

# Template Method

## Propósito

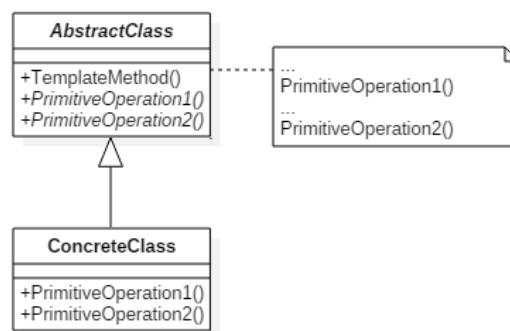
Define el esqueleto de un algoritmo en una operación, dejando que las subclases definan algunos de los pasos. El Template Method deja que las subclases redefinan ciertos pasos de un algoritmo sin variar la estructura del algoritmo.

## Aplicabilidad

El patrón Template Method se utiliza:

- Para implementar las partes invariantes de un algoritmo una sola vez y dejar que las subclases redefinan el comportamiento que puede variar
- Cuando factorizamos comportamiento común de varias subclases para evitar duplicación de código. Es un buen ejemplo de “refactoring por generalización”. Primero hay que identificar las diferencias, para luego separarlas en operaciones nuevas. Finalmente, reemplazas el código diferente con un método plantilla que llama a las nuevas operaciones.
- Para controlar las extensiones de las subclases. Se puede definir un Template Method que usa operaciones “gancho” en puntos específicos, permitiendo así extensiones sólo en esos puntos.

## Estructura



## Participantes

- **AbstractClass**
  - o Define las operaciones primitivas abstractas que las subclases deben definir para implementar los pasos de un algoritmo
  - o Implementa un método plantilla que define el esqueleto de un algoritmo. El Template Method llama tanto a las operaciones primitivas como otras operaciones definidas en **AbstractClass**
- **ConcreteClass**: implementa las operaciones primitivas que permiten llevar a cabo los pasos específicos-de-la-subclase del algoritmo

## Colaboraciones

- La ConcreteClass depende de AbstractClass para implementar las partes invariantes del algoritmo.

## Consecuencias

Los Template Method son una técnica fundamental para reutilizar código. Son particularmente importantes al implementar bibliotecas de clases (librerías), porque son el medio habitual de factorizar comportamiento común.

Los Template Method promueven el uso de una estructura de control invertido que en ocasiones se conoce como “el principio de Hollywood”, esto es, “no nos llame, nosotros le llamaremos”. Esto se refiere a cómo la clase padre llama las operaciones de las subclases y no al revés.

Los Template Method llaman a los siguientes tipos de operaciones:

- Operaciones concretas: definidas en ConcreteClass o en otras clases cliente
- Operaciones concretas definidas en AbstractClass
- Operaciones primitivas (abstractas o virtuales)
- Operaciones “gancho”, que proporcionan un comportamiento predeterminado y que las subclases lo pueden extender si lo consideran oportuno. Una operación “gancho” no hace nada de forma predeterminada

Es importante que los Template Method especifiquen qué operaciones son “gancho” (se pueden sobrescribir) y cuáles son abstractas (se deben sobrescribir). Para reutilizar una clase abstracta de forma efectiva hay que tener claro qué operaciones se han diseñado para que se sobrescriban.