

# DOCUMENTACIÓN PROYECTO INTEGRADO – ASIR

---

## DOCUMENTACIÓN PROYECTO INTEGRADO – ASIR

### 1. INTRODUCCIÓN

Este proyecto tiene como objetivo el despliegue automatizado de aplicaciones en contenedores usando Docker y Ansible, simulando un entorno real mediante Proxmox. Se busca estandarizar, automatizar y facilitar la puesta en marcha de entornos que puedan escalarse fácilmente. Se utilizarán tecnologías ampliamente adoptadas en entornos DevOps con un enfoque empresarial.

### 2. PLANIFICACIÓN

- **Inicio:** Marzo 2025
- **Fin estimado:** Junio 2025
- **Duración total:** 4 meses
- **Fases:**
  - Revisión de requisitos
  - Preparación del entorno
  - Desarrollo de los playbooks de Ansible
  - Creación y prueba de contenedores Docker
  - Configuración de Nginx como proxy reverso
  - Validación y pruebas
  - Redacción y presentación

### 3. ANÁLISIS

El sistema sustituye despliegues manuales y no homogéneos por una arquitectura automatizada. Los usuarios actuales (administradores de sistemas) se enfrentan a tareas repetitivas y propensas a errores. El nuevo sistema automatiza todo esto, permitiendo rapidez, estandarización y escalabilidad.

**Requisitos funcionales:**

- Despliegue de servicios web (por ejemplo, una app PHP)
- Base de datos relacional (MySQL)
- Proxy reverso con Nginx
- Gestión de servicios con contenedores Docker
- Automatización con Ansible

**Requisitos no funcionales:**

- Escalabilidad
- Trazabilidad mediante control de versiones (Git)
- Modularidad y reutilización de roles y playbooks
- Seguridad de acceso y configuración

## 4. DISEÑO

**Infraestructura:**

- 1 servidor Proxmox (virtualización)
- VM 1: Servidor principal (Ansible Master + Docker Host)
- VM 2: Cliente 1 (simulación de acceso a servicios)
- VM 3: Cliente 2 (opcional para pruebas)

**Tecnologías:**

- Ubuntu Server 22.04
- Docker + Docker Compose
- Ansible
- Nginx como proxy reverso
- MySQL como motor de base de datos

**Arquitectura lógica:**

Nginx → App (contenedor) ↔ MySQL (contenedor)

Todos gestionados por Docker, configurados con Ansible

## 5. IMPLEMENTACIÓN

### **Paso 1: Preparación del entorno**

- Instalación de Proxmox y creación de VMs
- Instalación de Ubuntu Server 22.04

### **Paso 2: Instalación de dependencias**

- Instalación de Docker y Docker Compose
- Instalación de Ansible
- Inicialización de repositorio Git

### **Paso 3: Desarrollo de Playbooks**

- Playbooks para instalación de dependencias
- Playbooks para despliegue de servicios (Nginx, MySQL, app PHP)

### **Paso 4: Configuración de contenedores**

- Dockerfiles personalizados
- docker-compose.yml para orquestación local

### **Paso 5: Configuración de Nginx**

- Hosts virtuales
- Redirección de tráfico y SSL si procede

### **Paso 6: Pruebas y validación**

- Conexión entre contenedores
- Acceso desde clientes
- Comprobación de reinicio automático y persistencia

## 6. PRUEBAS

Se realizarán pruebas unitarias y de integración:

- Verificar acceso a la app desde los clientes
- Comprobación de disponibilidad de MySQL
- Fallos simulados para validar autorecuperación
- Comprobación de reglas de seguridad (puertos abiertos, firewall)

## 7. DOCUMENTACIÓN TÉCNICA

- Manual de instalación de dependencias
- Manual de uso de playbooks
- Manual de gestión de contenedores y acceso a logs
- Diagrama de red y servicios

## 8. CONCLUSIONES

Este proyecto demuestra la viabilidad de automatizar el despliegue de servicios usando herramientas modernas de DevOps como Docker y Ansible. Además, muestra cómo una infraestructura sencilla puede escalarse y adaptarse a entornos reales. Proxmox ha permitido realizar pruebas sin coste adicional y simular un entorno empresarial realista.

**ESTA DOCUMENTACIÓN SIGUE EN DESARROLLO, CONSIDERAR QUE ALGUNOS DE LOS FORMATOS Y CONTENIDOS PUEDEN VARIAR CON LA VERSIÓN FINAL PRESENTADA Y QUE ESTA ES SIMPLEMENTE EL DESARROLLO DE LA IDEA PRINCIPAL.**