PATRONES ESTRUCTURALES

1. ADAPTER (ADAPTADOR)

- **PROPÓSITO**
  - PERMITE LA COLABORACIÓN ENTRE OBJETOS CON INTERFACES INCOMPATIBLES
- **APLICACIÓN**
  - ADAPTADOR DE OBJETOS
    - UTILIZA COMPOSICIÓN DE OBJETOS PARA CONVERTIR LA INTERFAZ DE UN OBJETO PARA QUE OTRO OBJETO PUEDA COMPRENDERLA
  - ADAPTADOR DE CLASE
    - UTILIZA HERENCIA MÚLTIPLE PARA CONVERTIR LA INTERFAZ DE UN OBJETO PARA QUE OTRO OBJETO PUEDA COMPRENDERLA
- **PRINCIPIOS**
  - RESPONSABILIDAD ÚNICA
    - SEPARA LA INTERFAZ O EL CÓDIGO DE CONVERSIÓN DE DATOS DE LA LÓGICA DE NEGOCIO
  - ABIERTO/CERRADO
    - INTRODUCE NUEVOS TIPOS DE ADAPTADORES SIN CAMBIAR EL CÓDIGO CLIENTE EXISTENTE
- **PROS Y CONTRAS**
  - PROS
    - FACILITA LA COLABORACIÓN ENTRE CLASES CON INTERFACES INCOMPATIBLES
  - CONTRAS
    - AUMENTA LA COMPLEJIDAD DEL CÓDIGO
- **COMPARACIONES**
  - ADAPTER VS DECORATOR
    - ADAPTER CAMBIA LA INTERFAZ, DECORATOR MEJORA UN OBJETO SIN CAMBIAR SU INTERFAZ
  - ADAPTER VS FACADE
    - ADAPTER TRABAJA CON UN SOLO OBJETO, FACADE TRABAJA CON UN SUBSISTEMA DE OBJETOS

PATRONES ESTRUCTURALES

2. BRIDGE (PUENTE)

- **PROPÓSITO**
  - DIVIDE UNA CLASE GRANDE O UN GRUPO DE CLASES ESTRECHAMENTE RELACIONADAS EN DOS JERARQUÍAS SEPARADAS (ABSTRACCIÓN E IMPLEMENTACIÓN)
- **COMPONENTES**
  - ABSTRACCIÓN
    - OFRECE LÓGICA DE CONTROL DE ALTO NIVEL, DEPENDE DE LA IMPLEMENTACIÓN PARA EL TRABAJO DE BAJO NIVEL
  - IMPLEMENTACIÓN
    - DECLARA LA INTERFAZ COMÚN PARA TODAS LAS IMPLEMENTACIONES CONCRETAS
- **APLICACIÓN**
  - SE UTILIZA CUANDO SE NECESITA EXTENDER UNA CLASE EN VARIAS DIMENSIONES ORTOGONALES O CAMBIAR IMPLEMENTACIONES DURANTE EL TIEMPO DE EJECUCIÓN
- **PROS Y CONTRAS**
  - PROS
    - FACILITA LA CREACIÓN DE CLASES INDEPENDIENTES DE PLATAFORMA
  - CONTRAS
    - PUEDE COMPLICAR EL CÓDIGO SI SE APLICA A UNA CLASE MUY COHESIONADA
- **COMPARACIONES**
  - BRIDGE VS ADAPTER
    - BRIDGE SE DISEÑA POR ANTICIPADO, ADAPTER SE UTILIZA CON APLICACIONES EXISTENTES

PATRONES ESTRUCTURALES

3. COMPOSITE (COMPUESTO)

- **PROPÓSITO**
  - COMPONER OBJETOS EN ESTRUCTURAS DE ÁRBOL Y TRABAJAR CON ESAS ESTRUCTURAS COMO SI FUERAN OBJETOS INDIVIDUALES
- **COMPONENTES**
  - **COMPONENTE**
    - DESCRIBE OPERACIONES COMUNES A ELEMENTOS SIMPLES Y COMPLEJOS
  - **HOJA**
    - ELEMENTO BÁSICO SIN SUBELEMENTOS
  - **CONTENEDOR**
    - TIENE SUBELEMENTOS Y TRABAJA CON ELLOS A TRAVÉS DE LA INTERFAZ COMPONENTE
- **APLICACIÓN**
  - SE UTILIZA CUANDO SE NECESITA IMPLEMENTAR UNA ESTRUCTURA DE OBJETOS CON FORMA DE ÁRBOL O CUANDO SE QUIERE QUE EL CÓDIGO CLIENTE TRATE ELEMENTOS SIMPLES Y COMPLEJOS DE LA MISMA FORMA
- **PROS Y CONTRAS**
  - **PROS**
    - FACILITA EL TRABAJO CON ESTRUCTURAS DE ÁRBOL COMPLEJAS
  - **CONTRAS**
    - PUEDE SER DIFÍCIL PROPORCIONAR UNA INTERFAZ COMÚN PARA CLASES CON FUNCIONALIDADES MUY DIFERENTES
- **COMPARACIONES**
  - COMPOSITE VS DECORATOR
    - AMBOS USAN COMPOSICIÓN RECURSIVA, PERO CON DIFERENTES PROPÓSITOS

PATRONES ESTRUCTURALES

4. DECORATOR (DECORADOR)

- **PROPÓSITO**
  - AÑADIR FUNCIONALIDADES A OBJETOS COLOCÁNDOLOS DENTRO DE OBJETOS ENCAPSULADORES ESPECIALES
- **COMPONENTES**
  - **COMPONENTE**
    - DECLARA LA INTERFAZ COMÚN PARA WRAPPERS Y OBJETOS ENVUELTOS
  - **COMPONENTE CONCRETO**
    - DEFINE EL COMPORTAMIENTO BÁSICO
  - **DECORADORA BASE**
    - REFERENCIA A UN OBJETO ENVUELTO
  - **DECORADORES CONCRETOS**
    - AÑADEN FUNCIONALIDADES ADICIONALES
- **APLICACIÓN**
  - SE UTILIZA CUANDO SE NECESITA ASIGNAR FUNCIONALIDADES ADICIONALES A OBJETOS DURANTE EL TIEMPO DE EJECUCIÓN O CUANDO EXTENDER EL COMPORTAMIENTO DE UN OBJETO MEDIANTE HERENCIA NO ES POSIBLE
- **PROS Y CONTRAS**
  - **PROS**
    - PERMITE EXTENDER EL COMPORTAMIENTO DE UN OBJETO SIN CREAR UNA NUEVA SUBCLASE
  - **CONTRAS**
    - PUEDE COMPLICARSE ELIMINAR UN WRAPPER ESPECÍFICO DE LA PILA DE WRAPPERS
- **COMPARACIONES**
  - DECORATOR VS ADAPTER
    - ADAPTER CAMBIA LA INTERFAZ, DECORATOR MEJORA EL OBJETO SIN CAMBIAR SU INTERFAZ

PATRONES ESTRUCTURALES

5. FACADE (FACHADA)

- **PROPÓSITO**
  - PROPORCIONAR UNA INTERFAZ SIMPLIFICADA A UN GRUPO COMPLEJO DE CLASES
- **COMPONENTES**
  - FACHADA
    - PROPORCIONA ACCESO A UNA PARTE ESPECÍFICA DE LA FUNCIONALIDAD DEL SUBSISTEMA
  - SUBSISTEMA COMPLEJO
    - CONJUNTO DE OBJETOS DIVERSOS
- **APLICACIÓN**
  - SE UTILIZA CUANDO SE NECESITA UNA INTERFAZ LIMITADA PERO DIRECTA A UN SUBSISTEMA COMPLEJO O CUANDO SE QUIERE ESTRUCTURAR UN SUBSISTEMA EN CAPAS
- **PROS Y CONTRAS**
  - PROS
    - AÍSLA EL CÓDIGO DE LA COMPLEJIDAD DE UN SUBSISTEMA
  - CONTRAS
    - PUEDE CONVERTIRSE EN UN OBJETO TODOPODEROSO ACOPLADO A TODAS LAS CLASES DE UNA APLICACIÓN
- **COMPARACIONES**
  - FACADE VS ADAPTER
    - FACADE TRABAJA CON UN SUBSISTEMA, ADAPTER CON UN SOLO OBJETO

PATRONES ESTRUCTURALES

6. FLYWEIGHT (PESO LIGERO)

- PROPÓSITO
  - MANTENER MÁS OBJETOS DENTRO DE LA CANTIDAD DISPONIBLE DE MEMORIA RAM COMPARTIENDO LAS PARTES COMUNES DEL ESTADO ENTRE VARIOS OBJETOS
- APLICACIÓN
  - SE UTILIZA CUANDO SE NECESITA GENERAR UNA CANTIDAD ENORME DE OBJETOS SIMILARES O CUANDO LOS OBJETOS CONTIENEN ESTADOS DUPLICADOS QUE SE PUEDEN COMPARTIR
- PROS Y CONTRAS
  - PROS
    - AHORRA MUCHA RAM
  - CONTRAS
    - COMPLICA EL CÓDIGO Y PUEDE CAMBIAR RAM POR CICLOS DE CPU

7. PROXY (APODERADO)

- PROPÓSITO
  - PROPORCIONAR UN SUSTITUTO O MARCADOR DE POSICIÓN PARA OTRO OBJETO Y CONTROLAR EL ACCESO AL OBJETO ORIGINAL
- TIPOS
  - PROXY VIRTUAL
    - INICIALIZACIÓN DIFERIDA
  - PROXY DE PROTECCIÓN
    - CONTROL DE ACCESO