Actividad 2: Identificación y Justificación de Patrones de Diseño

En la búsqueda de el desarrollo de una arquitectura escalable mantenible y flexible, es importante implementar ciertos patrones de diseño en el desarrollo. He escogido tres de estos que nos van a solucionar varios problemas a lo largo del desarrollo:

Factory Method (Patrón Creacional):

Permite la creación de objetos a través de una interfaz común para todos ellos, y delega la instanciación de estos a las subclases. En el caso de este proyecto, se usará para generar los diferentes tipos de proyectos según las necesidades del usuario lo que garantiza la flexibilidad en la creación de objetos sin depender de implementaciones concretas.

Adapter (Patrón Estructural):

Actúa como un puente entre interfaces incompatibles, permitiendo que clases con diferentes estructuras trabajen juntas sin modificar su código interno. Esto es importante en nuestro caso ya que nos facilitara la interoperabilidad entre herramientas de terceros sin necesidad de modificar su código.

Observer (Patrón de Comportamiento):

Define una especie de suscripción entre objetos, en la cual los observadores son notificados cuando el sujeto cambia su estado. Esto nos servirá para gestionar notificaciones en tiempo real.

Con estas herramientas respondemos a la necesidad de diseñar una arquitectura flexible y extensible. Factory Method facilita la creación de objetos, Adapter permite la interoperabilidad, y Observer habilita la comunicación entre objetos en tiempo real.