

## SESIÓN 14: PROYECTO INTEGRADOR

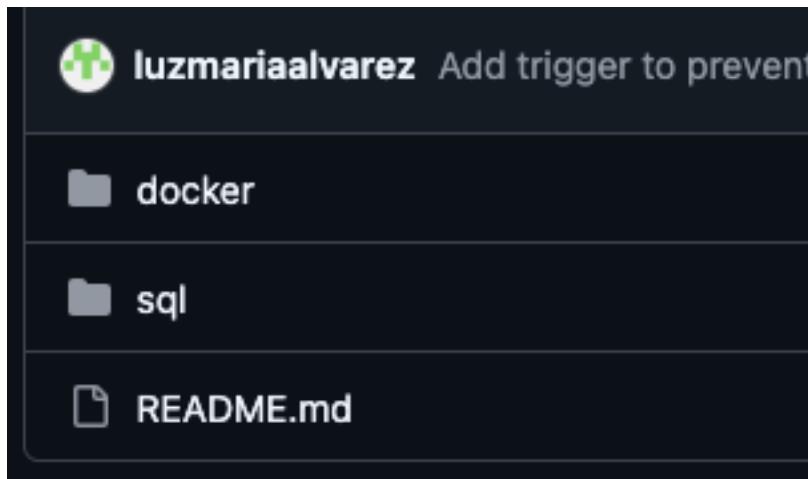
La empresa **PROMETEO** necesita una base de datos mínima para arrancar su nueva aplicación.

Tu misión es dejar un entorno funcional usando **Docker + MySQL**, con **usuarios**, una **vista** y un **trigger**, y subirlo todo a **GitHub** como entrega.

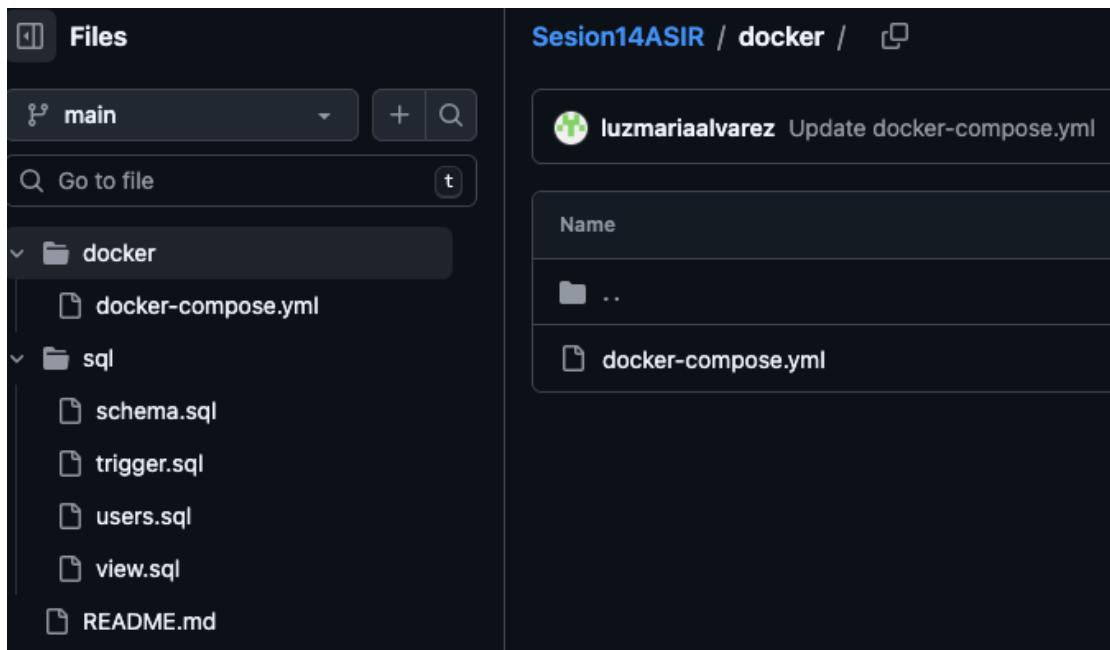
Crear un repositorio en GitHub llamado prometeo con:

1. Un **docker-compose** que levante MySQL y cree la BBDD prometeo
2. Scripts SQL para:
  - crear una tabla clientes
  - crear usuarios con permisos
  - crear una vista
  - crear un trigger
3. Un README.md que explique cómo se usa

### Estructura obligatoria



# PROMETEO



## Requisitos

- Base de datos: prometeo
- Tabla: clientes(id, nombre, email)
- Usuarios:
  - admin\_prometeo (todos los permisos sobre prometeo)
  - lector (solo SELECT sobre prometeo)
- Vista:
  - v\_clientes: muestra nombre y email
- Trigger:
  - impide insertar clientes con email vacío o NULL

## Guía:

### Crear repo y clonar

1. En GitHub: crea un repo prometeo
2. En tu PC: (opcional, en mi caso, lo voy a hacer desde el navegador)

```
git clone https://github.com/TU_USUARIO/prometeo.git  
cd prometeo
```

**Paso 1. Crear carpetas**

```
mkdir -p docker sql
```

**Paso 2. Crear docker/docker-compose.yml**

Crea el archivo:

services:

mysql:

  image: mysql:8.4

  container\_name: prometeo-db

  environment:

    MYSQL\_ROOT\_PASSWORD: root

    MYSQL\_DATABASE: prometeo

  ports:

    - "3306:3306"

  volumes:

    - ./sql:/docker-entrypoint-initdb.d

Esto hace que MySQL, al arrancar, ejecute automáticamente todos los .sql de sql/.

**Paso 3. Crear sql/schema.sql**

```
USE prometeo;
```

```
CREATE TABLE clientes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50),
  email VARCHAR(100)
);
```

```
INSERT INTO clientes(nombre, email)
VALUES ('Ana Alonso', 'ana@prometeo.com');
```

## Paso 4. Crear sql/users.sql

### **Crear usuarios y asignarles permisos**

```
CREATE USER 'admin_prometeo'@'%' IDENTIFIED BY 'Admin123';
GRANT ALL PRIVILEGES ON prometeo.* TO 'admin_prometeo'@'%';
```

```
CREATE USER 'lector'@'%' IDENTIFIED BY 'Read123';
GRANT SELECT ON prometeo.* TO 'lector'@'%';
```

```
FLUSH PRIVILEGES;
```

- **admin\_prometeo: control total sobre prometeo**
- **lector: solo puede consultar datos**
- **Ambos pueden conectarse desde cualquier IP (%)**

## Paso 5. Crear sql/view.sql

```
USE prometeo;
```

```
CREATE OR REPLACE VIEW v_clientes AS
SELECT nombre, email
FROM clientes;
```

## Paso 6. Crear sql/trigger.sql

```
USE prometeo;
```

```
DELIMITER $$
```

```
CREATE TRIGGER trg_email_no_vacio
BEFORE INSERT ON clientes
FOR EACH ROW
BEGIN
    IF NEW.email IS NULL OR NEW.email = " THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El email no puede estar vacío';
```

```
END IF;  
END$$
```

```
DELIMITER ;
```

## Paso 7. Levantar Docker

En la carpeta docker/:  
cd docker  
docker compose up -d

## Paso 8. Comprobar que funciona

### Entrar a MySQL

```
docker exec -it prometeo-db mysql -uroot -proot
```

### Ver tabla y datos

```
USE prometeo;  
SELECT * FROM clientes;  
SELECT * FROM v_clientes;
```

### Probar trigger (debe fallar)

```
INSERT INTO clientes(nombre, email) VALUES ('Sin Email', "");
```

Debe dar error: *El email no puede estar vacío*

## Para hacerlo online:

### GitHub Codespaces

1. En tu repo: **Code → Codespaces → Create codespace**
2. Cuando abra el VS Code web, abre **Terminal** y ejecuta:

```
cd docker  
docker compose up -d  
docker ps
```

PROMETEO

### 3. Ver logs por si algo falla:

```
docker logs -f prometeo-db
```

#### 4. Entrar a MySQL y comprobar:

```
docker exec -it prometeo-db mysql -uroot -proot
```

Dentro de MySQL:

USE prometeo;

```
SELECT * FROM clientes;
```

```
SELECT * FROM v_clientes;
```

-- Probar trigger (debe fallar)

```
INSERT INTO clientes(nombre, email) VALUES ('Sin Email', "");
```

Si todo está bien, verás el dato de “Ana Alonso”, la vista funcionará y el INSERT sin email dará error.

```
6->3306/tcp, [::]:3306->3306/tcp, 33060/tcp    prometeo-db
@luzmariaalvarez + /workspaces/Sesion14ASIR/docker (main) $ docker exec -it prometeo-db mysql
-uroot -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.8 MySQL Community Server - GPL

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

@luzmariaalvarez + /workspaces/Sesion14ASIR/docker (main) $ docker exec -it prometeo-db mysql
-uroot -proot
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM clientes;
+----+-----+-----+
| id | nombre | email        |
+----+-----+-----+
| 1  | Ana Alonso | ana@prometeo.com |
+----+-----+-----+
1 row in set (0.00 sec)

PROBLEMAS SALIDA TERMINAL ...
@luzmariaalvarez + /workspaces/Sesion14ASIR/docker (main) $ docker exec -it prometeo-db mysql
-uroot -proot
mysql> SELECT * FROM v_clientes;
+-----+-----+
| nombre | email        |
+-----+-----+
| Ana Alonso | ana@prometeo.com |
+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO clientes(nombre, email) VALUES ('Sin Email', '');
ERROR 1644 (45000): El email no puede estar vacío
mysql> 
```