

DPTO. INFORMÁTICA – I.E.S. LA MARISMA
MÓDULO PROYECTO
C.F.G.S.

ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED – GRADO SUPERIOR

**DESPLIEGUE AUTOMATIZADO DE UNA
INFRAESTRUCTURA POR RED**

David Faustino Benítez

19/03/2025 – XX/06/2025

Profesor responsable del seguimiento:

Gonzalo Cañadillas Rueda

Tutora del centro de trabajo:

ÍNDICE

INTRODUCCIÓN.....	3
ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO	3
OBJETIVO GENERAL DEL PROYECTO	3
OBJETIVOS ESPECÍFICOS	3
TAREAS	5
Tarea 1: Preparación del entorno.....	5
• Subtarea 1.1: Análisis de requisitos.....	5
• Subtarea 1.2: Selección de plataforma	6
• Subtarea 1.3: Provisionamiento inicial.....	6
• Subtarea 1.4: Verificación de la plataforma	6
Tarea 2: Contenerización con Docker	6
• Subtarea 2.1: Diseño de Dockerfiles	6
• Subtarea 2.2: Construcción y prueba de imágenes	6
• Subtarea 2.3: Orquestación en desarrollo	6
• Subtarea 2.4: Registro de imágenes.....	6
Tarea 3: Automatización con Ansible	6
• Subtarea 3.1: Estructura de roles y variables.....	6
• Subtarea 3.2: Playbooks de instalación	6
• Subtarea 3.3: Playbooks de despliegue.....	6
• Subtarea 3.4: Validación automatizada	6
Tarea 4: Pruebas y documentación final.....	6
• Subtarea 4.1: Pruebas funcionales	6
• Subtarea 4.2: Pruebas de resiliencia	7
• Subtarea 4.3: Corrección de errores	7
• Subtarea 4.4: Documentación técnica	7
• Subtarea 4.5: Revisión y entrega	7
RECURSOS HUMANOS	7
RECURSOS MATERIALES	9
CRONOGRAMA	9
PRESUPUESTO	10
Escenario 1: Entorno de pruebas con Proxmox.....	10
Infraestructura (entorno de pruebas).....	10
Escenario 2: Entorno simulado empresarial (infraestructura física y licencias).....	11
Infraestructura física.....	11
Licencias empresariales y software	11
Otros costes estimados	11
Total estimado del proyecto en entorno físico profesional:.....	12
ANEXOS.....	12
BIBLIOGRAFÍA.....	12
CONCLUSIÓN	13

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.

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

ORIGEN Y CONTEXTUALIZACIÓN DEL PROYECTO

OBJETIVO GENERAL DEL PROYECTO

Se pretende diseñar e implementar un sistema automatizado de despliegue de servicios utilizando contenedores Docker y herramientas de automatización como Ansible, con el propósito de facilitar la gestión, escalabilidad y reproducibilidad de aplicaciones en entornos profesionales. Este sistema permitirá simplificar tareas repetitivas, reducir errores humanos y establecer un modelo de despliegue eficiente, seguro y fácilmente replicable en contextos empresariales reales.

OBJETIVOS ESPECÍFICOS

A continuación, se presentan listados los diferentes objetivos desarrollados durante el proyecto:

- **Configuración de una infraestructura básica de red y servidores**

Esto permitirá simular un entorno profesional con despliegue de servicios.

- **Instalación y configuración de Docker y Docker Compose**

La instalación se realizaría sobre un servidor Linux, el cual serviría la dockerización de los servicios.

- **Automatización del proceso de despliegue**

Empleando Ansible mediante la creación de playbooks reutilizables y modulares.

- **Despliegue de servicios esenciales en contenedores**

Incluyendo un servidor Nginx como proxy reverso, una base de datos MySQL y una aplicación web como ejemplos de prueba.

- **Documentación del proceso de instalación, configuración y despliegue**

Lo que facilitaría futuras implementaciones y mantenimiento del sistema.

- **Simulación de accesos de usuarios desde clientes**

Así verificamos un correcto funcionamiento de los servicios desplegados.

- **Evaluación final sobre el rendimiento y la estabilidad del sistema**

A través de pruebas funcionales y técnicas, que nos permitirían identificar mejoras.

TAREAS

Tarea 1: (Descripción (nombre) de la tarea)

Objetivo y metodología

En este apartado se define tanto el objetivo que persigue la ejecución de esta tarea así como una definición de la misma, es decir, en qué consiste la tarea, qué se va a hacer durante el desarrollo de la misma.

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

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)

PASO 7 DOCUMENTACION

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

TAREAS

Este proyecto se llevará a cabo a través de cuatro tareas principales que combinan la contenedorización de servicios con Docker y la automatización mediante Ansible:

Tarea 1: Preparación del entorno

Establece y valida los cimientos técnicos antes de construir contenedores o playbooks.

- ***Subtarea 1.1: Análisis de requisitos***
Revisión de necesidades de hardware, redes, seguridad y rendimiento para soportar contenedores y automatización.

- **Subtarea 1.2: Selección de plataforma**
Comparación entre un entorno de pruebas virtualizado (Proxmox) y uno de producción (infraestructura física o nube).
- **Subtarea 1.3: Provisionamiento inicial**
Implementación del sistema operativo (Ubuntu Server) en el servidor principal y configuración de red y SSH.
- **Subtarea 1.4: Verificación de la plataforma**
Confirmación de que el entorno dispone de los recursos y configuraciones necesarias.

Tarea 2: Contenerización con Docker

Construye imágenes ligeras y portables de los servicios.

- **Subtarea 2.1: Diseño de Dockerfiles**
Creación de Dockerfiles optimizados para Nginx, MySQL y la aplicación web.
- **Subtarea 2.2: Construcción y prueba de imágenes**
Compilación de imágenes locales y pruebas funcionales individuales.
- **Subtarea 2.3: Orquestación en desarrollo**
Configuración de `docker-compose.yml` para ejecutar los servicios agrupados.
- **Subtarea 2.4: Registro de imágenes**
Publicación en un registry para facilitar despliegues automatizados.

Tarea 3: Automatización con Ansible

Define infraestructuras como código para desplegar y mantener contenedores.

- **Subtarea 3.1: Estructura de roles y variables**
Organización de playbooks en roles reutilizables con variables templated.
- **Subtarea 3.2: Playbooks de instalación**
Automatiza la instalación de Docker, Docker Compose y dependencias en hosts remotos.
- **Subtarea 3.3: Playbooks de despliegue**
Despliega contenedores y configura Nginx como proxy reverso y MySQL según definiciones.
- **Subtarea 3.4: Validación automatizada**
Scripts de comprobación incluidos en Ansible que confirman la salud de cada servicio.

Tarea 4: Pruebas y documentación final

Combina pruebas exhaustivas con la entrega de documentación completa.

- **Subtarea 4.1: Pruebas funcionales**
Acceso desde clientes a servicios web y base de datos, verificando rutas y autenticación.

- **Subtarea 4.2: Pruebas de resiliencia**
Simulación de fallos de contenedores y redes para probar la capacidad de autorecuperación.
- **Subtarea 4.3: Corrección de errores**
Ajuste de Dockerfiles, playbooks y configuraciones según resultados de pruebas.
- **Subtarea 4.4: Documentación técnica**
Manual de instalación, guía de uso de playbooks y mantenimiento de la infraestructura.
- **Subtarea 4.5: Revisión y entrega**
Validación final de la documentación, inclusión de anexos y preparación para la defensa.

Durante el desarrollo del proyecto, se han identificado los siguientes perfiles necesarios para su correcta ejecución. En este caso, el proyecto será desarrollado individualmente por el autor, por lo que un mismo alumno desempeñará todos los roles requeridos.

Subtarea 1.1:

	Duración (horas)	Fecha inicio	Fecha fin	Responsable
Tarea 1				
Subtarea 1.1				
Subtarea 1.2				

RECURSOS HUMANOS

En este caso toda la implementación e instalación será realizada por mí mismo, aunque enfocándolo a un entorno real necesitaríamos los siguientes perfiles.

Perfil	Función	Cantidad
---------------	----------------	-----------------

Administrador del sistema / DevOps	<ul style="list-style-type: none"> • Instalación y configuración del sistema. • Creación de contenedores Docker. • Automatización con Ansible. • Redacción de playbooks de Ansible. • Desarrollo de archivos Docker. • Orquestación de servicios. 	1
Tester / Quality Assurance	<ul style="list-style-type: none"> • Comprobación de la funcionalidad. • Pruebas de estrés. • Validación de la automatización. 	1
Redactor	<ul style="list-style-type: none"> • Manuales técnicos • Diagramas • Documentación de soporte 	1

RECURSOS MATERIALES

Se detallarán los recursos materiales necesarios para la ejecución o el éxito del proyecto.

Entorno de virtualización PROXMOX

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

CRONOGRAMA

- 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

Mediante un gráfico Gantt se explicitará la secuencia de las diferentes tareas a realizar durante el proyecto.

PRESUPUESTO

Escenario 1: Entorno de pruebas con Proxmox

Este escenario está enfocado a la implementación del proyecto en un entorno de pruebas, como el que se realiza en el entorno académico. Se basa en una sola máquina física capaz de virtualizar el resto de los elementos necesarios mediante Proxmox VE.

Infraestructura (entorno de pruebas)

Recurso	Coste estimado	Descripción
Servidor físico (host Proxmox)	700 € – 1.200 €	Equipo con CPU multinúcleo, 16–32 GB RAM, SSD o NVMe.
Almacenamiento adicional (opcional)	100 € – 200 €	Para snapshots, backups y expansión de VMs.
Electricidad y mantenimiento	50 € – 100 €/año	Estimación para uso doméstico.

En este caso he optado por software libre lo que no tendría costes.

Escenario 2: Entorno simulado empresarial (infraestructura física y licencias)

Nota: La conversión de este proyecto a un entorno empresarial real puede variar considerablemente en función de la infraestructura existente, el número de usuarios/clientes concurrentes, y las políticas de seguridad, soporte y escalabilidad específicas de cada organización.

Este escenario representa una simulación de un entorno profesional realista en el que el sistema se desplegaría sobre infraestructura física, con servicios preparados para ser accesibles desde un número indefinido de clientes. A diferencia del entorno de pruebas, este enfoque elimina Proxmox y cualquier virtualización, enfocándose en la replicación de un entorno empresarial real.

Infraestructura física

Recurso	Coste estimado	Descripción
Servidor principal (Ansible + Docker Host)	2.500 € – 3.000 €	Equipo dedicado con soporte empresarial.
Cliente(s) físicos de acceso (n ≥ 1)	Variable según escala	No se define un número exacto de clientes. Se considera como estimación mínima 1.500 €.
SAI y red profesional (switch, firewall, etc.)	500 € – 1.000 €	Infraestructura de red física y protección eléctrica.

Licencias empresariales y software

Software	Licencia empresarial estimada
Docker Business	21 €/usuario/mes (~250 €/año)
Ansible Automation Platform	Bajo presupuesto (Red Hat) ~ 1.500 €
Nginx Plus	2.500 €/año
MySQL Enterprise Edition	5.000 €/año

Otros costes estimados

- Electricidad, refrigeración y mantenimiento anual: **200 € – 300 €**
- Formación externa y documentación adicional: **500 € aprox.**

Total estimado del proyecto en entorno físico profesional:

11.000 € – 13.500 € (estimación ajustada para una empresa con infraestructura física y licencias activas). – 13.500 €** (incluyendo hardware, licencias anuales y soporte físico mínimo).

11.000 € – 13.500 € (incluyendo hardware, licencias anuales y soporte físico mínimo).

- Playbooks de Ansible utilizados para despliegue y configuración.
- Archivos de definición de contenedores (Dockerfile y docker-compose.yml).
- Capturas de pantalla de la ejecución del sistema.
- Diagrama de red y arquitectura del sistema.
- Comandos principales empleados y resultados de pruebas.

	Recursos humanos	Recursos materiales	Total
Tarea 1			
Subtarea 1.1			
Subtarea 1.2			
Subtarea 1. ...n			
Tarea 2			
Subtarea 2.1			
Subtarea 2. ...n			
Tarea 3			
Etc.			
Total:			

ANEXOS

BIBLIOGRAFÍA

- Documentación oficial de Docker: <https://docs.docker.com/>
- Documentación oficial de Ansible: <https://docs.ansible.com/>
- Manual de Nginx: <https://nginx.org/en/docs/>
- Manual de MySQL: <https://dev.mysql.com/doc/>
- Pautas del Proyecto Integrado - IES La Marisma (curso 2024/25)
- Repositorios y foros de comunidad (Stack Overflow, GitHub)

CONCLUSIÓN

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.