

Variables de Entorno - Referencia Rápida

Esta guía te permite copiar y pegar rápidamente todas las variables de entorno necesarias para el despliegue.

Índice

-
1. [Railway \(Backend\)](#)
 2. [Vercel \(Frontend\)](#)
 3. [Generar Nuevos Secrets](#)
 4. [Verificación](#)
-

Railway (Backend)

Copiar y Pegar en Railway

Ve a **Railway Dashboard** → **Tu Servicio** → **Variables** → **Raw Editor** y pega:

```

# =====
# RAILWAY BACKEND - VARIABLES DE ENTORNO
# =====

# DATABASE CONNECTION
# Abacus.AI PostgreSQL - Production
DATABASE_URL=postgresql://role_f804661bc:PK43sEVjwsuP1X1ArUp09gUYk4l4Ej48@db-f804661bc
.db002.hosteddb.reai.io:5432/f804661bc?connect_timeout=15

# JWT CONFIGURATION
# Access Token Secret (15 minutos)
JWT_SECRET=prod-secret-c8a9f3e2d1b4a5c7f9e8d2b1a4c6f8e9d2b5a7c9f1e3d5b7a9c1f3e5d7b9a1c
3

# Refresh Token Secret (7 días)
JWT_REFRESH_SECRET=prod-refresh-secret-a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x
4y5z6

# Token Expiration Times
JWT_EXPIRATION=15m
JWT_REFRESH_EXPIRATION=7d

# NEXTAUTH SECRET
# For NextAuth.js compatibility
NEXTAUTH_SECRET=qr6zNaFiLtKpLB6QjrPVPx4nEMbZFKeB

# ENCRYPTION
# Base64-encoded encryption key for sensitive data
ENCRYPTION_KEY=KN6X9ososWFJvm8WvNy2r7rXr3+B2XKc0oQTw0tV6qc=

# ENVIRONMENT CONFIGURATION
NODE_ENV=production
PORT=3001

# CORS ORIGINS
# ⚠ IMPORTANTE: Actualiza esto después de desplegar el frontend en Vercel
CORS_ORIGIN=https://admin.transparenciaciudadana.com

# RATE LIMITING (Throttle)
THROTTLE_TTL=60
THROTTLE_LIMIT=100

# LOGGING
LOG_LEVEL=info
LOG_FILE_ENABLED=false
LOG_FILE_PATH=./logs/app.log

# API DOCUMENTATION
SWAGGER_ENABLED=false
SWAGGER_PATH=api/docs

# DEFAULT SETTINGS
DEFAULT_MUNICIPALITY_NAME=Santiago
DEFAULT_MUNICIPALITY_COUNTRY=Chile
DEFAULT_MUNICIPALITY_REGION=Metropolitana
DEFAULT_CURRENCY=CLP
DEFAULT_TIMEZONE=America/Santiago
DEFAULT_LOCALE=es-CL

# SEED DATA (solo para desarrollo/testing)
SEED_ADMIN_EMAIL=admin@santiago.cl

```

```
SEED_ADMIN_PASSWORD=demo123
SEED_FISCAL_YEARS=2023,2025
```

⚙️ Configuración del Servicio en Railway

Además de las variables, configura:

Settings → Service

- **Root Directory:** apps/api
- **Build Command:** npm install && npm run build
- **Start Command:** npm run start:prod

Settings → Networking

- Genera un dominio público (ej: your-app.railway.app)
- **Guarda este URL** para usarlo en Vercel

Vercel (Frontend)

🚀 Copiar y Pegar en Vercel

Ve a **Vercel Dashboard → Tu Proyecto → Settings → Environment Variables** y agrega estas variables **una por una**:

```
# =====
# VERCEL FRONTEND - VARIABLES DE ENTORNO
# =====

# API BACKEND URL
# ⚠️ REEMPLAZA con tu URL de Railway
NEXT_PUBLIC_API_URL=https://your-app.railway.app

# DATABASE CONNECTION
# Mismo DATABASE_URL que el backend (para API routes internas)
DATABASE_URL=postgresql://role_f804661bc:PK43sEVjwsuP1X1ArUp09gUYk4l4Ej48@db-f804661bc
.db002.hosteddb.reai.io:5432/f804661bc?connect_timeout=15

# NEXTAUTH SECRET
# Debe ser el mismo que en Railway
NEXTAUTH_SECRET=qr6zNaFiLtKpLB6QjrPVPx4nEMbZFKeB

# ENVIRONMENT
NODE_ENV=production

# AWS STORAGE (opcional - si usas Abacus.AI storage)
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=abacusai-apps-19658dee6a168bd21a949e89-us-west-2
AWS_FOLDER_PREFIX=5986/

# ABACUS.AI API KEY (opcional - para funcionalidades avanzadas)
ABACUSAII_API_KEY=3e9aa270cc3642949a49f26cf746467b
```

⚙️ Configuración del Proyecto en Vercel

Project Settings

- **Framework Preset:** Next.js (auto-detectado)

- **Root Directory:** frontend/nextjs_space
- **Build Command:** npm run build
- **Output Directory:** .next
- **Install Command:** npm install

Domains

- Agrega: admin.transparenciaciudadana.com
 - Configura DNS según instrucciones de Vercel
-

Generar Nuevos Secrets

Si necesitas generar nuevos secrets para JWT o NEXTAUTH:

En Node.js

```
// Generar JWT_SECRET y JWT_REFRESH_SECRET
const crypto = require('crypto');

// 64 bytes = 512 bits
const jwtSecret = crypto.randomBytes(64).toString('hex');
console.log('JWT_SECRET:', jwtSecret);

const jwtRefreshSecret = crypto.randomBytes(64).toString('hex');
console.log('JWT_REFRESH_SECRET:', jwtRefreshSecret);

// NEXTAUTH_SECRET (32 bytes = 256 bits)
const nextAuthSecret = crypto.randomBytes(32).toString('base64');
console.log('NEXTAUTH_SECRET:', nextAuthSecret);

// ENCRYPTION_KEY (32 bytes = 256 bits para AES-256)
const encryptionKey = crypto.randomBytes(32).toString('base64');
console.log('ENCRYPTION_KEY:', encryptionKey);
```

En Bash/Terminal

```
# JWT_SECRET (64 bytes)
openssl rand -hex 64

# JWT_REFRESH_SECRET (64 bytes)
openssl rand -hex 64

# NEXTAUTH_SECRET (32 bytes base64)
openssl rand -base64 32

# ENCRYPTION_KEY (32 bytes base64)
openssl rand -base64 32
```

En Python

```

import secrets
import base64

# JWT_SECRET (64 bytes)
jwt_secret = secrets.token_hex(64)
print(f"JWT_SECRET: {jwt_secret}")

# JWT_REFRESH_SECRET (64 bytes)
jwt_refresh_secret = secrets.token_hex(64)
print(f"JWT_REFRESH_SECRET: {jwt_refresh_secret}")

# NEXTAUTH_SECRET (32 bytes base64)
nextauth_secret = base64.b64encode(secrets.token_bytes(32)).decode()
print(f"NEXTAUTH_SECRET: {nextauth_secret}")

# ENCRYPTION_KEY (32 bytes base64)
encryption_key = base64.b64encode(secrets.token_bytes(32)).decode()
print(f"ENCRYPTION_KEY: {encryption_key}")

```

Verificación

Checklist de Variables

Railway (Backend) - 22 variables

- [] DATABASE_URL
- [] JWT_SECRET
- [] JWT_REFRESH_SECRET
- [] JWT_EXPIRATION
- [] JWT_REFRESH_EXPIRATION
- [] NEXTAUTH_SECRET
- [] ENCRYPTION_KEY
- [] NODE_ENV
- [] PORT
- [] CORS_ORIGIN
- [] THROTTLE_TTL
- [] THROTTLE_LIMIT
- [] LOG_LEVEL
- [] LOG_FILE_ENABLED
- [] LOG_FILE_PATH
- [] SWAGGER_ENABLED
- [] SWAGGER_PATH
- [] DEFAULT_MUNICIPALITY_NAME
- [] DEFAULT_MUNICIPALITY_COUNTRY
- [] DEFAULT_MUNICIPALITY_REGION
- [] DEFAULT_CURRENCY
- [] DEFAULT_TIMEZONE

Vercel (Frontend) - 8 variables

- [] `NEXT_PUBLIC_API_URL` ⚠ Debe apuntar a Railway
- [] `DATABASE_URL`
- [] `NEXTAUTH_SECRET`
- [] `NODE_ENV`
- [] `AWS_PROFILE` (opcional)
- [] `AWS_REGION` (opcional)
- [] `AWS_BUCKET_NAME` (opcional)
- [] `AWS_FOLDER_PREFIX` (opcional)

💡 Probar Conexiones

1. Test Backend Health

```
curl https://your-app.railway.app/api/health
```

Respuesta esperada:

```
{
  "status": "ok",
  "timestamp": "2025-11-26T...",
  "uptime": 123.456,
  "environment": "production"
}
```

2. Test Frontend

```
curl https://admin.transparenciaciudadana.com
```

Debería devolver HTML de la aplicación Next.js

3. Test Database Connection

Crea un script temporal `test-db.js`:

```
const { PrismaClient } = require('@prisma/client');
require('dotenv').config();

const prisma = new PrismaClient();

async function test() {
  try {
    const users = await prisma.user.count();
    console.log('✅ Database connected! Users:', users);
  } catch (error) {
    console.error('❌ Database error:', error.message);
  } finally {
    await prisma.$disconnect();
  }
}

test();
```

Ejecuta:

```
node test-db.js
```

4. Test CORS

```
curl -X OPTIONS https://your-app.railway.app/api/health \
-H "Origin: https://admin.transparenciaciudadana.com" \
-H "Access-Control-Request-Method: GET" \
-v
```

Verifica que los headers de CORS estén presentes en la respuesta

5. Test Login

```
curl -X POST https://your-app.railway.app/api/auth/login \
-H "Content-Type: application/json" \
-d '{
  "email": "superadmin@transparencia.cl",
  "password": "demo12345"
}'
```

Respuesta esperada:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": "...",
    "email": "superadmin@transparencia.cl",
    "role": "super_admin"
  }
}
```



Actualizar Variables

Railway

1. Ve a **Variables** en el dashboard
2. Haz los cambios necesarios
3. Railway redesplegará automáticamente (2-3 minutos)

Vercel

1. Ve a **Settings → Environment Variables**
2. Haz los cambios necesarios
3. **IMPORTANTE:** Las variables nuevas/actualizadas solo se aplican en nuevos deployments
4. Ve a **Deployments → último deployment → Redeploy**

Notas Importantes

Seguridad

1. **NUNCA** compartas estos secrets en:

- Repositorios públicos
- Slack/Discord/Teams públicos
- Screenshots o videos

2. **Rota los secrets** cada 90 días:

- Genera nuevos valores
- Actualiza en Railway y Vercel
- Redesplegar ambas aplicaciones

3. **Usa diferentes secrets** para cada entorno:

- Development: `.env.development`
- Staging: `.env.staging`
- Production: `.env.production`

CORS

El `CORS_ORIGIN` en Railway debe incluir:

- El dominio de Vercel (ej: `your-app.vercel.app`)
- El dominio personalizado (ej: `admin.transparenciaciudadana.com`)
- Localhost para desarrollo local (opcional en producción)

Formato:

```
CORS_ORIGIN=https://your-app.vercel.app,https://admin.transparenciaciudadana.com
```

DATABASE_URL

- Asegúrate que incluya `?connect_timeout=15` al final
- Verifica que el usuario tenga permisos suficientes
- Si cambias la contraseña de la DB, actualiza en AMBOS lugares (Railway y Vercel)

Soporte

Si tienes problemas:

1. **Verifica los logs:**

- Railway: Dashboard → Deployments → View Logs
- Vercel: Dashboard → Deployments → View Function Logs

2. **Verifica las variables:**

```
```bash
En Railway (usando Railway CLI)
railway variables
```

```
En Vercel (usando Vercel CLI)
```

```
vercel env ls
```

```
```
```

1. **Consulta la guía completa:** `DEPLOYMENT_GUIDE.md`
-

Creado por: Municipal Transparency Platform Team

Última actualización: 26 de noviembre de 2025

Versión: 1.0.0