



ESCUELA DE  
INGENIERÍA EN CIENCIAS Y SISTEMAS  
FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



**Día, Fecha:**

Miércoles, 21/08/2024

**Hora de inicio:**

15:40

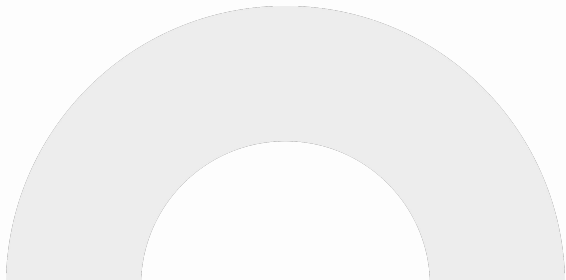
# Análisis y Diseño de Sistemas 2 [B]

Luis Angel Barrera Velásquez



# AGENDA

1	Avisos
2	Ejemplo Práctico patrones de diseño estructurales
3	Teoría Patrones de diseño de comportamiento
4	Ejemplo Práctico de Patrones de diseño de comportamiento



# Patrones de diseño de comportamiento



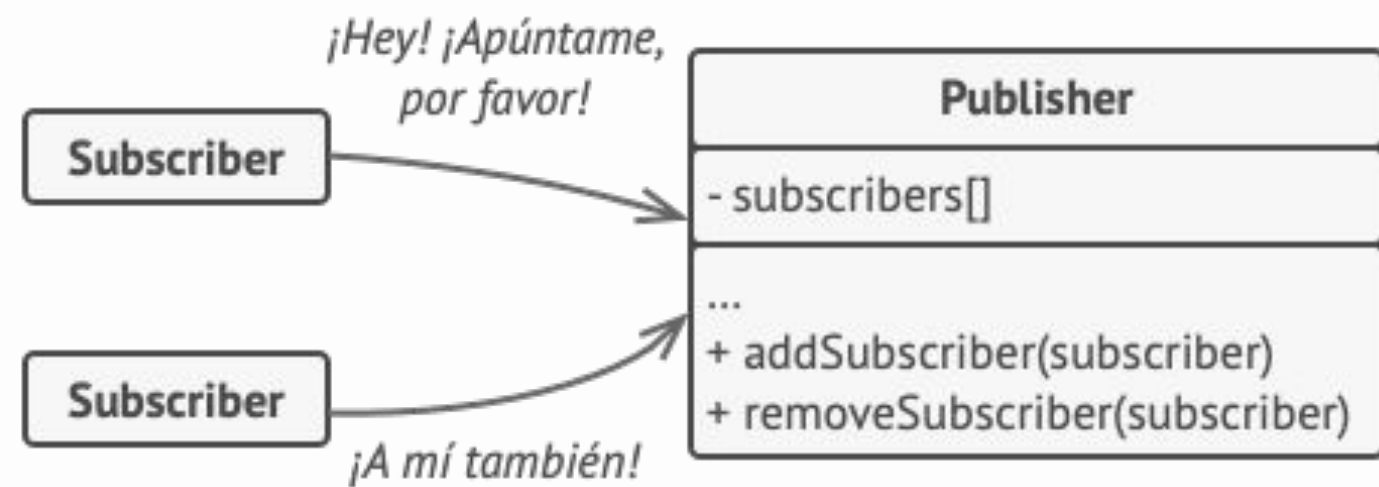
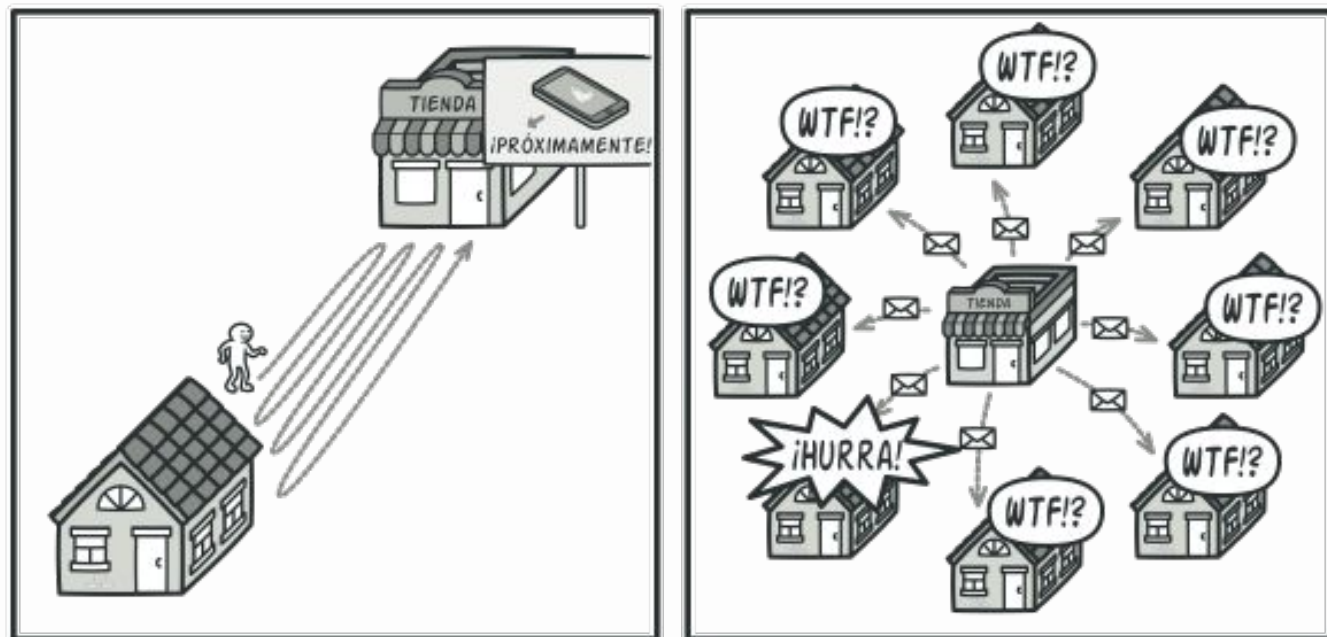
# Patrones de comportamiento

Los patrones de comportamiento tratan con algoritmos y la asignación de responsabilidades entre objetos.



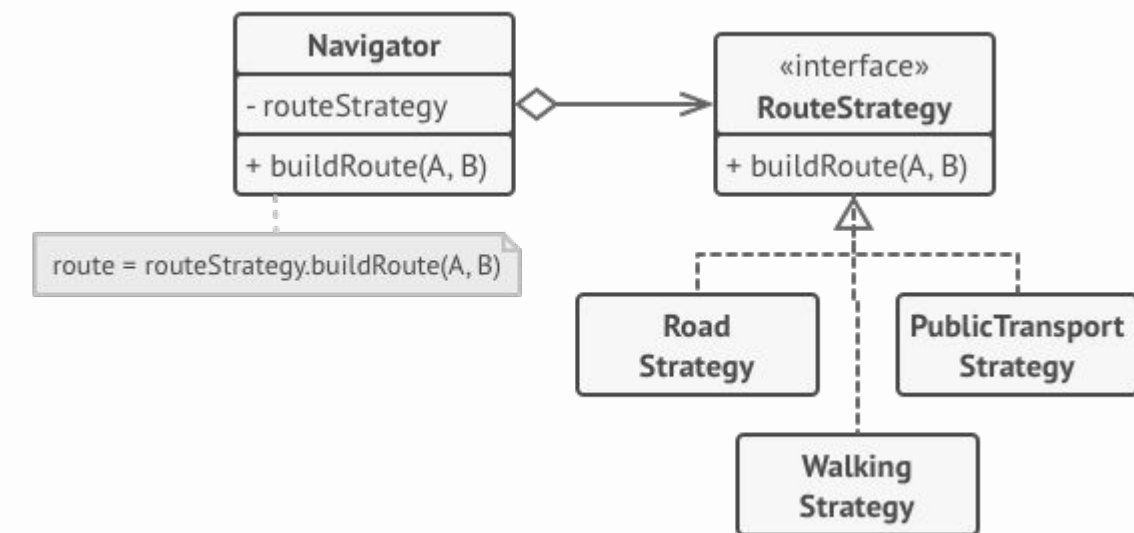
# Patrón Observer

es un patrón de diseño de comportamiento que te permite definir un mecanismo de suscripción para notificar a varios objetos sobre cualquier evento que le suceda al objeto que están observando.



# Patrón Strategy

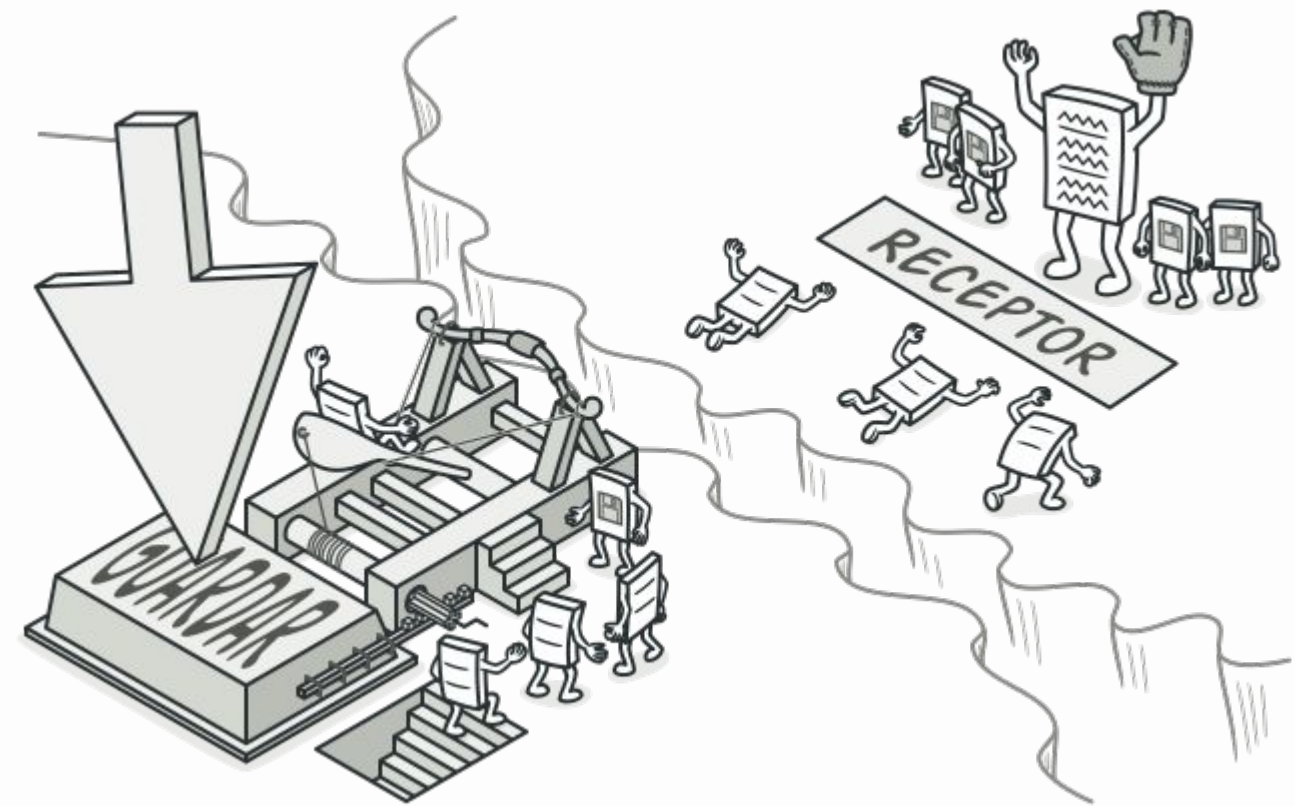
Strategy es un patrón de diseño de comportamiento que te permite definir una familia de algoritmos, colocar cada uno de ellos en una clase separada y hacer sus objetos intercambiables.





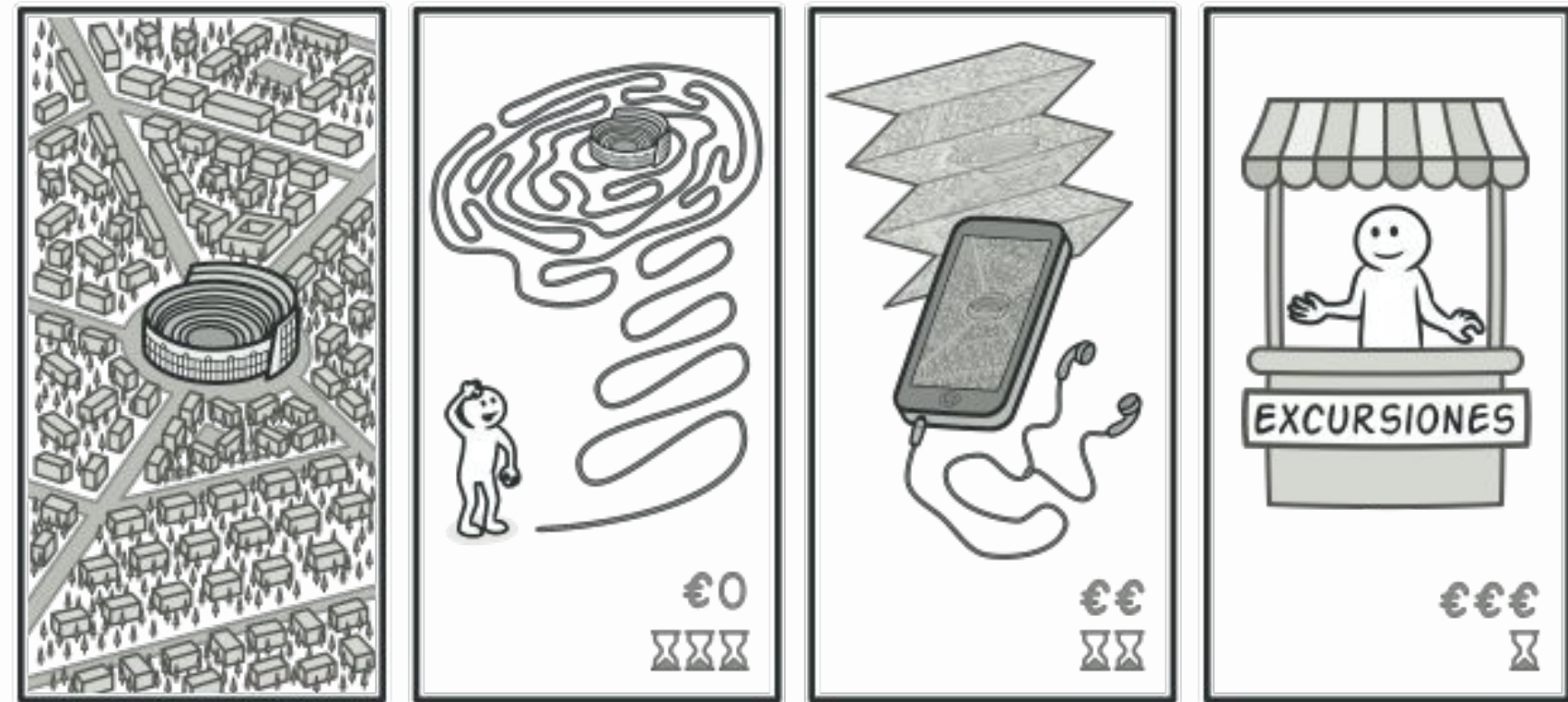
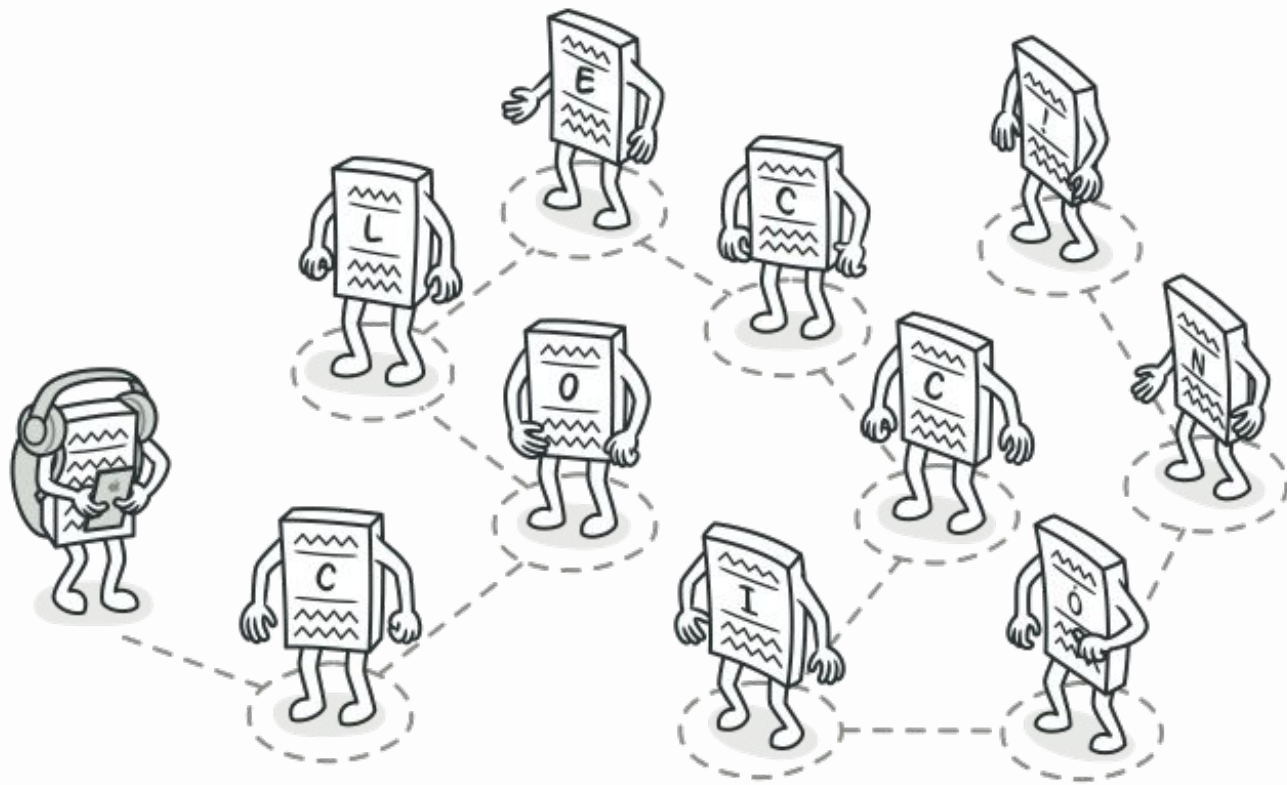
# Patrón Command

Factory Method es un patrón de diseño creacional que proporciona una interfaz para crear objetos en una superclase, mientras permite a las subclases alterar el tipo de objetos que se crearán.



# Patrones Iterator

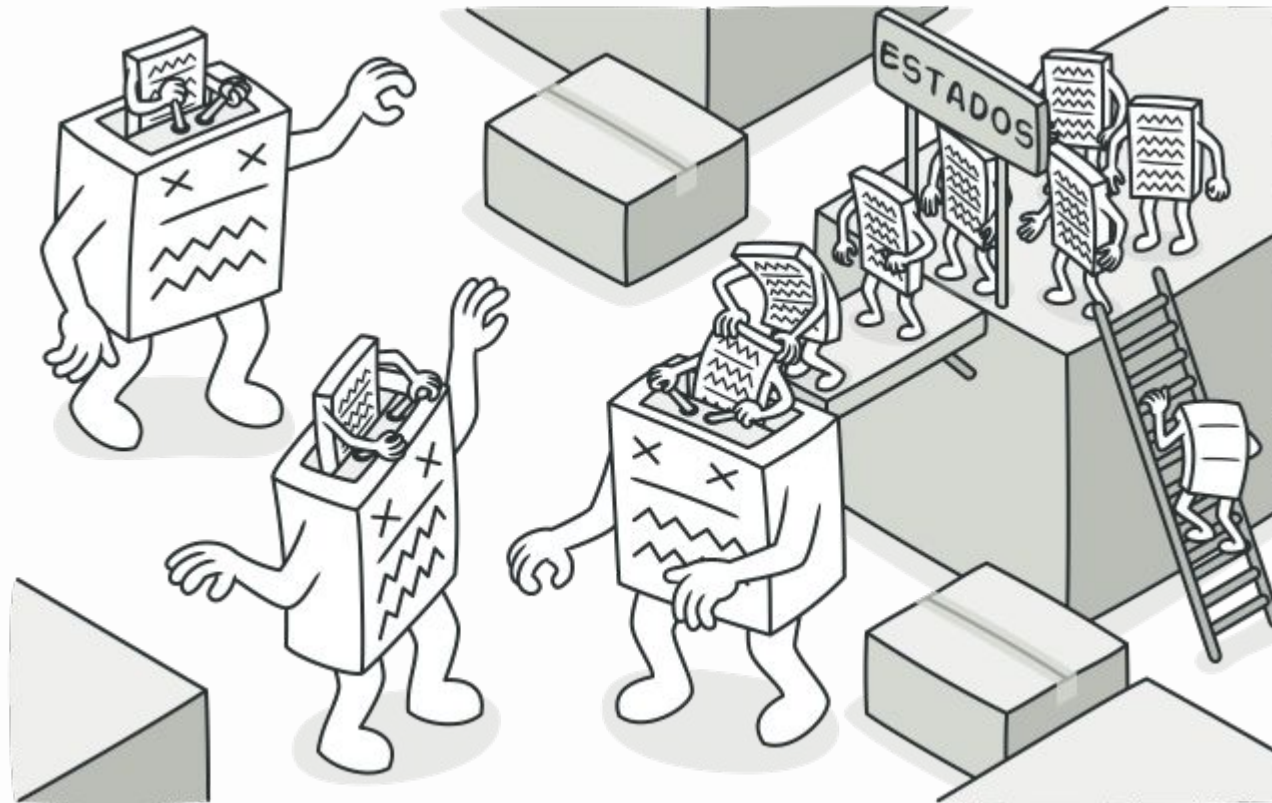
Iterator es un patrón de diseño de comportamiento que te permite recorrer elementos de una colección sin exponer su representación subyacente (lista, pila, árbol, etc.).





# Patrón State

State es un patrón de diseño de comportamiento que permite a un objeto alterar su comportamiento cuando su estado interno cambia. Parece como si el objeto cambiará su clase.



# ¿QUÉ SON LOS PATRONES DE DISEÑO?

Son técnicas para resolver problemas recurrentes de diseño de software.



## COMPORTAMENTALES

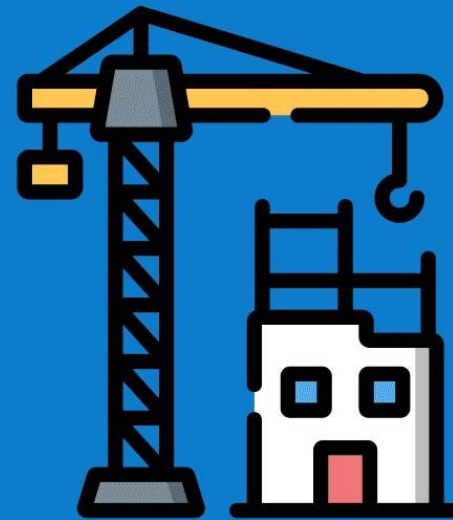
Establece soluciones relacionados con el comportamiento de la aplicación respecto a la interacción entre objetos y clases.



Ej. Observer, State.

## CREACIONALES

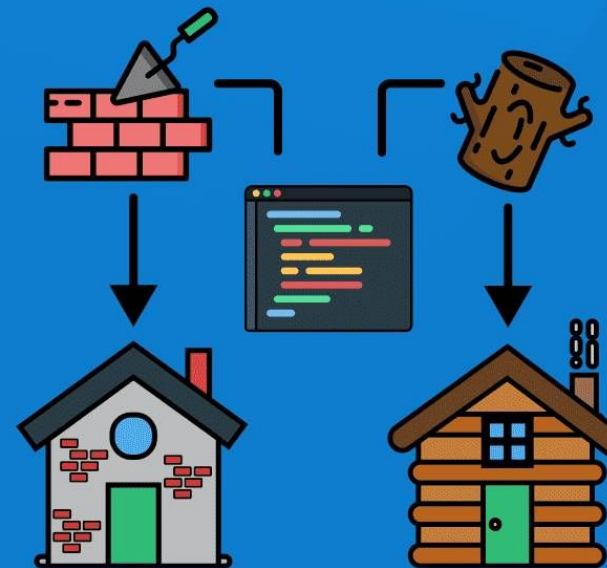
Soluciona la creación de objetos, hace un sistema independiente de cómo sus objetos son creados.



Ej. Singleton, Factory.

## ESTRUCTURALES

Describe cómo los objetos se componen para formar estructuras complejas.



Ej. Adapter, Decorator.

Si tienes un problema de diseño es muy probable que ya exista un patrón que lo solucione, no reinventes la rueda.



 Alejandro Rodriguez  
Backend Developer en EDteam

Domina los patrones de diseño de software en:  
 [ed.team/programacion](https://ed.team/programacion)

