

Patrones de Diseño en el Desarrollo de Software

Bienvenidos a esta presentación sobre los patrones de diseño, una herramienta esencial para cualquier desarrollador que busca crear software robusto, flexible y fácil de mantener. Acompáñenos en un recorrido por el mundo de los patrones de diseño.

 **by CYBER SECURITYTY**

¿Qué son los patrones de diseño?

Solución Recurrente

Son soluciones probadas y documentadas para problemas de diseño comunes que surgen durante el desarrollo de software.

Modelo de Diseño

Ofrecen un modelo reutilizable para resolver un problema específico, describiendo la interacción entre objetos y componentes.

Beneficios de utilizar patrones de diseño

■ Código Reutilizable

Promueven la reutilización del código, reduciendo la complejidad y el tiempo de desarrollo.

■ Mantenimiento Más Fácil

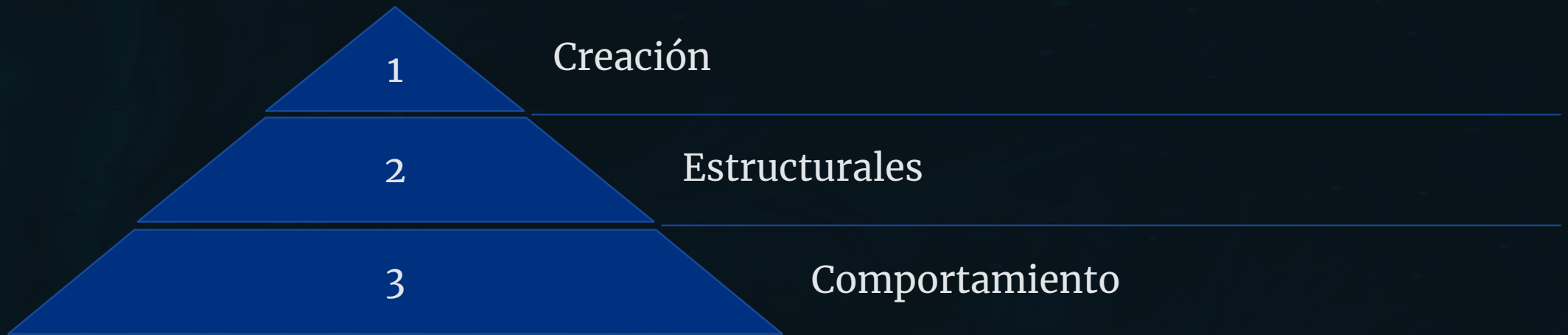
Facilitan el mantenimiento y la comprensión del código, al proporcionar una estructura clara y organizada.

■ Diseño Flexible

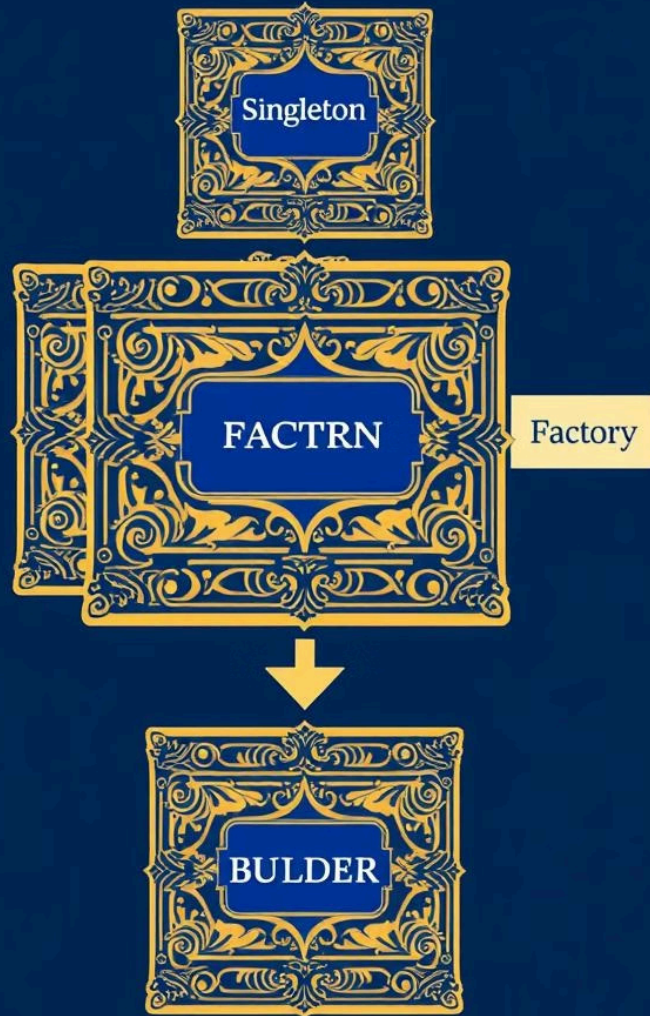
Permiten crear sistemas flexibles y adaptables a cambios futuros sin afectar el funcionamiento principal.



Clasificación de los patrones de diseño



CREATION PATTERNS



Patrones de Creación

Singleton

Garantiza que una clase tenga solo una instancia y proporciona un punto de acceso global a ella.

Factory

Define una interfaz para crear objetos, pero deja que las subclases decidan qué clase instanciar.

Builder

Separa la construcción de un objeto complejo de su representación, permitiendo construir diferentes variaciones.



Patrones Estructurales



Adaptador

Convierte la interfaz de una clase en otra que el cliente espera.



Decorador

Agrega responsabilidades adicionales a un objeto dinámicamente.



Puente

Desacopla una abstracción de su implementación para que puedan variar independientemente.

Patrones de Comportamiento

1

Observador

Define una dependencia de uno a muchos entre objetos, donde un cambio en un objeto notifica a todos sus dependientes.

2

Estrategia

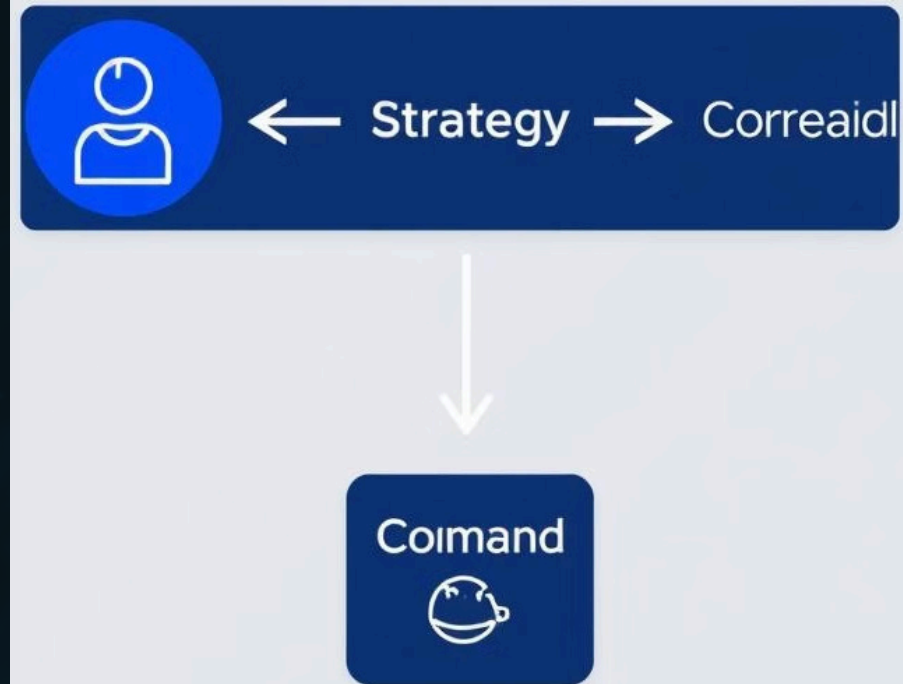
Define una familia de algoritmos, encapsula cada uno y los hace intercambiables.

3

Comando

Encapsula una solicitud como un objeto, permitiendo parametrizar clientes con diferentes solicitudes, colas o registros.

BEHAVIORAL PATTERNS



Cómo aplicar patrones de diseño en proyectos de software

1

Identificar el Problema

Comienza por determinar el problema de diseño específico que necesitas resolver.

2

Elegir el Patrón

Selecciona el patrón de diseño más adecuado para el problema identificado.

3

Implementar el Patrón

Implementa el patrón elegido en tu código, asegurándote de adaptarlo a tus necesidades.

