

Problema 1: Un cliente requiere utilizar sendgrid para envíos de email, pero otro cliente requiere enviarlos por mandril. Se quiere evitar el uso de IF, y realizar un diseño en donde podamos utilizar más servicios en caso de que un cliente requiera alguno en específico ¿Qué patrón de diseño utilizarías y por qué?

Opción 1: Strategy

Opción 2: Factory Method

Opción 3: Adapter.

Respuesta: Para mí el patrón de diseño más apropiado sería la opción 1: Strategy. Con este patrón de diseño puedes definir una familia de algoritmos, los encapsulas y los haces intercambiables. Esto permite que el usuario pueda elegir el algoritmo apropiado en tiempo de ejecución sin necesidad de condicionales.

Problema 2: Explica en tus propias palabras la diferencia entre Factory Method y Abstract Factory. Y proporciona un caso de uso.

Estos dos patrones de diseño son diferentes, pero ambos están relacionados con la creación de objetos.

Factory Method:

Este patrón define una interfaz para crear un objeto, pero deja que cada clase modifique el tipo de objetos que se crearán. La creación de los objetos se delega a las clases. La interfaz proporciona el patrón, que las clases deben de implementar para la creación de instancias.

Caso de Uso:

Supongamos que tienes una aplicación de procesamiento de documentos. Puedes tener una clase base DocumentProcessor que define un método abstracto llamado createDocument. Las subclases concretas (como PDFDocumentProcessor y WordDocumentProcessor) implementarán este método para crear instancias específicas de documentos según sus necesidades.

Abstract Factory:

Es una interfaz que crea familias de objetos relacionadas sin especificar sus clases. Permite que las clases proporcionen implementaciones específicas de estas familias.

Caso de Uso:

Imagina que estás construyendo un juego y necesitas manejar la creación de personajes y sus armas. Podrías tener una interfaz AbstractFactory que define métodos para crear personajes y armas. Luego, tendrías fábricas concretas que implementan esta interfaz para crear personajes y armas específicos para diferentes épocas o mundos del juego.