Anonimización de datos en Oracle ADB (RAE_CMDB)

Equipo Malackathon

15 de octubre de 2025

Resumen

El objetivo es proteger datos personales en la tabla `DIEGO.RAE_CMDB` usando:

- **Hash determinístico** SHA-256 con *pepper* para identificadores (`CIP_SNS_RECODIFICADO`, `CENTRO_RECODIFICADO`).
- **Seudonimización** estable de nombres (`NOMBRE` → `PACIENTE-XXXXXX`).
- **Trigger** que anonimiza en `INSERT/UPDATE` y **procedimiento de backfill** para registros existentes.
- **Checks** de formato e **vistas** "privadas" para consumo seguro.

Flujo de alto nivel

1. **Pepper** secreto guardado en `DIEGO.ANON_SECRETS`.
2. Paquete `DIEGO.ANON`: funciones utilitarias
   - `norm()` normaliza texto (trim + espacios + mayúsculas).
   - `sha256_hex()` calcula SHA-256 (vía `STANDARD_HASH` o `DBMS_CRYPTO`).
   - `token_cip()`, `token_centro()` ⇒ hash determinístico (*pepper* + valor normalizado).
   - `seudonimo_nombre()` ⇒ busca/crea alias en `DIEGO.PSEUDONIMO`.
3. **Trigger** `RAE_CMDB_ANON_BIU` aplica anonimización al escribir.
4. **Checks** validan que los hashes tengan formato hex de 64 caracteres.
5. **Backfill** (`DIEGO.ANON_BACKFILL`) anonimiza registros antiguos in–place (commits por lotes).
6. **Vistas**:
   - `V_RAE_CMDB_PRIV`: muestra el seudónimo, nunca el nombre real ni los hashes crudos.
   - `V_RAE_CMDB_AUD`: "hash tails" (últimos 6) para auditoría.

Campos afectados

CIP_SNS_RECODIFICADO
: Sustituido por SHA-256 (hex64) de *pepper + CIP normalizado*.

CENTRO_RECODIFICADO
: Sustituido por SHA-256 (hex64) de *pepper + ID centro*.

NOMBRE
: Sustituido por seudónimo estable `PACIENTE-000001`, gestionado en `DIEGO.PSEUDONIMO`.

Notas de seguridad y operación

- El *pepper* se guarda en tabla propia; no se expone en vistas ni código de aplicación.
- **Determinístico:** el mismo dato → mismo hash/seudónimo, útil para deduplicar sin revelar el original.
- Rotar el *pepper* implicaría recalcular hashes (política a definir); la tabla de seudónimos mantiene estabilidad de nombres.
- Usar siempre `V_RAE_CMDB_PRIV` o `V_RAE_CMDB_AUD` para analítica y *dashboards*.

SQL implementado

A continuación, el script completo (concediendo permisos a `ADMIN` y todos los objetos de anonimización).

```
GRANT SELECT, INSERT, UPDATE, DELETE ON diego.rae_cmdb TO admin;

-------------------------------------------------------------------------------
-- 0) PARAMETRIZACIN (pepper para el hash)
-------------------------------------------------------------------------------
CREATE TABLE DIEGO.ANON_SECRETS (
  PEPPER VARCHAR2(128) NOT NULL
);
BEGIN
  INSERT INTO DIEGO.ANON_SECRETS(PEPPER)
  SELECT DBMS_RANDOM.STRING('x', 32) FROM DUAL
  WHERE NOT EXISTS (SELECT 1 FROM DIEGO.ANON_SECRETS);
  COMMIT;
END;
/

-------------------------------------------------------------------------------
-- 1) TABLA DE PSEUDNIMOS
-------------------------------------------------------------------------------
CREATE TABLE DIEGO.PSEUDONIMO (
  SOURCE_SHA256 VARCHAR2(64) PRIMARY KEY,
  PSEUDONIMO VARCHAR2(50) UNIQUE NOT NULL,
  CREATED_AT DATE DEFAULT SYSDATE
);

CREATE SEQUENCE DIEGO.PSEUDONIMO_S START WITH 1 NOCACHE;

COMMENT ON TABLE DIEGO.PSEUDONIMO IS 'Mapa determinstico nombre->seudnimo';
COMMENT ON COLUMN DIEGO.PSEUDONIMO.SOURCE_SHA256 IS 'Hash SHA-256 (pepper+nombre
    normalizado)';
COMMENT ON COLUMN DIEGO.PSEUDONIMO.PSEUDONIMO IS 'Alias estable: Paciente-000001';

-------------------------------------------------------------------------------
-- 2) PAQUETE: funciones utilitarias de anonimizacin
-------------------------------------------------------------------------------
CREATE OR REPLACE PACKAGE BODY DIEGO.ANON IS
  FUNCTION norm(v VARCHAR2) RETURN VARCHAR2 IS
  BEGIN
    IF v IS NULL THEN RETURN NULL; END IF;
    RETURN UPPER(REGEXP_REPLACE(TRIM(v), '\s+', ' '));
  END;

  FUNCTION pepper RETURN VARCHAR2 IS
```

```
  p VARCHAR2(128);
BEGIN
  SELECT PEPPER INTO p FROM DIEGO.ANON_SECRETS FETCH FIRST 1 ROWS ONLY;
  RETURN p;
END;

-- SHA-256 en HEX, robusto a versiones/permisos
FUNCTION sha256_hex(v VARCHAR2) RETURN VARCHAR2 IS
  r VARCHAR2(64);
BEGIN
  IF v IS NULL THEN RETURN NULL; END IF;
  BEGIN
    EXECUTE IMMEDIATE
      q'[SELECT STANDARD_HASH(:b1,'SHA256') FROM DUAL]'
      INTO r USING v;
    RETURN r;
  EXCEPTION
    WHEN OTHERS THEN
      RETURN LOWER(
        RAWTOHEX(
          DBMS_CRYPTO.HASH(
            UTL_I18N.STRING_TO_RAW(v, 'AL32UTF8'),
            DBMS_CRYPTO.HASH_SH256
          )
        )
      );
  END;
END;

FUNCTION token_cip(v VARCHAR2) RETURN VARCHAR2 IS
BEGIN
  IF v IS NULL THEN RETURN NULL; END IF;
  IF REGEXP_LIKE(v, '^[0-9A-F]{64}$') THEN RETURN v; END IF;
  RETURN sha256_hex(pepper() || norm(v));
END;

FUNCTION token_centro(v VARCHAR2) RETURN VARCHAR2 IS
BEGIN
  IF v IS NULL THEN RETURN NULL; END IF;
  IF REGEXP_LIKE(v, '^[0-9A-F]{64}$') THEN RETURN v; END IF;
  RETURN sha256_hex(pepper() || norm(v));
END;

FUNCTION seudonimo_nombre(v VARCHAR2) RETURN VARCHAR2 IS
  k VARCHAR2(64);
  pse DIEGO.PSEUDONIMO.PSEUDONIMO%TYPE;
BEGIN
  IF v IS NULL THEN RETURN NULL; END IF;
  k := sha256_hex(pepper() || norm(v));
  BEGIN
    SELECT PSEUDONIMO INTO pse FROM DIEGO.PSEUDONIMO WHERE SOURCE_SHA256 = k;
    RETURN pse;
  EXCEPTION WHEN NO_DATA_FOUND THEN
    pse := 'PACIENTE-' || LPAD(DIEGO.PSEUDONIMO_S.NEXTVAL, 6, '0');
    INSERT INTO DIEGO.PSEUDONIMO(SOURCE_SHA256, PSEUDONIMO) VALUES (k, pse);
    RETURN pse;
  END;
END;
```

```
END ANON;
/

-------------------------------------------------------------------------------
-- 3) TRIGGER: anonimiza en INSERT/UPDATE
-------------------------------------------------------------------------------
CREATE OR REPLACE TRIGGER DIEGO.RAE_CMDB_ANON_BIU
BEFORE INSERT OR UPDATE ON DIEGO.RAE_CMDB
FOR EACH ROW
BEGIN
  :NEW.CIP_SNS_RECODIFICADO := DIEGO.ANON.token_cip(:NEW.CIP_SNS_RECODIFICADO);
  :NEW.CENTRO_RECODIFICADO := DIEGO.ANON.token_centro(:NEW.CENTRO_RECODIFICADO);

  IF :NEW.NOMBRE IS NOT NULL THEN
    :NEW.NOMBRE := DIEGO.ANON.seudonimo_nombre(:NEW.NOMBRE);
  END IF;
END;
/

-------------------------------------------------------------------------------
-- 4) CHECKS DE FORMATO
-------------------------------------------------------------------------------
ALTER TABLE DIEGO.RAE_CMDB ADD CONSTRAINT CK_RAE_CIP_HASH
  CHECK (CIP_SNS_RECODIFICADO IS NULL OR REGEXP_LIKE(CIP_SNS_RECODIFICADO, '^[0-9A-F
      ]{64}$'))
  ENABLE NOVALIDATE;

ALTER TABLE DIEGO.RAE_CMDB ADD CONSTRAINT CK_RAE_CENTRO_HASH
  CHECK (CENTRO_RECODIFICADO IS NULL OR REGEXP_LIKE(CENTRO_RECODIFICADO, '^[0-9A-F
      ]{64}$'))
  ENABLE NOVALIDATE;

COMMENT ON COLUMN DIEGO.RAE_CMDB.CIP_SNS_RECODIFICADO IS 'Hash SHA-256 (pepper + CIP
    normalizado)';
COMMENT ON COLUMN DIEGO.RAE_CMDB.CENTRO_RECODIFICADO IS 'Hash SHA-256 (pepper + ID de
    centro normalizado)';
COMMENT ON COLUMN DIEGO.RAE_CMDB.NOMBRE IS 'Seudnimo estable (no almacena el nombre
    real)';

-------------------------------------------------------------------------------
-- 5) BACKFILL: anonimiza registros ya existentes
-------------------------------------------------------------------------------
CREATE OR REPLACE PROCEDURE DIEGO.ANON_BACKFILL AS
  CURSOR c IS
    SELECT ROWID rid,
           CIP_SNS_RECODIFICADO,
           CENTRO_RECODIFICADO,
           NOMBRE
      FROM DIEGO.RAE_CMDB
     WHERE (CIP_SNS_RECODIFICADO IS NOT NULL AND NOT REGEXP_LIKE(CIP_SNS_RECODIFICADO,'
         ^[0-9A-F]{64}$'))
        OR (CENTRO_RECODIFICADO IS NOT NULL AND NOT REGEXP_LIKE(CENTRO_RECODIFICADO,'
            ^[0-9A-F]{64}$'))
        OR (NOMBRE IS NOT NULL AND NOT REGEXP_LIKE(NOMBRE, '^PACIENTE-\d{6}$'));
  TYPE t_rowid IS TABLE OF ROWID INDEX BY PLS_INTEGER;
  t t_rowid;
  i PLS_INTEGER := 0;
BEGIN
```

```
  FOR r IN c LOOP
    UPDATE DIEGO.RAE_CMDB
       SET CIP_SNS_RECODIFICADO = DIEGO.ANON.token_cip(r.CIP_SNS_RECODIFICADO),
           CENTRO_RECODIFICADO = DIEGO.ANON.token_centro(r.CENTRO_RECODIFICADO),
           NOMBRE = DIEGO.ANON.seudonimo_nombre(r.NOMBRE)
     WHERE ROWID = r.rid;

    i := i + 1;
    IF MOD(i, 5000) = 0 THEN COMMIT; END IF;
  END LOOP;
  COMMIT;
END;
/
BEGIN DIEGO.ANON_BACKFILL; END;
/

-------------------------------------------------------------------------------
-- 6) VISTAS PRIVADAS
-------------------------------------------------------------------------------
CREATE OR REPLACE VIEW DIEGO.V_RAE_CMDB_PRIV AS
SELECT NUM_REGISTRO_ANUAL,
       NOMBRE AS PACIENTE_SEUDONIMO,
       CCAA, COMUNIDAD_AUTONOMA, CCAA_RESIDENCIA,
       PAIS_NACIMIENTO, PAIS_RESIDENCIA, SEXO, FECHA_NACIMIENTO,
       FECHA_INICIO_CONTACTO, FECHA_INGRESO, FECHA_FIN_CONTACTO, ESTANCIA_DIAS,
       CIRCUNSTANCIA_CONTACTO, TIPO_ALTA, REGIMEN_FINANCIACION, PROCEDENCIA,
       CONTINUIDAD_ASISTENCIAL, INGRESO_UCI, DIAS_UCI, SERVICIO,
       DIAGNOSTICO_PRINCIPAL, CATEGORIA,
       DIAGNOSTICO_2, DIAGNOSTICO_3, DIAGNOSTICO_4, DIAGNOSTICO_5,
       DIAGNOSTICO_6, DIAGNOSTICO_7, DIAGNOSTICO_8, DIAGNOSTICO_9, DIAGNOSTICO_10,
       DIAGNOSTICO_11, DIAGNOSTICO_12, DIAGNOSTICO_13, DIAGNOSTICO_14,
       DIAGNOSTICO_15, DIAGNOSTICO_16, DIAGNOSTICO_17, DIAGNOSTICO_18, DIAGNOSTICO_19,
           DIAGNOSTICO_20,
       POA_DIAG_PRINCIPAL, POA_DIAG_2, POA_DIAG_3, POA_DIAG_4, POA_DIAG_5,
       POA_DIAG_6, POA_DIAG_7, POA_DIAG_8, POA_DIAG_9, POA_DIAG_10,
       POA_DIAG_11, POA_DIAG_12, POA_DIAG_13, POA_DIAG_14, POA_DIAG_15,
       POA_DIAG_16, POA_DIAG_17, POA_DIAG_18, POA_DIAG_19, POA_DIAG_20,
       FECHA_INTERVENCION,
       PROCEDIMIENTO_1, PROCEDIMIENTO_2, PROCEDIMIENTO_3, PROCEDIMIENTO_4,
           PROCEDIMIENTO_5,
       PROCEDIMIENTO_6, PROCEDIMIENTO_7, PROCEDIMIENTO_8, PROCEDIMIENTO_9,
           PROCEDIMIENTO_10,
       PROCEDIMIENTO_11, PROCEDIMIENTO_12, PROCEDIMIENTO_13, PROCEDIMIENTO_14,
           PROCEDIMIENTO_15,
       PROCEDIMIENTO_16, PROCEDIMIENTO_17, PROCEDIMIENTO_18, PROCEDIMIENTO_19,
           PROCEDIMIENTO_20,
       PROCEDIMIENTO_EXTERNO_1, PROCEDIMIENTO_EXTERNO_2, PROCEDIMIENTO_EXTERNO_3,
       PROCEDIMIENTO_EXTERNO_4, PROCEDIMIENTO_EXTERNO_5, PROCEDIMIENTO_EXTERNO_6,
       GRD_APR, CDM_APR, TIPO_GRD_APR, PESO_ESPANOL_APR, VALOR_PESO_AMERICANO_APR,
       NIVEL_SEVERIDAD_APR, RIESGO_MORTALIDAD_APR, COSTE_APR,
       CIE, EDAD, EDAD_EN_INGRESO, REINGRESO, MES_INGRESO
  FROM DIEGO.RAE_CMDB;

CREATE OR REPLACE VIEW DIEGO.V_RAE_CMDB_AUD AS
SELECT NUM_REGISTRO_ANUAL,
       SUBSTR(CIP_SNS_RECODIFICADO, -6) AS CIP_TAIL,
       SUBSTR(CENTRO_RECODIFICADO, -6) AS CENTRO_TAIL,
       NOMBRE AS PACIENTE_SEUDONIMO
```

```
  FROM DIEGO.RAE_CMDB;

-------------------------------------------------------------------------------
-- 7) NDICES TILES
-------------------------------------------------------------------------------
CREATE INDEX RAE_CMDB_IX_CIP_HASH ON DIEGO.RAE_CMDB (CIP_SNS_RECODIFICADO);
CREATE INDEX RAE_CMDB_IX_CENTRO_HASH ON DIEGO.RAE_CMDB (CENTRO_RECODIFICADO);
```