
INSTRUCCIONES PARA EL DESPLIEGUE DEL CONTENEDOR EN RENDER – E-CLOTHIFY (DJANGO)

1. OBJETIVO

Este documento describe el proceso para desplegar, actualizar y mantener en producción la aplicación E-Clothify, desarrollada en Django, utilizando la plataforma Render. Su objetivo es permitir que cualquier miembro del equipo pueda repetir el despliegue sin dificultades.

2. INFORMACIÓN GENERAL DEL SERVICIO

Proveedor: Render

Tipo de servicio: Web Service

Repositorio del proyecto: <https://github.com/Sergiootl/PGPI---1.15>

Branch de despliegue: main

URL pública: <https://ecommerce-store-65kd.onrender.com>

URL del panel de Render: <https://dashboard.render.com/web/srv-d4jmavqil9vc738iuld0>

3. REQUISITOS PREVIOS

- Acceso a la cuenta de Render
- Acceso al repositorio GitHub
- Archivo requirements.txt actualizado
- Archivo manage.py
- Base de datos PostgreSQL configurada
- Conocer el módulo del proyecto donde se encuentra wsgi.py

4. FLUJO DE DESPLIEGUE EN RENDER

4.1 Conexión del repositorio

1. Entrar en <https://dashboard.render.com>
2. Crear un nuevo servicio (Web Service)
3. Conectar con GitHub

4. Seleccionar el repositorio del proyecto
5. Elegir la rama que se desplegará

5. CONFIGURACIÓN DEL SERVICIO EN RENDER

Build Command

```
pip install -r requirements.txt  
python manage.py collectstatic --noinput  
python manage.py migrate
```

Start Command

```
gunicorn eclothify.wsgi:application --bind 0.0.0.0:$PORT
```

(Si el módulo principal tiene otro nombre distinto a “eclothify”, sustituirlo.)

6. VARIABLES DE ENTORNO NECESARIAS

SECRET_KEY → clave secreta de Django
DEBUG → False
ALLOWED_HOSTS → ecommerce-store-65kd.onrender.com
DATABASE_URL → URL de la base de datos PostgreSQL
DJANGO_SETTINGS_MODULE → eclothify.settings
PYTHON_VERSION → 3.11 (opcional)

Configuración de la base de datos en Django

En settings.py debe añadirse un gestor para DATABASE_URL (ejemplo común con dj-database-url).

7. DESPLIEGUE INICIAL

1. Crear el Web Service en Render
2. Configurar los comandos Build y Start
3. Añadir todas las variables de entorno
4. Guardar la configuración
5. Esperar a que Render complete el despliegue y muestre estado “Live”

8. MIGRACIONES DE DJANGO

Si es necesario ejecutarlas manualmente:

```
python manage.py migrate
```

9. ACTUALIZACIONES DEL SERVICIO

Cada vez que se hace push al branch configurado:

1. Render detecta el nuevo commit
2. Ejecuta un nuevo build
3. Reemplaza automáticamente la versión en producción
4. Revisar logs para confirmar que no hay errores

10. VERIFICACIÓN POST-DEPLOY

- Abrir la URL pública y comprobar funcionamiento
- Revisar logs desde el panel de Render
- Verificar vistas principales (inicio, productos, login...)
- Comprobar que los archivos estáticos se cargan correctamente

11. ROLLBACK

Si se detectan errores tras un despliegue:

1. Ir a la pestaña Deploys del servicio
2. Seleccionar un despliegue anterior que funcionara
3. Pulsar “Rollback”
4. Confirmar restauración

12. CHECKLIST FINAL

- ☐ SECRET_KEY configurada
- ☐ DEBUG = False
- ☐ ALLOWED_HOSTS correcto
- ☐ DATABASE_URL operativa
- ☐ Migraciones aplicadas
- ☐ Archivos estáticos funcionando
- ☐ Logs sin errores
- ☐ Servicio en estado “Live”