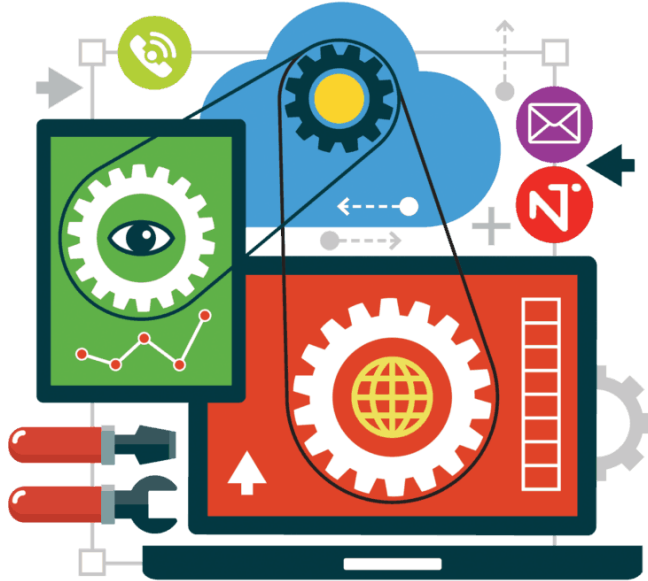


---

## Acceso a datos

Diego Calatayud Muñoz  
2º DAM  
IES Poeta Paco Mollá

# Patrones de diseño del Software



<b>INTRODUCCIÓN</b>	<b>1</b>
<b>DESCRIPCIÓN</b>	<b>2</b>
<b>TIPOS</b>	<b>2</b>
PATRONES CREACIONALES	3
PATRONES ESTRUCTURALES	3
PATRONES DE COMPORTAMIENTO	3
<b>VENTAJAS DEL USO DE PATRONES</b>	<b>3</b>
<b>BIBLIOGRAFÍA</b>	<b>4</b>

## INTRODUCCIÓN

El objetivo principal de la práctica es definir el concepto de “patrón de diseño de software” y repasar los tipos existentes. Además, también se comentarán las ventajas de su uso.

## DESCRIPCIÓN

Los patrones de diseño del software son una solución general, reutilizable y aplicable a diferentes problemas que pueden surgir a la hora de desarrollar un software.

De forma simplificada se pueden definir como plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas para problemas generales. Esto es gracias a que los desarrolladores se han enfrentado a ellos durante años a través de prueba y error.

## TIPOS

En el mundo de la informática, existen una gran variedad de patrones de diseño del software. Sin embargo, estos se agrupan en las siguientes 3 categorías principales que agrupan el resto de patrones.



## PATRONES CREACIONALES

Los patrones creacionales proporcionan mecanismos de creación de objetos para aumentar la flexibilidad y la posible reutilización del código.

Dentro de los patrones creacionales los más usados son:

- ❖ **Método de fábrica:** Crea objetos con una interfaz común.
- ❖ **Constructor:** Separa paso a paso la construcción y representación de objetos.
- ❖ **Singleton:** Restringe la creación de objetos para una clase a una sola instancia.

## PATRONES ESTRUCTURALES

Los patrones estructurales facilitan soluciones eficientes con respecto de la estructura de los objetos y las clases. Se hace uso del concepto de herencia para componer interfaces y objetos que permitan obtener nuevas funcionalidades.

El patrón estructural más importante es:

- ❖ **Adaptador:** Define cómo adaptar una interfaz a la de otra clase existente para permitir que las interfaces incompatibles trabajen de forma conjunta.

## PATRONES DE COMPORTAMIENTO

Este tipo de patrones se encarga de la comunicación entre objetos de la clase. Se utilizan para detectar patrones de comunicación ya presentes y manipularlos.

Los patrones de comportamiento más importantes son:

- ❖ **Observador:** Define cómo se han de notificar a los objetos los cambios en otros objetos.
- ❖ **Estrategia:** Permite agrupar algoritmos relacionados bajo una clase.

## VENTAJAS DEL USO DE PATRONES

Como hemos podido observar, el uso de patrones de diseño del software permite a los programadores ser mucho más eficientes, ya que se definen estándares para poder solucionar problemas muy comunes.

Otra de las principales ventajas de su uso es que estandarizan en cierta manera algunos aspectos de la programación. Esto permite que el código sea más legible por otros programadores.

## BIBLIOGRAFÍA

Para la realización de este trabajo se han consultado las siguientes páginas web.

- ❖ [Profile](#)
- ❖ [NetSolutions](#)
- ❖ [Dev.To](#)