Ejemplo: Juego de cocina mágica

Imaginá un juego donde el personaje principal es un chef mágico. Cada vez que el jugador elige una receta misteriosa, se genera automáticamente un plato mágico diferente: puede ser una Pizza Voladora, una Hamburguesa Explosiva, o una Sopa Infinita.

El chef no necesita saber cómo se cocinan ni qué ingredientes tienen estos platos. Solo necesita que el plato tenga una función común: servir(), que hace que el plato aparezca en la mesa con su efecto especial.

¿Cómo entra el patrón Factory Method?

- El chef es el CreationRequester: es quien necesita platos mágicos, pero no sabe cómo se hacen ni qué tipo va a recibir.
- El chef puede tener acceso a una o varias cocinas mágicas, cada una representada por un objeto que implementa la interfaz MagicKitchenFactory (FactoryIF). Esta interfaz declara un único metodo: createProduct().
- Esa interfaz es implementada por una única clase concreta llamada MagicKitchen. Esta clase es la Factory, y es la única responsable de decidir qué plato hacer (puede elegir al azar, según el nivel del chef, etc.).
- Todos los platos mágicos comparten una interfaz llamada MagicDish. Esta es la ProductIF, y define el método servir() que todos los platos deben implementar.
- Cada plato mágico concreto (por ejemplo, FlyingPizza, ExplodingBurger, EndlessSoup) implementa la interfaz MagicDish y tiene su propia versión de servir().

De este modo, el chef nunca conoce los tipos concretos de plato, solo se comunica con ellos a través de la interfaz MagicDish. Puede invocar cualquier cocina mágica disponible (una, varias o ninguna, según el caso) para crear un plato específico. La clase MagicKitchen se encarga de crear los platos, manteniendo el sistema desacoplado, extensible y fácil de mantener.