

Datu Maskaratze Prozedura

Data Masking Procedures

Bertsioa: 1.0

Data: 2026ko urtarrilaren 23a

Erakunde: Zabala Gailetak S.L.

Kontrol ISO 27001: A.8.11 - Datu Maskaratzea

GDPR: Artikulu 32 - Tratamenduaren Segurtasuna

Egoera: Indarrean

1. Xedea eta Eremu

1.1 Xedea

Dokumentu honek datu sentikorra maskaratzeko prozedurak definitzen ditu:

- Datu maskaratzea:** Datu errealak ordezeko datu faltsuez ordezkatzea
- Pseudonimizazioa:** Datu pertsonalak identifikatzailearekin ordezkatzea
- ISO 27001:2022 kontrolen (A.8.11) betetze
- GDPR artikulua 32 - Datu segurtasuna

1.2 Eremu

Prozedura hau aplikatu behar da:

Ingurune	Maskaratzea Beharrezkoa?	Arrazoia
Produkzioa	✗ EZ	Datu errealak behar dira
Staging (Pre-produkzioa)	✓ BAI	Test errealistak, babes sentikorra
Garapena (Development)	✓ BAI	Garatzaileek ez dute datu erreala behar
Testa (QA/Testing)	✓ BAI	Test datuak, ez errealak
Prestakuntza	✓ BAI	Prestakuntza datuak, ez errealak
Demostrazio	✓ BAI	Demo datuak, ez errealak

2. Maskaratzeko Datu Motak

2.1 Datu Mota Sailkapena

Datu mota bakoitzak maskaratzeko metodoa behar du:

Datu Mota	GDPR PII?	Maskaratzea Beharrezkoa?	Metodoa
NIF/DNI	✔ BAI	✔ BAI	Ordezko baliozkoa sortu
Izena eta Abizena	✔ BAI	✔ BAI	Izen faltsu sortzailea
Eposta	✔ BAI	✔ BAI	Eposta faltsua sortu
Telefonia	✔ BAI	✔ BAI	Telefono baliozkoa sortu
Helbidea	✔ BAI	✔ BAI	Helbide faltsua sortu
Bankuko Kontua (IBAN)	✔ BAI	✔ BAI	IBAN baliozkoa sortu
Kreditu Txartela	✔ BAI	✔ BAI	Luhn algoritmo baliozkoa
Pasahitza (hash)	✔ BAI	✔ BAI	Hash berri bat sortu
IP Helbidea	⚠ BATZUETAN	⚠ BATZUETAN	IP helbide pribatua
Data Jaiotza	✔ BAI	⚠ PARTZIALKI	Urtea mantendu, eguna/hilabetea aldatu
Soldata	✔ BAI	✔ BAI	Ausazko balioa tarte batean
Osasun Datuak	✔ BAI (kategoria berezia)	✔ BAI	Datu faltsua sortu
Argazkia	✔ BAI	✔ BAI	Aurpegi blur edo generatutako argazkia

2.2 Maskaratzeko Metodoak

2.2.1 Ordezko Balioa (Substitution)

Datu erreala ordezko balio faltsuaz ordezkatu:

Datu Erreala: Jon Azpiazu

Datu Maskaratua: Mikel Etxeberria

Abantailak:

- Erraza inplementatzea
- Formatu bera mantentzen da
- Test errealistak

Desabantailak:

- Erlazionatutako datuak sinkronizatu behar dira
- Ordezko datuen zerrenda behar da

2.2.2 Zatitzea (Truncation)

Datu erreala zatitu edo moztu:

```
Datu Erreala: 12345678A (NIF)
Datu Maskaratua: ****5678A

Datu Erreala: jon.azpiazu@zabalagailetak.com
Datu Maskaratua: j***u@zabalagailetak.com
```

Abantailak:

- Erraza inplementatzea
- Formatua mantentzen da (partzialki)

Desabantailak:

- Ez da guztiz anonimo
- Test batzuetan ez da erabilgarri

2.2.3 Nahasketea (Shuffling)

Datuak nahasi errenkada desberdinetatik:

```
Erabiltzailea 1: Jon Azpiazu, Donostia, 600123456
Erabiltzailea 2: Maite Etxeberria, Bilbo, 600654321

Maskaratua:
Erabiltzailea 1: Jon Azpiazu, Bilbo, 600654321
Erabiltzailea 2: Maite Etxeberria, Donostia, 600123456
```

Abantailak:

- Datu errealak dira (database constraints-ak betetzen dira)
- Erraz inplementatu

Desabantailak:

- Erlazioak hausten ditu
- Ez da beti erabilgarri

2.2.4 Ausazkoak (Randomization)

Ausazko balioak sortu:

```
Soldata errealak: 35.000 €  
Soldata maskaratua: 28.456 € (ausazkoa 20.000-50.000 tartean)
```

Abantailak:

- Guztiz anonimo
- Estatistika mantentzen da

Desabantailak:

- Datu batzuetan ez da erabilgarri (NIF, IBAN baliozkoak behar dira)

2.2.5 Pseudonimizazioa (Pseudonymization)

Datu errealak identifikatzaile bakarrik eta itzulezinezkin ordezkatu:

```
Datu Errealak: Jon Azpiazu  
Datu Pseudonimizatua: USER_8a7b3c9d  
  
Mapaketa:  
USER_8a7b3c9d → Jon Azpiazu (datu-base seguruan gordetzen da)
```

Abantailak:

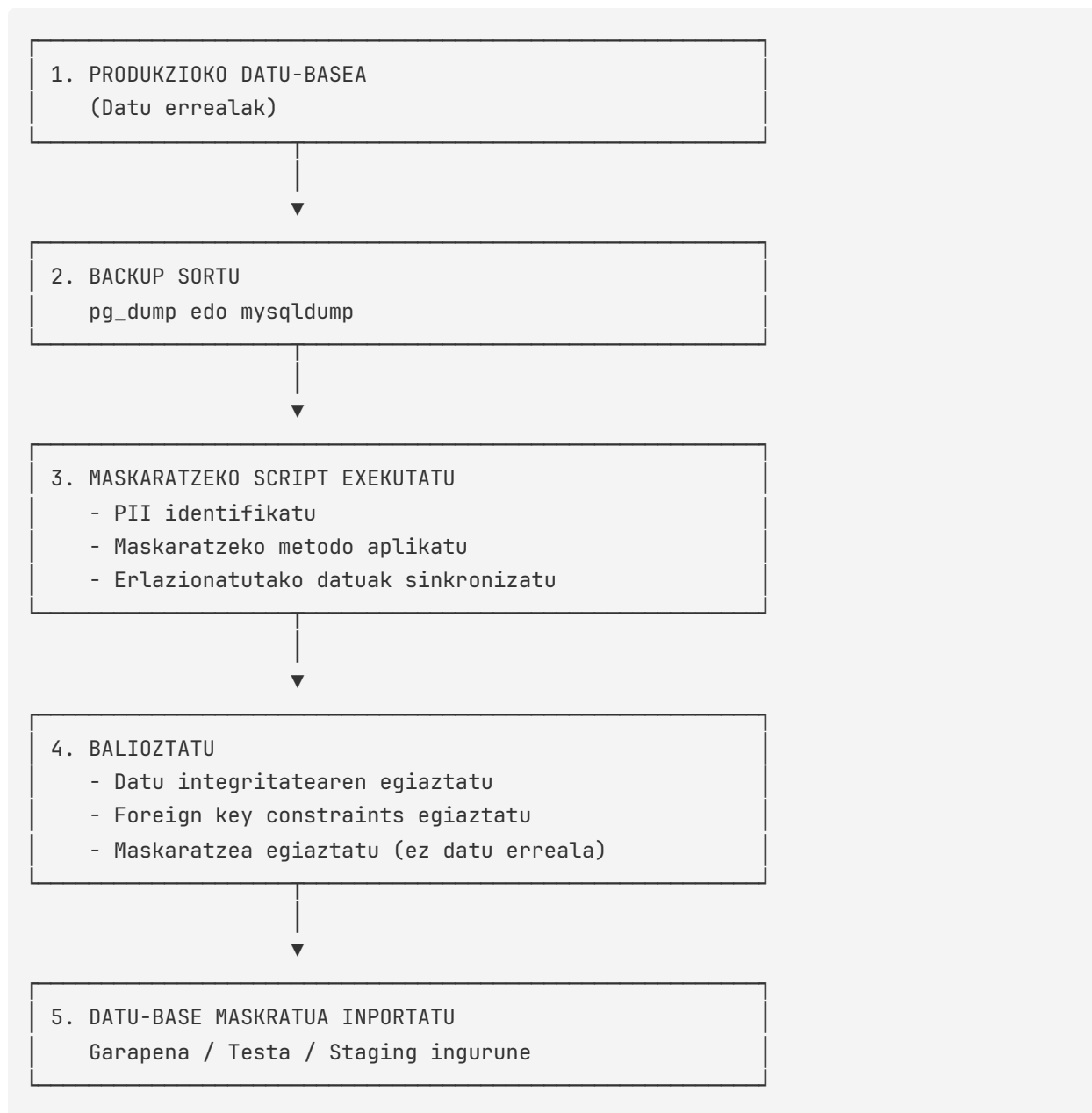
- GDPR-ren arabera onartua (Artikulu 4.5)
- Atzera itzul daiteke behar bada
- Erlazionatutako datuak mantentzen ditu

Desabantailak:

- Mapaketa taula behar da
- Mapaketa taula seguru mantendu behar da

3. Maskaratzeko Prozedura

3.1 Maskaratzeko Fluxua



3.2 Maskaratzeko Arauak Taula Maila

Taula bakoitzak maskaratzeko arauak behar ditu:

Adibidea: langileak Taula

Zutabea	Datu Mota	Maskaratzea?	Metodoa	Deskribapena
id	INTEGER	✗ EZ	-	Primary key, ez aldatu
izena	VARCHAR(100)	✓ BAI	Ordezko	Izen faltsu bat sortu
abizena	VARCHAR(100)	✓ BAI	Ordezko	Abizen faltsu bat sortu
nif	VARCHAR(9)	✓ BAI	NIF sortzailea	NIF baliozkoa sortu

Zutabea	Datu Mota	Maskaratzea?	Metodoa	Deskribapena
eposta	VARCHAR(255)	✅ BAI	Eposta sortzailea	eposta@test.zabalagailetak.com
telefonoa	VARCHAR(20)	✅ BAI	Telefono sortzailea	6XXXXXXXX baliozkoa
jaiotze_data	DATE	⚠️ PARTZIALKI	Urte mantendu	Urtea mantendu, eguna/hilabetea aldatu
soldata	DECIMAL(10,2)	✅ BAI	Ausazkoa	20000-60000 tartean
saila_id	INTEGER	❌ EZ	-	Foreign key, ez aldatu
kontratazio_data	DATE	❌ EZ	-	Negozio logika, ez aldatu
iban	VARCHAR(24)	✅ BAI	IBAN sortzailea	IBAN baliozkoa (ES)
osasun_datuak	TEXT	✅ BAI	NULL edo generiko	Osasun datu faltsua

4. Inplementazio Teknikoa

4.1 PHP Maskaratzeko Liburutegia

4.1.1 Klase Nagusia: DatuMaskaratzeZerbitzua

```

faker = Faker::create('es_ES');
}

/**
 * NIF maskaratu - NIF baliozkoa sortu
 *
 * @param string $nif Jatorrizko NIF
 * @return string NIF maskratua (baliozkoa)
 */
public function nif_maskaratu(string $nif): string
{
    // NIF baliozkoa sortu
    $zenbakia = rand(10000000, 99999999);
    $letra = $this->nif_letra_kalkulatu($zenbakia);

    return $zenbakia . $letra;
}

```

```

/**
 * NIF letra kalkulatu (algoritmo ofiziala)
 */
private function nif_letra_kalkulatu(int $zenbakia): string
{
    $letroak = 'TRWAGMYFPDXBNJZSQVHLCKE';
    $posizioa = $zenbakia % 23;
    return $letroak[$posizioa];
}

/**
 * Eposta maskaratu
 *
 * @param string $eposta Jatorrizko eposta
 * @return string Eposta maskaratua
 */
public function eposta_maskaratu(string $eposta): string
{
    // Izen faltsua sortu
    $izen = strtolower($this->faker->firstName);
    $abizen = strtolower($this->faker->lastName);

    // test domeinua erabili
    return "{$izen}.{$abizen}@test.zabalagailetak.com";
}

/**
 * Telefonoa maskaratu - telefono baliozkoa sortu
 *
 * @param string $telefonoa Jatorrizko telefonoa
 * @return string Telefono maskaratua (baliozkoa)
 */
public function telefonoa_maskaratu(string $telefonoa): string
{
    // Mugikor espainiar baliozkoa: 6XXXXXXX edo 7XXXXXXX
    $lehenengo_digitua = rand(6, 7);
    $gainerakoak = rand(10000000, 99999999);

    return $lehenengo_digitua . $gainerakoak;
}

/**
 * IBAN maskaratu - IBAN baliozkoa sortu (ES)
 *
 * @param string $iban Jatorrizko IBAN
 * @return string IBAN maskaratua (baliozkoa)
 */
public function iban_maskaratu(string $iban): string
{
    // IBAN espainiar: ESxx xxxx xxxx xx xxxxxxxxxxxx
    $bankoa = str_pad(rand(0, 9999), 4, '0', STR_PAD_LEFT);

```

```

    $bulegoa = str_pad(rand(0, 9999), 4, '0', STR_PAD_LEFT);
    $kontrola = str_pad(rand(0, 99), 2, '0', STR_PAD_LEFT);
    $kontua = str_pad(rand(0, 9999999999), 10, '0', STR_PAD_LEFT);

    $iban_base = "ES00{$bankoa}{$bulegoa}{$kontrola}{$kontua}";

    // IBAN egiaztapen digitua kalkulatu
    $iban_zuzena = $this->iban_egiaztapen_kalkulatu($iban_base);

    return $iban_zuzena;
}

/**
 * IBAN egiaztapen digitua kalkulatu
 */
private function iban_egiaztapen_kalkulatu(string $iban): string
{
    // IBAN algoritmo ofiziala
    $kodea = substr($iban, 4) . 'ES00';
    $kodea = str_replace(
        ['A','B','C','D','E','F','G','H','I','J','K','L','M',
        'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'],
        [10,11,12,13,14,15,16,17,18,19,20,21,22,
        23,24,25,26,27,28,29,30,31,32,33,34,35],
        $kodea
    );

    $egiaztapen = 98 - bccmod($kodea, '97');
    $egiaztapen_str = str_pad($egiaztapen, 2, '0', STR_PAD_LEFT);

    return "ES{$egiaztapen_str}" . substr($iban, 4);
}

/**
 * Kreditu txartel zenbakia maskaratu (Luhn algoritmoa)
 *
 * @param string $txartela Jatorrizko txartel zenbakia
 * @return string Txartel zenbaki maskaratua (baliozkoa)
 */
public function kreditu_txartela_maskaratu(string $txartela): string
{
    // 16 digituko zenbaki bat sortu Luhn algoritmoarekin
    $zenbakia = '';
    for ($i = 0; $i < 15; $i++) {
        $zenbakia .= rand(0, 9);
    }

    // Azken digitua Luhn algoritmoarekin kalkulatu
    $azken_digitua = $this->luhn_egiaztapen_kalkulatu($zenbakia);

    return $zenbakia . $azken_digitua;
}

```

```

/**
 * Luhn algoritmo egiaztapena kalkulatu
 */
private function luhn_egiaztapen_kalkulatu(string $zenbakia): int
{
    $batura = 0;
    $bikoitza = false;

    for ($i = strlen($zenbakia) - 1; $i ≥ 0; $i--) {
        $digitua = intval($zenbakia[$i]);

        if ($bikoitza) {
            $digitua *= 2;
            if ($digitua > 9) {
                $digitua -= 9;
            }
        }

        $batura += $digitua;
        $bikoitza = !$bikoitza;
    }

    return (10 - ($batura % 10)) % 10;
}

/**
 * Izena maskaratu - izen faltsua sortu
 *
 * @param string $izena Jatorrizko izena
 * @return string Izen maskaratua
 */
public function izena_maskaratu(string $izena): string
{
    return $this->faker->firstName;
}

/**
 * Abizena maskaratu - abizen faltsua sortu
 *
 * @param string $abizena Jatorrizko abizena
 * @return string Abizen maskaratua
 */
public function abizena_maskaratu(string $abizena): string
{
    return $this->faker->lastName;
}

/**
 * Helbidea maskaratu - helbide faltsua sortu
 *
 * @param string $helbidea Jatorrizko helbidea

```

```

    * @return string Helbide maskaratua
    */
    public function helbidea_maskaratu(string $helbidea): string
    {
        return $this->faker->streetAddress;
    }

    /**
     * Hiria maskaratu
     */
    public function hiria_maskaratu(string $hiria): string
    {
        $hiri_euskaldunak = [
            'Donostia', 'Bilbo', 'Gasteiz', 'Iruña', 'Baiona',
            'Zarautz', 'Getxo', 'Durango', 'Eibar', 'Tolosa'
        ];

        return $hiri_euskaldunak[array_rand($hiri_euskaldunak)];
    }

    /**
     * Posta kodea maskaratu
     */
    public function posta_kodea_maskaratu(string $posta_kodea): string
    {
        // Euskal Herrian: 20xxx (Gipuzkoa), 48xxx (Bizkaia), 01xxx (Araba)
        $lehenengo = [20, 48, 01][array_rand([20, 48, 01])];
        $gainerakoak = str_pad(rand(0, 999), 3, '0', STR_PAD_LEFT);

        return $lehenengo . $gainerakoak;
    }

    /**
     * Soldata maskaratu - ausazko soldata tarte batean
     *
     * @param float $soldata Jatorrizko soldata
     * @return float Soldata maskaratua
     */
    public function soldata_maskaratu(float $soldata): float
    {
        // Soldata errealista sortu 20.000-60.000 tartean
        return round(rand(20000, 60000) + rand(0, 99) / 100, 2);
    }

    /**
     * Data jaiotza maskaratu (partzialki - urtea mantendu)
     *
     * @param string $data Data jatorrizkoa (YYYY-MM-DD)
     * @return string Data maskaratua (urtea berdina)
     */
    public function jaiotze_data_maskaratu(string $data): string
    {

```

```

        $urtea = substr($data, 0, 4);
        $hilabetea = str_pad(rand(1, 12), 2, '0', STR_PAD_LEFT);
        $eguna = str_pad(rand(1, 28), 2, '0', STR_PAD_LEFT);

        return "{$urtea}-{$hilabetea}-{$eguna}";
    }

/**
 * Pasahitz hash-a maskaratu - hash berri bat sortu
 *
 * @param string $pasahitz_hash Jatorrizko hash-a
 * @return string Hash maskaratua
 */
public function pasahitz_hash_maskaratu(string $pasahitz_hash): string
{
    // Pasahitz faltsu bat sortu eta hash-a kalkulatu (bcrypt)
    $pasahitz_faltsua = 'TestPassword123!';
    return password_hash($pasahitz_faltsua, PASSWORD_BCRYPT, ['cost' => 12]);
}

/**
 * IP helbidea maskaratu - IP pribatu bat sortu
 *
 * @param string $ip Jatorrizko IP
 * @return string IP maskaratua (192.168.x.x)
 */
public function ip_maskaratu(string $ip): string
{
    $b = rand(0, 255);
    $c = rand(1, 254);
    return "192.168.{$b}.{$c}";
}

/**
 * Pseudonimizazioa - identifikatzaile bakarra sortu
 *
 * @param mixed $balio Jatorrizko balioa
 * @param string $eremu Eremua (adib: 'erabiltzailea')
 * @return string Identifikatzaile pseudonimizatua
 */
public function pseudonimizatu($balio, string $eremu): string
{
    $gakoa = "{$eremu}_{$balio}";

    // Dagoeneko badago mapeatu?
    if (isset($this->mapaketa[$gakoa])) {
        return $this->mapaketa[$gakoa];
    }

    // Identifikatzaile berri bat sortu
    $id = strtoupper($eremu) . '_' . bin2hex(random_bytes(8));

```

```

        // Mapaketa gorde
        $this->mapaketa[$gakoa] = $id;

        return $id;
    }

    /**
     * Mapaketa taula gorde (JSON fitxategi batean)
     */
    public function mapaketa_gorde(string $fitxategi_bidea): void
    {
        $json = json_encode($this->mapaketa, JSON_PRETTY_PRINT);
        file_put_contents($fitxategi_bidea, $json);
    }

    /**
     * Mapaketa taula kargatu
     */
    public function mapaketa_kargatu(string $fitxategi_bidea): void
    {
        if (file_exists($fitxategi_bidea)) {
            $json = file_get_contents($fitxategi_bidea);
            $this->mapaketa = json_decode($json, true);
        }
    }
}

```

4.1.2 Erabilpen Adibidea

```

nif_maskaratu($nif_jatorrizkoa);
echo "NIF maskratua: {$nif_maskratua}\n";
// Emaita: NIF maskratua: 45678912J

// Eposta maskaratu
$eposta_jatorrizkoa = 'jon.azpiazu@zabalagailetak.com';
$eposta_maskratua = $maskera->eposta_maskaratu($eposta_jatorrizkoa);
echo "Eposta maskratua: {$eposta_maskratua}\n";
// Emaita: Eposta maskratua: mikel.etxeberria@test.zabalagailetak.com

// IBAN maskaratu
$iban_jatorrizkoa = 'ES9121000418450200051332';
$iban_maskratua = $maskera->iban_maskaratu($iban_jatorrizkoa);
echo "IBAN maskratua: {$iban_maskratua}\n";
// Emaita: IBAN maskratua: ES7720384756320000051234

// Pseudonimizazioa
$erabiltzaile_id = 42;
$pseudonimatua = $maskera->pseudonimizatu($erabiltzaile_id, 'erabiltzailea');
echo "Pseudonimizatua: {$pseudonimatua}\n";
// Emaita: Pseudonimizatua: ERABILTZAILEA_A7B8C9D0E1F2A3B4

```



```

-- =====
-- 2. BEZEROAK TAULA MASKARATU
-- =====

UPDATE bezeroak SET
    izen_osea = (
        ARRAY['Mikel', 'Ane', 'Jon', 'Maite'][floor(random() * 4 + 1)]
    ) || ' ' || (
        ARRAY['Etxeberria', 'Azpiazu', 'Galdos'][floor(random() * 3 + 1)]
    ),
    eposta = 'bezero' || id || '@test.com',
    telefonoa = '6' || lpad(floor(random() * 100000000)::text, 8, '0'),
    nif = 'X' || lpad(floor(random() * 10000000)::text, 7, '0') ||
        substr('TRWAGMYFPDXBNJZSQVHLCKE', (floor(random() * 10000000)::int % 23) +

-- =====
-- 3. FAKTURAK - Kantitate ausazkoak
-- =====

UPDATE fakturak SET
    zenbatekoa = (zenbatekoa * (0.8 + random() * 0.4))::numeric(10,2);

-- =====
-- 4. EGIAZTATU MASKARAKETA
-- =====

-- Egiaztatu ez dagoela datu erreala
DO $$
BEGIN
    -- Egiaztatu ez dagoela 'Z' amaitzen duen NIF erreala
    IF EXISTS (SELECT 1 FROM langileak WHERE nif LIKE '%Z') THEN
        RAISE EXCEPTION 'ERROREA: Datu erreala aurkitu da (NIF Z)';
    END IF;

    -- Egiaztatu eposta guztiak @test domeinuak direla
    IF EXISTS (SELECT 1 FROM langileak WHERE eposta NOT LIKE '%@test.zabalagailetak.c
        RAISE EXCEPTION 'ERROREA: Eposta erreala aurkitu da';
    END IF;

    RAISE NOTICE 'ONDO: Maskaraketa egiaztatu da';
END $$;

-- =====
-- 5. AUDITORIA ERREGISTROA SORTU
-- =====

INSERT INTO auditoria_erregistroak (
    ekintza,
    deskribapena,
    erabiltzailea,
    data

```

```

) VALUES (
    'DATU_MASKARAKETA',
    'Datu-base osoa maskaratu da ingurune ez-produkzioarentzat',
    'SISTEMA',
    NOW()
);

COMMIT;

```

4.3 Bash Script Automatizazioa

```

#!/bin/bash
# maskaratu_datu_basea.sh
# Datu-base produkzioa maskaratu eta garapen ingurura inportatu

set -e # Errorean gelditu

# Konfigurazioa
PROD_DB="zabala_prod"
DEV_DB="zabala_dev"
PROD_HOST="prod-db.zabalagailetak.local"
DEV_HOST="dev-db.zabalagailetak.local"
BACKUP_DIR="/backup/maskaratuak"
DATA=$(date +%Y%m%d_%H%M%S)

echo "====="
echo " DATU MASKARAKETA SCRIPT-A"
echo " Data: $(date)"
echo "====="

# 1. Produkzioko backup sortu
echo "[1/6] Produkzioko backup sortzen..."
pg_dump -h $PROD_HOST -U postgres $PROD_DB \
    > $BACKUP_DIR/backup_prod_$DATA.sql

echo "✓ Backup sortuta: backup_prod_$DATA.sql"

# 2. Backup-a garapen datu-basean inportatu
echo "[2/6] Backup-a garapen datu-basean inportatzen..."
psql -h $DEV_HOST -U postgres -d $DEV_DB \
    < $BACKUP_DIR/backup_prod_$DATA.sql

echo "✓ Backup inportatuta"

# 3. Maskaratzeko script-a exekutatu
echo "[3/6] Maskaratzeko script-a exekutatzen..."
psql -h $DEV_HOST -U postgres -d $DEV_DB \
    < maskaratu_datubasea.sql

echo "✓ Datu-basea maskaratuta"

```

```

# 4. Integritatearen egiaztatu
echo "[4/6] Integritatearen egiaztatzen..."
psql -h $DEV_HOST -U postgres -d $DEV_DB -c "
    SELECT
        COUNT(*) as langile_kopurua,
        COUNT(DISTINCT saila_id) as saila_kopurua
    FROM langileak;
"

# 5. Foreign key constraints egiaztatu
echo "[5/6] Foreign key constraints egiaztatzen..."
psql -h $DEV_HOST -U postgres -d $DEV_DB -c "
    DO \$\$
    DECLARE
        constraint_record RECORD;
    BEGIN
        FOR constraint_record IN
            SELECT conname, conrelid::regclass AS table_name
            FROM pg_constraint
            WHERE contype = 'f'
        LOOP
            EXECUTE 'ALTER TABLE ' || constraint_record.table_name ||
                ' VALIDATE CONSTRAINT ' || constraint_record.conname;
            RAISE NOTICE 'Constraint % egiaztatu da', constraint_record.conname;
        END LOOP;
    END \$\$;
"

# 6. Maskratutako backup-a gorde
echo "[6/6] Maskratutako backup-a gordetzen..."
pg_dump -h $DEV_HOST -U postgres $DEV_DB \
    > $BACKUP_DIR/backup_maskratua_$DATA.sql

echo "✓ Maskratutako backup-a gordeta: backup_maskratua_$DATA.sql"

# Zaharrenak ezabatu (30 egun baino zaharragoak)
find $BACKUP_DIR -name "backup_maskratua_*.sql" -mtime +30 -delete

echo ""
echo "====="
echo " MASKARAKETA OSATUTA!"
echo " Maskratutako datu-basea: $DEV_DB"
echo " Backup fitxategia: backup_maskratua_$DATA.sql"
echo "====="

```

Cron job-a konfiguratu automatizatzeko:

```

# crontab -e
# Asteazkenetan 02:00etan exekutatu
0 2 * * 3 /opt/scripts/maskaratu_datu_basea.sh >> /var/log/maskaraketa.log 2>&1

```

5. Test Datu Sortzailea

5.1 Test Datu Sortzeko Zerbitzua

Test datu errealistak sortu, produkzioko datuak erabiliz:

```
maskera = new DatuMaskaratzeZerbitzua();
    $this->faker = Faker::create('es_ES');
}

/**
 * Test langileak sortu
 *
 * @param int $kopurua Zenbat langile sortu
 * @return array Langile maskaratuen array-a
 */
public function langile_sortu(int $kopurua = 50): array
{
    $langileak = [];

    for ($i = 1; $i ≤ $kopurua; $i++) {
        $langileak[] = [
            'id' => $i,
            'izena' => $this->faker->firstName,
            'abizena' => $this->faker->lastName,
            'nif' => $this->maskera->nif_maskaratu(''),
            'eposta' => $this->maskera->eposta_maskaratu(''),
            'telefonoa' => $this->maskera->telefonoa_maskaratu(''),
            'jaiotze_data' => $this->faker->date('Y-m-d', '-30 years'),
            'soldata' => $this->maskera->soldata_maskaratu(0),
            'saila_id' => rand(1, 5),
            'kontratazio_data' => $this->faker->date('Y-m-d', '-5 years'),
            'iban' => $this->maskera->iban_maskaratu(''),
            'helbidea' => $this->maskera->helbidea_maskaratu(''),
            'hiria' => $this->maskera->hiria_maskaratu(''),
            'posta_kodea' => $this->maskera->posta_kodea_maskaratu('')
        ];
    }

    return $langileak;
}

/**
 * Test datuak datu-basean txertatu
 */
public function datu_basean_txertatu(array $langileak, $db): void
{
    foreach ($langileak as $langileak) {
        $db->query(
            'INSERT INTO langileak (
                izena, abizena, nif, eposta, telefonoa,
```

```

        jaiotze_data, soldata, saila_id, kontratazio_data,
        iban, helbidea, hiria, posta_kodea
    ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)',
    array_values($langileak)
);
    }
}

/**
 * Test datuak JSON fitxategira esportatu
 */
public function json_esportatu(array $langileak, string $fitxategia): void
{
    $json = json_encode($langileak, JSON_PRETTY_PRINT);
    file_put_contents($fitxategia, $json);
}
}

```

Erabilpen adibidea:

```

langile_sortu(100);

// JSON fitxategira esportatu
$sortzailea→json_esportatu($langileak, 'test_langileak.json');

// Edo datu-basean txertatu
// $sortzailea→datu_basean_txertatu($langileak, $db);

echo "✓ 100 test langile sortu dira\n";

```

6. Maskaraketa Auditoria

6.1 Maskaraketa Egiaztapen Script-a

```

db = $db;
}

/**
 * Datu-basea osoa egiaztatu
 */
public function datu_basea_egiaztatu(): bool
{
    echo "Datu-basea maskaraketa egiaztatzen...\n\n";

    // 1. Eposta egiaztatu
    $this→eposta_egiaztatu();

    // 2. NIF egiaztatu

```

```

$this->nif_egiaztatu();

// 3. Telefono egiaztatu
$this->telefono_egiaztatu();

// 4. IBAN egiaztatu
$this->iban_egiaztatu();

// 5. IP helbidea egiaztatu
$this->ip_egiaztatu();

// Emaitzak
if (count($this->erroreak) > 0) {
    echo "\n✗ MASKARAKETA HUTSEGITEA\n";
    echo "Errore kopurua: " . count($this->erroreak) . "\n\n";

    foreach ($this->erroreak as $errorea) {
        echo "    • {$errorea}\n";
    }

    return false;
}

echo "\n✓ MASKARAKETA ZUZENA\n";
echo "Ez da datu erreala aurkitu\n";

return true;
}

private function eposta_egiaztatu(): void
{
    $emaitzak = $this->db->query(
        "SELECT COUNT(*) as kopurua
        FROM langileak
        WHERE eposta NOT LIKE '%@test.zabalagailetak.com'"
    );

    if ($emaitzak['kopurua'] > 0) {
        $this->erroreak[] = "Eposta erreala aurkitu da ({$emaitzak['kopurua']}) la";
    } else {
        echo "✓ Eposta: Guztiak maskaratuak\n";
    }
}

private function nif_egiaztatu(): void
{
    // Egiaztatu ez dagoela NIF ezagunik
    $nif_ezagunak = ['12345678Z', '87654321X']; // Adibideak

    $placeholders = implode(',', array_fill(0, count($nif_ezagunak), '?'));

    $emaitzak = $this->db->query(

```

```

        "SELECT COUNT(*) as kopurua
        FROM langileak
        WHERE nif IN ($placeholders)",
        $nif_ezagunak
    );

    if ($emaitzak['kopurua'] > 0) {
        $this->erroreak[] = "NIF ezaguna aurkitu da";
    } else {
        echo "✓ NIF: Ez da NIF ezaguna aurkitu\n";
    }
}

private function telefono_egiaztatu(): void
{
    // Egiaztatu telefono guztiak 6 edo 7-rekin hasten direla
    $emaitzak = $this->db->query(
        "SELECT COUNT(*) as kopurua
        FROM langileak
        WHERE telefonoa NOT LIKE '6%' AND telefonoa NOT LIKE '7%'"
    );

    if ($emaitzak['kopurua'] > 0) {
        $this->erroreak[] = "Telefono formatu okerra aurkitu da";
    } else {
        echo "✓ Telefono: Formatu zuzena\n";
    }
}

private function iban_egiaztatu(): void
{
    // Egiaztatu IBAN guztiak ES-rekin hasten direla
    $emaitzak = $this->db->query(
        "SELECT COUNT(*) as kopurua
        FROM langileak
        WHERE iban NOT LIKE 'ES%'"
    );

    if ($emaitzak['kopurua'] > 0) {
        $this->erroreak[] = "IBAN formatu okerra aurkitu da";
    } else {
        echo "✓ IBAN: Formatu zuzena\n";
    }
}

private function ip_egiaztatu(): void
{
    // Egiaztatu IP guztiak 192.168.x.x direla
    $emaitzak = $this->db->query(
        "SELECT COUNT(*) as kopurua
        FROM auditoria_erregistroak
        WHERE ip_helbidea NOT LIKE '192.168.%'"
    );
}

```

```

    );

    if ($emaitzak['kopurua'] > 0) {
        $this->erroreak[] = "IP publiko bat aurkitu da auditoria log-ean";
    } else {
        echo "✓ IP: Guztiak pribatuak\n";
    }
}
}

```

Erabilpen adibidea:

```

datu_basea_egiaztatu();

if (!$zuzena) {
    exit(1); // Error code
}

```

7. Erantzukizunak eta Rolak

7.1 Rolak

Rola	Erantzukizunak
IT Kudeatzailea	Maskaratzeko prozesua kudeatu, script-ak exekutatu
DBA (Database Administrator)	Datu-base backup eta inportazioa, integritatearen egiaztatu
Segurtasun Taldea	Maskaraketa auditatu, arau-haustea identifikatu
Garatzaileak	Test datuak erabiliz garatu, produkzioko datuak ez eskatu
QA Taldea	Test datu maskaratuak erabiliz probak egin

7.2 Arauak

ARAU ZORROTZA:

- ✗ EZ erabiliz inoiz produkzioko datuak garapen edo test ingurune lokaletan
- ✗ EZ deskargatu produkzioko datuak pertsonaletara edo USB gailuetara
- ✗ EZ partekatu produkzioko datuak Slack, email edo beste kanal publikoetan
- ✓ BETI erabiliz datu maskaratuak garapen/test ingurune
- ✓ BETI egiaztatu maskaraketa zuzen dagoela

7.3 Arau-hauste Kasuan

Produkzioko datuak erabil badira ingurune ez-produkzioan:

Larritasuna	Ekintza
1. aldiz	Ohartarazpen idatzia + prestakuntza berreskuratzea
2. aldiz	Diziplina espedientea + sarbide murriztea
3. aldiz	Kaleratzea + GDPR arau-hauste irekitzea

8. Jarraipen eta Auditoria

8.1 Maskaratzeko Maiztasuna

Ingurune	Maskaratze Maiztasuna	Arduraduna
Garapena	Astean behin	IT Kudeatzailea
Testa (QA)	Bi asterako behin	QA Kudeatzailea
Staging	Hilean behin	DevOps Taldea
Prestakuntza	3 hilean behin	RRHH Kudeatzailea

8.2 Auditoria Prozedura

Auditoria hilean behin:

- 1. Egiaztatu maskaraketa script-a exekutatu da
- 2. Egiaztatu ez dagoela datu erreala test/dev ingurune
- 3. Egiaztatu mapaketa taula seguru gordeta dago
- 4. Auditoria txostena sortu

Auditoria txantiloia:

```
# DATU MASKARATZEA AUDITORIA TXOSTENA

**Data:** 2026-01-23
**Auditoreak:** Maite Etxeberria (IT Kudeatzailea)

## 1. Maskaratzeko Exekuzioak

| Ingurune | Azken Maskaraketa | Egiaztatu? | Emaizta |
|-----|-----|-----|-----|
| Garapena | 2026-01-20 | ✓ BAI | ZUZENA |
| Testa | 2026-01-15 | ✓ BAI | ZUZENA |
```

| Staging | 2026-01-10 | ✓ BAI | ZUZENA |

2. Egiaztapen Emaitzak

- ✓ Eposta: Guztiak @test.zabalagailetak.com
- ✓ NIF: Ez da NIF ezaguna
- ✓ Telefono: Formatu zuzena
- ✓ IBAN: Formatu zuzena
- ✓ IP: Guztiak pribatuak (192.168.x.x)

3. Arau-haustekak

Ez da arau-hausterik aurkitu.

4. Gomendioak

- Mapaketa taula backup-ean gorde (enkriptatuta)
- Maskaratzeko script-a eguneratu liburu berri bat txertatzeko

Sinadura: Maite Etxeberria

Data: 2026-01-23

9. Erreferentzia eta Loturak

9.1 ISO 27001:2022 Kontrolak

- **A.8.11 - Datu Maskaratzea:** Test datuak maskaratu behar dira produkzioko datuetan oinarrituta daudenean

9.2 GDPR Artikuluak

- **Artikulu 32 - Tratamenduaren Segurtasuna:** Babes neurri teknikoak eta antolaketazkoak inplementatu behar dira
- **Artikulu 25 - Babes Diseinua eta Lehenetsitakoa:** Privacy by Design printzipioa aplikatu

9.3 Lotutako Dokumentuak

- Zabalagailetak/compliance/sgsi/sailkapen_eskuliburua.md - Informazio Sailkapen Eskuliburua
 - Zabalagailetak/compliance/gdpr/data_processing_register.md - Tratamendu Erregistroa
 - Zabalagailetak/devops/sop_secure_development.md - Garapen Seguru Prozedura
-

10. Berrikuste Historia

Bertsioa	Data	Aldaketak	Egilea
1.0	2026-01-23	Hasierako bertsioa sortuta	OpenCode AI + CISO

11. Onespena

Onartua:

Jon Azpiazu
CISO (Informazio Segurtasuneko Buruordea)
Data: 2026-01-23

Maite Etxeberria
IT Zuzendaria
Data: 2026-01-23

Ane Galdos
DPO (Datu Babeseko Ordezkaria)
Data: 2026-01-23

Dokumentu amaiera
Zabala Gailetak S.L. - ISO 27001:2022 / GDPR
Konfidentzialtasun Maila: KONFIDENTZIALA