

# Factory Method

## Propósito

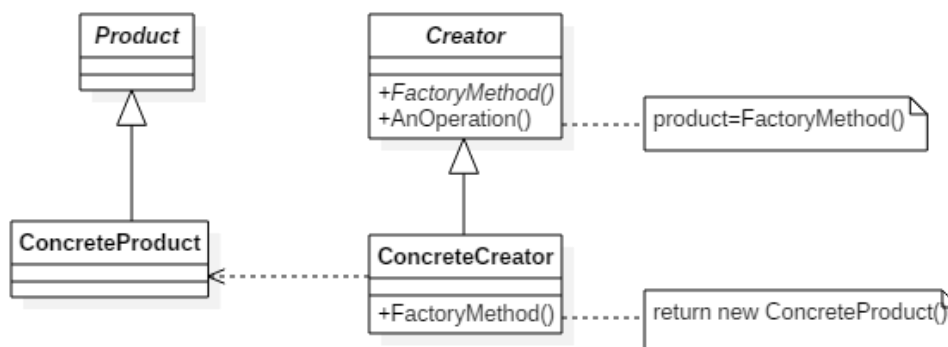
Define una interface para crear un objeto, pero deja que las subclases decidan qué clase instanciar. El Factory Method permite que una clase posponga la instanciación a las subclases.

## Aplicabilidad

El Factory Method se utiliza cuando:

- Una clase no puede anticipar la clase de los objetos que debe crear
- Una clase prefiere que sean las subclases las que especifiquen los objetos que crea
- Las clases delegan la responsabilidad a una o varias subclases auxiliares, y se desea centralizar el conocimiento sobre qué clase auxiliar se delega

## Estructura



## Participantes

- **Product**: define la interface de los objetos que crea el método de fabricación
- **ConcreteProduct**: implementa la interface **Product**
- **Creator**: declara el método de fabricación, que devuelve un objeto de tipo **Product**. **Creator** también puede proporcionar una implementación predeterminada del método de fabricación que devuelve un objeto **ConcreteProduct** predeterminado.
- **ConcreteCreator**: sobrescribe el método de fabricación para devolver una instancia de **ConcreteProduct**

## Colaboraciones

La clase **Creator** depende de sus subclases para definir el método de fabricación de modo que devuelva la instancia adecuada del **ConcreteProduct**.

## Consecuencias

Los métodos de fabricación eliminan la necesidad de enlazar en el código las clases específicas de aplicación. El código solamente conoce la interface de **Product**, con lo cual podría trabajar con cualquier clase **ConcreteProduct** definida por el usuario.

Una posible desventaja de los métodos de fabricación es que los clientes se ven obligados a subclasificar Creator sólo para crear un objeto ConcreteProduct particular. La subclasificación no es un problema si el cliente iba a subclasificar Creator (necesitaba un ConcreteCreator), de otro modo ahora el cliente se ve obligado a tratar dos puntos de evolución.

Otras dos consecuencias adicionales del patrón Factory Method:

1. Proporciona “enganches” para las subclases. Crear objetos dentro de una clase con un método de fabricación es siempre más flexible que crear el objeto directamente. El Factory Method proporciona a las subclases un enganche para proporcionar una versión extendida de un objeto.
2. Conecta jerarquías paralelas de clases. Las jerarquías paralelas de clases aparecen cuando una clase delega alguna de sus responsabilidades en una clase separada. Supongamos que tenemos una jerarquía de figuras geométricas que se pueden manipular interactivamente (desplazar, mover, rotar, etc). La implementación de esas operaciones puede ser complejo porque requiere almacenar y actualizar información que registre el estado de la manipulación en un momento determinado. Ese estado solo se necesita durante la manipulación, por lo que no es necesario almacenarlo en el objeto figura. Con lo cual, podría ser interesante separar toda la interacción y manipulación en un objeto diferente, de modo que diferentes figuras tendrán su propio manipulador (el objeto Linea tendrá su propio ManipularLinea).