

UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y  
COMPUTACIÓN



**U P L A**

**“SEMANA 10”**

**ASIGNATURA:** Base de Datos II

**ESTUDIANTE:** Luis Enrique Espejo Quispe

**DOCENTE:** Mg. Raúl Bejarano Fernández

**CICLO:** V

**SECCIÓN:** A1

**HYO-2025**

## 1. ESTRUCTURA DE ALMACENAMIENTO Y ARCHIVOS DE DATOS

Explicar la estructura de almacenamiento en SQL Server y cómo se organizan los archivos de datos. Documentar la estructura actual de la base de datos QhatuPERU.

### SCRIPT T-SQL:

```
15 -- Consultar la estructura de almacenamiento de QhatuPERU
16 SELECT
17     mf.file_id,
18     mf.name AS NombreArchivo,
19     mf.physical_name AS RutaFisica,
20     CAST((mf.size * 8.0) / 1024 AS DECIMAL(10,2)) AS TamanioMB,
21     CASE
22         WHEN mf.type = 0 THEN 'Datos (MDF/NDF)'
23         WHEN mf.type = 1 THEN 'Log (LDF)'
24     END AS TipoArchivo,
25     CAST((mf.max_size * 8.0) / 1024 / 1024 AS DECIMAL(10,2)) AS TamanioMaximoGB
26 FROM sys.master_files mf
27 WHERE database_id = DB_ID('QhatuPERU')
28 ORDER BY mf.type, mf.file_id;
29 GO
30
```

81 %

Results Messages

	file_id	NombreArchivo	RutaFisica	TamanioMB	TipoArchivo	TamanioMaximoGB
1	1	QhatuPeru_Data	D:\BaseDatos2025\QhatuPeru_Data.mdf	8.00	Datos (MDF/NDF)	0.02
2	2	QhatuPeru_Log	D:\BaseDatos2025\QhatuPeru_Log.ldf	2.00	Log (LDF)	0.01

```
31 -- Consultar filegroups disponibles de una base específica
32 SELECT
33     fg.name AS NombreFileGroup,
34     fg.type_desc AS Tipo,
35     COUNT(mf.file_id) AS CantidadArchivos
36     FROM sys.filegroups fg
37     LEFT JOIN sys.master_files mf
38         ON fg.data_space_id = mf.data_space_id
39     WHERE mf.database_id = DB_ID('QhatuPERU')
40     GROUP BY fg.name, fg.type_desc;
41 GO
42
```

1 %

Results Messages

	NombreFileGroup	Tipo	CantidadArchivos
1	PRIMARY	ROWS_FILEGROUP	1

## JUSTIFICACIÓN TÉCNICA

COMPONENTES:

**1. MDF (Primary Data File):**

- Archivo principal de datos
- Contiene tablas, índices, vistas indexadas
- Máximo 1 por base de datos

**2. NDF (Secondary Data File):**

- Archivos de datos secundarios
- Distribuyen carga I/O
- Mejoran rendimiento en bases de datos grandes

**3. LDF (Log File):**

- Archivo de registro de transacciones
- Crítico para recuperación
- Puede ser múltiple (hasta 32 GB por archivo)

**VENTAJAS DE LA ESTRUCTURA:**

- Distribuir I/O entre múltiples discos
- Mejor rendimiento en consultas
- Recuperación eficiente
- Mantenimiento sin downtime

**BUENAS PRÁCTICAS**

**RECOMENDACIONES:**

**1. UBICACIÓN:**

- MDF/NDF: Discos SSD de alta velocidad
- LDF: Disco separado (RAID 1 o 10)
- Logs: Nunca en red lenta

**2. TAMAÑO:**

- MDF inicial: Mínimo tamaño esperado
- Crecimiento automático: 10-15%
- LDF: 25% del tamaño de datos

**3. FILEGROUPS:**

- Usar para distribuir I/O
- Índices clustered en separate filegroup
- Datos históricos en filegroup readonly

## 2. PROPIEDADES Y CONFIGURACIONES DE BASES DE DATOS

Mostrar y modificar propiedades y configuraciones de la base de datos QhatuPERU, así como explicar el impacto de cada configuración.

### SCRIPT T-SQL:

```
-- Consultar propiedades actuales de QhatuPERU
54
55 SELECT
56     db.name AS NombreBaseDatos,
57     db.state_desc AS Estado,
58     db.recovery_model_desc AS ModeloRecuperacion,
59     db.is_read_only AS EsSoloLectura,
60     db.is_auto_close_on AS CierreAutomatico,
61     db.is_auto_shrink_on AS EncogimientoAutomatico,
62     db.compatibility_level AS NivelCompatibilidad,
63     db.is_db_chaining_on AS CadenasDB,
64     db.is_trustworthy_on AS Confiable,
65     CAST(ls.cntr_value * 8.0 / 1024 / 1024 AS DECIMAL(10,2)) AS TamanioDB_GB
66 FROM sys.databases db
67 LEFT JOIN sys.dm_os_performance_counters ls
68     ON ls.counter_name = 'Data File(s) Size (KB)'
69     AND db.name = CASE
70         WHEN CHARINDEX('_', ls.instance_name) > 0
71             THEN LEFT(ls.instance_name, CHARINDEX('_', ls.instance_name) - 1)
72         ELSE ls.instance_name
73     END
74 WHERE db.name = 'QhatuPERU';
75 GO
76
```

Results Messages

	NombreBaseDatos	Estado	ModeloRecuperacion	EsSoloLectura	CierreAutomatico	EncogimientoAutomatico	NivelCompatibilidad	CadenasDB	Confiable	TamanioDB_GB
1	QhatuPeru	ONLINE	FULL	0	0	0	160	0	0	0.06

```

78 -- Consultar opciones de configuración detalladas
79 USE QhatuPERU;
80 GO
81
82 SELECT
83     name AS Propiedad,
84     CASE value
85         WHEN 1 THEN 'ACTIVADO'
86         WHEN 0 THEN 'DESACTIVADO'
87         ELSE CAST(value AS VARCHAR(50))
88     END AS Valor
89     FROM sys.database_scoped_configurations
90     ORDER BY name;
91 GO

```

31 %

Results Messages

	Propiedad	Valor
1	ACCELERATED_PLAN_FORCING	ACTIVADO
2	ASYNC_STATS_UPDATE_WAIT_AT_LOW_PRIORITY	DESACTIVADO
3	BATCH_MODE_ADAPTIVE_JOINS	ACTIVADO
4	BATCH_MODE_MEMORY_GRANT_FEEDBACK	ACTIVADO
5	BATCH_MODE_ON_ROWSTORE	ACTIVADO
6	CE_FEEDBACK	ACTIVADO
7	DEFERRED_COMPILATION_TV	ACTIVADO
8	DOP_FEEDBACK	DESACTIVADO
9	DW_COMPATIBILITY_LEVEL	DESACTIVADO
10	ELEVATE_ONLINE	OFF
11	ELEVATE_RESUMABLE	OFF
12	EXEC_QUERY_STATS_FOR_SCALAR_FUNCTIONS	ACTIVADO
13	FORCE_SHOWPLAN_RUNTIME_PARAMETER_COLLECTION	DESACTIVADO
14	GLOBAL_TEMPORARY_TABLE_AUTO_DROP	ACTIVADO
15	IDENTITY_CACHE	ACTIVADO
16	INTERLEAVED_EXECUTION_TVF	ACTIVADO
17	ISOLATE_SECURITY_POLICY_CARDINALITY	DESACTIVADO
18	LAST_QUERY_PLAN_STATS	DESACTIVADO
19	LEDGER_DIGEST_STORAGE_ENDPOINT	OFF
20	LEGACY_CARDINALITY_ESTIMATION	DESACTIVADO
21	LIGHTWEIGHT_QUERY_PROFILING	ACTIVADO
22	MAXDOP	DESACTIVADO
23	MEMORY_GRANT_FEEDBACK_PERCENTILE_GRANT	ACTIVADO
24	MEMORY_GRANT_FEEDBACK_PERSISTENCE	ACTIVADO
25	OPTIMIZE_FOR_AD_HOC_WORKLOADS	DESACTIVADO
26	OPTIMIZED_PLAN_FORCING	ACTIVADO
27	PARAMETER_SENSITIVE_PLAN_OPTIMIZATION	ACTIVADO
28	PARAMETER_SNIFFING	ACTIVADO
29	PAUSED_RESUMABLE_INDEX_ABORT_DURATION_MINU...	1440
30	QUERY_OPTIMIZER_HOTFIXES	DESACTIVADO
31	ROW_MODE_MEMORY_GRANT_FEEDBACK	ACTIVADO
32	TSQL_SCALAR_UDF_INLINING	ACTIVADO
33	VERBOSE_TRUNCATION_WARNINGS	ACTIVADO
34	XTP PROCEDURE EXECUTION_STATISTICS	DESACTIVADO
35	XTP QUERY EXECUTION_STATISTICS	DESACTIVADO

## PRÁCTICA:

The screenshot shows a SQL Server Management Studio window with two panes. The top pane displays a script of T-SQL commands numbered 94 to 111. The bottom pane shows the execution results in the 'Messages' tab, indicating successful completion and providing the completion time.

```
94 -- 2. PRÁCTICA: CAMBIAR PROPIEDADES
95
96 -- Práctica 2.1: Cambiar a modelo FULL
97 ALTER DATABASE QhatuPERU SET RECOVERY FULL;
98 GO
99
100 -- Práctica 2.2: Desactivar cierre automático
101 ALTER DATABASE QhatuPERU SET AUTO_CLOSE OFF;
102 GO
103
104 -- Práctica 2.3: Desactivar encogimiento automático
105 ALTER DATABASE QhatuPERU SET AUTO_SHRINK OFF;
106 GO
107
108 -- Práctica 2.4: Establecer compatibilidad a SQL Server 2019
109 ALTER DATABASE QhatuPERU SET COMPATIBILITY_LEVEL = 150;
110 GO
111
```

Commands completed successfully.  
Completion time: 2025-12-10T00:51:05.5708421-05:00

```
112 -- Práctica 2.5: Verificar cambios
113 SELECT
114     name,
115     recovery_model_desc,
116     is_auto_close_on,
117     is_auto_shrink_on,
118     compatibility_level
119 FROM sys.databases
120 WHERE name = 'QhatuPERU';
121 GO
122
```

Results

	name	recovery_model_desc	is_auto_close_on	is_auto_shrink_on	compatibility_level
1	QhatuPeru	FULL	0	0	150

## JUSTIFICACIÓN TÉCNICA

### PROPIEDADES CRÍTICAS:

#### 1. RECOVERY MODEL (Modelo de Recuperación):

MODELO	BACKUPS	PÉRDIDA	LOG
SIMPLE	FULL/DIFF	Hasta último FULL backup	Truncado automáticamente
BULK_LOGGED	FULL/DIFF	Minutos	Truncado en checkpoints
FULL	FULL/DIFF+LOG	Segundos (granular)	Manual/Log backups

## 2. AUTO\_CLOSE:

- ON: Base de datos se cierra si no hay conexiones
- Ventaja: Libera memoria en dev
- Desventaja: Performance penalty en producción
- RECOMENDACIÓN: OFF en producción

## 3. AUTO\_SHRINK:

- ON: Encoge archivos automáticamente
- Ventaja: Economiza espacio
- Desventaja: Causa fragmentación, I/O intensivo
- RECOMENDACIÓN: OFF + mantenimiento manual

## 4. COMPATIBILITY\_LEVEL:

- 150 = SQL Server 2019/2022
- 140 = SQL Server 2017
- Permite usar funciones nuevas manteniendo compatibilidad

## **BUENAS PRÁCTICAS**

CONFIGURACIÓN RECOMENDADA PARA QhatuPERU (Producción):

RECOVERY FULL (Recuperación completa)

AUTO\_CLOSE OFF (Conexión permanente)

AUTO\_SHRINK OFF (Mantenimiento manual)

COMPATIBILITY\_LEVEL 150 (Última versión)

TRUSTWORTHY OFF (Seguridad por defecto)

## 3. TIPOS DE RECUPERACIÓN (Simple, Full, Bulk-Logged)

Explicar los tres modelos de recuperación en SQL Server y cuándo usar cada uno e implementar cambios de recuperación en QhatuPERU.

## **SCRIPT T-SQL:**

```
180 -- Consultar modelo actual
181 SELECT
182     recovery_model_desc AS ModeloActual,
183     CASE recovery_model_desc
184         WHEN 'SIMPLE' THEN 'Backup FULL/DIFF. No LOG. Para dev/test'
185         WHEN 'FULL' THEN 'Backup FULL/DIFF/LOG. Recuperación granular'
186         WHEN 'BULK_LOGGED' THEN 'Backup FULL/DIFF. Operaciones bulk optimizadas'
187     END AS Descripcion
188 FROM sys.databases
189 WHERE name = 'QhatuPERU';
190 GO
191
```

81 %

Results Messages

	ModeloActual	Descripcion
1	FULL	Backup FULL/DIFF/LOG. Recuperación granular

SIMPLE:

```
-- Práctica 3.1: Cambiar a SIMPLE (solo para demo, NO en producción)
ALTER DATABASE QhatuPERU SET RECOVERY SIMPLE;
GO
```

```
181 -- Consultar modelo actual
182 SELECT
183     recovery_model_desc AS ModeloActual,
184     CASE recovery_model_desc
185         WHEN 'SIMPLE' THEN 'Backup FULL/DIFF. No LOG. Para dev/test'
186         WHEN 'FULL' THEN 'Backup FULL/DIFF/LOG. Recuperación granular'
187         WHEN 'BULK_LOGGED' THEN 'Backup FULL/DIFF. Operaciones bulk optimizadas'
188     END AS Descripcion
189 FROM sys.databases
190 WHERE name = 'QhatuPERU';
191 GO
192
```

81 %

Results Messages

	ModeloActual	Descripcion
1	SIMPLE	Backup FULL/DIFF. No LOG. Para dev/test

BULK\_LOGGED:

```
-- Práctica 3.2: Cambiar a BULK_LOGGED (para operaciones masivas)
ALTER DATABASE QhatuPERU SET RECOVERY BULK_LOGGED;
GO
```

```
181 -- Consultar modelo actual
182 SELECT
183     recovery_model_desc AS ModeloActual,
184     CASE recovery_model_desc
185         WHEN 'SIMPLE' THEN 'Backup FULL/DIFF. No LOG. Para dev/test'
186         WHEN 'FULL' THEN 'Backup FULL/DIFF/LOG. Recuperación granular'
187         WHEN 'BULK_LOGGED' THEN 'Backup FULL/DIFF. Operaciones bulk optimizadas'
188     END AS Descripcion
189     FROM sys.databases
190     WHERE name = 'QhatuPERU';
191 GO
192
```

81 %

Results Messages

ModeloActual	Descripcion
BULK_LOGGED	Backup FULL/DIFF. Operaciones bulk optimizadas

FULL:

```
-- Práctica 3.3: Volver a FULL (recomendado para QhatuPERU)
ALTER DATABASE QhatuPERU SET RECOVERY FULL;
GO
```

```
180
181 -- Consultar modelo actual
182 SELECT
183     recovery_model_desc AS ModeloActual,
184     CASE recovery_model_desc
185         WHEN 'SIMPLE' THEN 'Backup FULL/DIFF. No LOG. Para dev/test'
186         WHEN 'FULL' THEN 'Backup FULL/DIFF/LOG. Recuperación granular'
187         WHEN 'BULK_LOGGED' THEN 'Backup FULL/DIFF. Operaciones bulk optimizadas'
188     END AS Descripcion
189     FROM sys.databases
190     WHERE name = 'QhatuPERU';
191 GO
192
```

81 %

Results Messages

ModeloActual	Descripcion
FULL	Backup FULL/DIFF/LOG. Recuperación granular

BACKUP DE RESPALDO:

```

207 -- Práctica 3.4: Crear backup de log después de cambio
208 BACKUP DATABASE QhatuPERU
209   TO DISK = 'D:\BackupSQL\QhatuPERU_FULL.bak'
210   WITH INIT, COMPRESSION;
211   GO
212
213 BACKUP LOG QhatuPERU
214   TO DISK = 'D:\BackupSQL\QhatuPERU_LOG_PostSwitch.trn'
215   WITH INIT, COMPRESSION;
216   GO
217
81 % ▾
Messages
Processed 3 pages for database 'QhatuPERU', file 'QhatuPeru_Log' on file 1.
BACKUP LOG successfully processed 3 pages in 0.006 seconds (3.906 MB/sec).

Completion time: 2025-12-10T01:04:03.0357029-05:00

```

## JUSTIFICACIÓN TÉCNICA

### COMPARATIVA DE MODELOS:

CARACTERISTICA	SIMPLE	FULL	BULK_LOGGED
TAMAÑO LOG	PEQUEÑO	GRANDE	MEDIO
FREQ. BACKUP	SEMANAL	DIARIO/HORA	DIARIO
PITR	NO	SI	LIMITADO
OPERACIÓN BULK	COMPLETO	LOGGING	MINIMAL LOG
RECOVERY TIME	RÁPIDO	MÁS LENTO	RÁPIDO
MEJOR PARA	DEV/TEST	PRODUCCIÓN	OPERACIÓN BULK

#### SIMPLE:

- Logs se truncan automáticamente en checkpoints
- No soporta backups de log transaccionales
- Pérdida de datos si falla entre FULL backups
- NUNCA usar en producción

#### FULL:

- Logs se mantienen hasta backup de log
- Soporta Point-In-Time Recovery (PITR)
- Recuperación granular por timestamp
- Recomendado para sistemas críticos

#### BULK\_LOGGED:

- Optimiza operaciones masivas (INSERT, BULK INSERT)
- Logs minimales para operaciones bulk
- Útil durante operaciones ETL
- Usar temporalmente, volver a FULL

#### 4. ADMINISTRACIÓN DE USUARIOS Y ROLES DE SEGURIDAD

Crear usuarios, roles y asignar permisos en SQL Server e implementar modelo de seguridad por rol para QhatuPERU.

#### SCRIPT T-SQL:

```
184  -- Práctica 4.1: Crear Login a nivel servidor
185  CREATE LOGIN LoginGestorInventario WITH PASSWORD = 'P@ssw0rd2024Segura!';
186  GO
187
188  -- Práctica 4.2: Crear Usuario en base de datos QhatuPERU
189  CREATE USER UsuarioGestorInventario FOR LOGIN LoginGestorInventario;
190  GO
191
192  -- Práctica 4.3: Crear rol personalizado
193  CREATE ROLE RolGestorInventario;
194  GO
195
196  -- Práctica 4.4: Crear rol solo lectura
197  CREATE ROLE RolAuditor;
198  GO
199
200  -- Práctica 4.5: Asignar usuarios a roles
201  ALTER ROLE RolGestorInventario ADD MEMBER UsuarioGestorInventario;
202  GO
203
```

31 % ▾

Messages

Commands completed successfully.

Completion time: 2025-12-10T01:11:01.6293586-05:00

```

204  -- Consultar usuarios y roles creados
205  USE QhatuPERU;
206  GO
207
208  SELECT
209      name AS NombreEntidad,
210      type_desc AS Tipo
211  FROM sys.database_principals
212  WHERE type IN ('U', 'R')    -- U = User, R = Database Role
213  ORDER BY type_desc, name;
214  GO
215

```

81 %

Results Messages

	NombreEntidad	Tipo
1	db_accessadmin	DATABASE_ROLE
2	db_backupoperator	DATABASE_ROLE
3	db_datareader	DATABASE_ROLE
4	db_datawriter	DATABASE_ROLE
5	db_ddladmin	DATABASE_ROLE
6	db_denydatareader	DATABASE_ROLE
7	db_denydatawriter	DATABASE_ROLE
8	db_owner	DATABASE_ROLE
9	db_securityadmin	DATABASE_ROLE
10	public	DATABASE_ROLE
11	RolAuditor	DATABASE_ROLE
12	RolGestorInventario	DATABASE_ROLE
13	dbo	WINDOWS_USER

```

216
217  -- Consultar miembros de roles
218  SELECT |
219      r.name AS NombreRol,
220      m.name AS MiembroDelRol
221  FROM sys.database_role_members drm
222  INNER JOIN sys.database_principals r ON drm.role_principal_id = r.principal_id
223  INNER JOIN sys.database_principals m ON drm.member_principal_id = m.principal_id
224  ORDER BY r.name;
225  GO
226

```

81 %

Results Messages

	NombreRol	MiembroDelRol
1	db_owner	dbo
2	RolGestorInventario	UsuarioGestorInventario

## JUSTIFICACIÓN TÉCNICA: PRINCIPIOS DE SEGURIDAD

**1. LEAST PRIVILEGE (Menor Privilegio):**

- Asignar solo permisos necesarios
- No usar sa (admin) para aplicaciones
- Crear roles específicos por función

**2. SEPARATION OF DUTIES (Separación de Funciones):**

- Gestor Inventario: CRUD en ARTICULO
- Auditor: SELECT en todas las tablas
- Admin: SELECT en tablas de sistema

**3. DEFENSE IN DEPTH (Defensa en Profundidad):**

- Login + User + Role + Permissions
- Validación en múltiples capas
- Auditoría de acceso

**BUENAS PRÁCTICAS**

**CONTRASEÑAS SEGURAS:**

- Mínimo 12 caracteres
- Mayúsculas + minúsculas + números + símbolos
- Cambio cada 90 días
- Nunca guardar en scripts

**ROLES RECOMENDADOS PARA QhatuPERU:**

- RolGestorInventario: CRUD en ARTICULO, LINEA, PROVEEDOR
- RolVentas: SELECT en ARTICULO, INSERT en GUIA\_ENVIO
- RolAuditor: SELECT en todas las tablas
- RolAdmin: Todos los permisos

**5. ASIGNACIÓN DE PERMISOS Y POLÍTICAS DE ACCESO**

Asignar permisos granulares a roles y usuarios e implementar políticas de acceso por objeto.

**SCRIPT T-SQL:**

```
238 | -- Práctica 5.1: Otorgar SELECT en tabla ARTICULO
239 | GRANT SELECT ON ARTICULO TO RolGestorInventario;
240 | GO
241 |
242 | -- Práctica 5.2: Otorgar INSERT en tabla ARTICULO
243 | GRANT INSERT ON ARTICULO TO RolGestorInventario;
244 | GO
245 |
246 | -- Práctica 5.3: Otorgar UPDATE en tabla ARTICULO
247 | GRANT UPDATE ON ARTICULO TO RolGestorInventario;
248 | GO
249 |
250 | -- Práctica 5.4: Otorgar DELETE en tabla ARTICULO
251 | GRANT DELETE ON ARTICULO TO RolGestorInventario;
252 | GO
253 |
254 | -- Práctica 5.5: Otorgar SELECT en LINEA y PROVEEDOR (relaciones)
255 | GRANT SELECT ON LINEA TO RolGestorInventario;
256 | GRANT SELECT ON PROVEEDOR TO RolGestorInventario;
257 | GO
258 |
259 | -- Práctica 5.6: Crear rol Auditor con permisos de lectura
260 | GRANT SELECT ON TIENDA TO RolAuditor;
261 | GRANT SELECT ON LINEA TO RolAuditor;
262 | GRANT SELECT ON PROVEEDOR TO RolAuditor;
263 | GRANT SELECT ON ARTICULO TO RolAuditor;
264 | GRANT SELECT ON ORDEN_COMPRA TO RolAuditor;
265 | GRANT SELECT ON ORDEN_DETALLE TO RolAuditor;
266 | GRANT SELECT ON GUIA_ENVIO TO RolAuditor;
267 | GRANT SELECT ON GUIA_DETALLE TO RolAuditor;
268 | GRANT SELECT ON TRANSPORTISTA TO RolAuditor;
269 | GO
270 |
```

81 % ▶

#### Messages

Commands completed successfully.

Completion time: 2025-12-10T01:16:18.7634289-05:00

```

271 -- Consultar permisos asignados
272 SELECT
273     r.name AS Rol,
274     p.permission_name AS Permiso,
275     p.state_desc AS Estado,
276     OBJECT_NAME(p.major_id) AS Tabla
277 FROM sys.database_permissions p
278 INNER JOIN sys.database_principals r ON p.grantee_principal_id = r.principal_id
279 WHERE r.name IN ('RolGestorInventario', 'RolAuditor')
280 ORDER BY r.name, OBJECT_NAME(p.major_id);
281 GO

```

81 %

	Rol	Permiso	Estado	Tabla
1	RolAuditor	SELECT	GRANT	ARTICULO
2	RolAuditor	SELECT	GRANT	GUIA_DETALLE
3	RolAuditor	SELECT	GRANT	GUIA_ENVIO
4	RolAuditor	SELECT	GRANT	LINEA
5	RolAuditor	SELECT	GRANT	ORDEN_COMPRA
6	RolAuditor	SELECT	GRANT	ORDEN_DETALLE
7	RolAuditor	SELECT	GRANT	PROVEEDOR
8	RolAuditor	SELECT	GRANT	TIENDA
9	RolAuditor	SELECT	GRANT	TRANSPORTISTA
10	RolGestorInventario	DELETE	GRANT	ARTICULO
11	RolGestorInventario	INSERT	GRANT	ARTICULO
12	RolGestorInventario	SELECT	GRANT	ARTICULO
13	RolGestorInventario	UPDATE	GRANT	ARTICULO
14	RolGestorInventario	SELECT	GRANT	LINEA
15	RolGestorInventario	SELECT	GRANT	PROVEEDOR

## JUSTIFICACIÓN TÉCNICA:

### PERMISOS EN SQL SERVER:

#### NIVEL SERVIDOR:

- CONTROL SERVER
- ALTER ANY DATABASE
- ALTER ANY LOGIN
- CREATE ENDPOINT

#### NIVEL DATABASE:

- CONTROL (Total)
- ALTER (Modificar BD)
- CREATE TABLE (Crear tablas)
- CREATE PROCEDURE (Crear procedimientos)

#### NIVEL OBJETO:

- SELECT (Lectura)
- INSERT (Agregar datos)
- UPDATE (Modificar datos)

- DELETE (Eliminar datos)
- EXECUTE (Ejecutar procedimientos)
- REFERENCES (Claves foráneas)

## **BUENAS PRÁCTICAS**

MODELO RBAC (Role-Based Access Control):

1. CREAR ROLES por función:

RolGestorInventario (CRUD inventario)  
RolGestorCompras (Órdenes)  
RolGestorVentas (Guías de envío)  
RolAuditor (Solo lectura)  
NO otorgar permisos directos a usuarios

2. AUDITAR CAMBIOS:

- sys.database\_permissions: Permisos actuales  
- Registrar en tabla de auditoria cada cambio  
- REVOKE: Remover permisos antiguos

3. REVISAR PERIÓDICAMENTE:

- Eliminar usuarios inactivos  
- Actualizar permisos según cambios de rol  
- Documentar justificación de permisos

6. MONITOREO BÁSICO CON SQL SERVER ACTIVITY MONITOR

Usar SQL Server Activity Monitor para monitorear la actividad de la base de datos e identificar sesiones lentas, bloqueos y recursos intensivos.

## **SCRIPT T-SQL:**

```

300  -- Práctica 6.1: Ver sesiones activas
301  SELECT
302      s.session_id,
303      s.login_name,
304      s.host_name,
305      DB_NAME(r.database_id) AS database_name,
306      s.status,
307      r.command,
308      DATEDIFF(SECOND, s.login_time, GETDATE()) AS TiempoConexion_Seg
309  FROM sys.dm_exec_sessions s
310  LEFT JOIN sys.dm_exec_connections c ON s.session_id = c.session_id
311  LEFT JOIN sys.dm_exec_requests r ON s.session_id = r.session_id
312  WHERE DB_NAME(r.database_id) = 'QhatuPERU'
313  AND s.session_id > 50    -- evita procesos del sistema
314  ORDER BY TiempoConexion_Seg DESC;
315  GO
316

```

81 % ▶

Results Messages

	session_id	login_name	host_name	database_name	status	command	TiempoConexion_Seg
1	60	sa	NULL	QhatuPeru	sleeping	SYSTEM_HEALTH_MONITOR	17801
2	75	sa	NULL	QhatuPeru	sleeping	HADR_AR_MGR_NOTIFICATION_WORKER	17801
3	58	HP123\USUARIO	HP123	QhatuPeru	running	SELECT	2819

31/ -- Práctica 6.2: Ver procesos bloqueados

```

319  SELECT
320      er.session_id,
321      er.status,
322      er.wait_type,
323      er.wait_time AS TiempoEspera_ms,
324      er.blocking_session_id,
325      SUBSTRING(st.text, 1, 100) AS ConsultaSQL
326  FROM sys.dm_exec_requests er
327  CROSS APPLY sys.dm_exec_sql_text(er.sql_handle) st
328  WHERE er.database_id = DB_ID('QhatuPERU')
329  AND er.blocking_session_id > 0;
330  GO
331

```

31 % ▶

Results Messages

session_id	status	wait_type	TiempoEspera_ms	blocking_session_id	ConsultaSQL
------------	--------	-----------	-----------------	---------------------	-------------

```

332 -- Práctica 6.3: Ver recursos de las sesiones
333 SELECT
334     session_id,
335     login_name,
336     reads,
337     writes,
338     logical_reads,
339     memory_usage,
340     CAST(memory_usage * 8.0 / 1024 AS DECIMAL(10,2)) AS MemoriaMB
341 FROM sys.dm_exec_sessions
342 WHERE database_id = DB_ID('QhatuPERU')
343 AND session_id > 50
344 ORDER BY logical_reads DESC;
345 GO
346

```

81 %

Results Messages

	session_id	login_name	reads	writes	logical_reads	memory_usage	MemoriaMB
1	52	HP123\USUARIO	2639	70	288724	4	0.03
2	58	HP123\USUARIO	7840	91	29690	4	0.03
3	78	HP123\USUARIO	0	0	0	4	0.03

81 %

Results Messages

```

347 -- Práctica 6.4: Ver tamaño de la base de datos
348 SELECT
349     mf.name AS NombreArchivo,
350     CAST((mf.size * 8.0) / 1024 AS DECIMAL(10,2)) AS TamañoMB,
351     CASE WHEN mf.type = 0 THEN 'DATOS' ELSE 'LOG' END AS Tipo
352     FROM sys.master_files mf
353 WHERE database_id = DB_ID('QhatuPERU')
354 ORDER BY mf.type, mf.name;
355 GO
356

```

	NombreArchivo	TamañoMB	Tipo
1	QhatuPeru_Data	8.00	DATOS
2	QhatuPeru_Log	2.00	LOG

```

556
357 -- Práctica 6.5: Ver conexiones por usuario
358 SELECT
359     s.login_name,
360     COUNT(s.session_id) AS CantidadConexiones
361 FROM sys.dm_exec_sessions s
362 LEFT JOIN sys.dm_exec_requests r
363     ON s.session_id = r.session_id
364 WHERE DB_NAME(r.database_id) = 'QhatuPERU'
365 AND s.session_id > 50    -- evita procesos del sistema
366 GROUP BY s.login_name
367 ORDER BY CantidadConexiones DESC;
368 GO
369

```

81 %

	login_name	CantidadConexiones
1	sa	2
2	HP123\USUARIO	1

## JUSTIFICACIÓN TÉCNICA

### VISTAS DINÁMICAS DE MONITOREO:

sys.dm\_exec\_sessions:

- Session\_id: Identificador único de conexión
- Status: RUNNING, SLEEPING, SUSPENDED
- Wait\_type: Qué espera (PAGEIOLATCH, LOCK, CPU, etc.)
- Memory\_usage: KB de memoria asignada
- Logical\_reads: Lecturas de página en caché
- Reads: Lecturas físicas del disco

sys.dm\_exec\_requests:

- Procesos ACTUALMENTE EJECUTÁNDOSE
- Blocking\_session\_id: Sesión que causa bloqueo
- Wait\_time: Tiempo esperando en ms
- Status: RUNNING, SLEEPING, SUSPENDED

### MÉTRICAS CRÍTICAS:

- Wait\_type = PAGEIOLATCH\_SH: I/O disk lento
- Wait\_type = LCK\_M\_IX: Bloqueos de índice
- Logical\_reads alto: Consultas ineficientes

- CPU high: Falta de índices

## BUENAS PRÁCTICAS

### MONITOREO RUTINARIO:

#### 1. REVISAR DIARIAMENTE:

- Sesiones largas (>1 hora)
- Bloqueos activos (blocking\_session\_id > 0)
- Crecimiento del LOG
- Espacio en disco

#### 2. ALERTAS:

- Si logical\_reads > 1M en una consulta
- Si wait\_time > 5 minutos
- Si database\_size > 80% del disco
- Si más de 100 conexiones simultáneas

#### 3. HERRAMIENTAS:

- SQL Server Management Studio: Activity Monitor
- Extended Events para análisis profundo
- Trace logs para auditoría

#### 7. INTRODUCCIÓN AL USO DE SQL SERVER AGENT (TAREAS AUTOMÁTICAS)

Crear y programar tareas automáticas con SQL Server Agent e implementar backup automático diario de QhatuPERU.

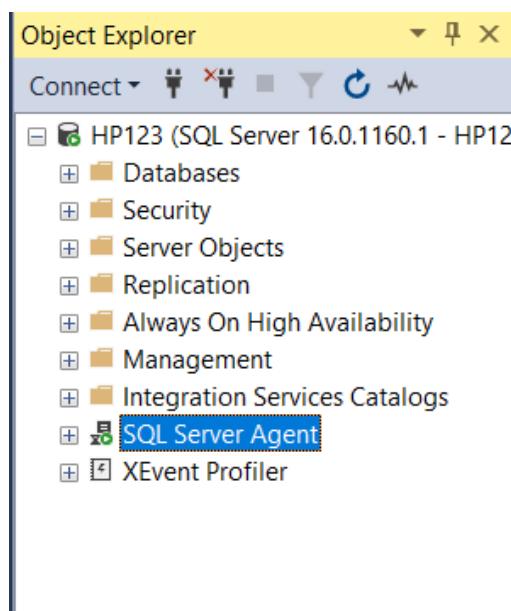
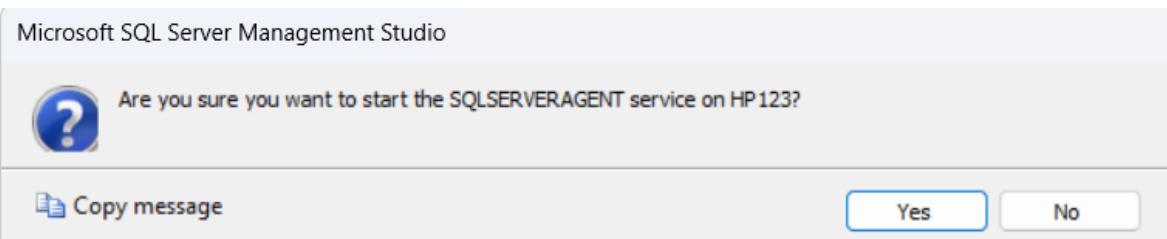
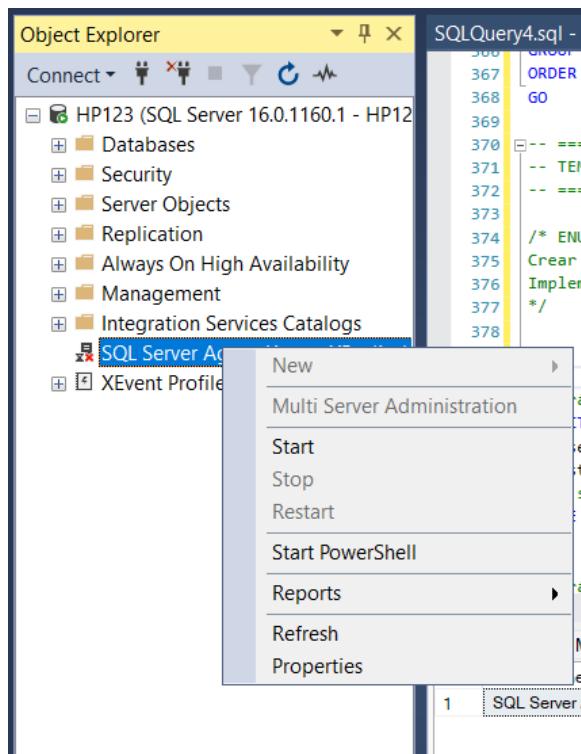
### SCRIPT T-SQL

The screenshot shows a SQL Server Management Studio (SSMS) window. The query pane contains the following T-SQL script:

```
381 -- Práctica 7.1: Verificar que SQL Server Agent esté activo
382 SELECT
383     servicename,
384     status_desc
385 FROM sys.dm_server_services
386 WHERE servicename = 'SQL Server Agent (MSSQLSERVER)';
387 GO
388
```

The status bar at the bottom left indicates "81 %". Below the query pane are two tabs: "Results" and "Messages". The "Results" tab displays the following table:

	servicename	status_desc
1	SQL Server Agent (MSSQLSERVER)	Stopped



```
381 -- Práctica 7.1: Verificar que SQL Server Agent esté activo
382 SELECT
383     servicename,
384     status_desc
385 FROM sys.dm_server_services
386 WHERE servicename = 'SQL Server Agent (MSSQLSERVER)';
387 GO
388
```

31 %

Results Messages

	servicename	status_desc
1	SQL Server Agent (MSSQLSERVER)	Running

```
389 -- Práctica 7.2: Crear Job de Backup Completo
390 USE msdb;
391 GO
392
393 EXEC sp_add_job
394     @job_name = 'Job_Backup_QhatuPERU_FULL',
395     @enabled = 1,
396     @description = 'Backup completo diario de QhatuPERU a las 22:00',
397     @notify_level_eventlog = 2;
398 GO
399
```

31 %

Messages

Commands completed successfully.

Completion time: 2025-12-10T01:29:59.4777773-05:00

```
299  
400 -- Práctica 7.3: Agregar paso del job  
401 EXEC sp_add_jobstep  
402     @job_name = 'Job_Backup_QhatuPERU_FULL',  
403     @step_name = 'Backup_Database',  
404     @subsystem = 'TSQL',  
405     @command = N'  
        BACKUP DATABASE QhatuPERU  
        TO DISK = N''C:\Backups\QhatuPERU\QhatuPERU_FULL_'' +  
        CONVERT(VARCHAR(8), GETDATE(), 112) + '''.bak''  
        WITH INIT, COMPRESSION, STATS = 5, CHECKSUM;  
        ',  
411     @on_success_action = 1,  
412     @retry_attempts = 3,  
413     @retry_interval = 5;  
414 GO  
415
```

81 % ▾

Messages

Commands completed successfully.

Completion time: 2025-12-10T01:30:24.5954879-05:00

```
416 -- Práctica 7.4: Crear Job de Backup de Log  
417 EXEC sp_add_job  
418     @job_name = 'Job_Backup_QhatuPERU_LOG',  
419     @enabled = 1,  
420     @description = 'Backup de log cada 4 horas',  
421     @notify_level_eventlog = 2;  
422 GO  
423  
424 EXEC sp_add_jobstep  
425     @job_name = 'Job_Backup_QhatuPERU_LOG',  
426     @step_name = 'Backup_Log',  
427     @subsystem = 'TSQL',  
428     @command = N'  
        BACKUP LOG QhatuPERU  
        TO DISK = N''C:\Backups\QhatuPERU\QhatuPERU_LOG_'' +  
        CONVERT(VARCHAR(8), GETDATE(), 112) + ''_'' +  
        CONVERT(VARCHAR(6), GETDATE(), 108) + '''.trn''  
        WITH INIT, COMPRESSION;  
        ',  
435     @on_success_action = 1;  
436 GO  
437
```

81 % ▾

Messages

Commands completed successfully.

Completion time: 2025-12-10T01:30:49.2069331-05:00

```

431
438 -- Práctica 7.5: Crear Schedule para FULL (22:00, lunes-viernes)
439 EXEC sp_add_schedule
440     @schedule_name = 'Schedule_FULL_Diario_22h',
441     @freq_type = 8,          -- Semanal
442     @freq_interval = 62,      -- Lun(2)+Mar(4)+Mie(8)+Jue(16)+Vie(32) = 62
443     @freq_recurrence_factor = 1,    -- Cada 1 semana
444     @active_start_time = 220000,    -- 22:00:00
445     @active_end_time = 235959,
446     @active_start_date = 20250101,
447     @active_end_date = 20991231;
448 GO
449

81 % ▾ Messages
Commands completed successfully.

Completion time: 2025-12-10T01:32:11.7990515-05:00

450
451 -- Práctica 7.6: Crear Schedule para LOG (cada 4 horas)
452 EXEC sp_add_schedule
453     @schedule_name = 'Schedule_LOG_Cada4h',
454     @freq_type = 4,          -- Diaria
455     @freq_interval = 1,      -- Cada día
456     @freq_subday_type = 8,    -- Cada N horas
457     @freq_subday_interval = 4,  -- Cada 4 horas
458     @active_start_time = 000000,  -- Desde las 00:00
459     @active_end_time = 235959,   -- Hasta 23:59
460     @active_start_date = 20250101,
461     @active_end_date = 20991231;
462 GO
463

81 % ▾ Messages
Commands completed successfully.

Completion time: 2025-12-10T01:33:28.4516095-05:00

```

## JUSTIFICACIÓN TÉCNICA

El respaldo de bases de datos en SQL Server se basa en un modelo de recuperación que define cómo se registran las transacciones. En este caso, QhatuPERU utiliza el modelo de recuperación completa (FULL), lo que permite recuperar la base hasta el último punto de fallo siempre que existan copias del log de transacciones.

Por ello, la estrategia aplicada incluye:

Tipo de backup	Frecuencia	Justificación técnica

<b>FULL</b>	<b>Diario a las 22:00</b>	<b>Copia completa del estado de la BD, momento de baja carga (menor impacto en rendimiento).</b>
<b>LOG</b>	<b>Cada 4 horas</b>	<b>Evita crecimiento descontrolado del log y permite recuperar hasta el último punto posible.</b>

**Se programan mediante SQL Server Agent, lo que reduce intervención manual, facilita auditorías, y permite configuración granular de alertas, notificaciones y reintentos.**

## **BUENAS PRÁCTICAS**

✓ Compresión de backups (WITH COMPRESSION)

Reduce el tamaño hasta 70% y disminuye I/O de disco.

✓ CHECKSUM en el Backup FULL

Brinda detección temprana de corrupción de páginas en disco:

WITH CHECKSUM

✓ Nombrado dinámico con fecha y hora

Permite rastrear fácilmente versiones históricas:

CONVERT(VARCHAR(8), GETDATE(), 112) -- AAAAMMDD

CONVERT(VARCHAR(6), GETDATE(), 108) -- HHMMSS

✓ Programación fuera del horario laboral (22:00)

Minimiza impacto en el rendimiento durante horas productivas.

✓ Backup de log cada 4 horas

Evita saturación del archivo .ldf y protege transacciones recientes.

✓ Reintentos y tolerancia a fallos

Se configuró:

@retry\_attempts = 3, @retry\_interval = 5

✓ Separación de FULL y LOG

Facilita recuperación granular con cadena de logs consistente.