

Ejemplos de Variables de Entorno

Archivo `.env.example`

Crea un archivo `.env.example` en `nextjs_space/` con este contenido:

```

# =====
# DATABASE
# =====

# Supabase PostgreSQL Connection String
# Get from: Supabase Dashboard → Settings → Database → Connection string
DATABASE_URL="postgresql://postgres.[PROJECT-REF]:[YOUR-PASSWORD]@aws-0-[RE-
GION].pooler.supabase.com:5432/postgres"

# =====
# NEXTAUTH
# =====

# Generate with: openssl rand -base64 32
NEXTAUTH_SECRET="tu-secret-key-super-segura-de-al-menos-32-caracteres"

# Development
NEXTAUTH_URL="http://localhost:3000"

# Production (actualiza con tu dominio)
# NEXTAUTH_URL="https://tu-dominio.com"

# =====
# AWS S3
# =====

# AWS Configuration
AWS_PROFILE="hosted_storage"
AWS_REGION="us-west-2"
AWS_BUCKET_NAME="tu-bucket-name"
AWS_FOLDER_PREFIX="tu-carpeta/"

# Optional: Si no usas perfil de AWS
# AWS_ACCESS_KEY_ID="AKIA..."
# AWS_SECRET_ACCESS_KEY="tu-secret-key"

# =====
# STRIPE
# =====

# Get from: https://dashboard.stripe.com/apikeys

# Test Keys (para desarrollo)
STRIPE_SECRET_KEY="sk_test_51..."
STRIPE_PUBLISHABLE_KEY="pk_test_51..."
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY="pk_test_51..."

# Production Keys (para producción)
# STRIPE_SECRET_KEY="sk_live_51..."
# STRIPE_PUBLISHABLE_KEY="pk_live_51..."
# NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY="pk_live_51..."

# Webhook Secret (get from: Stripe Dashboard → Developers → Webhooks)
STRIPE_WEBHOOK_SECRET="whsec_..."

# =====
# PUSH NOTIFICATIONS (VAPID)
# =====

# Generate with: npx web-push generate-vapid-keys
NEXT_PUBLIC_VAPID_PUBLIC_KEY="BE162iUYgUivxIkv69yViEuiBIa-Ib27SzV9p3F-Jq-6-kxq9R-
wD9qdL4U3JfYxSh_Vu_WG2cEg8u7kJ7-vQTmE"

```

```

VAPID_PRIVATE_KEY="p-K-PxeghWxVyGxvxHYVsT3xhp5fKWvUqNfNqN-J4XM"

# =====
# ABACUS AI
# =====

ABACUSAI_API_KEY="tu-api-key-de-abacus-ai"

# =====
# VIDEO
# =====

NEXT_PUBLIC_VIDEO_URL="https://www.youtube.com/embed/VIDEO_ID"

# =====
# DOCUSIGN (OPCIONAL)
# =====

DOCUSIGN_ACCOUNT_ID="tu-account-id-aqui"
DOCUSIGN_PRIVATE_KEY="-----BEGIN RSA PRIVATE KEY-----\n...\n-----END RSA PRIVATE KEY-----"
DOCUSIGN_BASE_PATH="https://demo.docusign.net/restapi"

# =====
# REDSYS (OPCIONAL - Para integración bancaria)
# =====

REDSYS_API_URL="https://apis-i.redsys.es:20443/psd2/xs2a/api-entrada-xs2a/services"
REDSYS_OAUTH_URL="https://apis-i.redsys.es:20443/psd2/xs2a/api-oauth-xs2a"
REDSYS_BANKINTER_CODE="bankinter"
REDSYS_CLIENT_ID="your_client_id_here"
REDSYS_CLIENT_SECRET="your_client_secret_here"
REDSYS_CERTIFICATE_PATH="/path/to/qwac_certificate.pem"
REDSYS_CERTIFICATE_KEY_PATH="/path/to/qwac_private_key.pem"
REDSYS_SEAL_CERTIFICATE_PATH="/path/to/qseal_certificate.pem"
REDSYS_SEAL_KEY_PATH="/path/to/qseal_private_key.pem"

# =====
# SECURITY
# =====

# Generate with: openssl rand -hex 32
CRON_SECRET="tu-cron-secret-key-aqui"
ENCRYPTION_KEY="tu-encryption-key-de-64-caracteres-hexadecimales-aqui"

# =====
# EMAIL (OPCIONAL)
# =====

# SMTP Configuration
# EMAIL_SERVER_HOST="smtp.gmail.com"
# EMAIL_SERVER_PORT="587"
# EMAIL_SERVER_USER="tu-email@gmail.com"
# EMAIL_SERVER_PASSWORD="tu-app-password"
# EMAIL_FROM="noreply@inmova.com"

# =====
# SMS (OPCIONAL)
# =====

# Twilio Configuration
# TWILIO_ACCOUNT_SID="AC..."
# TWILIO_AUTH_TOKEN="..."

```

```
# TWILIO_PHONE_NUMBER="+1234567890"

# =====
# MONITORING (OPCIONAL)
# =====

# Sentry
# NEXT_PUBLIC_SENTRY_DSN="https://...@sentry.io/..."
# SENTRY_AUTH_TOKEN="..."

# Google Analytics
# NEXT_PUBLIC_GA_ID="G-..."
```

Cómo Obtener las Credenciales

1. Supabase (Database)

1. Ve a <https://supabase.com/dashboard> (<https://supabase.com/dashboard>)
2. Crea un nuevo proyecto o selecciona uno existente
3. Ve a **Settings** → **Database**
4. En “Connection string” selecciona “URI”
5. Copia la cadena de conexión
6. Reemplaza `[YOUR-PASSWORD]` con tu contraseña

Ejemplo:

```
postgresql://postgres.abcdefgh:MiPassword123@aws-0-us-west-1.pooler.supabase.com:5432/
postgres
```

2. NextAuth Secret

Generar localmente:

```
openssl rand -base64 32
```

O usa este generador online: generate-secret.vercel.app (<https://generate-secret.vercel.app/32>)

3. AWS S3

Opción A: Crear Bucket en AWS

1. Ve a [AWS S3 Console](https://s3.console.aws.amazon.com/s3/) (<https://s3.console.aws.amazon.com/s3/>)
2. Crea un nuevo bucket
3. Configura permisos públicos si necesitas archivos públicos
4. Crea un usuario IAM con permisos S3
5. Genera Access Keys

Opción B: Usar Servicio Compatible con S3

Puedes usar servicios como:

- **DigitalOcean Spaces**
- **Wasabi**

- **Cloudflare R2**
- **MinIO** (self-hosted)

Todos son compatibles con la API de S3.

4. Stripe

1. Ve a <https://dashboard.stripe.com/register> (<https://dashboard.stripe.com/register>)
2. Crea una cuenta o inicia sesión
3. Ve a **Developers** → **API keys**
4. Copia las claves de prueba (test) o producción (live)

Para Webhooks:

1. Ve a **Developers** → **Webhooks**
2. Click en "Add endpoint"
3. URL: `https://tu-dominio.com/api/webhooks/stripe`
4. Eventos a escuchar:
 - `checkout.session.completed`
 - `payment_intent.succeeded`
 - `payment_intent.payment_failed`
 - `customer.subscription.created`
 - `customer.subscription.updated`
 - `customer.subscription.deleted`
5. Copia el "Signing secret"

5. Push Notifications (VAPID)

Generar VAPID Keys:

```
cd nextjs_space
npx web-push generate-vapid-keys
```

Esto generará:

```
Public Key: BEl62i...
Private Key: p-K-Px...
```

6. Abacus AI

1. Ve a <https://abacus.ai> (<https://abacus.ai>)
2. Crea una cuenta
3. Ve a API Keys
4. Genera una nueva API Key



Configurar Variables en Vercel

Método 1: Por la UI

1. Ve a tu proyecto en Vercel
2. Settings → Environment Variables

3. Agrega cada variable:

- **Key:** Nombre de la variable
- **Value:** Valor de la variable
- **Environments:** Production, Preview, Development

Método 2: Con Vercel CLI

```
# Instalar Vercel CLI
npm i -g vercel

# Login
vercel login

# Link al proyecto
vercel link

# Agregar variables una por una
vercel env add DATABASE_URL production
vercel env add NEXTAUTH_SECRET production
# ... etc

# O importar desde .env
vercel env pull .env.local
```

Método 3: Archivo JSON (Masivo)

1. Crea un archivo `env.json` :

```
{
  "DATABASE_URL": {
    "type": "plain",
    "value": "postgresql://...",
    "target": ["production", "preview", "development"]
  },
  "NEXTAUTH_SECRET": {
    "type": "plain",
    "value": "tu-secret-aqui",
    "target": ["production", "preview", "development"]
  }
}
```

1. Importar:

```
vercel env import env.json
```

Seguridad

NUNCA hacer:

- ✗ Subir `.env` a GitHub
- ✗ Compartir las keys en Slack/Discord
- ✗ Usar las mismas keys en desarrollo y producción
- ✗ Hardcodear keys en el código
- ✗ Exponer keys en el frontend (excepto las `NEXT_PUBLIC_*`)

SÍ hacer:

- ✓ Usar `.env.local` para desarrollo
- ✓ Usar diferentes keys para test y producción
- ✓ Rotar keys regularmente
- ✓ Usar servicios de secretos (Vercel Env, AWS Secrets Manager)
- ✓ Agregar `.env` al `.gitignore`

Verificar Seguridad

```
# Verificar que .env no está trackeado
git ls-files | grep .env

# Si aparece, removerlo:
git rm --cached .env
git commit -m "Remove .env from tracking"

# Verificar en GitHub
# No debería haber archivos .env en el repositorio
```



Prioridad de Variables

Las variables se cargan en este orden (las últimas sobrescriben las anteriores):

1. `.env`
2. `.env.local`
3. `.env.development` (en dev)
4. `.env.production` (en prod)
5. Variables del sistema (Vercel, etc.)



Debugging Variables

Verificar localmente:

```
cd nextjs_space
node scripts/check-env.js
```

Verificar en Vercel:

```
# Ver todas las variables
vercel env ls

# Ver una específica
vercel env pull .env.vercel
cat .env.vercel | grep DATABASE_URL
```

Verificar en runtime (solo desarrollo):

Crea un archivo temporal `nextjs_space/app/api/debug/env/route.ts` :

```
import { NextResponse } from 'next/server';

export async function GET() {
  // SOLO PARA DESARROLLO - NUNCA EN PRODUCCIÓN
  if (process.env.NODE_ENV !== 'development') {
    return NextResponse.json({ error: 'Not allowed' }, { status: 403 });
  }

  return NextResponse.json({
    NODE_ENV: process.env.NODE_ENV,
    DATABASE_URL: process.env.DATABASE_URL ? '✅ Set' : '❌ Missing',
    NEXTAUTH_SECRET: process.env.NEXTAUTH_SECRET ? '✅ Set' : '❌ Missing',
    NEXTAUTH_URL: process.env.NEXTAUTH_URL,
    AWS_REGION: process.env.AWS_REGION,
    STRIPE_SECRET_KEY: process.env.STRIPE_SECRET_KEY ? '✅ Set' : '❌ Missing',
    // Agregar más según necesites
  });
}
```

Luego visita: <http://localhost:3000/api/debug/env>

⚠️ IMPORTANTE: Elimina este archivo antes de hacer deploy a producción



Referencias

- [Next.js Environment Variables](https://nextjs.org/docs/basic-features/environment-variables) (https://nextjs.org/docs/basic-features/environment-variables)
- [Vercel Environment Variables](https://vercel.com/docs/concepts/projects/environment-variables) (https://vercel.com/docs/concepts/projects/environment-variables)
- [Prisma Connection Strings](https://www.prisma.io/docs/reference/database-reference/connection-urls) (https://www.prisma.io/docs/reference/database-reference/connection-urls)
- [Stripe API Keys](https://stripe.com/docs/keys) (https://stripe.com/docs/keys)
- [AWS IAM Best Practices](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html) (https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html)

Última actualización: Diciembre 2024