

Trabajo Práctico: *"Mini Plataforma Web con Persistencia y Escalabilidad"*

Objetivo general

Diseñar e implementar una arquitectura básica basada en Docker Compose que simule una pequeña plataforma web con backend, base de datos y un componente opcional de visualización o administración.

Enunciado

Situación:

Una pequeña pyme quiere tener una plataforma interna donde sus empleados puedan cargar tareas, ver su estado y generar reportes simples. Como consultores, ustedes deberán:

- Diseñar la solución con servicios dockerizados.
- Usar `docker-compose` para orquestar los servicios.
- Montar volúmenes para persistencia de datos.
- Documentar sus decisiones tecnológicas y justificar por qué eligieron ciertas imágenes o herramientas.

Requisitos del trabajo

1. Servicios mínimos que deben montar

Servicio	Requisitos
Backend API	Debe manejar tareas y estados (<code>ToDo</code> , <code>Doing</code> , <code>Done</code>).
Base de Datos	PostgreSQL o MySQL/MariaDB, con persistencia en volumen.
Admin Web	(Opcional pero recomendado) Usar una app para ver o administrar los datos.

2. Requerimientos técnicos

- El `docker-compose.yml` debe incluir:
 - Al menos **dos contenedores** (API + DB).
 - **Volumen montado** para la persistencia de datos de la base.
 - Uso de **redes docker** para la comunicación entre servicios.
- Opcional: utilizar `.env` para variables sensibles (puertos, contraseñas).

 **Guía para resolver (no incluida en el enunciado del TP, pero podés usarla como corrección o ayuda en clase)**

Paso	Tarea
1	Elegir stack tecnológico (Flask + PostgreSQL, Express + MySQL, etc).
2	Buscar imágenes oficiales o confiables de DockerHub.
3	Crear <code>docker-compose.yml</code> con volúmenes y red.
4	Probar conexión del backend a la base de datos.
5	Validar persistencia con reinicio de contenedores.
6	(Opcional) Añadir Adminer, pgAdmin o Metabase para visualización.

 **Ideas de tecnologías posibles (ellos eligen, vos podés guiar)**

Función	Alternativas sugeridas
Backend API	Flask, FastAPI, Node.js (Express), etc.
Base de datos	PostgreSQL, MariaDB, MySQL
Admin Web opcional	Adminer, pgAdmin, Metabase

Entregables

- `docker-compose.yml` bien estructurado y comentado.
- Instrucciones de uso en `README.md`.
- Capturas o evidencia del sistema funcionando.
- Documento de decisiones: justificación de las tecnologías elegidas.
- (Bonus) Endpoint de prueba con curl o Postman documentado.