

**Hablamos de**

# **PATRONES DE DISEÑO DE SOFTWARE**



# ¿Qué son?

Un patrón es una solución general y reutilizable aplicable a diferentes problemas de diseño de software. Son plantillas con soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error.



# ¿Por qué usarlos?

Los patrones de diseño de software ayudan a estar seguro de la validez del código, ya que son soluciones que funcionan y han sido probadas por muchísimos desarrolladores, siendo menos propensas a errores.



# Tipos de patrones

Hay un total de 23 patrones de diseño individuales principales, integrados en tres categorías fundamentales:

1. Patrones creacionales
2. Patrones estructurales
3. Patrones de comportamiento

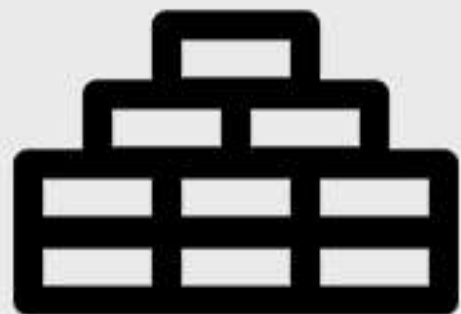
# 1. Patrones creacionales

Proporcionan diversos mecanismos de creación de objetos, que aumentan la flexibilidad y la reutilización del código existente de una manera adecuada a la situación. Algunos son: Abstract Factory, Builder Patterns y Factory Method.



## 2. Patrones estructurales

Facilitan soluciones y estándares eficientes con respecto a las composiciones de clase y las estructuras de objetos. El concepto de herencia se utiliza para componer interfaces y definir formas de componer objetos. Destacan Adapter, Bridge y Composite, entre otros.



### **3. Patrones de comportamiento**

Se ocupan de la comunicación entre objetos de clase. Chain of responsibility, Command e Interpreter son algunos de ellos.



# Resumen





→ Más info:

[www.profile.es](http://www.profile.es)



MADRID BARCELONA SEVILLA



*Imágenes: Diseñadas by Freepik*