

```

-- Procedimiento para agregar empleados
CREATE PROCEDURE sp_InsertarEmpleado
    @Nombre NVARCHAR(100),
    @Apellido NVARCHAR(100),
    @Cargo NVARCHAR(100),
    @Salario DECIMAL(10,2),
    @Rendimiento NVARCHAR(50)
AS
BEGIN
    BEGIN TRY
        INSERT INTO Empleados (Nombre, Apellido, Cargo, Salario, Rendimiento)
        VALUES (@Nombre, @Apellido, @Cargo, @Salario, @Rendimiento);

        INSERT INTO Logs (Procedimiento, Mensaje, Error)
        VALUES ('sp_InsertarEmpleado', CONCAT('Empleado insertado: ', @Nombre, ' ', @Apellido), 0);
    END TRY
    BEGIN CATCH
        INSERT INTO Logs (Procedimiento, Mensaje, Error)
        VALUES ('sp_InsertarEmpleado', ERROR_MESSAGE(), 1);
    END CATCH
END;
GO

EXEC sp_InsertarEmpleado N'Carlos', N'Ramírez', N'Analista', 3500.00, N'Bajo';
EXEC sp_InsertarEmpleado N'María', N'González', N'SubGerente', 4000.00, N'Alto';
EXEC sp_InsertarEmpleado N'Luis', N'Torres', N'Gerente', 7500.00, N'Bueno';

SELECT * FROM Empleados;

```

	EmpleadoID	Nombre	Apellido	Cargo	Salario	Rendimiento	FechaIngreso
1	1	Carlos	Ramírez	Analista	3500.00	Bajo	2025-05-13
2	2	María	González	SubGerente	4000.00	Alto	2025-05-13
3	3	Luis	Torres	Gerente	7500.00	Bueno	2025-05-13

```

-- Crear un sp (un trigger más bien) para notificar al administrador si el rendimiento
CREATE TRIGGER tr_NotificarRendimientoEmpleado
ON Empleados
AFTER INSERT, UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO Logs (Procedimiento, Mensaje, Error)
    SELECT
        'tr_NotificarRendimientoEmpleado',
        CONCAT('Empleado ID ', i.EmpleadoID, ': Advertencia por bajo rendimiento'),
        0
    FROM inserted i
    WHERE i.Rendimiento = 'Bajo';

END;
GO

```

	LogID	Fecha	Procedimiento	Mensaje	Error
1	1	2025-05-13 16:24:02.003	sp_InsertarEmpleado	Empleado insertado: Carlos Ramírez	0
2	2	2025-05-13 16:24:02.003	sp_InsertarEmpleado	Empleado insertado: María González	0
3	3	2025-05-13 16:24:02.003	sp_InsertarEmpleado	Empleado insertado: Luis Torres	0
4	4	2025-05-13 16:24:02.120	tr_NotificarRendimientoEmpleado	Empleado ID 4: Advertencia por bajo rendimiento	0
5	5	2025-05-13 16:24:02.120	sp_InsertarEmpleado	Empleado insertado: Javier Martínez	0

-- Procedimiento para agregar productos

```
CREATE PROCEDURE sp_InsertarProducto
    @Nombre NVARCHAR(100),
    @Tipo NVARCHAR(50),
    @Costo DECIMAL(10,2),
    @Stock INT
AS
BEGIN
    BEGIN TRY
        INSERT INTO Productos (Nombre, Tipo, Costo, Stock)
        VALUES (@Nombre, @Tipo, @Costo, @Stock);

        INSERT INTO Logs (Procedimiento, Mensaje, Error)
        VALUES ('sp_InsertarProducto', CONCAT('Producto insertado: ', @Nombre), 0);
    END TRY
    BEGIN CATCH
        INSERT INTO Logs (Procedimiento, Mensaje, Error)
        VALUES ('sp_InsertarProducto', ERROR_MESSAGE(), 1);
    END CATCH
END;
```

```
EXEC sp_InsertarProducto 'Cemento', 'Importado', 5.50, 50;
EXEC sp_InsertarProducto 'Arena', 'Nacional', 3.25, 25;
EXEC sp_InsertarProducto 'Ladrillo', 'Importado', 0.45, 35;
```

	ProductoID	Nombre	Tipo	Costo	Stock	FechaIngreso
1	1	Cemento	Importado	5.50	50	2025-05-13
2	2	Arena	Nacional	3.25	25	2025-05-13
3	3	Ladrillo	Importado	0.45	35	2025-05-13

```

3 CREATE TRIGGER tr_NotificarStockProducto
ON Productos
AFTER INSERT, UPDATE, DELETE
AS
3 BEGIN

    SET NOCOUNT ON;

3 BEGIN TRY
    INSERT INTO Logs (Procedimiento, Mensaje, Error)
    SELECT
        'tr_NotificarStockProducto',
        CONCAT('Advertencia: Producto ', i.Nombre, ' con stock bajo (', i.Stock, ' unidades)'),
        0
    FROM inserted i
    WHERE i.Stock <= 10;
- END TRY
BEGIN CATCH
3 INSERT INTO Logs (Procedimiento, Mensaje, Error)
VALUES ('tr_NotificarStockProducto', ERROR_MESSAGE(), 1);
- END CATCH

- END;
- GO

3 EXEC sp_InsertarProducto N'Cortadora de sesped', 'Importado', 1750.00, 5;
SELECT * FROM Logs;
- GO

```

	LogID	Fecha	Procedimiento	Mensaje	Error
3	3	2025-05-13 16:24:02.003	sp_InsertarEmpleado	Empleado insertado: Luis Torres	0
4	4	2025-05-13 16:24:02.120	tr_NotificarRendimientoEmpleado	Empleado ID 4: Advertencia por bajo rendimiento	0
5	5	2025-05-13 16:24:02.120	sp_InsertarEmpleado	Empleado insertado: Javier Martínez	0
6	6	2025-05-13 16:24:02.177	sp_InsertarProducto	Producto insertado: Cemento	0
7	7	2025-05-13 16:24:02.177	sp_InsertarProducto	Producto insertado: Arena	0
8	8	2025-05-13 16:24:02.177	sp_InsertarProducto	Producto insertado: Ladrillo	0
9	9	2025-05-13 16:24:02.230	tr_NotificarStockProducto	Advertencia: Producto Cortadora de sesped con stock bajo (5 unidades)	0
10	10	2025-05-13 16:24:02.233	sp_InsertarProducto	Producto insertado: Cortadora de sesped	0

```

-- Cursor para actualizar salarios por cargo
DECLARE @EmpleadoID INT

DECLARE cursor_salario CURSOR STATIC FOR
    SELECT EmpleadoID FROM Empleados;

OPEN cursor_salario

FETCH NEXT FROM cursor_salario INTO @EmpleadoID
WHILE @@FETCH_STATUS = 0
BEGIN
    EXEC sp_AumentarSalarioEmpleado @EmpleadoID
    FETCH NEXT FROM cursor_salario INTO @EmpleadoID
END

CLOSE cursor_salario
DEALLOCATE cursor_salario

```

	EmpleadoID	Nombre	Apellido	Cargo	Salario	Rendimiento	FechaIngresc
1	1	Carlos	Ramírez	Analista	4025.00	Bajo	2025-05-13
2	2	María	González	SubGerente	4800.00	Alto	2025-05-13
3	3	Luis	Torres	Gerente	9750.00	Bueno	2025-05-13
4	4	Javier	Martínez	Analista	4025.00	Bajo	2025-05-13

```

-- Cursor para actualizar salarios por rendimiento
] DECLARE cursor_salario_rendimiento CURSOR STATIC FOR
    SELECT EmpleadoID FROM Empleados;

OPEN cursor_salario_rendimiento

FETCH NEXT FROM cursor_salario_rendimiento INTO @EmpleadoID
] WHILE @@FETCH_STATUS = 0
] BEGIN
    EXEC sp_AumentarSalarioRendimiento @EmpleadoID
    FETCH NEXT FROM cursor_salario_rendimiento INTO @EmpleadoID
END

CLOSE cursor_salario_rendimiento
DEALLOCATE cursor_salario_rendimiento

```

	EmpleadoID	Nombre	Apellido	Cargo	Salario	Rendimiento	FechaIngreso
1	1	Carlos	Ramírez	Analista	4025.00	Bajo	2025-05-13
2	2	María	González	SubGerente	5136.00	Alto	2025-05-13
3	3	Luis	Torres	Gerente	9945.00	Bueno	2025-05-13
4	4	Javier	Martínez	Analista	4025.00	Bajo	2025-05-13

```
-- Cursor para aplicar aumento a productos importados y nacionales
DECLARE @TipoProducto NVARCHAR(50)

DECLARE cursor_productos CURSOR STATIC FOR
    SELECT DISTINCT Tipo FROM Productos WHERE LOWER(Tipo) IN ('importado', 'nacional');

OPEN cursor_productos

FETCH NEXT FROM cursor_productos INTO @TipoProducto
WHILE @@FETCH_STATUS = 0
BEGIN
    EXEC sp_AumentoPrecioTipoProducto @TipoProducto
    FETCH NEXT FROM cursor_productos INTO @TipoProducto
END

CLOSE cursor_productos
DEALLOCATE cursor_productos
```

	ProductoID	Nombre	Tipo	Costo	Stock	FechaIngreso
1	1	Cemento	Importado	7.92	50	2025-05-13
2	2	Arena	Nacional	3.64	25	2025-05-13
3	3	Ladrillo	Importado	0.65	35	2025-05-13
4	4	Cortad...	Importado	252...	5	2025-05-13