

## **Identificación y Justificación de Patrones de Diseño**

**Para la arquitectura de la plataforma de gestión de proyectos, seleccionamos los siguientes patrones de diseño, cada uno perteneciente a una categoría distinta:**

### **1. Patrón Creacional: Factory Method**

#### **Descripción:**

**El Factory Method permite la creación de objetos sin especificar la clase exacta a instanciar, delegando esta responsabilidad a subclases.**

#### **Justificación:**

**En nuestra plataforma, este patrón se aplicará en la creación de tareas y usuarios. Por ejemplo, podríamos tener una fábrica que genere instancias de TareaSimple, TareaCompleja o TareaConIA según los requerimientos del proyecto. Esto facilitará la extensibilidad y mantenimiento del código.**

---

### **2. Patrón Estructural: Facade**

#### **Descripción:**

**Facade proporciona una interfaz simplificada para un conjunto de interfaces en un subsistema, reduciendo la complejidad para los clientes.**

#### **Justificación:**

**Este patrón se empleará para unificar el acceso a servicios internos, como la gestión de proyectos, tareas y usuarios. Una fachada central podría ofrecer una interfaz sencilla para acceder a la lógica interna sin exponer la complejidad del sistema subyacente.**

---

### **3. Patrón de Comportamiento: Observer**

#### **Descripción:**

**El patrón Observer permite que varios objetos sean notificados de cambios en otro objeto sin acoplamiento directo.**

#### **Justificación:**

**Se aplicará en la gestión de notificaciones. Por ejemplo, cuando una tarea cambie de estado, los usuarios suscritos (como el responsable o el administrador del proyecto) recibirán una notificación de manera automática, garantizando una comunicación eficiente y en tiempo real.**