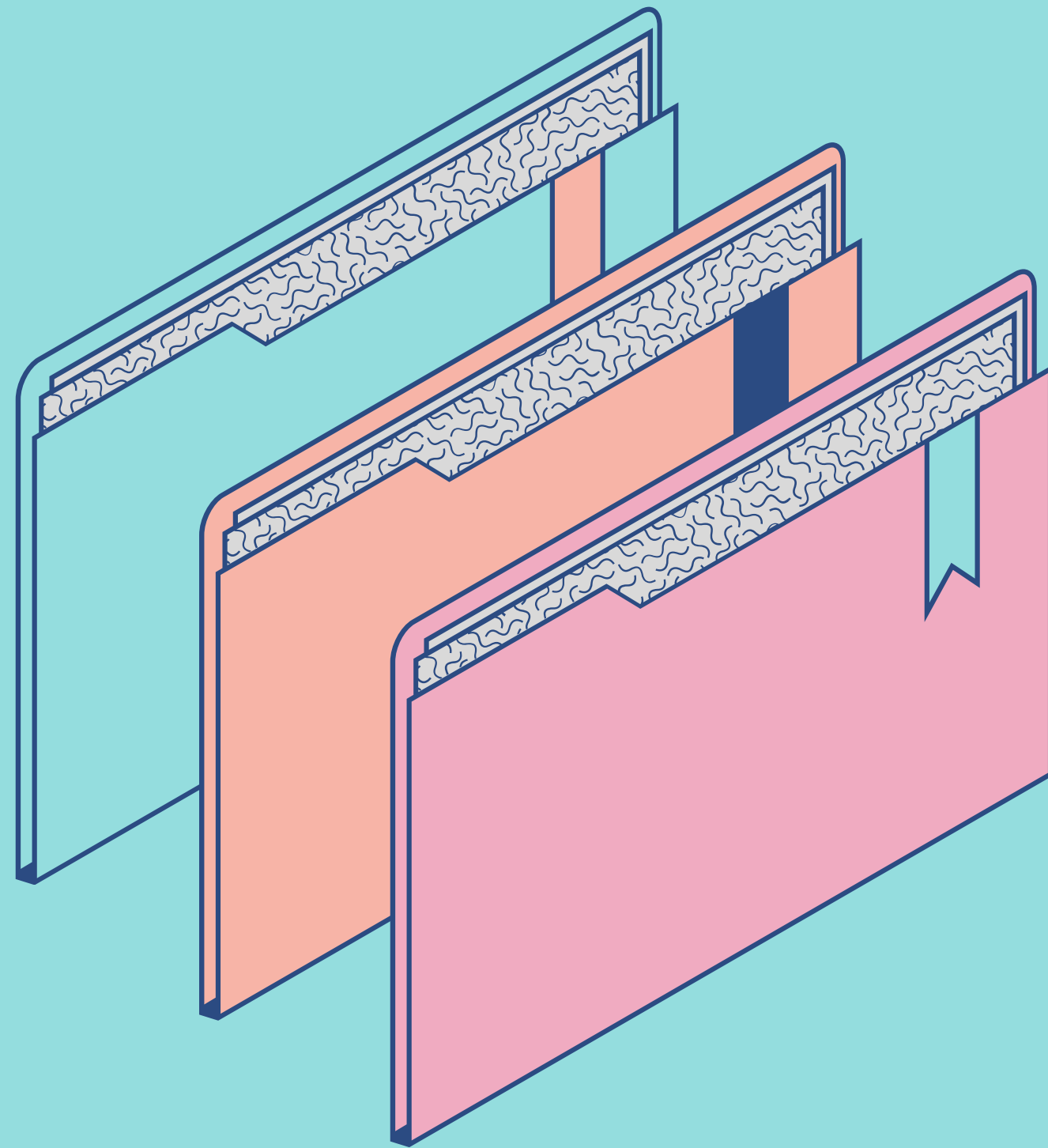




SOLUCIONES INTELIGENTES PARA EL  
DISEÑO DE SOFTWARE

# Patrones De Diseño (GOF)

Hecho por:  
Juan Felipe Garcia  
Juan David Lopez

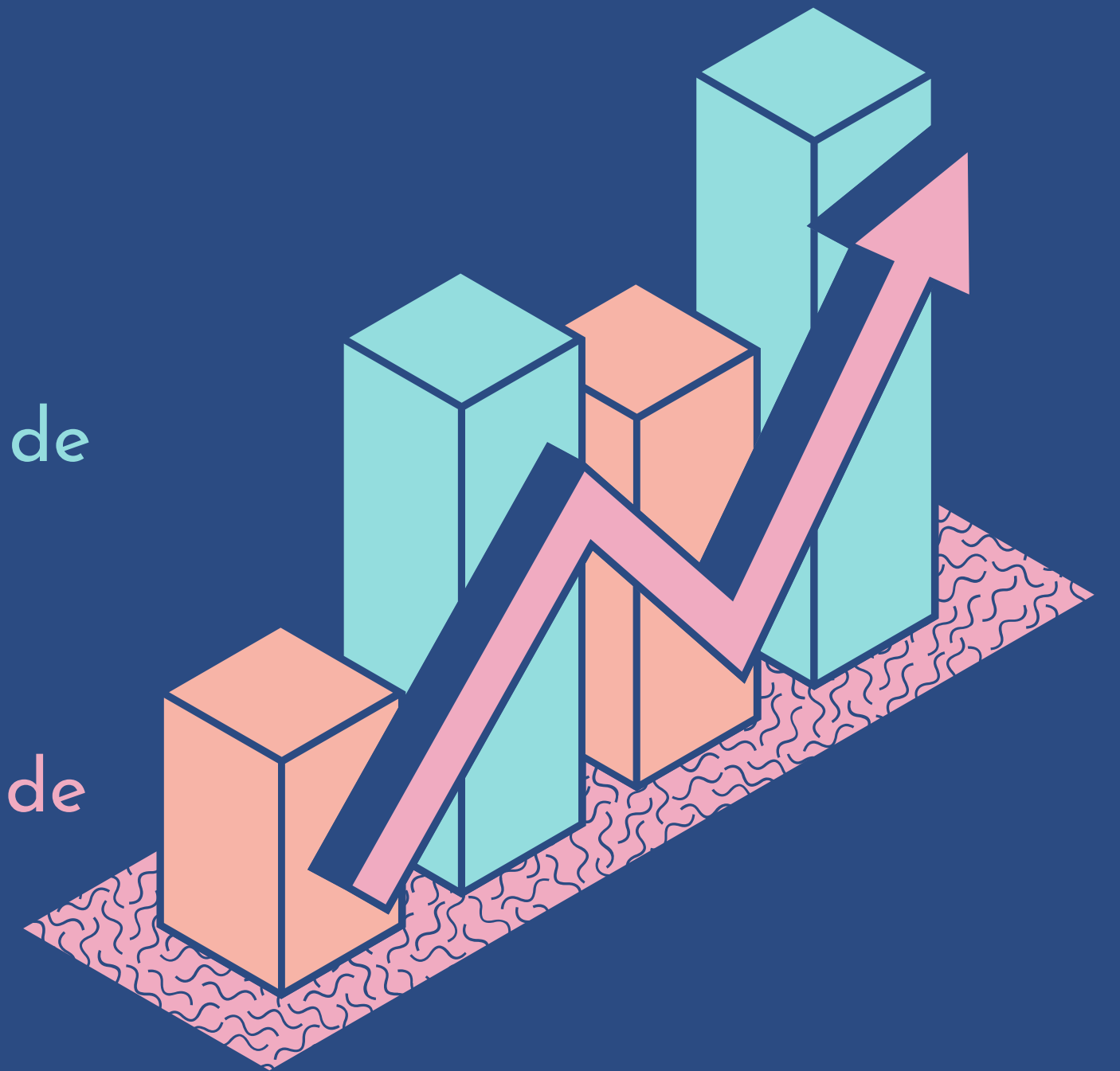


# Patrones De Diseño

SOLUCIONES INTELIGENTES PARA EL DISEÑO DE SOFTWARE

Son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en tu código. El patrón no es una porción específica de código, sino un concepto general para resolver un problema particular.

- El **propósito** del patrón explica brevemente el problema y la solución.
- La **motivación** explica en más detalle el problema y la solución que brinda el patrón.
- La **estructura** de las clases muestra cada una de las partes del patrón y el modo en que se relacionan.
- El **ejemplo de código** en uno de los lenguajes de programación populares facilita la asimilación de la idea que se esconde tras el patrón.



# Patrones De Diseño (GOF)

## Influencia

Ha trascendido el ámbito de la programación orientada a objetos, siendo aplicables a diversos lenguajes de programación y paradigmas de diseño. ofreciendo soluciones efectivas a problemas comunes de diseño y promoviendo la creación de software más flexible.

## Significado

GOF, siglas de Gang of Four (pandilla de cuatro), hace referencia a un grupo de cuatro autores de software, son reconocidos por su libro publicado en 1994 titulado Design Patterns: Elements of Reusable Object-Oriented Software

## Funcionalidad

- Escribir código más flexible y adaptable a cambios futuros.
- Mejorar la comunicación y comprensión entre los miembros del equipo de desarrollo.
- Crear software más fácil de mantener y extender.
- Reducir el tiempo de desarrollo al reutilizar soluciones ya probadas..

# Patrones De Diseño (GOF)

PATRONES DE DISEÑO GOF: SOLUCIONES  
INTELIGENTES PARA UN SOFTWARE ROBUSTO Y  
FLEXIBLE.

## Comportamentales

Se enfocan en la forma en que los objetos interactúan y se comunican entre sí. Su objetivo es mejorar la asignación de responsabilidades entre los objetos, facilitando la comunicación y coordinación de acciones.

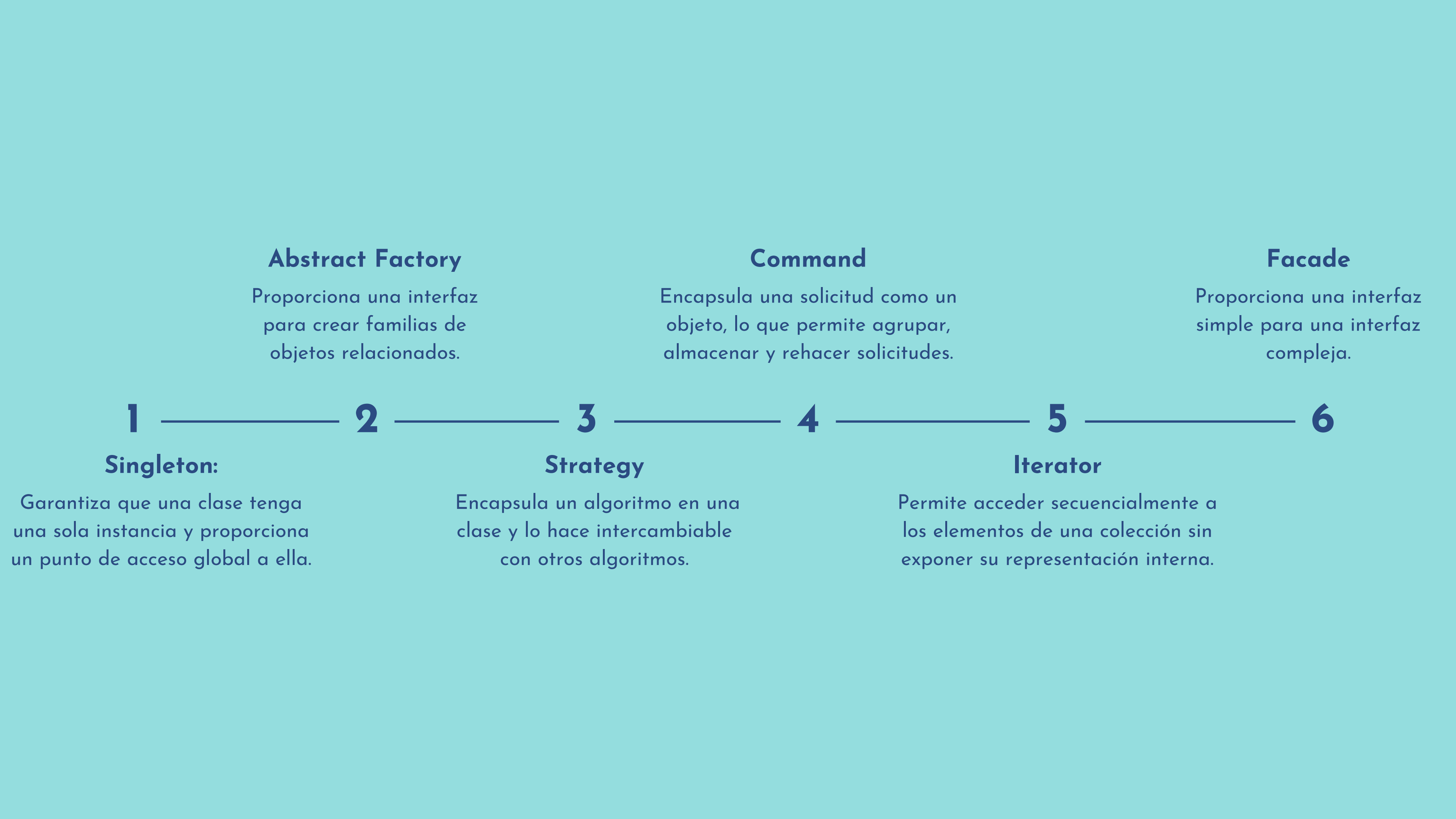
## Creacionales

Se centran en los mecanismos de creación de objetos, proporcionando maneras de crear objetos que aumenten la flexibilidad y reutilización del código.

## Estructurales

Se ocupan de cómo se componen las clases y los objetos para formar estructuras más grandes y complejas. Estos patrones ayudan a garantizar que si un componente se cambia, el sistema general no se vea afectado.





**¡GRACIAS POR SU  
ATENCIÓN!**

