

Aquí tienes un ejercicio integrador que combina los temas 5 (bases de datos documentales en MongoDB) y 6 (programación orientada a componentes), cumpliendo con los criterios de evaluación proporcionados:

Ejercicio Integrador Contexto:

Vas a desarrollar una aplicación que permita gestionar una biblioteca digital utilizando MongoDB como base de datos documental. Además, deberás diseñar componentes reutilizables orientados a gestionar la conexión, las consultas, y las operaciones CRUD (crear, leer, actualizar, eliminar) sobre los datos almacenados.

Requisitos del ejercicio:

- Evaluar tecnologías y diseñar componentes:
- Describe en un documento breve las ventajas e inconvenientes de usar MongoDB para esta aplicación. Justifica por qué MongoDB es adecuado para esta solución.
- Menciona las ventajas de usar programación orientada a componentes en este proyecto.
- Implementar componentes reutilizables:
- Desarrolla un componente en Python para la conexión a MongoDB (deberá manejar credenciales y controlar excepciones).
- Crea un componente para realizar consultas genéricas a la base de datos. Este componente debe admitir condiciones dinámicas.
- CRUD sobre la base de datos:
- Desarrolla un componente que permita:
- Añadir documentos representando libros. Cada libro debe tener un título, autor, género, y un campo opcional como "resumen".
- Modificar la información de un libro específico (por ejemplo, cambiar el resumen o actualizar el género).
- Eliminar libros de la base de datos según condiciones específicas (por ejemplo, todos los libros de un género dado).
- Consultar libros con filtros dinámicos y una proyección que solo muestre ciertos campos (por ejemplo, título y autor).
- Gestión de colecciones:
- Implementa un componente que permita crear y eliminar colecciones.
- Mapeo objeto-relacional:
- Diseña un modelo de clase para representar libros en Python y transforma los documentos de MongoDB a instancias de esta clase utilizando un componente específico.
- Pruebas y documentación:
- Prueba todos los componentes y documenta los resultados. Asegúrate de manejar errores de conexión, operaciones fallidas, y mostrar mensajes adecuados al usuario.
- Escribe un manual breve para explicar cómo integrar estos componentes en otras aplicaciones.

**Instalar mongodb community ubuntu**

Tener instalado docker e introducir el siguiente comando en la cmd

- docker run -d -p 27017:27017 --name mongodb + nombre de la imagen:mongodb-community-server

docker run -d -p 27017:27017 --name mongodb mongodb-community-server

### **Entrar en su terminal**

mongosh

docker exec -it mongodb mongosh

### **Crear base de datos, y configurar usuario y contraseña**

use 2dam

```
db.createUser({
  user: "usuario",
  pwd: "usuario", // Cambia por una contraseña segura
  roles: [ { role: "readWrite", db: "1dam" } ]
})
```

### **Select a mi tabla motocicletas**

db.motocicletas.find().pretty()

### **Arrancar mongodb**

docker start mongodb

docker exec -it mongodb mongosh

docker start mongodb/mongodb-community-server

show databases

use 2dam

show tables

db.motocicletas.drop()

db.motocicletas.find().pretty()