociados a la falta de control o duplicación de datos.
  - Mejorar la capacidad de análisis y planificación para la toma de decisiones informadas.
- 2) Implemente un Script para crear una **vista** para crear utilizando tres tablas

```
CREATE VIEW vista_delosi_ventas_inventario AS
SELECT
    d.ID,
    d.NOMBRE,
    d.APELLIDO,
    d.CARGO,
    i.PRODUCTO,
    i.TALLA,
    i.COLOR,
    i.CANTIDAD,
    v.CANTIDAD AS CANTIDAD_VENDIDA
FROM
    delosi d
JOIN
    ventas v ON v.PRODUCTO = i.PRODUCTO
JOIN
    inventario i ON v.CODIGO = i.CODIGO;
```

- 3) Implemente un Script para crear un **procedimiento almacenado** para modificar el ingreso de datos en forma secuencial

Din:

```

CREATE PROCEDURE InsertarDatosSecuenciales
    @Nombre VARCHAR(50),
    @Apellido VARCHAR(50),
    @Cargo VARCHAR(50),
    @Producto VARCHAR(50),
    @Talla VARCHAR(4),
    @Color VARCHAR(20),
    @Cantidad INT
AS
BEGIN
    -- Insertar en la tabla delosi
    INSERT INTO delosi (NOMBRE, APELLIDO, CARGO)
    VALUES (@Nombre, @Apellido, @Cargo);

    -- Insertar en la tabla inventario
    DECLARE @Codigo INT = (SELECT MAX(CODIGO) FROM inventario) + 1;
    INSERT INTO inventario (CODIGO, PRODUCTO, TALLA, COLOR, CANTIDAD)
    VALUES (@Codigo, @Producto, @Talla, @Color, @Cantidad);

    -- Insertar en la tabla ventas
    INSERT INTO ventas (PRODUCTO, CODIGO, CANTIDAD)
    VALUES (@Producto, @Codigo, @Cantidad);
END;

```

- 4) Implemente un Script para crear un **disparador** para verificar el control de datos (Ejemplo: que la nota ingresada este entre 0 y 20)

```

CREATE TRIGGER VerificarNota
ON ventas
AFTER INSERT
AS
BEGIN
    DECLARE @Cantidad INT;
    SELECT @Cantidad = CANTIDAD FROM inserted;

    IF (@Cantidad < 0 OR @Cantidad > 20)
    BEGIN
        RAISERROR('La cantidad debe estar entre 0 y 20', 16, 1);
        ROLLBACK;
    END
END;

```

- 5) Utilizando Script Crear 03 usuarios con nombres de sus compañeros y uno suyo

Mg. Ing. Raúl Fernández Bejarano

```
CREATE LOGIN compañero1 WITH PASSWORD = 'Password123';
CREATE USER compañero1 FOR LOGIN compañero1;
```

```
CREATE LOGIN compañero2 WITH PASSWORD = 'Password123';
CREATE USER compañero2 FOR LOGIN compañero2;
```

```
CREATE LOGIN Jaime WITH PASSWORD = 'Password123';
CREATE USER Jaime FOR LOGIN Jaime;
```

- 6) Utilizando un script, copiar la base de datos (creada anteriormente) y compartir en cada uno de los usuarios

```
CREATE USER Jennifer FOR LOGIN Jennifer;
-- Crear la base de datos de copia
BACKUP DATABASE delosibd TO DISK = 'C:\backup\delosibd.bak';
RESTORE DATABASE delosibd_copy FROM DISK = 'C:\backup\delosibd.bak';

-- Compartir con los usuarios
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::delosibd_copy TO compañero1, compañero2, Jennifer;
```

- 7) Utilizando un script, generar una copia de seguridad de la base de datos y compartir a cada uno de los usuarios

```
BACKUP DATABASE delosibd TO DISK = 'C:\backup\delosibd.bak';
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::delosibd TO compañero1, compañero2, Jennifer;
```

- 8) Utilizando un script, encriptar una de las tablas para que no se puedan ver los datos

```
-- Encriptar la columna de la tabla delosi
CREATE COLUMN ENCRYPTION KEY MyColumnEncryptionKey
WITH VALUES (
    ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_512',
    ENCRYPTION PRIMARY KEY = 'MyPrimaryKey',
    ENCRYPTION KEY = 'MyEncryptionKey'
);

CREATE TABLE delosi_encriptado (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    NOMBRE VARCHAR(50) ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = MyColumnEncryptionKey),
    APELLIDO VARCHAR(50),
    CARGO VARCHAR(50)
);
```

- 9) Utilizando un script, aplique la seguridad a nivel de columna, restringiendo el acceso a la columna DNI de la tabla empleado en el usuario con nombre de su compañero

```
DENY SELECT ON dbo.empleado (DNI) TO compañero1;
```

- 10) Utilizando un script, implementé seguridad a nivel de columna restringiendo el acceso a una de las columnas de una tabla.

```
DENY SELECT ON dbo.inventario (COLOR) TO compañero2;
```

- 11) Utilizando un script, realice el cifrado transparente de datos (TDE) para una las tablas.

Din:

```
-- Crear un certificado
CREATE CERTIFICATE MyCertificate
WITH SUBJECT = 'MyEncryptionCertificate';

-- Crear una base de datos cifrada
CREATE DATABASE delosibd_encriptado;
ALTER DATABASE delosibd_encriptado SET ENCRYPTION ON;
```

Página | 4

- 12) Utilizando un script, configure el usuario con el nombre de su compañero para otorgar permisos de SELECT, INSERT, UPDATE y DELETE en la base de datos.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON DATABASE::delosibd TO compañero1;
```

- 13) Utilizando un script, configure la auditoría para el seguimiento y registro de acciones en la base de datos

```
-- Crear auditoría
CREATE SERVER AUDIT DelosiAudit
TO FILE (FILEPATH = 'C:\Audit\')
WITH (ON_FAILURE = CONTINUE);

-- Crear especificación de auditoría
CREATE SERVER AUDIT SPECIFICATION DelosiAuditSpec
FOR SERVER AUDIT DelosiAudit
ADD (SCHEMA_OBJECT_ACCESS_GROUP)
WITH (STATE = ON);
```

- 14) Utilizando un script, configure de la memoria y el disco duro

```
-- Configurar la memoria
EXEC sp_configure 'max server memory', 4096;
RECONFIGURE;

-- Configurar el disco
ALTER DATABASE delosibd
MODIFY FILE (NAME = 'delosibd_data', SIZE = 10GB);
```

- 15) Utilizando un script, genere una copia de seguridad de la base de datos

```
BACKUP DATABASE delosibd TO DISK = 'C:\backup\delosibd.bak';
```

- 16) Utilizando un script, genere la restauración de la base de datos

```
RESTORE DATABASE delosibd FROM DISK = 'C:\backup\delosibd.bak';
```

- 17) Utilizando un script, cree un espejo de la base de datos

```
-- Configurar el espejo de la base de datos
ALTER DATABASE delosibd
SET PARTNER = 'TCP://ServidorSecundario:5022';
```

- 18) Utilizando un script, realice la replicación de bases de datos

Mg. Ing. Raúl Fernández Bejarano

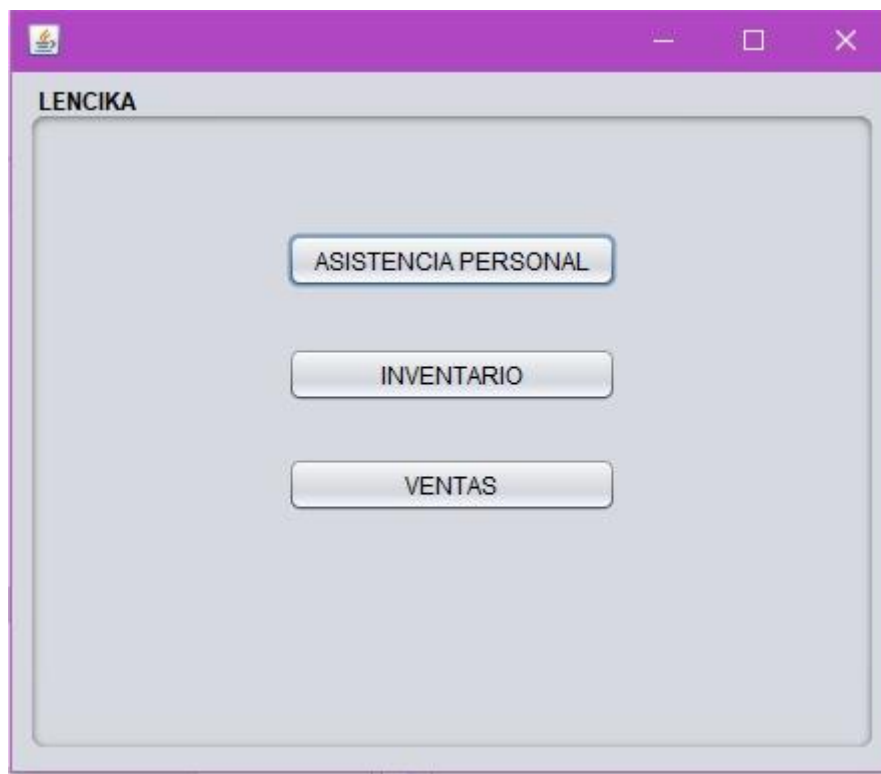
```
-- Configurar publicación para replicación
EXEC sp_replicationdboption
@dbname = N'delosibd',
@optname = N'publish',
@value = N'true';
```

- 19) Explique que es Always On Availability Groups

**Always On Availability Groups** es una característica de alta disponibilidad en SQL Server. Permite la creación de grupos de bases de datos que están replicados en múltiples servidores, garantizando que la base de datos primaria esté siempre disponible incluso si uno de los servidores fallara.

- 20) Explique que es Log Shipping

**Log Shipping** es una técnica de recuperación ante desastres que se utiliza para mantener una base de datos de respaldo sincronizada con la base de datos principal. En este proceso, los registros de transacciones de la base de datos primaria se copian y aplican a la base de datos secundaria en intervalos regulares.



ASISTENCIA DEL PERSONAL

**DATOS**

ID

NOMBRE

APELLIDO

CARGO

**OPCIONES**

AGREGAR ELIMINAR

MODIFICAR NUEVO

REGRESAR

**ASISTENCIA**

ID	NOMBRE	APELLIDO	CARGO
2	mario	mario	Al
3	JENNIFER	JJ	1
4	jennifer	pv	4
5	jennifer	pv	4
6	AAA	BBB	AA

INVENTARIO LENCICA

**DATOS**

CODIGO

PRODUCTO

TALLA

COLOR

CANTIDAD

**OPCIONES**

AGREGAR ELIMINAR

MODIFICAR NUEVO

REGRESAR

**INVENTARIO**

CODIGO	PRODUCTO	TALLA	COLOR	CANTIDAD
301	Int. merc	s	negro	30
57	Int. clasico...	xl	negro	22

—

□

×

LENCIKA - VENTAS

**VENTAS**

N° TIKECT:

PRODUCTO

CODIGO

CANTIDAD:

**OPCIONES**

AGREGAR ORDEN

ELIMINAR ORDEN

LIMPIAR

REGRESAR

**DATOS**

N° TICKET	PRODUCTO	CODIGO	CANTIDAD
2	int. Jacky F...	1331	2