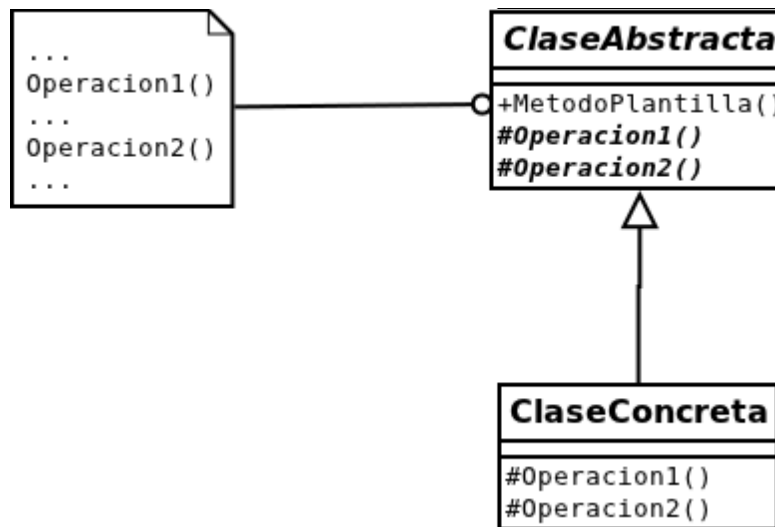


# Introduccion

El patrón Template Method (Método Plantilla o Método Modelo) es un patrón de diseño enmarcado dentro de los llamados patrones de comportamiento, que se caracteriza por la definición, dentro de una operación de una superclase, de los pasos de un algoritmo, de forma que todos o parte de estos pasos son redefinidos en las subclases herederas de la citada superclase.



El *template method* define el esqueleto del programa de un algoritmo. Uno o mas de los pasos del algoritmo pueden ser sobreescritos por las subclases para implementar diferentes comportamientos a la vez que nos aseguramos que el algoritmo principal es respetado.

Primero se crea una clase que provee los pasos basicos del algoritmo deseado. Estos pasos se implementan usando metodos abstractos. Mas tarde, las subclases cambian los metodos abstractos para implementar acciones reales. De esta manera el algoritmo principal se mantiene intacto en un lugar pero los pasos concretos pueden ser cambiados en las subclases.

El template method maneja en general las tareas a realizar. Este llama metodos abstractos y no abstractos para realizar su objetivo. Los metodos no abstractos son controlados completamente por el template method pero los metodos abstractos, implementados en las subclases son los que dan la flexibilidad y libertad de accion necesaria. Algunos o todos los metodos abstractos pueden especializarse para cada subclase permitiendo al autor de la subclase dotar

de un comportamiento particular sin tener que modificar el algoritmo a gran escala. El template method (que es no abstracto) permanece sin modificaciones, asegurando que los metodos abstractos y no abstractos son llamados en la secuencia originalmente planeada.

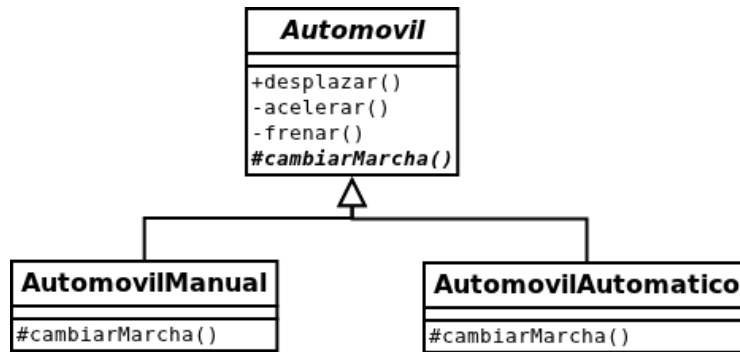
El template method se usa frecuentemente, al menos en su implementacion mas simple, cuando un metodo llama solo a un metodo abstracto. Si un programador usa un metodo polimorfo este patron de diseño es una consecuencia natural. Esto sucede porque un metodo llamando a un metodo abstracto o una funcion polimorfica es simplemente la razon de ser de el metodo abstracto/polimorfico. El template method puede ser usado para agregar valor inmediato al software o con vision pensando en mejoras futuras.

## Uso

El template method es usado para:

- permitir a las subclases implementar (sobreescribiendo metodos) un comportamiento que puede variar
- evitar duplicar codigo: el flujo general es implementado una vez en la clase abstracta y las variaciones necesarias son implementadas en cada subclase
- controlar hasta que punto se puede usar subclases. Al contrario de en una simple sobreescritura polimorfica donde el metodo es completamente sobreescrito, solo los detalles especificos del flujo pueden ser cambiados.

La estructura de control (inversion de control) que es el resultado de la aplicacion de el template pattern is a menudo mencionada como el Principio de Hollywood "No nos llames, nosotros te llamamos." Usando este principio, el template method en una clase padre controla el proceso llamando los metodos de las subclases como sea requerido.



```
public abstract class Automovil
{
    public void desplazar()
    {
        acelerar();
        cambiarMarcha();
        frenar();
    }
    private void acelerar()
    {
        System.out.println("Acelerando...");
    }

    private void frenar()
    {
        System.out.println("Frenando...");
    }

    protected abstract void cambiarMarcha()
}

public class AutomovilManual extends Automovil
{
    protected void cambiarMarcha()
    {
        System.out.println("Cambiando de marcha de forma manual");
    }
}

public class AutomovilAutomatico extends Automovil
{
    protected void cambiarMarcha()
    {
        System.out.println("Cambiando de marcha de forma automática");
    }
}
```