**Desplegar el Frontend (Angular)**

El frontend en Angular se compila a archivos estáticos (Html, Css y Javascript). En el caso de angular para compilar hacia producción colocamos el Comando:  
  
ng build --prod

Generara la aplicación de forma optimizada en una carpeta “dist” que se usara para desplegar.

Luego debemos elegir un servicio de alojamiento. Al ser archivos estáticos podemos alojarlos en plataformas como Netlify, Vercel o Aws.

Una vez seleccionada nuestra plataforma, subimos los archivos. Vercel y netlify ofrecen una facilidad para desplegar ya que podemos vincularlo con github. Proyectos como react, netlify y github (github-pages) son muy utilizados para desplegar.

**Desplegar el Backend (Django)**

En el caso del backend debemos seleccionar una plataforma donde podamos ser capaz de manejar solicitudes HTTP y servir la api del proyecto.

Como usamos django, debemos configurar el archivo settings.py para que este configurado correctamente para producción. Configuraciones obvias: Configuración de la base de datos, configurar el cors, entre otras.

Un punto no menos importante es verificar que las migraciones esten listadas para preparar la base de datos “Python manage.py migrae”

Luego debemos configurar un servidor web paraDjango. Un servidor recomendado es Gunicorn junto con un servidor proxy como Nginx o Apache. Gunicorn: “gunicorn myproject.wsgi:application”

Elegimos una Plataforma de alojamiento para nuestro backend. Algunas de las mas usadas puedes ser Heroku, DigitalOcean (en mi caso llegue a ver por muy arriba esta opción, pero tambien es una muy bueno con Wordpress), Azure o Google Cloud plataforma.

No debemos olvidar configurar nuestras variables de entorno para producción. Podemos usar Docker para facilitar el empaquetado y despliegue del backend en diferentes entornos de producción.

En el caso de que vinculemos github con heroku, podemos utilizar este comando para un despliegue automatico “git push heroku main”

**Manejo de Variables de Entorno y Secretos**

Para manejar las variables de entorno en producción, podemos usar herramientas como **django-environ** para cargar las variables de un archivo .env que tenga configuraciones de la base de datos, claves secretas, etc.

**Migraciones de Bases de Datos**

**python manage.py migrate**

Podemos usar **Django migrations** para actualizar los esquemas de las bases de datos entre entornos de desarrollo y producción. Otra cosa no menos importante es la creación de un backup para la recuperación, podemos crear copias diarias y antes de una actualización o cambio en producción es tambien una buena opción el realizar una copia de seguridad.

Plataformas que tambien pueden alojar base de datos: firebase, supabase (en el caso de supabase se debe manejar con el cli sus migraciones).

**Programa de migración.**

**Preparar el Proyecto Backend (Django con Channels).**

* **Instalar dependencias:**
* **Configurar settings.py de Django**
* **Configurar CHANNEL\_LAYERS con redis**
* **Configuracion del archive asgy.py**
* **Creacion de las vistas y rutas para los websockets.**
* **Hacer las migraciones en la base de datos.**

**Preparar el proyecto frontend (angulr).**

* **Compilar la aplicación Angular.**
* **Configurar un servidor web para Angular**

**Configurar Redis en Docker.**

* **Agregar Redis al archivo docker-compose.yml.**
* **Configurar Dockerfile para Django.**
* **Configurar Docker Compose para Backend y Frontend**

**Configurar y Desplegar en un Servidor.**

* **Configurar un servicio de despliegue.**
* **Teer en cuenta la seguridad de la aplicación (https).**
* **Subir el código al servidor.**
* **Desplegar la aplicación en prod.**