Archivo de Configuración de Entorno (.env)

Configuración Centralizada del Sistema IoT

🎯 Descripción General

Este archivo .env contiene todas las variables de configuración necesarias para el funcionamiento del ecosistema IoT. Organizado en secciones lógicas, permite una gestión centralizada y segura de la configuración across todos los servicios.

🚀 CONFIGURACIÓN DEL SERVIDOR NODE.JS

Entorno de Ejecución

env

NODE_ENV=development
PORT=4000

NODE_ENV: Define el ambiente como development para debugging y logs detallados

PORT: Puerto 4000 para el servidor backend (configuración personalizada)

Modo Desarrollo

env

DEV_MODE=true
DEV_USER_EMAIL=admin@localhost.com
DEV_USER_NAME=Administrador Local

DEV_MODE: Habilita autenticación simplificada sin Google OAuth

Credenciales de desarrollo: Usuario por defecto para testing y desarrollo

☁️ CONFIGURACIÓN CLOUDFLARE TUNNEL

Token de Túnel

env

CLOUDFLARE_TUNNEL_TOKEN=eyJhIjoiYzk3ZDU0ZjhiNzQyNWUwM2VkM2YzODk3YjlmNGJlODIiLCJ0IjoiZmVjMTIzNmQtYTJkOS00MTU5LTgzOWYtMDE5MDk2MjI5OTFkIiwicyI6Ik4yWTJOakk0TWpndE1UWmlPU>
# CLOUDFLARE_TUNNEL_TOKEN=tu_token_aqui

Túnel seguro: Exposición de servicios sin abrir puertos en el firewall

Token embebido: Configuración directa (alternativa menos segura)

Alternativa comentada: Para uso con variables de entorno

🔐 SISTEMA DE AUTENTICACIÓN JWT

Configuración de Tokens

env

JWT_SECRET=2fe600301bc1e4e129c5519d32bed987cb521c3281b5e40027e37fee64a
bd7ae706f56388a8f1c507889b09b9ef1de3822f232bc798b03680f7bb943849910fc
JWT_EXPIRES_IN=15m
REFRESH_EXPIRES_IN=7d

JWT_SECRET: Clave criptográfica de 64 bytes (256 bits) generada seguramente

JWT_EXPIRES_IN: Expiración corta de 15 minutos para tokens de acceso

REFRESH_EXPIRES_IN: Expiración larga de 7 días para tokens de refresco

🔗 INTEGRACIÓN GOOGLE OAUTH2

Identidad de Google

env

GOOGLE_CLIENT_ID=113014965393-
9t3h4eg2jr4aj7mfs4q78kkajln16m79.apps.googleusercontent.com

Client ID: Identificador único de la aplicación en Google Cloud Console

OAuth2 Flow: Autenticación social para usuarios

🌐 CONFIGURACIÓN CORS (CROSS-ORIGIN RESOURCE SHARING)

Orígenes Permitidos

env

CORS_ORIGIN=http://127.0.0.1:8080,http://localhost:8080,https://iot-
opalo.work,http://app.ispciot.org

Desarrollo local: localhost y 127.0.0.1 en puerto 8080

Producción: Dominios públicos de la aplicación

Protocolos mixtos: HTTP/HTTPS para transición gradual a SSL

👥 SISTEMA DE ROLES Y AUTORIZACIÓN

Listas de Control de Acceso

env

```
ADMIN_WHITELIST=vittodutti@gmail.com
ACTION_WHITELIST=
```

ADMIN_WHITELIST: Usuarios con rol de administrador (acceso completo)

ACTION_WHITELIST: Usuarios con permisos de acción (operaciones específicas)

Formato: Emails separados por comas, sin espacios

📡 CONFIGURACIÓN MQTT PARA IOT

Broker MQTT

env

```
MQTT_BROKER_HOST=mqtt.ispciot.org
MQTT_BROKER_PORT=80
MQTT_BROKER_USERNAME=
MQTT_BROKER_PASSWORD=
MQTT_TOPICS=sensors/temperature,sensors/humidity
```

Host público: Broker accesible externamente via subdominio

Puerto 80: Para evitar bloqueos de firewall corporativos

Topics por defecto: Sensores de temperatura y humedad

🗄️ CONFIGURACIÓN DE BASES DE DATOS

MariaDB/MySQL

env

```
MYSQL_HOST=172.18.0.4
MYSQL_ROOT_PASSWORD=root@siloiot2025
MYSQL_DATABASE=silo_db
```

MYSQL_USER=silo_user
MYSQL_PASSWORD=user@siloiot2015

IP estática: 172.18.0.4 en red Docker bridge

Credenciales seguras: Contraseñas complejas con fechas de referencia

Usuario dedicado: silo_user con permisos específicos de aplicación

InfluxDB para Series Temporales

env

INFLUXDB_DB=metricas_silo
INFLUXDB_ADMIN_USER=admin
INFLUXDB_ADMIN_PASSWORD=influx@siloiot2025
INFLUXDB_USER=telegraf_user
INFLUXDB_USER_PASSWORD=telegraf@siloiot2025

Base de métricas: metricas_silo para datos de sensores

Separación de usuarios: Admin vs Telegraf para seguridad

Contraseñas específicas: Diferentes por rol de usuario

📊 SISTEMA DE VISUALIZACIÓN GRAFANA

Acceso Administrativo

env

GRAFANA_ADMIN_USER=admin
GRAFANA_ADMIN_PASSWORD=grafana@siloiot2025

Credenciales default: Usuario admin con contraseña segura

Dashboards: Para visualización de métricas IoT y monitoreo del sistema

💾 CONFIGURACIÓN DE BACKUP AUTOMÁTICO

Políticas de Respaldo

env

BACKUP_ENABLED=true
BACKUP_SCHEDULE=0 2 * * *
BACKUP_RETENTION_DAYS=30
BACKUP_PATH=/backups

Automático: Sistema de backup activado

Horario: 2:00 AM diariamente (formato cron: minuto hora dia mes dia_semana)

Retención: 30 días de backups históricos

Ruta: Directorio de almacenamiento dentro del contenedor

📧 SISTEMA DE NOTIFICACIONES

Configuración SMTP para Email

env

```
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=notifications@tudominio.com
SMTP_PASS=your-smtp-password
```

Servidor: Gmail como proveedor SMTP

Puerto TLS: 587 para conexiones seguras

Cuenta dedicada: Para notificaciones del sistema (placeholder)

Webhooks para Integraciones

env

```
WEBHOOK_URL=https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK
```

Slack Integration: URL para notificaciones en tiempo real

Placeholder: Configuración a reemplazar con valores reales

📈 SISTEMA DE MONITOREO Y LOGS

Configuración de Logging

env

```
LOG_LEVEL=info
LOG_FILE=/var/log/iot-app.log
```

Nivel info: Balance óptimo entre detalle y rendimiento

Archivo centralizado: Todos los logs en ubicación específica

Métricas del Sistema

env

METRICS_ENABLED=true
METRICS_PORT=9090

Métricas activadas: Recolección de datos de performance

Puerto específico: 9090 para scraping de métricas por Prometheus

🌍 CONFIGURACIÓN DE PRODUCCIÓN

Dominios y SSL

env

DOMAIN=app.ispciot.org
SSL_EMAIL=vittodutti@gmail.com

Dominio principal: app.ispciot.org para acceso público

Email SSL: Para certificados Let's Encrypt y notificaciones de renovación

Reverse Proxy

env

REVERSE_PROXY_HOST=nginx-proxy-manager
REVERSE_PROXY_PORT=80

Proxy manager: Servicio centralizado de reverse proxy

Puerto interno: 80 para comunicación entre contenedores

⚠️ CONSIDERACIONES DE SEGURIDAD

Variables Críticas a Proteger

JWT_SECRET: Debe ser único por ambiente y mantenerse secreto

Contraseñas de BD: Diferentes para cada usuario y servicio

Tokens externos: Cloudflare token debe protegerse adecuadamente

Configuración por Ambiente

Desarrollo: DEV_MODE=true, logs detallados, autenticación simplificada

Producción: DEV_MODE=false, configuración optimizada, SSL obligatorio

Mejores Prácticas Implementadas

Contraseñas complejas: Combinación de palabras, símbolos y años

Separación de usuarios: Diferentes credenciales por servicio

Tokens seguros: JWT secret generado criptográficamente

CORS restrictivo: Orígenes explícitamente definidos

🛠️ NOTAS DE IMPLEMENTACIÓN

Valores por Defecto vs Personalizados

Topics MQTT: Configuración básica para empezar (extensible)

Credenciales: Placeholders para reemplazar con valores reales

IPs de red: Específicas del despliegue Docker (pueden cambiar)

Variables Opcionales vs Requeridas

Cloudflare: Opcional para exposición pública

Notificaciones: Configurables según necesidades del proyecto

Backup: Activado por defecto para protección de datos críticos