

## FORMATO DE ESTADARIZACION PARA CODIGO SQL

### Estilo de escritura:

- **S1. Palabras reservadas:**  
Todas las palabras reservadas de SQL deben ir en MAYÚSCULAS.  
Ejemplo: SELECT, FROM, WHERE, ORDER BY, BEGIN TRY, END TRY, RETURN, etc.
- **S2. Objetos de base de datos:**  
Siempre se debe prefijar el nombre de los objetos (tablas, SP, vistas, etc.) con el nombre del esquema: dbo..  
Ejemplo: dbo.Empleado, dbo.TipoMovimiento, dbo.InsertarEmpleado.
- **S3. Nombres de tablas:**  
Los nombres de tablas deben ser en singular.  
Ejemplo correcto: Empleado, Movimiento.  
Ejemplo incorrecto: Empleados, Movimientos.
- **S4. Nombres de columnas (atributos):**  
Usar camelCase (primera letra en mayúscula, resto minúscula).  
Ejemplo: Nombre, ValorDocumentoidentidad, SaldoVacaciones.

### Estructura de llaves primarias y foráneas:

- **S5. Llaves primarias (PK):**  
Toda tabla debe tener una columna llamada Id como PK, tipo INT IDENTITY(1,1).
- **S6. Llaves foráneas (FK):**  
El nombre de la FK debe ser id<NombreTablaReferencia>.  
Ejemplo: en Empleado, la FK sería idPuesto.

### Reglas para los SPs:

- **S7. Bloques BEGIN...END:**  
Todo SP debe tener **BEGIN** y **END** después del AS. Siempre.
- **S8. SET NOCOUNT ON / OFF:**  
Inmediatamente después de BEGIN, se debe poner SET NOCOUNT ON;.  
Antes del END, se debe poner SET NOCOUNT OFF;.  
Esto evita que el servidor mande mensajes de filas afectadas.
- **S9. Punto y coma (;):**  
Todas las sentencias SQL deben terminar en ;.  
Incluso SELECT, INSERT, UPDATE, DELETE, SET, BEGIN TRANSACTION, etc.
- **S10. Prefijo de parámetros:**  
Parámetros de entrada: comienzan con @in.

Parámetros de salida: comienzan con @out.

- **S11. Parámetro de resultado obligatorio:**  
Todo SP debe tener un parámetro de salida llamado @outResultCode.  
0 si el SP ejecutó bien.  
Código mayor a 50000 si hubo error lógico.  
Si ocurre un error de SQL Server, se captura el ERROR\_NUMBER().

#### Buenas prácticas de formato:

- **S12. No sentencias largas:**  
No esconder código hacia la derecha.  
Separar en varias líneas los SELECT, FROM, JOIN, WHERE.\
- **S13. Espacios en operadores:**  
Los operadores deben estar separados por espacios.  
Ejemplo correcto: WHERE Id = @inIdEmpleado.
- **S14. INSERTs ordenados:**  
Listar las columnas y los valores uno por línea, separados por comas.  
Ejemplo:

```
INSERT INTO dbo.Empleado (  
    IdPuesto  
    , ValorDocumentoIdentidad  
    , Nombre  
)  
VALUES (  
    @inIdPuesto  
    , @inValorDocumentoIdentidad  
    , @inNombre  
);
```

- **S15. INSERT completo:**  
Siempre especificar todas las columnas que no tengan valores por DEFAULT

#### Reglas para consultas:

- **S16. No usar SELECT \*:**  
Siempre indicar explícitamente qué columnas quieres traer con alias.
- **S17. Alias en consultas:**  
Siempre utilizar alias de tabla para columnas (e.Nombre, m.Monto).

#### Manejo de errores:

- **S18. Manejar errores siempre:**  
Usar bloques BEGIN TRY...END TRY y BEGIN CATCH...END CATCH.

Insertar el error en dbo.DBError dentro del CATCH.

Aunque el SP solo haga SELECT, se debe capturar errores.

- **S19. Manejo de transacciones:**

Si el SP afecta **más de una tabla**, se debe hacer:

BEGIN TRANSACTION

COMMIT TRANSACTION

(En caso de error, opcionalmente hacer ROLLBACK).