

GUÍA MAESTRA DE DESPLIEGUE - GPRO LOGISTIC

Esta guía cubre el proceso paso a paso para desplegar el sistema completo en producción usando Railway (Backend + BD) y Vercel (Frontend), con almacenamiento seguro en AWS S3.

FASE 1: Configuración de AWS S3 (Almacenamiento de Archivos)

Necesitamos esto para guardar PDFs de facturas y comprobantes de forma segura y barata.

1. Crear el Bucket S3

1. Ve a [AWS Console > S3](#).
2. Clic en **Create bucket**.
3. **Bucket name:** `gpro-logistic-prod` (o un nombre único).
4. **Region:** `us-east-1` (o la más cercana).
5. **Block Public Access settings:** DEJA MARCADO "Block all public access" (Queremos que sea privado).
6. Clic en **Create bucket**.

2. Crear Usuario IAM (Llaves de Acceso)

1. Ve a [AWS Console > IAM > Users](#).
2. Clic en **Create user**.
3. **User name:** `gpro-s3-user`.
4. Clic **Next**.
5. Selecciona **Attach policies directly**.
6. Busca y marca: `AmazonS3FullAccess`.
7. Clic **Next > Create user**.

3. Obtener Credenciales

1. Haz clic en el usuario creado (`gpro-s3-user`).
2. Ve a la pestaña **Security credentials**.
3. Baja a **Access keys > Create access key**.
4. Elige **Application running outside AWS**.
5. Descarga el archivo `.csv` o copia las claves INMEDIATAMENTE:
 - **Access key ID:** (Ej: AKIAIOSFODNN7EXAMPLE)
 - **Secret access key:** (Ej: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY)

FASE 2: Despliegue del Backend (Railway)

1. Crear Proyecto

1. Entra a [Railway.app](#).
2. **New Project > Deploy from GitHub repo**.
3. Selecciona el repositorio de GPRO Logistic.
4. Railway detectará el `Procfile` y la carpeta `backend`.

2. Crear Base de Datos

1. En el panel del proyecto en Railway, clic derecho (o botón "New") > **Database > PostgreSQL**.
2. Espera unos segundos a que se cree. Railway inyectará automáticamente la variable **DATABASE_URL** a tu proyecto Django.

3. Configurar Variables de Entorno (Environment Variables)

Ve a la pestaña **Variables** de tu servicio Django y agrega:

Variable	Valor
PORT	8000
DEBUG	False
SECRET_KEY	(Genera una cadena larga aleatoria)
ALLOWED_HOSTS	* (O tu dominio de Railway: xxx.up.railway.app)
CORS_ALLOWED_ORIGINS	https://gpro-logistic.vercel.app (La URL de tu frontend, ver Fase 3)
AWS_ACCESS_KEY_ID	(Tu Access Key de AWS)
AWS_SECRET_ACCESS_KEY	(Tu Secret Key de AWS)
AWS_STORAGE_BUCKET_NAME	gpro-logistic-prod

4. Configurar Comandos de Inicio

Ve a la pestaña **Settings > Build & Deploy**:

- **Build Command:** `pip install -r backend/requirements.txt && python backend/manage.py collectstatic --noinput`
- **Start Command:** `python backend/manage.py migrate && gunicorn backend.config.wsgi`
(Nota: Ajusta las rutas si 'backend' es la raíz o subcarpeta. Si tu repo TIENE la carpeta 'backend', usa los comandos de arriba. Si el repo ES la carpeta backend, quita el prefijo 'backend/'.)

▲ FASE 3: Despliegue del Frontend (Vercel)

1. Crear Proyecto

1. Entra a [Vercel](#).
2. **Add New... > Project**.
3. Importa el repositorio de GPRO Logistic.

2. Configuración

1. **Framework Preset:** Vite.
2. **Root Directory:** Haz clic en "Edit" y selecciona la carpeta **frontend**.
3. **Environment Variables:**

- **VITE_API_URL:** (La URL de tu backend en Railway, ej: <https://web-production-xxx.up.railway.app/api>)
- *Importante:* Asegúrate de agregar `/api` al final si tu código frontend lo espera así (revisa `axios.js`).

4. Clic en **Deploy**.

FASE 4: Verificación y Post-Deploy

1. Crear Superusuario:

- En Railway, ve a tu servicio Django.
- Pestaña **CLI** (o Connect).
- Ejecuta: `python backend/manage.py createsuperuser`.

2. Conexión Frontend-Backend:

- Abre tu app en Vercel.
- Intenta hacer login.
- Si falla por CORS, revisa que `CORS_ALLOWED_ORIGINS` en Railway sea EXACTAMENTE la URL de Vercel (sin slash al final).

3. Prueba de Archivos:

- Sube un PDF en una Orden de Servicio.
- Verifica que se guarde y se pueda descargar (el link vendrá firmado por AWS S3).