**Ejercicios Conceptuales**

1. **Beneficios de los patrones de diseño**
   * **Estandarización**: Permite que diferentes desarrolladores trabajen con un mismo enfoque estructurado.  
     ***Ejemplo****:* En un equipo de desarrollo grande, el uso del patrón *Singleton* para la configuración de la base de datos evita múltiples instancias y problemas de concurrencia.
   * **Reutilización**: Evita la necesidad de reinventar soluciones para problemas comunes.  
     ***Ejemplo****:* Un sistema de autenticación puede utilizar el patrón *Factory Method* para crear distintos tipos de autenticación (OAuth, JWT, LDAP).
   * **Mantenibilidad**: Reduce el acoplamiento y mejora la organización del código.  
     ***Ejemplo****:* En un sistema de notificaciones, usar *Observer* permite que los módulos de email, SMS y push notifications sean independientes.
2. **Identificación de patrones en código existente**
3. **Clasificación de patrones y cuándo usarlos**
   * **Creacionales**: Cuando se necesita flexibilidad en la creación de objetos sin acoplamiento directo. (*Ejemplo: Factory Method en sistemas de pago para diferentes pasarelas*).
   * **Estructurales**: Para organizar mejor la comunicación entre clases y simplificar interacciones. (*Ejemplo: Facade en un sistema de reservas como EasyBook*).
   * **De comportamiento**: Cuando se debe gestionar la comunicación y flujo de datos entre objetos. (*Ejemplo: Observer en sistemas de eventos en tiempo real*).