Documentación instalador

## 📄 Documentación de despliegue de producción de **SIG CRA**

### 🔧 Repositorio de despliegue

url del repositoprio: [GitHub - lcrojano/sig-cra-dist](https://github.com/lcrojano/sig-cra-dist)

Este repositorio contiene los archivos necesarios para ejecutar en **producción** el sistema SIG CRA (Cliente, API Laravel, Tileserver, Geostyler, etc).

## ⚙️ ¿Qué contiene este repositorio?

Los siguientes archivos y carpetas son sincronizados desde el repositorio principal sig-cra:

dist/  
├── apps/  
│ ├── api-laravel/ → Aplicación Laravel (sin .env)  
│ ├── client-ui/ → Aplicación Angular compilada  
│ └── tileserver-gl/ → Servicio de mapas  
├── libs/  
│ └── geostyler-cli/ → Herramienta de estilo de mapas  
├── tools/  
│ ├── docker/ → Archivos de configuración para Docker (nginx, mysql, traefik, etc.)  
│ └── scripts/ → Scripts auxiliares (.sh)  
├── docker-compose.yml  
├── docker-compose.cra.yml  
├── docker-compose.monitoring.yml  
├── deploy-production.sh → Script principal de despliegue

⚠️ **Importante:** El archivo .env de Laravel **no se copia** por seguridad. Se genera dinámicamente desde .env.example durante el despliegue.

## 🚀 ¿Cómo desplegar la plataforma?

### 1. Clonar este repositorio

git clone https://github.com/lcrojano/sig-cra-dist.git  
cd sig-cra-dist

### 2. Configurar el archivo .env

Antes de ejecutar el despliegue, debe agregar un archivo .env en la raíz del proyecto (junto al deploy-production.sh) con las siguientes variables necesarias:

DOMAIN=sig-cra.midominio.com  
DB\_USERNAME=root  
DB\_PASSWORD=secret  
DB\_ROOT\_PASSWORD=supersecret  
DB\_DATABASE=sigcra

🔐 Estos valores serán usados para crear las configuraciones correctas en Laravel y MySQL.

### 3. Ejecutar el script de despliegue

chmod +x deploy-production.sh  
./deploy-production.sh

El script realiza las siguientes tareas automáticamente:

* Verifica que Docker y Docker Compose estén instalados y activos.
* Genera el archivo .env de Laravel desde .env.example, reemplazando el dominio.
* Crea el archivo app.config.json para Angular.
* Actualiza las URLs de migración para MySQL.
* Crea el archivo de contraseña de MySQL.
* Detiene contenedores anteriores (si existen) y limpia recursos no usados.
* Construye y levanta los contenedores de Docker necesarios.
* Verifica la salud de los servicios (client-ui, api-laravel, tileserver).
* Configura Laravel: genera clave APP\_KEY, cachea configuración y rutas.
* Muestra los accesos finales y comandos útiles.

## 🧪 Verificación al finalizar

Al final del script, deberías ver:

🎉 Deployment completed!  
🌍 Main App: sig-cra.midominio.com  
🔧 API: https://api.sig-cra.midominio.com  
🗺️ Tiles: https://tiles.sig-cra.midominio.com  
📊 Traefik Dashboard: https://traefik.sig-cra.midominio.com

💡 Puedes verificar manualmente que los servicios estén corriendo usando:

docker compose ps

Y revisar los logs:

docker compose logs -f

## 🧠 Consideraciones

* El despliegue genera certificados SSL automáticamente usando Traefik y Let's Encrypt.
* Los archivos .env deben mantenerse fuera del repositorio por seguridad.
* Siempre usa el dominio completo (sig-cra.midominio.com) en la variable DOMAIN.
* Nota **para pruebas en entorno local**: Si se desea ejecutar la plataforma en un entorno local, es necesario agregar las siguientes entradas al archivo hosts del sistema (generalmente ubicado en /etc/hosts en Linux/macOS o en C:\Windows\System32\drivers\etc\hosts en Windows):

127.0.0.1 app.localhost  
127.0.0.1 tiles.localhost  
127.0.0.1 traefik.localhost  
127.0.0.1 pma.localhost

* Para el correcto funcionamiento de la plataforma, es necesario crear los siguientes subdominios en el proveedor de DNS utilizado (por ejemplo, Cloudflare, Route 53, etc.):
  + api.: Este subdominio apunta al backend desarrollado en Laravel.
  + tiles.: Este subdominio sirve para exponer el servidor de mapas y capas geoespaciales.
  + traefik.: Este subdominio está asociado al panel de monitoreo y configuración del proxy inverso Traefik, utilizado para enrutar tráfico y gestionar certificados SSL.
* Adicionalmente, se debe agregar la clave de API de SendGrid en el entorno (.env) para habilitar el envío de correos desde la plataforma. Esta clave debe ser generada desde el panel de SendGrid y asignada a la variable correspondiente (por ejemplo, SENDGRID\_API\_KEY). Asegúrate de configurar adecuadamente el dominio verificado en SendGrid para garantizar una entrega segura y efectiva de los correos.

## 📬 ¿Preguntas?

Para dudas técnicas, contactar con el equipo de desarrollo o revisar el archivo deploy-production.sh que contiene muchos comentarios útiles y validaciones.