

Patrón	Descripción
Abstract Factory	Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas.
Adapter	Convierte la interfaz de una clase en otra interfaz que los clientes esperan. Permite que las clases trabajen juntas aunque tengan interfaces incompatibles.
Bridge	Desacopla una abstracción de su implementación para que ambas puedan variar independientemente.
Builder	Separa la construcción de un objeto complejo de su representación, permitiendo que el mismo proceso de construcción pueda crear representaciones diferentes.
Chain of Responsibility	Evita el acoplamiento entre el remitente de una solicitud y su receptor, dándole a más de un objeto la oportunidad de manejar la solicitud. Pasa la solicitud a lo largo de una cadena de objetos hasta que uno la maneje.
Command	Encapsula una solicitud como un objeto, permitiendo la parametrización de clientes con diferentes solicitudes, la encola o registra solicitudes y admite operaciones deshacer.
Composite	Compone objetos en estructuras de árbol para representar jerarquías parte-todo. Permite que los clientes traten objetos individuales y composiciones de objetos de manera uniforme.
Decorator	Adjunta responsabilidades adicionales a un objeto de manera dinámica. Los decoradores ofrecen una alternativa flexible a la subclase para extender la funcionalidad.
Facade	Proporciona una interfaz unificada para un conjunto de interfaces en un subsistema. Define una interfaz de nivel superior que facilita el uso del subsistema.
Factory Method	Define una interfaz para crear un objeto, pero permite que las subclases decidan qué clase instanciar. Permite que una clase difiera la instanciación a sus subclases.
Flyweight	Utiliza el intercambio para admitir eficientemente un gran número de objetos finos y pequeñas porciones de estado.
Interpreter	Dado un lenguaje, define una representación para su gramática junto con un intérprete que utiliza la representación para interpretar oraciones en el lenguaje.
Iterator	Proporciona una manera de acceder a los elementos de un objeto agregado secuencialmente sin exponer su representación interna.
Mediator	Define un objeto que encapsula cómo interactúan un conjunto de objetos. Promueve un acoplamiento débil al evitar que los objetos se refieran explícitamente entre sí.
Memento	Sin violar la encapsulación, captura y externaliza el estado interno de un objeto para que pueda restaurarse a este estado más tarde.
Observer	Define una dependencia uno a muchos entre objetos para que cuando un objeto cambie de estado, todos sus dependientes sean notificados y actualizados automáticamente.
Prototype	Especifica los tipos de objetos a crear utilizando una instancia prototípica y crea nuevos objetos copiando este prototipo.

Patrón	Descripción
Proxy	Proporciona un sustituto o espacio reservado para otro objeto para controlar el acceso a él.
Singleton	Asegura que una clase tenga solo una instancia y proporciona un punto de acceso global a ella.
State	Permite que un objeto altere su comportamiento cuando cambia su estado interno, haciendo que el objeto parezca cambiar de clase.
Strategy	Define una familia de algoritmos, encapsula cada uno y los hace intercambiables. Permite que el algoritmo varíe independientemente de los clientes que lo usan.
Template Method	Define el esqueleto de un algoritmo en una operación, delegando algunos pasos a las subclasses. Permite que las subclasses redefinan ciertos pasos sin cambiar la estructura del algoritmo.
Visitor	Representa una operación a realizar en los elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera.