

Patrones de diseño creacionales



Proporcionan mecanismos de creación de objetos que incrementan la flexibilidad y la reutilización de código existente. Tienen por objetivo hacer un mejor uso de los recursos y brindar una metodología estructurada para trabajar una serie de problemas interrelacionados.

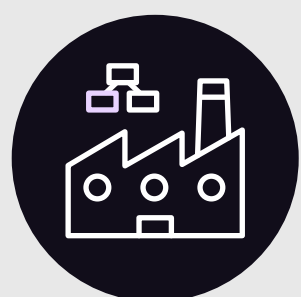
Los principales patrones de diseño creacionales son:



Singleton: nos permite crear una única instancia de una clase y a su vez proporciona un punto de acceso global a dicha instancia, pero puede ser visto como un antipatrón a su vez.



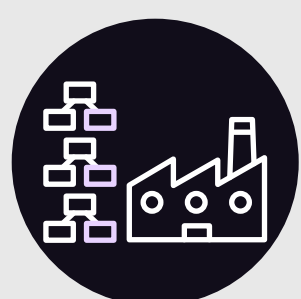
Prototype: permite clonar objetos sin que nuestro código dependa de sus clases.



Factory Method: proporciona una interfaz para crear objetos en una superclase mientras permite a las subclasses alterar el tipo de objetos que se crean.



Builder: permite construir objetos complejos paso a paso. Permite producir diferentes tipos y representaciones de un objeto usando el mismo código de construcción.



Abstract Factory: permite encapsular un grupo de objetos relacionados sin especificar sus clases concretas. Útil cuando se quiere utilizar una única interfaz para crear familias de objetos relacionados, sin tener que especificar las clases de los mismos.

Algunos desafíos:



Implementación compleja: que puede dificultar la comprensión.



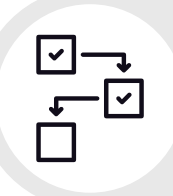
Sobrecarga de código: puede añadir clases y abstracciones innecesarias.



Flexibilidad vs simplicidad: lograr el equilibrio puede complicar el código.



Escalabilidad: problemas en sistemas muy grandes.



Dependencia de la estructura de clases: puede dificultar modificaciones futuras.



Compatibilidad con otros patrones: integrar con otros patrones puede ser complejo.



Links de referencia

1



Cursos relacionados



Más recursos

