Perfecto, Chema. Vamos a diseñar un entorno de trabajo y pruebas **profesional, modular y reproducible**, listo para desplegar el proyecto SentinelCore dentro de un servidor (bare-metal, virtual o cloud). Aquí va el **paso a paso completo**, y al final te daré también un checklist de verificación.

**🛠️ PASOS PARA IMPLEMENTAR EL ENTORNO EN EL SERVIDOR**

**✅ 1. Requisitos del servidor**

* **Sistema operativo**: Ubuntu Server 22.04 LTS (o equivalente)
* **CPU mínima**: 4 cores
* **RAM recomendada**: 8 GB
* **Almacenamiento**: 50 GB (extensible)
* **Acceso a Internet** y permisos de sudo

**✅ 2. Instalar herramientas básicas**

sudo apt update && sudo apt upgrade -y

sudo apt install -y curl git make zip unzip ufw

**✅ 3. Instalar Docker y Docker Compose**

# Docker

curl -fsSL https://get.docker.com | sh

sudo usermod -aG docker $USER

# Docker Compose v2

sudo apt install docker-compose-plugin -y

🔁 Cierra sesión y vuelve a entrar o ejecuta: newgrp docker

**✅ 4. Clonar el repositorio del proyecto**

Si ya tienes los .zip, súbelos por SCP o usa git si lo has subido a GitHub:

git clone https://github.com/tu\_usuario/sentinelcore.git

cd sentinelcore

o bien:

mkdir sentinelcore && cd sentinelcore

unzip platsentinel\_backend.zip

unzip platsentinel\_ui\_auth.zip

mv platsentinel\_backend backend

mv platsentinel\_ui\_auth frontend

**✅ 5. Colocar los archivos auxiliares**

* docker-compose.yml
* Makefile
* .env (ajustado si cambias claves JWT o puertos)

📁 Estructura deseada final:

sentinelcore/

├── backend/

├── frontend/

├── docker-compose.yml

├── Makefile

└── informes/

**✅ 6. Abrir puertos del firewall**

sudo ufw allow 8000 # API backend

sudo ufw allow 3000 # Frontend

sudo ufw allow 8080 # ZAP

sudo ufw allow ssh

sudo ufw enable

**✅ 7. Levantar el entorno**

make up

Y verificar:

* <http://localhost:8000/docs> → Backend API
* [http://localhost:3000](http://localhost:3000/) → Dashboard Vue
* [http://localhost:8080](http://localhost:8080/) → ZAP daemon

**✅ 8. (Opcional) Configurar NGINX + SSL (Producción)**

Puedes usar NGINX como proxy inverso con certificados Let’s Encrypt.

**📋 CHECKLIST FINAL**

| **Elemento** | **Estado** |
| --- | --- |
| Sistema operativo actualizado | ✅ |
| Docker y Compose instalados | ✅ |
| Proyecto cargado y estructurado | ✅ |
| .env configurado | ✅ |
| Carpetas backend, frontend | ✅ |
| docker-compose.yml operativo | ✅ |
| Puertos abiertos en firewall | ✅ |
| Servicios accesibles vía navegador | ✅ |
| Logs disponibles vía make logs | ✅ |

¿Quieres que te genere un script .sh para automatizar todo este proceso desde cero en un servidor limpio?