

# BRIDGE

## PATRÓN DE DISEÑO

# DESCRIPCIÓN

- NOMBRE Y CLASIFICACIÓN:

BRIDGE(PUENTE), PATRONES DE DISEÑO ESTRUCTURALES.

- INTENCIÓN:

\_DESACOPLAR UNA ABSTRACCIÓN DE SU IMPLEMENTACIÓN DE MODO QUE LOS DOS PUEDAN SER MODIFICADOS DE FORMA INDEPENDIENTE.

- OTROS NOMBRES (A.K.A.):

TAMBIÉN ES CONOCIDO COMO HANDLE/BODY.

- MOTIVACIÓN:

DADO QUE CUALQUIER CAMBIO QUE SE REALICE SOBRE UNA ABSTRACCIÓN AFECTARÁ A TODAS LAS CLASES QUE LA IMPLEMENTAN, *BRIDGE* PROPONE AÑADIR UN NUEVO NIVEL DE ABSTRACCIÓN ENTRE AMBOS ELEMENTOS QUE PERMITAN QUE PUEDAN DESARROLLARSE CADA UNO POR SU LADO.



# DESCRIPCIÓN

- APLICACIÓN:

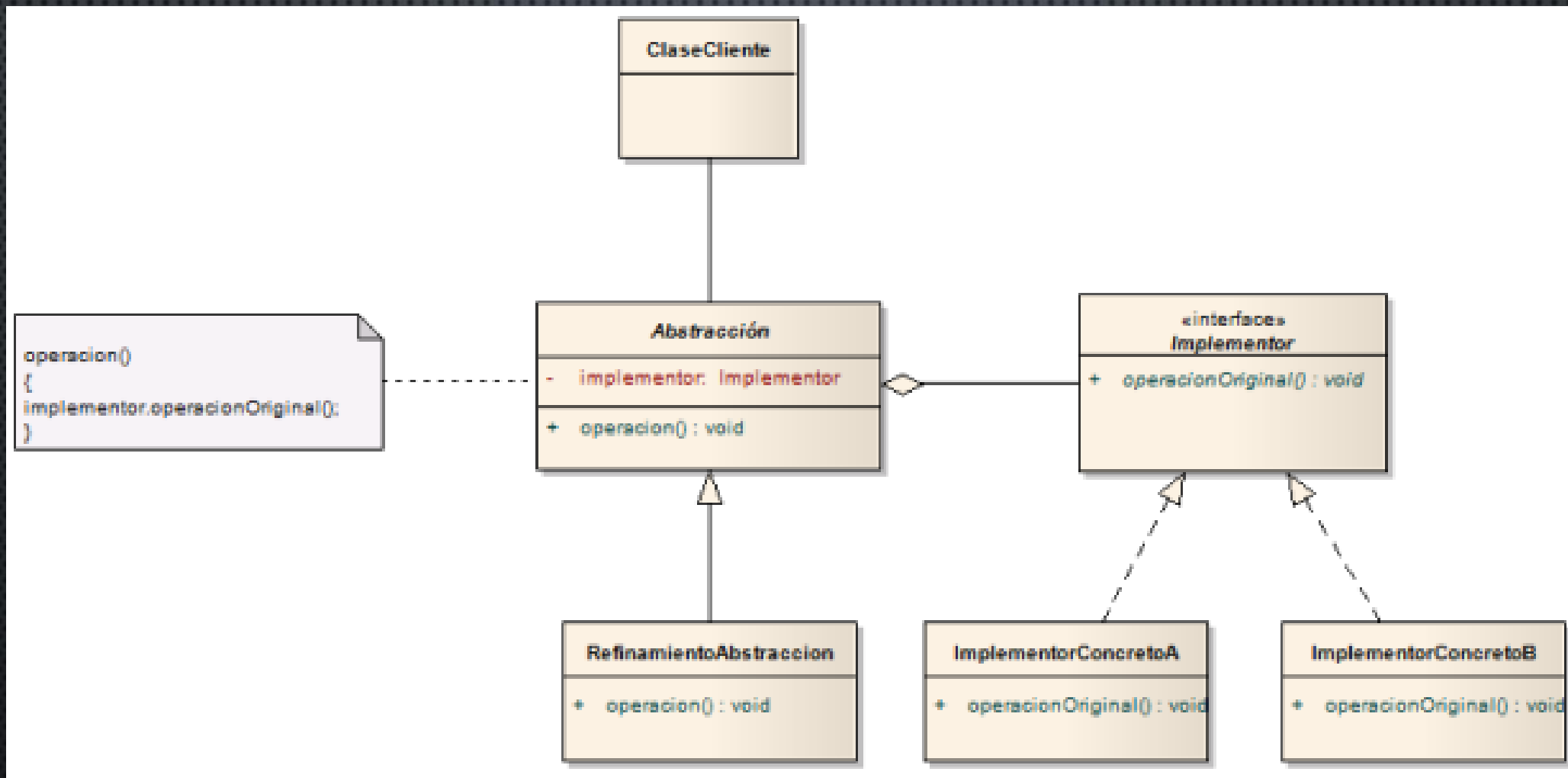
LAS SITUACIONES ÓPTIMAS EN LOS QUE SE DEBE UTILIZAR ESTE PATRÓN SERÁN, POR TANTO:

- CUANDO SE DESEA EVITAR UN ENLACE PERMANENTE ENTRE LA ABSTRACCIÓN Y (TODA O PARTE DE) SU IMPLEMENTACIÓN.
- CUANDO LOS CAMBIOS EN LA IMPLEMENTACIÓN DE UNA ABSTRACCIÓN NO DEBE AFECTAR A LAS CLASES QUE HACE USO DE ELLA.
- CUANDO SE DESEA COMPARTIR UNA IMPLEMENTACIÓN ENTRE MÚLTIPLES OBJETOS.

- PATRONES RELACIONADOS:

ADAPTER: TIENE UNA ESTRUCTURA SIMILAR AUNQUE UN FIN DIFERENTE. MIENTRAS BRIDGE BUSCA SEPARAR LA INTERFACE DE SU IMPLEMENTACIÓN ADAPTER ESTA DESTINADO A CAMBIAR LA INTERFACE.

# ESTRUCTURA





# DESCRIPCIÓN

- PARTICIPANTES:
  - **ABSTRACCIÓN** DEFINE UNA INTERFAZ ABSTRACTA. MANTIENE UNA REFERENCIA A UN OBJETO DE TIPO *IMPLEMENTOR*.
  - **REFINAMIENTOABSTRACCIÓN** EXTIENDE LA INTERFAZ DEFINIDA POR *ABSTRACCIÓN*.
  - **IMPLEMENTOR** DEFINE LA INTERFAZ PARA LA IMPLEMENTACIÓN DE CLASES. ESTA INTERFAZ NO SE TIENE QUE CORRESPONDER EXACTAMENTE CON LA INTERFAZ DE *ABSTRACCIÓN*; DE HECHO, LAS DOS INTERFACES PUEDEN SER BASTANTE DIFERENTE. TÍPICAMENTE LA INTERFAZ *IMPLEMENTOR* PROVEE SÓLO OPERACIONES PRIMITIVAS, Y *ABSTRACCIÓN* DEFINE OPERACIONES DE ALTO NIVEL BASADAS EN ESTAS PRIMITIVAS.
  - **IMPLEMENTORCONCRETO** IMPLEMENTA LA INTERFAZ DE *IMPLEMENTOR* Y DEFINE SU IMPLEMENTACIÓN CONCRETA.
- COLABORACIONES:

*ABSTRACCIÓN* REENVÍA LAS PETICIONES DE LOS CLIENTES A SU OBJETO *IMPLEMENTOR*.



# DESCRIPCIÓN

- CONSECUENCIAS:

1. *ESCONDE LOS DETALLES DE LA IMPLEMENTACIÓN A LOS CLIENTES.*
2. *MEJORA LA EXTENSIBILIDAD: SE PUEDE EXTENDER LAS JERARQUÍAS DE ABSTRACCIÓN E IMPLEMENTOR INDEPENDIENTEMENTE.*
3. *DESACOPLANDO LA INTERFAZ DE LA IMPLEMENTACIÓN: LA IMPLEMENTACIÓN NO ESTÁ LIMITADA PERMANENTEMENTE A UNA INTERFAZ, LA IMPLEMENTACIÓN DE UNA ABSTRACCIÓN SE PUEDE CONFIGURAR EN RUN-TIME. ADEMÁS LE ES POSIBLE A UN OBJETO CAMBIAR SU IMPLEMENTACIÓN EN TIEMPO DE EJECUCIÓN.*

- IMPLEMENTACIÓN:

*SOLAMENTE UN EJECUTOR. EN LAS SITUACIONES DONDE HAY SOLAMENTE UNA IMPLEMENTACIÓN, CREAR UNA CLASE ABSTRACTA IMPLEMENTOR NO ES NECESARIO.*

*CREANDO EL OBJETO IMPLEMENTOR ADECUADO: ¿CÓMO, CUÁNDO Y DÓNDE QUE CLASE IMPLEMENTOR INSTANCIAR CUANDO HAY MÁS DE UNA? SI ABSTRACCIÓN CONOCE TODAS LAS CLASES CONCRETE IMPLEMENTOR PUEDE INSTANCIAR UNA DE ELLAS EN SU CONSTRUCTOR; PUEDE DECIDIR CUÁL INSTANCIAR DEPENDIENDO DE LOS PARÁMETROS DEL CONSTRUCTOR.*