

# Procedimientos almacenados

---

Un procedimiento almacenado es una porcion de codigo que se puede almacenar y reutilizar, es util para hacer inserciones o actualizaciones a tablas que se suelen repetir, este es un metodo para encapsular codigo siguiendo las propiedades **ACID**.

- Atomicidad:
  - Todo **procedimiento almacenado** debe tratar las operaciones como única unidad de trabajo. Si una operacion falla, todas las demás operaciones dentro de la transaccion deben deshacerse.

```
-- Ejemplo de una transacción
START TRANSACTION;
-- SENTENCIAS SQL

-- CASO DE EXITO
COMMIT;
-- CASO DE FRACASO
ROLLBACK;
```

- Consistencia:
  - Al final del procedimiento almacenado debe de estar en un estado consistente, cumplir con las llaves foraneas, unicidad, etc.
- Aislamiento (Isolation):
  - Durante la ejecucion del procedimiento, los cambios que realizaron no deben ser visibles para otras transacciones hasta que se complete.
- Durabilidad:
  - Una vez que la transaccion se ha comprometido ( `COMMIT;` ), los cambios deben ser persistidos de manera que sobrevivan incluso a fallos del sistema.
  - Esto depende de gran medida del **SGBD** y la configuracion que se le de a este.

## Ejemplo de un procedimiento almacenado.

---

```

-- Cambiamos la estructura de la sentencia para que
-- finalice el procedimiento no con ; si no con //
DELIMITER //

-- Define el procedimiento almacenado con 4 parametros
-- los parametros son de entrada (IN)
CREATE PROCEDURE ActualizarPuntos(
    IN p_idTarjeta INT,
    IN p_puntos INT,
    IN p_idTransaccion INT,
    IN p_montoCompra DECIMAL(10,2)
)
-- Iniciamos el procedimiento almacenado
BEGIN
    -- Definimos el bloque en caso de errores sql
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        -- Rollback en caso de error
        ROLLBACK;
    END;

    -- Iniciar la transacción
    START TRANSACTION;

    -- Actualizar puntos en la tabla tarjeta_puntos
    UPDATE tarjeta_puntos
    SET puntos_acumulados = puntos_acumulados + p_puntos,
        fecha_ultima_actualizacion = NOW()
    WHERE id_tarjeta = p_idTarjeta;

    -- Registrar la transacción
    INSERT INTO transacciones (id_tarjeta, monto_compra, puntos_obtenidos,
fecha_transaccion)
    VALUES (p_idTarjeta, p_montoCompra, p_puntos, NOW());

    -- Confirmar los cambios
    COMMIT;
END //

DELIMITER ;

```