Sección 1: Preguntas Teóricas

1. Optimización de consultas

Para optimizar una consulta lenta, revisaría el plan de ejecución, usaría índices en columnas filtradas, evitaría SELECT * y solo consultaría los campos necesarios. También limitaría resultados cuando sea posible.
Para cachear resultados frecuentes, usaría memoria o archivos temporales para guardar las respuestas y evitar ejecutar la misma consulta varias veces.

2. Seguridad de datos

Para prevenir inyección SQL, usaría consultas preparadas, validaría los datos de entrada y evitaría concatenar valores directamente en las consultas.
Las credenciales de la base de datos las almacenaría en variables de entorno o archivos seguros fuera del código, con acceso restringido.

3. Automatización

Usaría patrones como repositorio para el acceso a datos, servicios para la lógica y un programador de tareas para ejecutar procesos automáticamente.
Los reintentos se implementarían con un número máximo de intentos y tiempos de espera entre fallos, registrando los errores para control y seguimiento.

Ejercicio 3: Optimización

Rendimiento

- Usar índices en columnas filtradas.
- Evitar SELECT * y traer solo lo necesario.
- Limitar resultados y reutilizar consultas frecuentes.

Lectura de resultados

- Validar si hay datos antes de procesar.
- Procesar por bloques si son muchos registros.

Mantenibilidad

- Separar el código por funciones o módulos.
- Usar configuración externa .env, JSON.

Documentar y manejar errores de forma centralizada.