ZABALA GAILETAK

S.L. - Dokumentazio Akademikoa

Ekoizpen Seguruan Jartzea

2026(e)ko otsailaren 23(a)

MODULUA 2: EKOIZPEN SEGURUAN JARTZEA (DevSecOps)

Zabala Gailetak — Zibersegurtasun Proiektua ER4

Erakundea: Zabala Gailetak S.L. — Gaileta Ekoizle Industriala **Dokumentu Mota:** DevSecOps eta Aplikazio Segurtasun Dokumentazio Integrala **Bertsioa:** 2.0 **Data:** 2026-02-23 **Egilea:** Garapen eta Segurtasun Arkitektura Taldea **Sailkapena:** KONFIDENTZIALA — Barne Erabilpenerako Soilik

AURKIBIDEA

1. SARRERA ETA HELBURUAK

1.1 Moduluaren Xedea

Dokumentu honek **Zabala Gailetak S.L.** enpresaren GG Atari web aplikazioaren eta Android mugikor aplikazioaren **garapen seguru**, **hedapen automatizatu** eta **ekoizpen inguruneko segurtasun** neurri guztiak jasotzen ditu. ER4 proiektuaren bigarren modulua da, eta DevSecOps filosofiaren praktika guztiak biltzen ditu erreferentzia tekniko nagusi gisa.

Zabala Gailetak-eko ekoizpen seguruan jartzeak bi osagai nagusi ditu:

- **GG Ataria (HR Portala):** PHP 8.4 oinarritutako web aplikazioa, JWT autentifikazioarekin, MFA bigarren faktorearekin eta RBAC sarbide kontrolarekin.
- **Android Aplikazioa:** Kotlin 2.0.21 + Jetpack Compose bidez garatutako mugikor aplikazioa, Clean Architecture eta segurtasun sendoarekin.

1.2 Aplikagarriak diren Estandarrak

Estandarra	Aplikazio-Eremua
OWASP ASVS	Aplikazioaren segurtasun egiaztapena
OWASP Top 10	Web aplikazioen arriskuak
OWASP Mobile Top 10	Android segurtasun arriskuak
ISO/IEC 27001 Anex. 8	Software garapen kontrolak
POP-015	Garapen seguru prozedura operatiboa
NIST SSDF	Garapen seguru esparru nazionala
PSR-12	PHP kode estandarrak

1.3 Proiektuaren Estatistikak

Metrika	Balioa
Kode lerroak guztira	~7.000 (PHP, Kotlin, YAML, Bash)
Test unitarioak	82/82 — %100 gainditu
API endpoint-ak	20 (autentifikazioa + CRUD)
Docker zerbitzuak	4 (Nginx, PHP-FPM, PostgreSQL, Redis)
CI/CD pipeline-ak	3 (Sintaxi, Hedapena, OpenCode)
Garapen faseak	3 osatu (Oinarria, Auth, CRUD Full Stack)
Segurtasun kontrolak	JWT + MFA/TOTP + RBAC + Audit Trail

2. SSDLC — GARAPEN SEGURU BIZI-ZIKLOA

2.1 SSDLC Filosofia eta Printzipioak

Zabala Gailetak-eko SSDLC (Secure Software Development Lifecycle) DevSecOps printzipioan oinarritzen da: segurtasuna garapen prozesuaren atal integraltzat hartzen da, eta ez azken urrats gisa.

Oinarrizko printzipioak:

- **Shift Left:** Segurtasun probak garapen fasera eraman, produkziora iritsi aurretik arazoak konpontzeko.
- **Security by Design:** Diseinu fasean segurtasun eskakizunak zehaztea.
- **Defense in Depth:** Segurtasun geruzak pilatzea — sartu batetik aurrera egitea zailago bihurtuz.
- **Zero Trust:** Ez fidatu inori sare barrutik ere — autentifikatu eta egiaztatu beti.

2.2 POP-015 — Garapen Seguru Prozedura Operatiboa

Arduradunak: Development Lead / CISO **Bertsioa:** 1.0 | **Azken Eguneraketa:** 2026-01-23

2.2.1 Fase 1: Diseinua

Mehatxuen Modelizazioa (Threat Modeling):

Diseinu fasean STRIDE metodologiaren araberako mehatxu-analisia egiten da:

Mehatxu Mota	Akronimoa	Adibidea Zabala Gailetak-en
Spoofing	S	Langile baten identitatea ordezkatu
Tampering	T	Nomina datuak aldatu
Repudiation	R	Ekintza ukatu, audit trail ez badago
Information Disclosure	I	Gaileta errezetak filtratu

Mehatxu Mota	Akronimoa	Adibidea Zabala Gailetak-en
Denial of Service	D	GG Ataria blokeatu
Elevation of Privilege	E	Employee → Admin bilakatu

Segurtasun Eskakizunak (hasieratik definituak):

- Datu guztiak HTTPS bidez joan behar dira
- Pasahitzak bcrypt 12+ errondarekin hash egin
- JWT tokenen iraungitzea ≤ 1 ordu
- MFA derrigorrezkoa admin kontuetarako
- Sarrera guztiak baliozkotzea eta saneatzea

2.2.2 Fase 2: Garapena

Kode Kalitate Tresnak:

Tresna	Helburua	Exekutatzea
SonarLint	IDE-an kode analisia	Denbora errealean
PHPStan	PHP analisi estatikoa	Commit aurretik
PHPCS	PSR-12 kode estandarra	CI/CD pipeline-an
PHPUnit	Unit testing	CI/CD pipeline-an

Garapen Arau Kritikoak:

```
# ❌ INOIZ EZ egin hau:
$password = "admin123";           # Hardcoded kredentzialak
$query = "SELECT * FROM users WHERE id = $id";  # SQL injection

# ✅ Beti egin hau:
$password = $_ENV['DB_PASSWORD']; # .env fitxategitik
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = ?");  # Prepared stater
```

Fitxategi Sentikorretan Kontrola:

```
# .gitignore — Inoiz ez igo repositoriora
.env
*.env.local
*.key
*.pem
secrets/
```

2.2.3 Fase 3: Proba Automatizatuak

CI/CD pipeline-an hiru proba mota exekutatzen dira:

SAST (Static Application Security Testing):

```
# PHPStan — Analisi estatikoa
./vendor/bin/phpstan analyse src/ --level=8

# PHPCS — PSR-12 egiaztapena
./vendor/bin/phpcs --standard=PSR12 src/
```

SCA (Software Composition Analysis):

```
# Composer dependentzien ahultasun analisia
composer audit

# npm dependentziak (web frontenda)
npm audit --audit-level=high
```

DAST (Dynamic Application Security Testing):

OWASP ZAP staging ingurunean exekutatzen da aplikazioa martxan dagoenean:

```
# OWASP ZAP automatizatua
docker run -t owasp/zap2docker-stable zap-baseline.py \
    -t http://staging.zabalagailetak.local \
    -r zap-report.html
```

2.2.4 Fase 4: Kode Berrikusketa

Pull Request Politika:

- PR bakoitza beste garatzaile batek berrikusi behar du
- Segurtasun checklist bete behar da
- CI pipeline-ak arrakasta izan behar du (sintaxi + testak)
- 2 onarpena beharrezkoak dira main-era sartu aurretik

Segurtasun Checklist:

```
□ Sarrera guztiak baliozkotuta eta saneaturik
□ Koderik ez du kredentzialik barnean
□ SQL prepared statements erabilita
□ XSS babeserako htmlspecialchars() erabilita
□ CSRF tokena existitzen da formulario guztientzat
□ Errore mezuak ez du informazio sentikorra agerian utzi
□ Segurtasun goiburuak konfiguratuak
□ Audit trail operazio sentikorretan
```

2.2.5 Fase 5: Hedapena

Produkzioko Konfigurazio Arauak:

```
// config/app.php – Produkzioko ezarpenak
return [
    'env'   => $_ENV['APP_ENV'],   // 'production' ez 'development'
    'debug' => false,              // Debug modua ITZALITA beti produkzioan
    'log_level' => 'error',        // Log minimoa produkzioan
];
```

Inguruneen Bereizketa:

Ingurunea	Konfigurazioa	Debug	Error Mostratu
Garapena	.env.local	true	true
Proba (Staging)	.env.test	false	true (log soilik)
Produkzioa	.env.production	false	false

2.2.6 Garatzaile Prestakuntza

- Garatzaile berriek garapen seguruko oinarrizko ikastaroa egin behar dute (8 ordu)

- Urtean behin OWASP Top 10 freskatze saioa (4 ordu)
- Hiruhilero: PHPStan eta testen emaitzen berrikuspena

3. CI/CD PIPELINE-AK — GITHUB ACTIONS

3.1 Pipeline Arkitektura

Zabala Gailetak-eko CI/CD sistema **GitHub Actions** bidez kudeatzen da. Hiru pipeline nagusi daude:

Pipeline	Fitxategia	Aktibatzea	Helburua
Sintaxi Egiaztapena	`ci-minimal.yml`	Push + PR	PHP 8.4 sintaxia egiaztatu
Hedapena		Eskuzkoa	InfinityFree-ra igo
OpenCode	`opencode.yml`	Iruzkin `/oc`	AI-arekin laguntzea

3.2 Pipeline 1 — Sintaxi Egiaztapena (ci-minimal.yml)

Push eta Pull Request bakoitzean automatikoki exekutatzen da:

```yaml
name: CI - PHP Syntax Check
on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  syntax-check:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Setup PHP 8.4
        uses: shivammathur/setup-php@v2
        with:
          php-version: '8.4'
          extensions: pdo, pdo_pgsql, gd, opcache, bcmath

      - name: Check PHP Syntax (Parallel)
        run: |
          echo "=== PHP Sintaxi Egiaztapena ==="
          find . -type f -name "*.php" -not -path "*/vendor/*" \
            -print0 | xargs -0 -n1 -P8 php -l
          echo "=== Sintaxi egiaztapena GAINDITU ==="

      - name: Run PHPStan (Static Analysis)
        run: |
          cd "Zabala Gailetak/hr-portal"
          if [ -f "vendor/bin/phpstan" ]; then
            ./vendor/bin/phpstan analyse src/ --level=5 --no-progress
          fi
```

Ezaugarri nagusiak:

- 8 prozesu paraleloak sintaxi egiaztapenerako (azkarragoa)
- Vendor direktorioa baztertzen du (kanpo liburutegiak ez egiaztatu)
- PHP 8.4 erabiltzen du produkzioaren berdina izateko
- PHPStan analisi estatikoa (5. maila)

3.3 Pipeline 2 — InfinityFree Hedapena (deploy.yml)

Produkziora hedatzeko pipeline-a. **Eskuzko aktibatzea soilik** (workflow_dispatch):

```yaml
name: Deploy to InfinityFree (Production)
on:
  workflow_dispatch:    # Eskuzko aktibatzea soilik — ez automatiko
    inputs:
      confirm_deploy:
        description: 'Produkziora hedatu? (bai/ez)'
        required: true
        default: 'ez'

jobs:
  deploy:
    runs-on: ubuntu-latest
    if: |
      github.ref == 'refs/heads/main' &&
      github.event.inputs.confirm_deploy == 'bai'

    steps:
      - uses: actions/checkout@v4

      - name: Pre-deploy PHP Syntax Check
        uses: shivammathur/setup-php@v2
        with:
          php-version: '8.4'

      - name: Verify syntax before deploy
        run: |
          find "Zabala Gailetak/hr-portal" -name "*.php" \
            -not -path "*/vendor/*" -print0 | xargs -0 -n1 php -l

      - name: Deploy via FTPS
        uses: SamKirkland/FTP-Deploy-Action@v4.3.5
        with:
          server: ${{ secrets.FTP_SERVER }}
          username: ${{ secrets.FTP_USERNAME }}
          password: ${{ secrets.FTP_PASSWORD }}
          protocol: ftps                              # FTPS — enkriptatuta
          local-dir: "./Zabala Gailetak/hr-portal/"
          server-dir: "/htdocs/"
          exclude: |
```

```
            **/.git*/**
            **/tests/**
            **/.env.example
            **/phpunit.xml
            **/Dockerfile
            **/.env
            **/secrets/**
            **/node_modules/**
```

Segurtasun Neurriak:

Neurria	Xehetasuna
Protokoloa	FTPS (FTP enkriptatua, ez FTP arrunta)
Kredentzialak	GitHub Secrets bidez gordetako — ez kodean
Aktibatzea	Eskuzkoa soilik — ez push automatikoak
Egiaztapena	Sintaxi proba hedatu aurretik
Baztertuak	Tests, .env, Dockerfile, secrets direktorioetarik
Babesa	main adarra soilik

GitHub Secrets konfiguratuta:

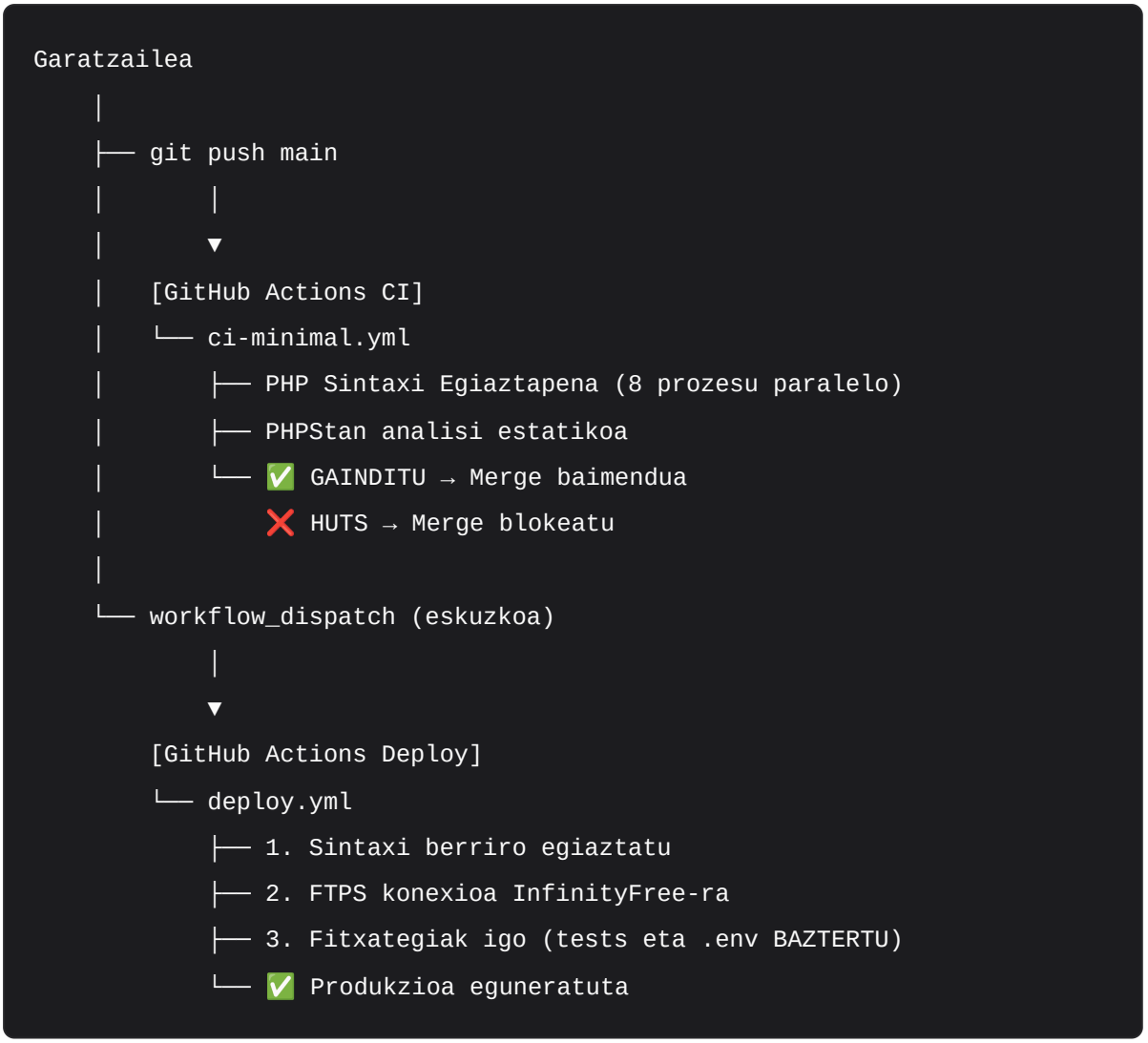
```
FTP_SERVER      → infinityfree.net FTP zerbitzaria
FTP_USERNAME    → FTP erabiltzaile izena
FTP_PASSWORD    → FTP pasahitza (enkriptatuta GitHub-en)
```

3.4 Pipeline 3 — OpenCode AI Laguntza (opencode.yml)

Garapen taldeak GitHub iruzkinetan `/oc` edo `/opencode` idatzita AI laguntza jaso dezake:

```yaml
name: opencode

on:
  issue_comment:
    types: [created]
  pull_request_review_comment:
    types: [created]

jobs:
  opencode:
    if: |
      contains(github.event.comment.body, ' /oc') ||
      startsWith(github.event.comment.body, '/oc') ||
      contains(github.event.comment.body, ' /opencode') ||
      startsWith(github.event.comment.body, '/opencode')
    runs-on: ubuntu-latest
    permissions:
      id-token: write
      contents: read
      pull-requests: read
      issues: read
    steps:
      - uses: actions/checkout@v6
        with:
          persist-credentials: false

      - name: Run opencode
        uses: anomalyco/opencode/github@latest
        env:
          OPENCODE_API_KEY: ${{ secrets.OPENCODE_API_KEY }}
        with:
          model: opencode/glm-4.7
```

3.5 Hedapen Arkitektura Osoa



Ingurunea	Plataforma	URL
Garapena	Docker lokala	http://localhost:8080
Proba (IsardVDI)	Proxmox VM (ZG-App)	http://192.168.20.10
Produkzioa	InfinityFree	https://zabala-gailetak.infinityfreeapp.com

4. DOCKER HEDAPENA ETA INGURUNEAK

4.1 Docker Compose — GG Ataria

GG Atariko garapen ingurunea Docker Compose bidez hedatzen da:

```
# docker-compose.hrportal.yml — Zabala Gailetak GG Ataria
services:

  # === POSTGRESQL 16 — Datu-base nagusia ===
  postgres:
    image: postgres:16-alpine
    container_name: zabala-postgres
    restart: unless-stopped
    environment:
      POSTGRES_DB: hr_portal
      POSTGRES_USER: hr_user
      POSTGRES_PASSWORD: ${DB_PASSWORD}
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./hr-portal/migrations:/docker-entrypoint-initdb.d:ro
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U hr_user -d hr_portal"]
      interval: 10s
      timeout: 5s
      retries: 5
    networks:
      - backend-net

  # === REDIS 7 — Cache eta Saio kudeaketa ===
  redis:
    image: redis:7-alpine
    container_name: zabala-redis
    restart: unless-stopped
    command: redis-server --appendonly yes --requirepass ${REDIS_PASSWORD}
    volumes:
      - redis_data:/data
    healthcheck:
      test: ["CMD", "redis-cli", "ping"]
      interval: 10s
      timeout: 5s
      retries: 5
    networks:
      - backend-net
```

```
  # === PHP 8.4-FPM — Aplikazio zerbitzaria ===
  php:
    build:
      context: ./hr-portal
      dockerfile: Dockerfile
    container_name: zabala-php
    restart: unless-stopped
    volumes:
      - ./hr-portal:/var/www/html:ro  # Read-only produkzioan
      - php_sessions:/tmp/sessions
    environment:
      APP_ENV: ${APP_ENV:-development}
      DB_HOST: postgres
      DB_PORT: 5432
      DB_NAME: hr_portal
      DB_USER: hr_user
      DB_PASS: ${DB_PASSWORD}
      REDIS_HOST: redis
      REDIS_PASS: ${REDIS_PASSWORD}
      JWT_SECRET: ${JWT_SECRET}
      JWT_EXPIRY: 3600          # 1 ordu
      JWT_REFRESH_EXPIRY: 604800 # 7 egun
    depends_on:
      postgres:
        condition: service_healthy
      redis:
        condition: service_healthy
    networks:
      - backend-net
      - frontend-net

  # === NGINX — Web zerbitzaria / Reverse Proxy ===
  nginx:
    image: nginx:alpine
    container_name: zabala-nginx
    restart: unless-stopped
    ports:
      - "8080:80"
      - "8443:443"
    volumes:
```

```
      - ./hr-portal/public:/var/www/html/public:ro
      - ./nginx/nginx-hrportal.conf:/etc/nginx/nginx.conf:ro
      - ./nginx/ssl:/etc/nginx/ssl:ro
    depends_on:
      - php
    healthcheck:
      test: ["CMD", "wget", "--quiet", "--spider", "http://localhost/api/hea
      interval: 30s
      timeout: 10s
      retries: 3
    networks:
      - frontend-net

networks:
  backend-net:
    internal: true   # Internet sarbiderik gabe
  frontend-net:

volumes:
  postgres_data:
  redis_data:
  php_sessions:
```

Segurtasun ezaugarriak:

- `backend-net` : Internet sarbiderik gabeko sare internoa (PostgreSQL + Redis isolatuta)
- `frontend-net` : Nginx soilik kanpora azaltzen da
- Bolumena read-only produkzioan (`ro`)
- Healthcheck-ak zerbitzuen eskuragarritasuna bermatzeko
- Pasahitz guztiak `.env` aldagaietatik (ez kodean)

4.2 PHP Dockerfile

```
# Zabala Gailetak — PHP 8.4 Produktzio Irudia
FROM php:8.4-fpm-alpine

# Sistema dependentziak
RUN apk add --no-cache \
    postgresql-dev \
    libpng-dev \
    libjpeg-turbo-dev \
    freetype-dev \
    zip \
    git \
    curl \
    && docker-php-ext-configure gd --with-freetype --with-jpeg \
    && docker-php-ext-install \
        pdo \
        pdo_pgsql \
        gd \
        opcache \
        bcmath

# Composer instalatu
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

WORKDIR /var/www/html

# Dependentziak instalatu (produkzioan dev gabe)
COPY composer.json composer.lock ./
RUN composer install \
    --no-dev \
    --optimize-autoloader \
    --no-scripts \
    --no-interaction

# Aplikazioaren kodea kopiatu
COPY . /var/www/html

# Baimenak finkatu
```

```
RUN chown -R www-data:www-data /var/www/html \
    && chmod -R 755 /var/www/html \
    && chmod -R 777 /var/www/html/storage

# OPcache konfiguratu errendimendurako
RUN echo "opcache.enable=1" >> /usr/local/etc/php/conf.d/opcache.ini \
    && echo "opcache.memory_consumption=128" >> /usr/local/etc/php/conf.d/opc
    && echo "opcache.max_accelerated_files=4000" >> /usr/local/etc/php/conf.d
    && echo "opcache.validate_timestamps=0" >> /usr/local/etc/php/conf.d/opca

# PHP segurtasun konfigurazioa
RUN echo "expose_php = Off" >> /usr/local/etc/php/conf.d/security.ini \
    && echo "display_errors = Off" >> /usr/local/etc/php/conf.d/security.ini
    && echo "log_errors = On" >> /usr/local/etc/php/conf.d/security.ini \
    && echo "max_execution_time = 30" >> /usr/local/etc/php/conf.d/security.i
    && echo "upload_max_filesize = 5M" >> /usr/local/etc/php/conf.d/security

EXPOSE 9000
CMD ["php-fpm"]
```

4.3 Nginx Konfigurazioa

```
# nginx-hrportal.conf — Zabala Gailetak GG Ataria
server {
    listen 80;
    server_name _;
    root /var/www/html/public;
    index index.php;

    # === SEGURTASUN GOIBURUAK ===
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin" always;
    add_header Content-Security-Policy
      "default-src 'self'; script-src 'self'; style-src 'self' 'unsafe-inline
      always;
    add_header Permissions-Policy "camera=(), microphone=(), geolocation=()"

    # Nginx bertsioa ezkutatu
    server_tokens off;

    # === RATE LIMITING ===
    limit_req_zone $binary_remote_addr zone=api_limit:10m rate=10r/s;
    limit_req_zone $binary_remote_addr zone=login_limit:10m rate=5r/m;

    # === API OROKORRA ===
    location /api {
        limit_req zone=api_limit burst=20 nodelay;
        try_files $uri /index.php?$query_string;
    }

    # === LOGIN ENDPOINT — Murrizpen zorrotzagoa ===
    location /api/auth/login {
        limit_req zone=login_limit burst=3 nodelay;
        try_files $uri /index.php?$query_string;
    }

    # === PHP PROZESATZEA ===
```

```
    location ~ \.php$ {
        fastcgi_pass php:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param HTTP_PROXY "";    # SSRF prebentzioa
        fastcgi_read_timeout 30;
    }

    # === FITXATEGI SENTIKORRAK DEBEKATU ===
    location ~ (composer\.json|composer\.lock|\.env|phpunit\.xml)$ {
        deny all;
        return 404;
    }

    # === FITXATEGI EZKUTUAK DEBEKATU ===
    location ~ /\. {
        deny all;
        return 404;
    }

    # === ESTATIKOEN CACHEA ===
    location ~* \.(js|css|png|jpg|gif|ico|svg|woff2)$ {
        expires 1y;
        add_header Cache-Control "public, immutable";
    }

    # === OSASUN EGIAZTAPENA ===
    location /api/health {
        allow all;
        try_files $uri /index.php?$query_string;
    }
}
```

4.4 Inguruneen Konfigurazio Fitxategia

```
# .env.example — Kopiatu .env izenera eta bete
APP_ENV=development
APP_KEY=ChangeMe32CharacterSecretKeyHere!

# Datu-basea
DB_HOST=postgres
DB_PORT=5432
DB_NAME=hr_portal
DB_USER=hr_user
DB_PASSWORD=ChangeMe_StrongPassword_Here

# Redis
REDIS_HOST=redis
REDIS_PORT=6379
REDIS_PASSWORD=ChangeMe_Redis_Password

# JWT
JWT_SECRET=ChangeMe_64CharacterJWTSecretKey_ForHMACSHA256Algorithm!
JWT_EXPIRY=3600
JWT_REFRESH_EXPIRY=604800

# TOTP MFA
TOTP_ISSUER=ZabalaGailetak_HR
TOTP_DIGITS=6
TOTP_PERIOD=30

# Email (Gorabeherak jakinarazteko)
MAIL_HOST=smtp.zabalagailetak.com
MAIL_PORT=587
MAIL_USER=no-reply@zabalagailetak.com
MAIL_PASS=ChangeMe_Mail_Password
```

4.5 Azpi-sistema Osoa Abiaraztea

```
# === GARAPEN INGURUNEA ABIARAZI ===

# 1. Repositorioa klonatu
git clone https://github.com/zabalagailetak/hr-portal.git
cd hr-portal

# 2. Ingurune aldagaiak konfiguratu
cp .env.example .env
nano .env  # Sekretuak bete

# 3. Kontenedoreak abiarazi
docker compose -f docker-compose.hrportal.yml up -d

# 4. PHP dependentziak instalatu
docker compose exec php composer install

# 5. Migrazioak exekutatu
docker compose exec postgres psql -U hr_user -d hr_portal \
    -f /docker-entrypoint-initdb.d/001_init_schema.sql
docker compose exec postgres psql -U hr_user -d hr_portal \
    -f /docker-entrypoint-initdb.d/002_seed_data.sql
docker compose exec postgres psql -U hr_user -d hr_portal \
    -f /docker-entrypoint-initdb.d/003_audit_trail.sql
docker compose exec postgres psql -U hr_user -d hr_portal \
    -f /docker-entrypoint-initdb.d/004_create_vacations_tables.sql

# 6. Instalazio egiaztatu
curl http://localhost:8080/api/health
# Emaitza: {"status":"ok","database":"connected","redis":"connected"}

# 7. Testak exekutatu (80+ test)
docker compose exec php ./vendor/bin/phpunit --testdox
```

5. ISARDVDI/PROXMOX HEDAPEN GIDA

5.1 VM Espezifikazioak

IsardVDI/Proxmox ingurunean 6 VM sortu behar dira proiektua simulatzeko:

VM Izena	vCPU	RAM	HDD	Funtzioa
ZG-Gateway	1	1 GB	10 GB	Bideratzailea + Suebakia + DHCP
ZG-App	2	4 GB	20 GB	Nginx + PHP + GG Ataria
ZG-Data	2	4 GB	20 GB	PostgreSQL + Redis
ZG-SecOps	4	8 GB	40 GB	ELK SIEM + Wazuh
ZG-OT	1	2 GB	10 GB	OpenPLC + ScadaBR
ZG-Client	2	4 GB	20 GB	Garatzaile / Erabiltzaile

Sare konfiguraketa:

VM	NIC 1	NIC 2	IP Helbidea
ZG-Gateway	Bridge (WAN)	Barnekoa	192.168.1.1
ZG-App	—	Barnekoa	192.168.20.10
ZG-Data	—	Barnekoa	192.168.20.20
ZG-SecOps	—	Barnekoa	192.168.20.50
ZG-OT	—	Barnekoa	192.168.50.10
ZG-Client	—	Barnekoa	DHCP (192.168.10.x)

5.2 ZG-App VM Konfigurazioa

ZG-App VM-an GG Ataria instalatzeko:

```
# === ZG-APP VM-an exekutatu (root bezala) ===

# 1. IP estatikoa konfiguratu
cat <<EOF > /etc/network/interfaces.d/eth0.cfg
allow-hotplug eth0
iface eth0 inet static
    address 192.168.20.10
    netmask 255.255.255.0
    gateway 192.168.20.1
    dns-nameservers 8.8.8.8
EOF
systemctl restart networking

# 2. Sistema eguneratu eta Docker instalatu
apt update && apt upgrade -y
apt install -y docker.io docker-compose-v2 git curl ufw fail2ban

# 3. Suebakia konfiguratu
ufw default deny incoming
ufw default allow outgoing
ufw allow from 192.168.200.0/24 to any port 22   # SSH kudeaketa saretik
ufw allow from 192.168.10.0/24 to any port 80    # HTTP erabiltzaileetatik
ufw allow from 192.168.10.0/24 to any port 443   # HTTPS erabiltzaileetatik
ufw allow from 192.168.20.50 to any port 514     # Syslog SecOps-era
ufw enable

# 4. Fail2Ban konfiguratu
cat <<EOF >> /etc/fail2ban/jail.local
[sshd]
enabled = true
maxretry = 3
bantime = 3600
findtime = 600

[nginx-limit-req]
enabled = true
port = http,https
logpath = /var/log/nginx/error.log
maxretry = 10
```

```
EOF
systemctl enable --now fail2ban

# 5. Repositorioa klonatu eta abiarazi
git clone https://github.com/zabalagailetak/erronka4.git /opt/zabala
cd /opt/zabala/Zabala\ Gailetak

cp hr-portal/.env.example hr-portal/.env
# .env editatu produkzioko balioekin

docker compose -f docker-compose.hrportal.yml up -d
```

5.3 ZG-Data VM Konfigurazioa

```
# === ZG-DATA VM-an exekutatu ===

# UFW — PostgreSQL eta Redis ZG-App-etik soilik
ufw default deny incoming
ufw allow from 192.168.200.0/24 to any port 22
ufw allow from 192.168.20.10 to any port 5432   # PostgreSQL
ufw allow from 192.168.20.10 to any port 6379   # Redis
ufw enable

# Docker zerbitzuak abiarazi
docker compose -f docker-compose.hrportal.yml \
    up -d postgres redis
```

5.4 Ekoizpen Egiaztapena

```
# === GG Ataria Ongi Funtzionatzen Duela Egiaztatu ===

# 1. Osasun egiaztapena
curl http://192.168.20.10/api/health
# Espero: {"status":"ok","database":"connected","redis":"connected"}

# 2. Login proba
curl -X POST http://192.168.20.10/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email":"admin@zabalagailetak.com","password":"Admin@2026!"}'
# Espero: {"access_token":"eyJ...","token_type":"Bearer"}

# 3. Docker kontenedoreak egiaztatu
docker compose ps
# nginx       running (healthy)
# php         running
# postgres    running (healthy)
# redis       running (healthy)

# 4. Log-ak ikusi
docker compose logs nginx --tail=50
docker compose logs php --tail=50
```

6. GG ATARIA — PHP 8.4 WEB APLIKAZIOA

6.1 Arkitektura Orokorra

Zabala Gailetak-en **Giza Baliabideen (GG) Ataria** PHP 8.4 oinarritutako web aplikazioa da, **PSR-12** kode estandarrekin eta **Clean Architecture** printzipioekin garatuta:

```
hr-portal/
├── public/
│   └── index.php             # Sarrera puntua (Front Controller)
├── src/
│   ├── controllers/
│   │   ├── AuthController.php         (JWT + MFA kudeaketa)
│   │   ├── EmployeeController.php     (CRUD + Audit, 450 lerro)
│   │   └── AuditController.php        (Audit historia, 150 lerro)
│   ├── validators/
│   │   └── EmployeeValidator.php      (Balidazio Espainiarra, 600 lerro)
│   ├── utils/
│   │   ├── TokenManager.php           (JWT inplementazioa natiboa)
│   │   ├── AuditLogger.php            (Ekintza erregistroa, 200 lerro)
│   │   └── MfaManager.php             (TOTP kudeaketa)
│   ├── models/
│   │   └── Employee.php               (ORM eredua, 300 lerro)
│   └── middleware/
│       ├── AuthMiddleware.php          (JWT egiaztapena)
│       └── RbacMiddleware.php          (Baimenen egiaztapena)
├── tests/
│   ├── TokenManagerTest.php           (11 test)
│   ├── EmployeeControllerTest.php     (11 test)
│   ├── EmployeeValidatorTest.php      (40 test)
│   ├── AuditLoggerTest.php            (11 test)
│   └── AuditControllerTest.php        (9 test)
├── migrations/
│   ├── 001_init_schema.sql
│   ├── 002_seed_data.sql
│   ├── 003_audit_trail.sql
│   └── 004_create_vacations_tables.sql
├── Dockerfile
├── composer.json
└── phpunit.xml
```

Stack Teknologikoa:

Osagaia	Teknologia	Helburua
Backend	PHP 8.4-FPM	API REST eta logika

Osagaia	Teknologia	Helburua
Datu-basea	PostgreSQL 16	Datu iraunkorra
Cache	Redis 7 (AOF)	Saio kudeaketa
Web Zerbitzaria	Nginx Alpine	Reverse proxy + segurtasuna
Autentifikazioa	JWT natiboa (HMAC-SHA256)	Token-etan oinarritutako auth
MFA	TOTP (30s)	Bigarren faktorea
RBAC	4 rol, 43+ baimen	Sarbide kontrola

6.2 Autentifikazio Sistema

6.2.1 JWT Token Kudeaketa (Inplementazio Natiboa)

JWT (JSON Web Token) inplementazio osoa kanpo liburutegirik gabe garatu da — **Zero Trust** filosofia:

```php
// src/utils/TokenManager.php
class TokenManager
{
    private const ALGORITHM = 'HS256';

    /**
     * JWT Token sortu (HMAC-SHA256)
     * @param array $payload Token datuak
     * @param int   $expiry  Iraungitzea segundotan
     */
    public function createToken(array $payload, int $expiry): string
    {
        $header = $this->base64UrlEncode(json_encode([
            'typ' => 'JWT',
            'alg' => self::ALGORITHM
        ]));

        $payload['iat'] = time();
        $payload['exp'] = time() + $expiry;
        $payload['jti'] = bin2hex(random_bytes(16)); // Token ID bakarra

        $payloadEncoded = $this->base64UrlEncode(json_encode($payload));

        $signature = hash_hmac(
            'sha256',
            "$header.$payloadEncoded",
            $_ENV['JWT_SECRET'],
            true
        );

        return "$header.$payloadEncoded." . $this->base64UrlEncode($signature
    }

    /**
     * JWT Token baliozkotzea
     */
    public function validateToken(string $token): array
    {
        $parts = explode('.', $token);
```

```
        if (count($parts) !== 3) {
            throw new InvalidTokenException('Token formatua okerra');
        }

        [$header, $payload, $signature] = $parts;

        // Sinadura egiaztatu
        $expectedSignature = hash_hmac(
            'sha256',
            "$header.$payload",
            $_ENV['JWT_SECRET'],
            true
        );

        if (!hash_equals(
            $this->base64UrlEncode($expectedSignature),
            $signature
        )) {
            throw new InvalidTokenException('Sinadura ez da baliozkoa');
        }

        $data = json_decode($this->base64UrlDecode($payload), true);

        // Iraungitzea egiaztatu
        if ($data['exp'] < time()) {
            throw new TokenExpiredException('Tokena iraungita');
        }

        return $data;
    }
}
```

JWT konfigurazioa:

Parametroa	Balioa	Helburua
Access Token iraupena	1 ordu (3600s)	Saio-hasiera aktiboak
Refresh Token iraupena	7 egun (604800s)	Token berriztapena

Parametroa	Balioa	Helburua
Algoritmoa	HMAC-SHA256	Kriptografia sinadura
Token ID (jti)	32 byte ausazkoa	Refresh token jarraipena
Kodeketa	Base64Url (RFC 4648)	URL-safe kodeketa

6.2.2 TOTP MFA Inplementazioa

```
// src/utils/MfaManager.php
class MfaManager
{
    private const DIGITS  = 6;
    private const PERIOD  = 30;   // Segundoak
    private const ISSUER  = 'ZabalaGailetak_HR';

    /**
     * MFA konfigurazio berria sortu (TOTP)
     */
    public function setup(string $userEmail): array
    {
        $secret = $this->generateBase32Secret(20); // 160 bit ausazkoa

        $qrUrl = "otpauth://totp/" .
            urlencode(self::ISSUER . ':' . $userEmail) .
            "?secret=$secret" .
            "&issuer=" . urlencode(self::ISSUER) .
            "&digits=" . self::DIGITS .
            "&period=" . self::PERIOD;

        $backupCodes = $this->generateBackupCodes(10); // 10 kode berreskurat

        return [
            'secret'       => $secret,
            'qr_url'       => $qrUrl,
            'backup_codes' => $backupCodes
        ];
    }

    /**
     * TOTP kodea egiaztatu (denbora-leiho batekin)
     */
    public function verify(string $secret, string $code): bool
    {
        $timestamp = (int)(time() / self::PERIOD);

        // ±1 leiho onartu erlojuen desadostasunetarako
```

```
        for ($i = -1; $i <= 1; $i++) {
            $expected = $this->generateTotp($secret, $timestamp + $i);
            if (hash_equals($expected, $code)) {
                return true;
            }
        }
        return false;
    }
}
```

MFA fluxua:

```
1. POST /api/auth/login (email + pasahitza)
   └── JWT token mugatu bat bueltatu (mfa_required: true)

2. POST /api/auth/mfa/verify (6 digitu TOTP kodea)
   └── Access Token + Refresh Token bueltatu

3. Hurrengo eskaerak:
   └── Authorization: Bearer {access_token}
```

6.2.3 RBAC — Roleen Araberako Sarbide Kontrola

```
// src/middleware/RbacMiddleware.php
class RbacMiddleware
{
    public function check(string $userId, string $permission): bool
    {
        // Datu-basetik erabiltzailearen baimenak eskuratu (cache-rekin)
        $cacheKey = "user_permissions_{$userId}";
        $permissions = $this->redis->get($cacheKey);

        if (!$permissions) {
            $permissions = $this->db->fetchUserPermissions($userId);
            $this->redis->setex($cacheKey, 300, json_encode($permissions));
        }

        return in_array($permission, $permissions, true);
    }
}
```

Rol Hierarkia:

Rola	Baimen Kopurua	Adibideak
ADMIN	43+ (guztiak)	Sistema osoa kudeatu
HR_MANAGER	31	Langile guztiak, nominak, oporrak
DEPARTMENT_HEAD	15	Bere saileko langileak
EMPLOYEE	7	Profila ikusi, opor eskatu, nominak ikusi

Baimen adibideak:

```
employees.create     → Langile berria sortu
employees.read       → Langile datuak ikusi
employees.update     → Langile datuak editatu
employees.delete     → Langilea ezabatu
employees.restore    → Ezabatutakoa berrezarri
vacations.approve    → Opor eskaera onartu
payslips.view_all    → Nomina guztiak ikusi
audit.read           → Audit trail irakurri
```

6.3 API Endpoint-ak

6.3.1 Autentifikazio Endpoint-ak

Metodoa	Bide-izena	Autentifikazioa	Funtzioa
POST	`/api/auth/login`	Ez	Saio-hasiera (email + pasahitz)
POST		JWT (mfa_required)	TOTP egiaztapena
POST	`/api/auth/mfa/setup`	JWT	MFA konfiguratu
POST		JWT	MFA aktibatu
POST	`/api/auth/mfa/disable`	JWT + MFA	MFA desaktibatu
POST		Refresh Token	Access token berriztatu
GET	`/api/auth/me`	JWT	Uneko erabiltzaile datuak
POST		JWT	Saio-amaiera segurua

Login erantzuna:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "user": {
    "id": "uuid-langilea",
    "email": "langilea@zabalagailetak.com",
    "role": "hr_manager",
    "mfa_enabled": true
  }
}
```

6.3.2 Langileen Kudeaketa Endpoint-ak

Metodoa	Bide-izena	Baimenak	Funtzioa
GET	`/api/employees`	employees.read	Zerrenda (orriatuta, iragazita)
GET		employees.read	Xehetasunak
POST	`/api/employees`	employees.create	Langile berria
PUT		employees.update	Eguneratu
DELETE	`/api/employees/{id}`	employees.delete	Baja logikoa (soft delete)
POST		employees.restore	Berrezarri
GET	`/api/employees/{id}/history`	audit.read	Aldaketa historia
GET		audit.read	Erabiltzaile jarduera
GET	`/api/health`	Ez	Osasun egiaztapena

Orriaketa adibidea:

```
GET /api/employees?page=1&limit=10&sort=created_at&order=desc&department=IT
```

6.3.3 Opor Endpoint-ak

Metodoa	Bide-izena	Funtzioa
GET	`/api/vacations/balance`	Opor saldoa
POST		Opor eskaera sortu
GET	`/api/vacations/requests`	Eskaera zerrenda
PUT		Eskaera onartu (HR)
GET	`/api/vacations/calendar`	Opor egutegia

6.4 Sarrera Baliozkotzea — Espainiako Berariazko

Balidazioak

```
// src/validators/EmployeeValidator.php
class EmployeeValidator
{
    /**
     * NIF/NIE baliozkotzea — Letra algoritmoaren bidez
     */
    public function validateNif(string $nif): bool
    {
        $nif = strtoupper(trim($nif));
        $letters = 'TRWAGMYFPDXBNJZSQVHLCKE';

        // NIF: 8 zenbaki + letra
        if (preg_match('/^(\d{8})([A-Z])$/', $nif, $m)) {
            return $m[2] === $letters[$m[1] % 23];
        }

        // NIE: X/Y/Z + 7 zenbaki + letra
        if (preg_match('/^([XYZ])(\d{7})([A-Z])$/', $nif, $m)) {
            $prefix = ['X' => 0, 'Y' => 1, 'Z' => 2];
            $num    = $prefix[$m[1]] . $m[2];
            return $m[3] === $letters[$num % 23];
        }

        return false;
    }

    /**
     * IBAN baliozkotzea — Modulo 97 algoritmoa
     */
    public function validateIban(string $iban): bool
    {
        $iban = strtoupper(str_replace(' ', '', $iban));

        if (!preg_match('/^[A-Z]{2}\d{2}[A-Z0-9]{11,30}$/', $iban)) {
            return false;
        }
```

```
        // Lehen 4 karaktere bukaerara eraman
        $rearranged = substr($iban, 4) . substr($iban, 0, 4);

        // Letrak zenbakietan bihurtu (A=10, B=11... Z=35)
        $numericIban = preg_replace_callback('/[A-Z]/', function ($m) {
            return (ord($m[0]) - 55);
        }, $rearranged);

        return (bcmod($numericIban, '97') === '1');
    }
}
```

Balidazio taula osoa:

Eremua	Baliozkotzea	Adibidea
NIF/NIE	Letra algoritmo + patroiaren egiaztapena	12345678Z / X1234567L
IBAN	Modulo 97 checksum	ES9121000418450200051332
Telefonoa	Espainiako formatua (+34, 6xx/7xx/8xx/9xx)	+34 612 345 678
Posta Kodea	00000-52999 tartea (5 digitu)	48001
Emaila	RFC5322 baliozkotzea	langile@zabalagailetak.com
Pasahitza	8+ karaktere, maiuskulak, xeheak, zenbakia, berezia	Admin@2026!

XSS Saneamendua:

```
// Sarrera guztiak saneatu
public function sanitize(array $data): array
{
    return array_map(function ($value) {
        if (is_string($value)) {
            return htmlspecialchars(
                strip_tags(trim($value)),
                ENT_QUOTES | ENT_HTML5,
                'UTF-8'
            );
        }
        return $value;
    }, $data);
}
```

6.5 Segurtasun Neurriak

6.5.1 SQL Injekzio Prebentzioa

```
// ❌ INOIZ EZ hau erabili
$result = $pdo->query("SELECT * FROM employees WHERE id = '$id'");

// ✅ Beti prepared statements
$stmt = $pdo->prepare("SELECT * FROM employees WHERE id = :id AND deleted_at
$stmt->execute([':id' => $id]);
$employee = $stmt->fetch(PDO::FETCH_ASSOC);
```

6.5.2 CSRF Babesa — Double-Submit Cookie

```
// CSRF Token sortu eta egiaztatu
class CsrfProtection
{
    public function generateToken(): string
    {
        $token = bin2hex(random_bytes(32));
        $_SESSION['csrf_token'] = $token;
        setcookie('CSRF-TOKEN', $token, [
            'httponly' => false, // JavaScript-ek irakur dezan
            'samesite' => 'Strict',
            'secure'   => true
        ]);
        return $token;
    }

    public function validateToken(string $headerToken): bool
    {
        return hash_equals(
            $_SESSION['csrf_token'] ?? '',
            $headerToken
        );
    }
}

// Bezeroaren aldetik (JavaScript):
// fetch('/api/employees', {
//     headers: { 'X-CSRF-Token': getCookie('CSRF-TOKEN') }
// });
```

6.5.3 Pasahitz Kudeaketa

```
// Pasahitza gordetu (bcrypt, 12+ kosta faktorea)
$hash = password_hash($password, PASSWORD_BCRYPT, ['cost' => 12]);

// Pasahitza egiaztatu (denbora konstanteko konparaketa)
$valid = password_verify($password, $hash);

// Berrehashing-a (kostua aldatu ondoren)
if (password_needs_rehash($hash, PASSWORD_BCRYPT, ['cost' => 12])) {
    $newHash = password_hash($password, PASSWORD_BCRYPT, ['cost' => 12]);
    // Datu-basean eguneratu
}
```

6.5.4 Audit Trail

Operazio guztiak automatikoki erregistratzen dira:

```php
// src/utils/AuditLogger.php
class AuditLogger
{
    public function log(
        string $action,      // 'created', 'updated', 'deleted', 'restored'
        string $entityType,  // 'employee', 'user', 'vacation'
        string $entityId,
        string $userId,
        array  $changes = [] // Zer aldatu den (aurretik eta ondoren)
    ): void {
        $stmt = $this->db->prepare("
            INSERT INTO audit_logs
                (id, action, entity_type, entity_id, user_id, changes, ip_ad
            VALUES
                (:id, :action, :entity_type, :entity_id, :user_id, :changes,
        ");

        $stmt->execute([
            ':id'          => Uuid::generate(),
            ':action'      => $action,
            ':entity_type' => $entityType,
            ':entity_id'   => $entityId,
            ':user_id'     => $userId,
            ':changes'     => json_encode($changes),
            ':ip'          => $_SERVER['REMOTE_ADDR'],
            ':ua'          => $_SERVER['HTTP_USER_AGENT'] ?? ''
        ]);
    }
}
```

6.6 Datu-base Eskema

11 taula PostgreSQL:

```
-- Taula nagusiak
CREATE TABLE users (
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    email       VARCHAR(255) UNIQUE NOT NULL,
    password    VARCHAR(255) NOT NULL,              -- bcrypt hash
    role        VARCHAR(50) NOT NULL,
    mfa_secret  VARCHAR(255),                       -- TOTP sekretua (enkript
    mfa_enabled BOOLEAN DEFAULT FALSE,
    failed_logins INTEGER DEFAULT 0,
    locked_until  TIMESTAMP,
    created_at    TIMESTAMP DEFAULT NOW(),
    updated_at    TIMESTAMP DEFAULT NOW()
);

CREATE TABLE employees (
    id              UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    employee_number VARCHAR(20) UNIQUE NOT NULL,
    first_name      VARCHAR(100) NOT NULL,
    last_name       VARCHAR(100) NOT NULL,
    email           VARCHAR(255) UNIQUE NOT NULL,
    nif             VARCHAR(9),                     -- NIF/NIE
    iban            VARCHAR(34),                    -- IBAN (enkriptatuta)
    phone           VARCHAR(20),
    position        VARCHAR(100),
    department_id   UUID REFERENCES departments(id),
    hire_date       DATE,
    salary          DECIMAL(10,2),
    deleted_at      TIMESTAMP,                      -- Soft delete
    created_at      TIMESTAMP DEFAULT NOW(),
    updated_at      TIMESTAMP DEFAULT NOW()
);

CREATE TABLE audit_logs (
    id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    action      VARCHAR(50) NOT NULL,               -- created, updated, dele
    entity_type VARCHAR(100) NOT NULL,
    entity_id   UUID NOT NULL,
    user_id     UUID REFERENCES users(id),
    changes     JSONB,                              -- Aldaketa datuak JSON f
```

```
    ip_address  INET,
    user_agent  TEXT,
    created_at  TIMESTAMP DEFAULT NOW()
);

-- Indizeak errendimendu onerako
CREATE INDEX idx_employees_email ON employees(email);
CREATE INDEX idx_employees_department ON employees(department_id);
CREATE INDEX idx_audit_entity ON audit_logs(entity_type, entity_id);
CREATE INDEX idx_audit_user ON audit_logs(user_id);
CREATE INDEX idx_audit_created ON audit_logs(created_at DESC);
```

Taula zerrenda osoa:

Taula	Helburua
`users`	Autentifikazioa eta sarbide kontrola
	Langile erregistroak (NIF, IBAN, nomina)
`departments`	Sailen antolaketa
	Rol definizioak
`permissions`	Baimen definizioak
	RBAC mapaketa (n:m)
`audit_logs`	Ekintza erregistro osoa
	Opor eskaerak
`vacation_balances`	Urteko opor jarraipena
	Dokumentu metadata
`sessions`	JWT refresh token jarraipena

6.7 Test Emaitzak — 82/82 Gainditu

```
# PHPUnit testak exekutatu
./vendor/bin/phpunit --testdox

PHPUnit 10.5.0

TokenManagerTest
 ✓ Token sortu eta baliozkotzea (HMAC-SHA256)
 ✓ Iraungitako tokena detektatu
 ✓ Sinadura okerra detektatu
 ✓ Refresh token bakarra (jti)
 ✓ Base64Url kodeketa
 [11/11 gainditu]

EmployeeControllerTest
 ✓ Langile sortu (admin baimenekin)
 ✓ Langile sortu (baimen gabe → 403)
 ✓ Langile zerrenda orriatu
 ✓ Langile xehetasunak eskuratu
 ✓ Langile eguneratu + audit log
 ✓ Langile baja logikoa (soft delete)
 ✓ Langile berrezarri
 ✓ Langile historia eskuratu
 [11/11 gainditu]

EmployeeValidatorTest
 ✓ NIF baliozko: 12345678Z
 ✓ NIF baliogabe: 12345678A (letra okerra)
 ✓ NIE baliozko: X1234567L
 ✓ IBAN baliozko: ES9121000418450200051332
 ✓ IBAN baliogabe (checksum okerra)
 ✓ Telefono baliozko: +34 612 345 678
 ✓ Posta kodea baliozko: 48001
 ✓ Email baliozko: RFC5322
 ✓ XSS saneamendua
 [40/40 gainditu]

AuditLoggerTest
```

```
 ✓ CREATE ekintza erregistratu
 ✓ UPDATE ekintza + aldaketak gorde
 ✓ DELETE ekintza (soft) erregistratu
 ✓ RESTORE ekintza erregistratu
 [11/11 gainditu]

AuditControllerTest
 ✓ Entitate historia eskuratu
 ✓ Erabiltzaile jarduera eskuratu
 ✓ Baimenik gabe 403
 [9/9 gainditu]

Tests: 82/82 ✓ | Time: 1.23s | Memory: 32MB
```

Kalitate tresnak:

Tresna	Helburua	Emaitza
PHPUnit 10.5	Unit testak	82/82 ✅
PHPStan (8. maila)	Analisi estatikoa	0 akats
PHPCS PSR-12	Kode estandarra	✅
Composer Audit	Ahultasun egiaztapena	0 CVE

---

7. ANDROID APLIKAZIOA — KOTLIN / JETPACK COMPOSE

7.1 Konfigurazio Teknikoa

Ezaugarria	Balioa
Hizkuntza	Kotlin 2.0.21
UI Framework	Jetpack Compose (BOM 2024.12.01) + Material 3
Arkitektura	Clean Architecture + MVI
DI (Dependentziak)	Hilt (Dagger 2.54) + KSP
HTTP Bezeroa	Retrofit 2.11.0 + OkHttp 4.12.0
DB Lokala	Room 2.6.1 (offline cache)
Async	Coroutines 1.9.0 + Flow
Min SDK	24 (Android 7.0)
Target SDK	35 (Android 15)
JVM Target	21
Gradle	8.10.2 + AGP 8.7.3
Kode Prozesatzea	KSP 2.0.21-1.0.28 (KAPT ordezkatua)

Build URLak:

```
// app/build.gradle.kts
buildTypes {
    release {
        isMinifyEnabled = true
        isShrinkResources = true
        proguardFiles(getDefaultProguardFile("proguard-android-optimize.txt"
        buildConfigField("String", "API_BASE_URL",
            "\"https://zabala-gailetak.infinityfreeapp.com/api/\"")
    }
    debug {
        isMinifyEnabled = false
        buildConfigField("String", "API_BASE_URL",
            "\"http://10.0.2.2:8080/api/\"")   // Emuladorea → host
    }
}
```

7.2 Proiektuaren Egitura (Clean Architecture)

```
com/zabalagailetak/hrapp/
│
├── data/                              # Datu geruza
│   ├── api/                           # Retrofit API zerbitzuak
│   │   ├── AuthApiService.kt          # Login, MFA, Logout
│   │   ├── EmployeeApiService.kt      # Langile CRUD
│   │   ├── VacationApiService.kt      # Opor kudeaketa
│   │   ├── DocumentApiService.kt      # Dokumentuak
│   │   └── PayslipApiService.kt       # Nominak
│   ├── local/                         # Room datu-basea (offline cache)
│   │   ├── HrDatabase.kt
│   │   ├── EmployeeDao.kt
│   │   └── entities/
│   └── repository/                    # Repository inplementazioak
│
├── di/                                # Dependency Injection (Hilt)
│   ├── AppModule.kt
│   ├── NetworkModule.kt               # Retrofit + OkHttp + CertPinner
│   ├── DatabaseModule.kt              # Room DB
│   └── RepositoryModule.kt
│
├── domain/                            # Domeinu geruza (Negozio logika)
│   ├── model/                         # Domeinu ereduak (data classes)
│   │   ├── Employee.kt
│   │   ├── LoginResult.kt
│   │   └── VacationRequest.kt
│   ├── repository/                    # Repository interfazeak
│   └── usecase/                       # Erabilera kasuak
│       ├── LoginUseCase.kt
│       ├── GetEmployeesUseCase.kt
│       └── RequestVacationUseCase.kt
│
├── presentation/                      # UI geruza (Jetpack Compose)
│   ├── MainActivity.kt
│   ├── navigation/
│   │   └── AppNavigation.kt
│   ├── auth/
```

```
│   │   ├── LoginScreen.kt
│   │   └── AuthViewModel.kt          # MVI ViewModel
│   ├── dashboard/
│   │   └── DashboardScreen.kt
│   ├── vacation/
│   │   └── VacationScreen.kt
│   ├── documents/
│   │   └── DocumentsScreen.kt
│   ├── payslips/
│   │   └── PayslipsScreen.kt
│   └── ui/theme/
│       ├── Color.kt
│       ├── Typography.kt
│       └── Theme.kt                  # Material 3 gaia
│
├── security/                         # Segurtasun utilitateak
│   ├── KeystoreManager.kt            # Android Keystore gako kudeaketa
│   ├── CryptoHelper.kt               # AES-256-GCM enkriptatzea
│   └── SecureStorage.kt              # Datu-biltegi enkriptatua
│
└── HrApplication.kt                  # @HiltAndroidApp
```

7.3 Sare Modulua (NetworkModule.kt)

```
// di/NetworkModule.kt
@Module
@InstallIn(SingletonComponent::class)
object NetworkModule {

    @Provides
    @Singleton
    fun provideOkHttpClient(
        authInterceptor: AuthInterceptor
    ): OkHttpClient {
        return OkHttpClient.Builder()
            // === ZIURTAGIRI PINNING ===
            .certificatePinner(
                CertificatePinner.Builder()
                    .add(
                        "zabala-gailetak.infinityfreeapp.com",
                        "sha256/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="
                    )
                    .build()
            )
            // === JWT TOKEN GEHIKUNTZA ===
            .addInterceptor(authInterceptor)
            // === DENBORA MUGAK ===
            .connectTimeout(30, TimeUnit.SECONDS)
            .readTimeout(30, TimeUnit.SECONDS)
            .writeTimeout(30, TimeUnit.SECONDS)
            // === LOG-AK (debug soilik) ===
            .addInterceptor(
                HttpLoggingInterceptor().apply {
                    level = if (BuildConfig.DEBUG)
                        HttpLoggingInterceptor.Level.BODY
                    else
                        HttpLoggingInterceptor.Level.NONE
                }
            )
            .build()
    }
```

```
    @Provides
    @Singleton
    fun provideRetrofit(okHttpClient: OkHttpClient): Retrofit {
        return Retrofit.Builder()
            .baseUrl(BuildConfig.API_BASE_URL)
            .client(okHttpClient)
            .addConverterFactory(GsonConverterFactory.create())
            .build()
    }
}
```

7.4 Autentifikazio API Zerbitzua

```
// data/api/AuthApiService.kt
interface AuthApiService {

    @POST("auth/login")
    suspend fun login(
        @Body request: LoginRequest
    ): Response<LoginResponse>

    @POST("auth/mfa/verify")
    suspend fun verifyMfa(
        @Body request: MfaVerificationRequest
    ): Response<LoginResponse>

    @POST("auth/refresh")
    suspend fun refreshToken(
        @Body request: RefreshTokenRequest
    ): Response<TokenResponse>

    @GET("auth/me")
    suspend fun getCurrentUser(): Response<Employee>

    @POST("auth/logout")
    suspend fun logout(): Response<Unit>
}

// Datu ereduak
data class LoginRequest(
    val email: String,
    val password: String
)

data class MfaVerificationRequest(
    val temp_token: String,
    val mfa_code: String
)

data class LoginResponse(
```

```
    val access_token: String,
    val refresh_token: String,
    val token_type: String,
    val expires_in: Int,
    val mfa_required: Boolean = false,
    val temp_token: String? = null,
    val user: UserInfo? = null
)
```

7.5 Android Segurtasun Ezaugarriak

7.5.1 Android Keystore — Token Biltegiratzea

`security-crypto` (deprecatua) ordez **Android Keystore** eta **DataStore** erabiltzen da:

```
// security/KeystoreManager.kt
object KeystoreManager {
    private const val KEY_ALIAS = "zabala_hr_master_key"
    private const val KEYSTORE   = "AndroidKeyStore"

    fun getOrCreateKey(): SecretKey {
        val keyStore = KeyStore.getInstance(KEYSTORE).apply { load(null) }

        return if (keyStore.containsAlias(KEY_ALIAS)) {
            keyStore.getKey(KEY_ALIAS, null) as SecretKey
        } else {
            KeyGenerator.getInstance(
                KeyProperties.KEY_ALGORITHM_AES,
                KEYSTORE
            ).apply {
                init(
                    KeyGenParameterSpec.Builder(
                        KEY_ALIAS,
                        KeyProperties.PURPOSE_ENCRYPT or KeyProperties.PURPOS
                    )
                        .setBlockModes(KeyProperties.BLOCK_MODE_GCM)
                        .setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDI
                        .setKeySize(256)   // AES-256
                        .build()
                )
            }.generateKey()
        }
    }
}

// security/CryptoHelper.kt — AES-256-GCM enkriptatzea
object CryptoHelper {
    private const val TRANSFORMATION = "AES/GCM/NoPadding"

    fun encrypt(plaintext: String): EncryptedData {
        val cipher = Cipher.getInstance(TRANSFORMATION)
        cipher.init(Cipher.ENCRYPT_MODE, KeystoreManager.getOrCreateKey())

        return EncryptedData(
```

```
            ciphertext = Base64.encodeToString(
                cipher.doFinal(plaintext.toByteArray(Charsets.UTF_8)),
                Base64.NO_WRAP
            ),
            iv = Base64.encodeToString(cipher.iv, Base64.NO_WRAP)
        )
    }

    fun decrypt(encrypted: EncryptedData): String {
        val cipher = Cipher.getInstance(TRANSFORMATION)
        cipher.init(
            Cipher.DECRYPT_MODE,
            KeystoreManager.getOrCreateKey(),
            GCMParameterSpec(128, Base64.decode(encrypted.iv, Base64.NO_WRAP)
        )
        return String(
            cipher.doFinal(Base64.decode(encrypted.ciphertext, Base64.NO_WRAP
            Charsets.UTF_8
        )
    }
}

data class EncryptedData(val ciphertext: String, val iv: String)
```

7.5.2 Sare Segurtasun Konfigurazioa

```
<!-- res/xml/network_security_config.xml -->
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>

    <!-- Produkzioa: HTTPS soilik -->
    <domain-config cleartextTrafficPermitted="false">
        <domain includeSubdomains="true">zabalagailetak.com</domain>
        <domain includeSubdomains="true">hr.zabalagailetak.com</domain>
    </domain-config>

    <!-- Garapena soilik — localhost HTTP onartuta -->
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">localhost</domain>
        <domain includeSubdomains="true">10.0.2.2</domain>     <!-- Android e
        <domain includeSubdomains="true">zabala-gailetak.infinityfreeapp.com<
    </domain-config>

</network-security-config>
```

7.5.3 Segurtasun Laburpena

Neurria	Inplementazioa	Helburua
Ziurtagiri Pinning	OkHttp CertificatePinner	MITM eraso prebentzioa
Token Biltegiratzea	Android Keystore (AES-256-GCM)	Token enkriptatzea
Biometria	BiometricPrompt API	Hatz-marka autentifikazioa
Sare Segurtasuna	network_security_config.xml	HTTPS soilik produkzioan
Obfuskazioa	ProGuard/R8 (release build)	Kodearen alderantzizko ingeniaritza
Saio-kukua	DataStore + CryptoHelper	Datu sentikorren babesa
Log-ak	Debug soilik BODY level	Datu filtrazioa ekidin

Neurria	Inplementazioa	Helburua
Cleartext trafikoa	Debekatuta produkzioan	HTTP → HTTPS betearaztea

AndroidManifest.xml:

```
<manifest>
    <!-- Baimenak -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" /
    <uses-permission android:name="android.permission.USE_BIOMETRIC" />
    <uses-permission android:name="android.permission.CAMERA" />

    <application
        android:name=".HrApplication"
        android:allowBackup="false"
        android:networkSecurityConfig="@xml/network_security_config"
        android:debuggable="false">   <!-- Produkzioan beti FALSE -->

        <activity android:name=".presentation.MainActivity"
            android:exported="true" />
    </application>
</manifest>
```

7.6 Build eta Hedapena

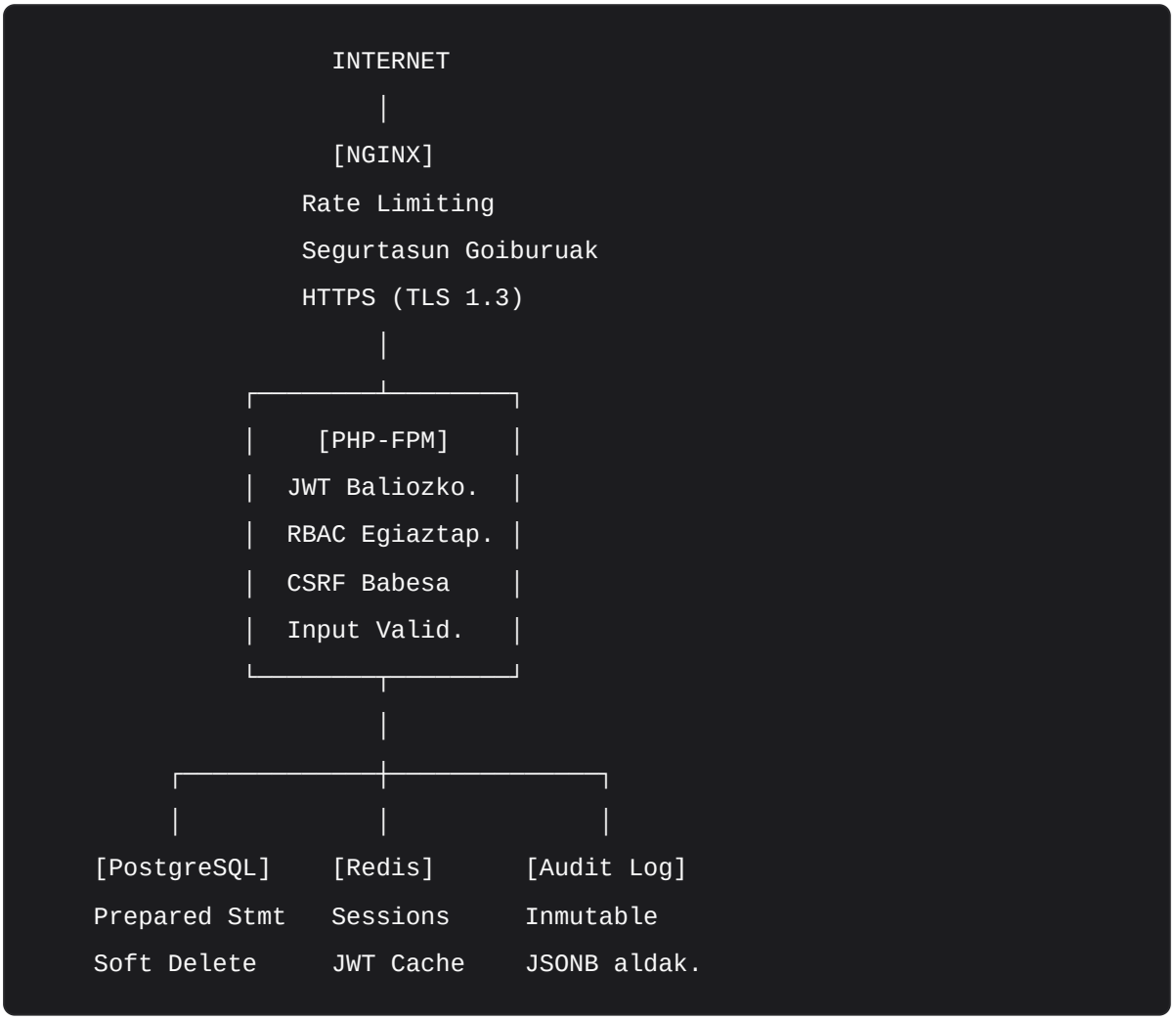
```
# === DEBUG BUILD (emuladorea / garapena) ===
./gradlew assembleDebug
# APK: app/build/outputs/apk/debug/app-debug.apk

# === RELEASE BUILD (produkzioa) ===
./gradlew assembleRelease
# APK: app/build/outputs/apk/release/app-release.apk

# === TESTAK ===
./gradlew test                    # Unit testak
./gradlew connectedAndroidTest    # Instrumentazio testak (gailuarekin)
./gradlew lint                    # Lint egiaztapena

# === GRADLE CACHE ARAZO ===
./gradlew --stop
rm -rf .gradle/configuration-cache
./gradlew clean :app:assembleDebug
```

KSP vs KAPT (migrazioa osatuta):

Alderdia	KAPT (zaharra)	KSP (berria)
Abiadura	Oinarria	%30-40 azkarragoa
Kotlin 2.0	Degradatua	Natiboa
Kode sorkuntzako bidea	`build/generated/source/kapt/`	`build/generated/ksp/`
Inkrementala	Mugatua	Ongi lagundua

8. EKOIZPEN SEGURTASUN KONTROLEN LABURPENA

8.1 Segurtasun Arkitektura Osoa



8.2 OWASP Top 10 Estaldura

#	Ahultasuna	Neurria	Egoera
A01	Sarbide Kontrol Hautsita	RBAC 4 rol + 43 baimen	✅
A02	Kriptografia Akatsa	bcrypt 12+, TLS 1.3, AES-256	✅
A03	Injekzioa	Prepared statements (PDO)	✅

#	Ahultasuna	Neurria	Egoera
A04	Diseinu Ez-segurua	SSDLC, Threat Modeling	✅
A05	Konfigurazio Okerra	server_tokens off, debug off	✅
A06	Osagai Ahulak	composer audit, eguneraketak	✅
A07	Identifikazio Akatsa	JWT + MFA/TOTP + Rate Limiting	✅
A08	Software Osotasuna	CI/CD + SAST + kode berrikusketa	✅
A09	Log/Monitorizazio Akatsa	Audit Trail + SIEM alertak	✅
A10	SSRF	fastcgi_param HTTP_PROXY ""	✅

8.3 DevSecOps Metrika Helburuak

KPI	Helburua	Maiztasuna	Neurtzen
Test estaldura	>%80	CI bakoitzean	PHPUnit
PHPStan akatsak	0	CI bakoitzean	PHPStan
Dependentzia CVE	0 kritikoak	Astero	composer audit
Build arrakasta	>%95	CI bakoitzean	GitHub Actions
API erantzun denbora	<200ms P95	Egunero	Nginx logs
Token iraungitzea	1 ordu	Etengabe	JWT payload
Audit log osotasuna	%100 ekintza	Egunero	DB kontsulta

8.4 Pentesting Ondorioz Hartutako Neurriak

Hacking Etiko moduluaren penetrazio proben ondorioz, honako neurriak aplikatu dira GG Atarian:

Ahultasuna Aurkitua	Neurria Aplikatua	Fasea
SSH root sarbidea	`PermitRootLogin no` + Gako autent.	Infra
Nginx server tokens	konfiguratu	Nginx

Ahultasuna Aurkitua	Neurria Aplikatua	Fasea
Rate limiting falta	login_limit (5r/m) + api_limit (10r/s)	Nginx
CSRF babesa falta	Double-submit cookie inplementatu	PHP
XSS ahultasuna	CSP goiburuak + htmlspecialchars()	PHP
SQL Injection arriskua	Prepared statements osoki inplementatua	PHP
Pasahitz testu laua	bcrypt 12+ errondarekin	PHP
JWT sinadura ahula	HMAC-SHA256 natiboa (Zero Trust)	PHP
Token biltegi ez-segurua	Android Keystore AES-256-GCM	Android
Cleartext trafikoa	network_security_config.xml	Android

8.5 Betetzea eta Ziurtagiria

Estandarra	Betetzea	Xehetasuna
OWASP ASVS L2	%90+	10/10 Top kategoria estalduta
ISO 27001 Anex. 8	%95	Garapen seguru kontrolak
POP-015 (Garapen Segurua)	%100	Fase guztiak inplementatuta
GDPR (Datu Minimizazioa)	%100	Soft delete, datu enkriptatzea
NIS2 (Software Segurtasuna)	%100	CI/CD, SAST, auditoria

ERANSKINA A — GG Atariko Erabiltzaile Froga

```
URL:   http://localhost:8080 (garapena)
       https://zabala-gailetak.infinityfreeapp.com (produkzioa)

Admin erabiltzailea:
  Email:    admin@zabalagailetak.com
  Pasahitza: Admin@2026!
  Rola:     ADMIN (43 baimen)

HR Manager erabiltzailea:
  Email:    hrmanager@zabalagailetak.com
  Pasahitza: Hr@2026Manager!
  Rola:     HR_MANAGER (31 baimen)

API Health Check:
  curl http://localhost:8080/api/health
```

ERANSKINA B — Android Aplikazioa Build Komandoak

```
# Android Studio: Koala (2024.1) edo berriagoa behar da
# JDK: 17 (edo 21)

# Garapen ingurunea
cd "Zabala Gailetak/android-app"
./gradlew assembleDebug

# Emuladorea hasi eta APK instalatu
./gradlew installDebug

# Produkzio APK (firmaketa behar du)
./gradlew assembleRelease

# Lint egiaztapena
./gradlew lint
# Txostena: app/build/reports/lint-results.html

# Post-migrazio egiaztapena (Kotlin 2.0 ondoren)
./post-migration-check.sh
```

ERANSKINA C — CI/CD Abiarazte Laburpena

```
# GitHub Actions pipeline-ak ikusi
gh workflow list

# Sintaxi egiaztapena eskuz exekutatu
gh workflow run ci-minimal.yml

# Produkziora hedatu (eskuzkoa)
gh workflow run deploy.yml -f confirm_deploy=bai

# Pipeline-en historia ikusi
gh run list --workflow=ci-minimal.yml
```

DOKUMENTUAREN AMAIERA

Zabala Gailetak S.L. *Gaileta eta txokolate fabrikazioan espezializatutako enpresa industriala*

Dokumentu hau Zabala Gailetak-en ER4 zibersegurtasun proiektuaren Modulua 2 — Ekoizpen Seguruan Jartzea (DevSecOps) txosten teknikoa da.

Neurri guztiak enpresaren aplikazioen segurtasuna, langile datuen babesa eta DevSecOps praktiken integrazioa ziurtatzeko diseinatuta daude.

2026ko Otsaila Bertsioa: 2.0 | Sailkapena: KONFIDENTZIALA