

Design Patterns: Prototype

DIEGO ARAYA UREÑA

Patrones de Diseño

DEFINICIÓN

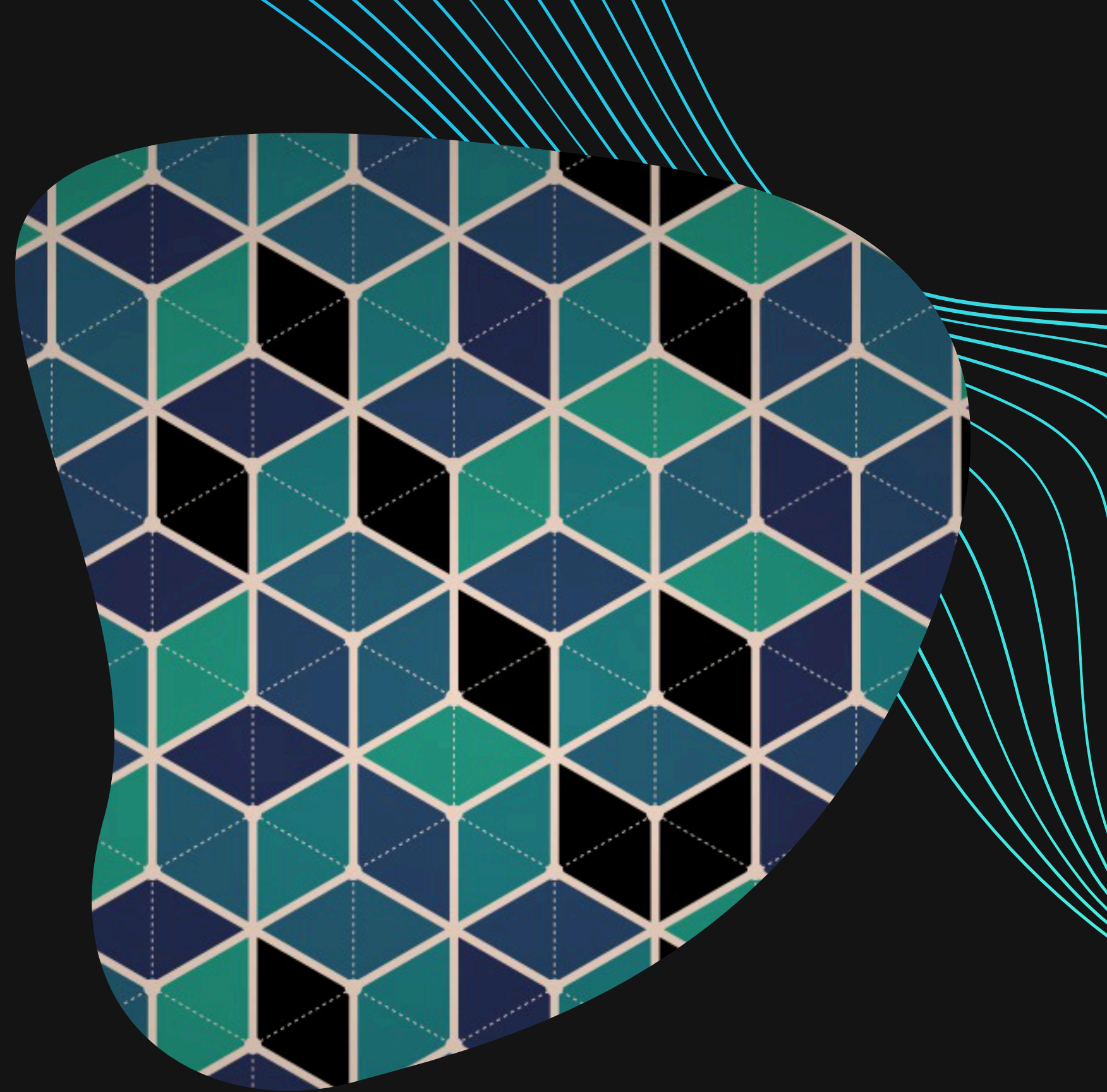
- Soluciones reutilizables a problemas comunes en diseño de software
- No son código, sino modelos conceptuales

CHRISTOPHER ALEXANDER:

- "Cada patrón describe un problema recurrente y el núcleo de su solución"

PROPÓSITO:

- Evitar reinventar soluciones
- Estandarizar vocabulario técnico





Elementos de un Patrón (GoF)

NOMBRE

Prototype



PROBLEMA

- Creación de objetos costosa o con estructura compleja
- Evitar jerarquías de clases rígidas

SOLUCIÓN

Clonar instancias existentes (prototipos)

CONSECUENCIAS

-  Flexibilidad en runtime
-  Complejidad en clonación profunda

¿Qué es el Patrón Prototype?

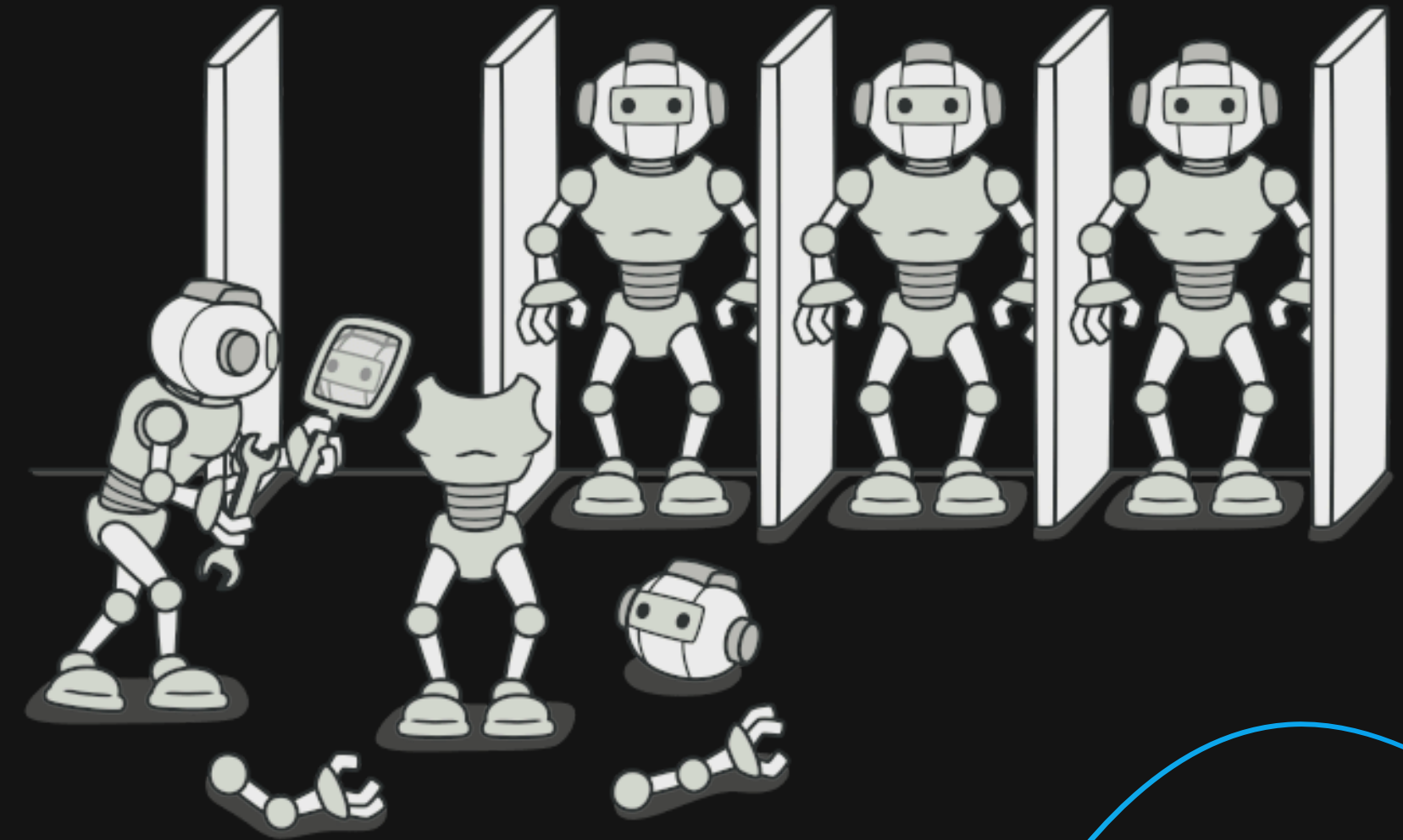
PROTOTYPE

DEFINICIÓN

- Mecanismo para crear objetos copiando una instancia existente (prototipo)
- (Como usar una plantilla de documento en lugar de empezar desde cero)

INTENCION

- "Especificar los tipos de objetos a crear usando una instancia prototípica, y crear nuevos objetos copiando este prototipo."



Clonación Simple (Shallow Copy)

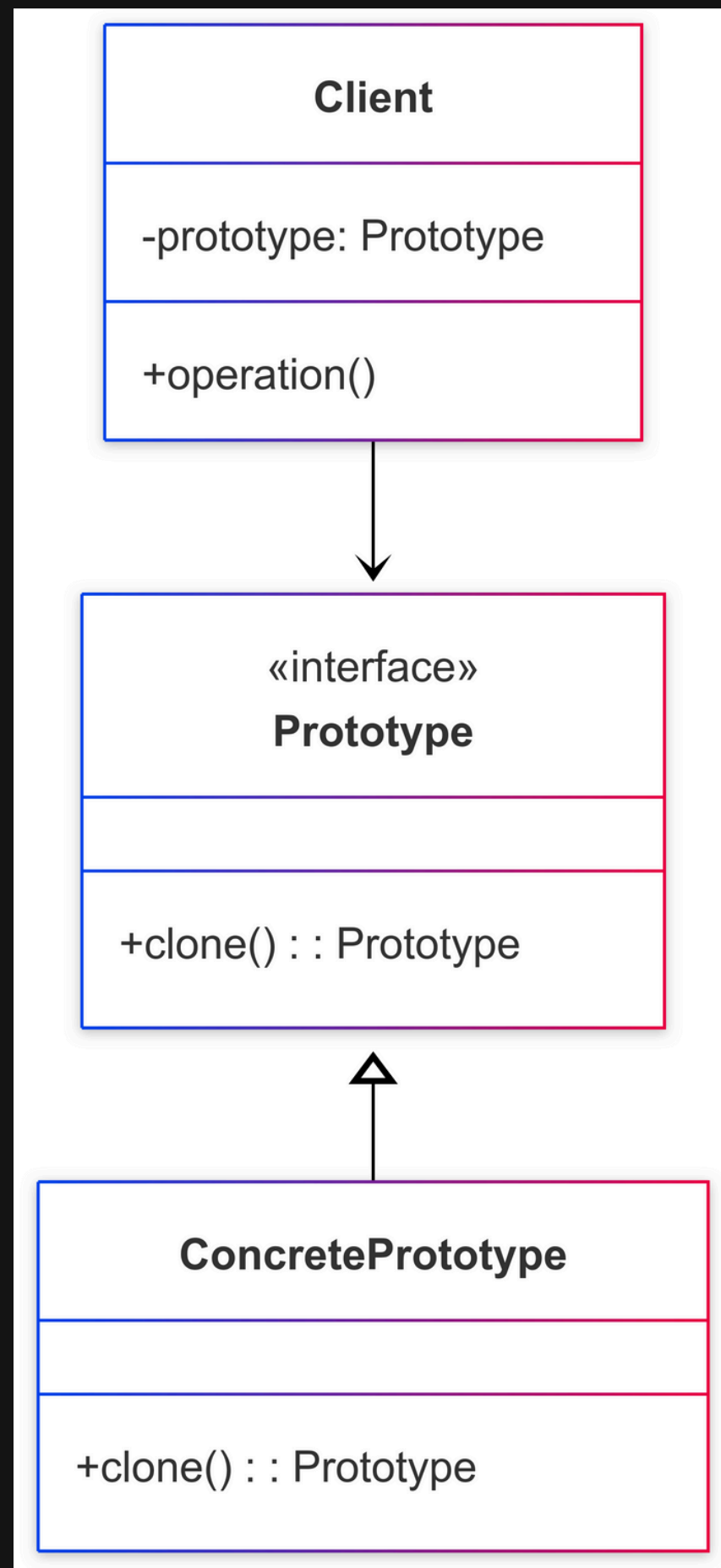
- Copia solo primitivos y referencias
- Rápida pero peligrosa con objetos mutables

Clonación Profunda (DeepCopy)

- Copia todos los objetos anidados recursivamente
- Costosa pero segura

Estructura de Prototype

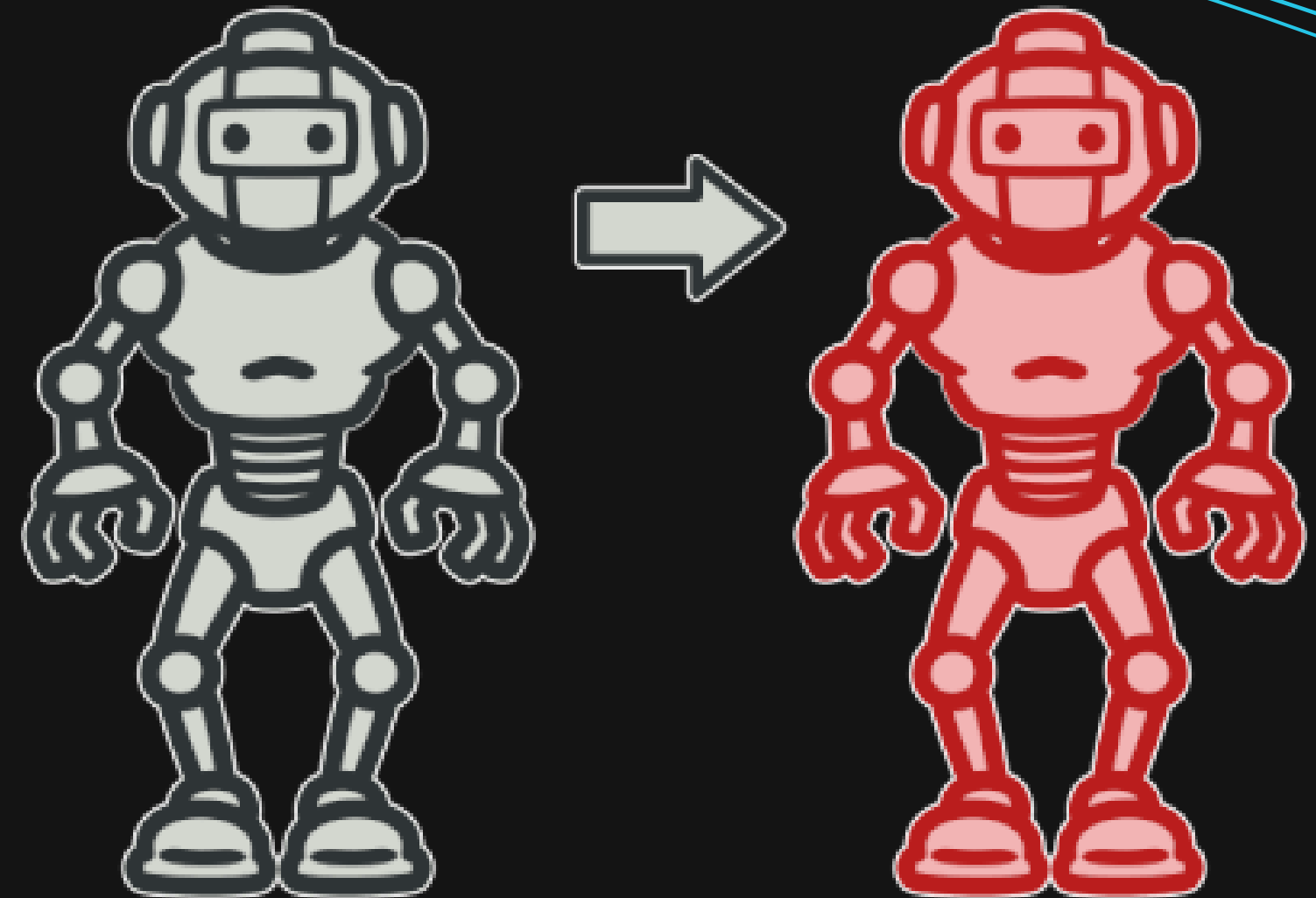
- Prototype: Interfaz con método clone()
- ConcretePrototype: Implementa la clonación
- Client: Usa el prototipo para crear nuevos objetos



Aplicabilidad

USE PROTOTYPE CUANDO:

- Un sistema debe ser independiente de cómo se crean sus objetos
- Las clases a instanciar se deciden en tiempo de ejecución
- Evitar jerarquías paralelas de factories
- Los objetos tienen pocas combinaciones de estado (mejor prototipos que subclasses)





Codigo de Ejemplo (Product List)