

ACTIVIDAD 2

Para desarrollar nuestra plataforma de gestión de proyectos con inteligencia artificial hemos elegido tres patrones de diseño que ayudan a que el sistema sea más fácil de usar y de mantener. Cada patrón nos ayuda a resolver problemas concretos y a evitar errores comunes en el desarrollo. El primero es el Factory Method que es un patrón creacional y permite crear objetos sin depender de clases concretas. Por ejemplo si necesitamos generar distintos tipos de notificaciones como correos electrónicos, SMS o avisos dentro de la aplicación el Factory Method centraliza toda la lógica de creación de manera que cuando se añada un nuevo tipo de notificación solo tendremos que ampliar la fábrica sin cambiar el código base. Es muy importante que este método se mantenga sencillo y bien documentado para evitar acoplamientos innecesarios ya que un error frecuente es mezclar la lógica de negocio con la creación de objetos.

El segundo patrón es el Adapter que es de tipo estructural y se utiliza para integrar sistemas que en principio tienen interfaces incompatibles. Por ejemplo si la plataforma debe conectarse con un servicio externo de almacenamiento en la nube y su API es diferente a la que usamos nosotros el Adapter actúa como un traductor permitiendo que ambos sistemas se comuniquen sin tener que modificar sus implementaciones originales. Una buena práctica es diseñar el adapter de forma genérica para minimizar posibles errores en la conversión de datos.

El tercer patrón es el Observer que es un patrón de comportamiento y gestiona la comunicación basada en eventos. En nuestro caso cuando se actualiza el estado de un proyecto o se termina una tarea el Observer notifica a todos los módulos y usuarios suscritos permitiendo que se actualicen en tiempo real sin acoplar los componentes del sistema. Por ejemplo se puede actualizar automáticamente el panel de control de cada usuario y es muy importante gestionar correctamente las suscripciones y las cancelaciones para evitar problemas como fugas de memoria.

En resumen la adopción de Factory Method, Adapter y Observer se justifica porque ayudan a crear objetos de forma flexible, a integrar sistemas externos de forma limpia y a gestionar las notificaciones de forma eficiente. Estos patrones no solo mejoran la organización del código sino que también facilitan la evolución del sistema asegurando una plataforma robusta y preparada para futuros cambios. Con estos patrones nuestra plataforma estará lista para adaptarse a nuevas necesidades y evitará muchos problemas que se suelen dar en otros proyectos.