Laboratorio Frappe Seguro - Modelo Preproductivo (Windows 11 / Linux)

# Índice de Contenido

1. 1. Introducción
2. 2. Objetivo del Laboratorio Seguro
3. 3. Requisitos Técnicos
4. 4. Estructura del Proyecto

5. Cómo subir o apuntar al paquete .zip del laboratorio

### Opción 1: Descargar desde navegador (local Windows)  
1. Descarga el archivo desde el siguiente enlace o fuente oficial: `Paquete\_Laboratorio\_Seguro\_Frappe.zip`  
2. Guárdalo en tu PC, por ejemplo: `C:\Users\TU\_USUARIO\Downloads`  
3. En Ubuntu (WSL), copia el archivo:  
```bash  
mkdir -p ~/frappe\_secure  
cp /mnt/c/Users/TU\_USUARIO/Downloads/Paquete\_Laboratorio\_Seguro\_Frappe.zip ~/frappe\_secure/  
cd ~/frappe\_secure  
unzip Paquete\_Laboratorio\_Seguro\_Frappe.zip  
```  
  
### Opción 2: Subir desde servidor o URL  
```bash  
cd ~  
mkdir frappe\_secure && cd frappe\_secure  
wget https://tu-dominio.com/Paquete\_Laboratorio\_Seguro\_Frappe.zip  
unzip Paquete\_Laboratorio\_Seguro\_Frappe.zip  
```  
  
### Opción 3: Usar SCP si lo tienes en otra máquina  
```bash  
scp usuario@ip:/ruta/Paquete\_Laboratorio\_Seguro\_Frappe.zip .  
unzip Paquete\_Laboratorio\_Seguro\_Frappe.zip  
```  
  
### Opción 4: Azure Storage (requiere cuenta configurada)  
```bash  
az login  
az storage blob download --account-name MI\_STORAGE --container-name MI\_CONTENEDOR \   
--name Paquete\_Laboratorio\_Seguro\_Frappe.zip --file Paquete\_Laboratorio\_Seguro\_Frappe.zip --auth-mode login  
unzip Paquete\_Laboratorio\_Seguro\_Frappe.zip  
```

1. 5. Despliegue Seguro Paso a Paso
2. 6. Eliminación de Prácticas Inseguras  
   Paso 1: Crear carpeta de trabajo y descargar el paquete  
   ```bash  
   mkdir ~/frappe\_secure && cd ~/frappe\_secure  
   # Copia el ZIP desde Windows (WSL):  
   cp /mnt/c/Users/TU\_USUARIO/Downloads/Paquete\_Laboratorio\_Seguro\_Frappe.zip .  
   unzip Paquete\_Laboratorio\_Seguro\_Frappe.zip  
   ```  
     
   Paso 2: Instalar Docker en Ubuntu (si no está instalado)  
   ```bash  
   sudo apt update  
   sudo apt install docker.io docker-compose -y  
   sudo systemctl enable docker  
   sudo usermod -aG docker $USER  
   newgrp docker  
   ```  
     
   Paso 3: Verifica que Docker funciona  
   ```bash  
   docker --version  
   docker-compose --version  
   ```  
     
   Paso 4: Crear archivo de variables de entorno  
   ```bash  
   cp .env.example .env  
   nano .env # Edita y cambia contraseñas seguras  
   ```  
     
   Paso 5: Ejecutar los servicios con Docker  
   ```bash  
   docker-compose down  
   docker-compose up --build -d  
   ```  
     
   Paso 6: Ingresar al contenedor Frappe e inicializar  
   ```bash  
   docker exec -it frappe bash  
   source .env  
   ./init.sh  
   ```  
     
   Paso 7: Validar servicios  
   ```bash  
   docker ps  
   curl http://localhost  
   docker logs nginx  
   ```  
     
   Paso 8: Acceder desde navegador  
   http://localhost
3. 7. HTTPS con Renovación Automática
4. 8. Configuración de Red Segura
5. 9. Observaciones y Buenas Prácticas
6. 10. Conclusión

# 1. Introducción

Este laboratorio está diseñado para emular un entorno seguro de producción, desplegado sobre Windows 11 con WSL2 o una VM Ubuntu, con Docker y NGINX. Se incluyen medidas preventivas contra malas prácticas, orientadas a seguridad y continuidad operativa.

# 2. Objetivo del Laboratorio Seguro

Proveer una base sólida y reutilizable con configuración similar a un entorno productivo, contemplando seguridad en servicios, eliminación de contraseñas en texto plano, y buenas prácticas de despliegue y acceso.

# 3. Requisitos Técnicos

- WSL2 en Windows 11 o Ubuntu  
- Docker y Docker Compose  
- Acceso a terminal Bash  
- Opcional: Dominio registrado para habilitar HTTPS

# 4. Estructura del Proyecto

- `docker-compose.yml`  
- `nginx.conf`  
- `init.sh`  
- `renew\_ssl.sh`  
- `README\_EXTENDIDO.md`  
- `certbot/` (conf y www para certificados)  
- Red interna Docker: `frappe-net`

# 5. Despliegue Seguro Paso a Paso

1. Clonar o descomprimir el entorno:  
```bash  
mkdir ~/frappe\_secure && cd ~/frappe\_secure  
unzip Paquete\_Final\_Adviters\_Frappe.zip  
```  
2. Crear archivo `.env` con credenciales seguras  
3. Editar `docker-compose.yml` para usar variables de entorno  
4. Iniciar:  
```bash  
docker-compose up --build -d  
```  
5. Ejecutar `init.sh` sin hardcodear contraseñas  
6. Validar acceso solo por NGINX:  
```bash  
curl http://localhost  
```

# 6. Eliminación de Prácticas Inseguras

- ❌ Se evita `bench start`, se debe usar `bench start` solo en desarrollo, se puede reemplazar por `supervisor` externo o `pm2`  
- ❌ Se eliminan contraseñas hardcodeadas  
- ✅ Se incluye `.env` y carga de variables con `source .env`  
- ❌ Puerto 9000 no se expone externamente; sólo accesible por red Docker (`frappe-net`)

# 7. HTTPS con Renovación Automática

- Certificados generados con `certbot`  
- Script `renew\_ssl.sh` incluido y automatizado con `cron`  
- Volúmenes persistentes configurados en `docker-compose.yml`  
- NGINX recargado automáticamente tras renovación  
- Redirección permanente de HTTP a HTTPS activada

# 8. Configuración de Red Segura

- Se define red Docker `frappe-net` aislada  
- Redis y MariaDB no expuestos fuera de Docker  
- Solo NGINX publica puertos 80 y 443  
- Comunicación entre contenedores se da por hostname interno (`mariadb`, `redis`)

# 9. Observaciones y Buenas Prácticas

- Separación por servicios garantiza escalabilidad  
- Uso de entorno `.env` para evitar secretos expuestos  
- Documentación clara en Word y Markdown  
- Listo para adaptar a Azure, Dev/Test Labs o CI/CD pipelines

# 10. Conclusión

Este laboratorio establece una base sólida, segura y modular para Frappe ERP. Puede escalarse a productivo con mínimas adaptaciones, y permite integrar monitoreo, backups y pipelines automatizados sin rediseñar la arquitectura base.