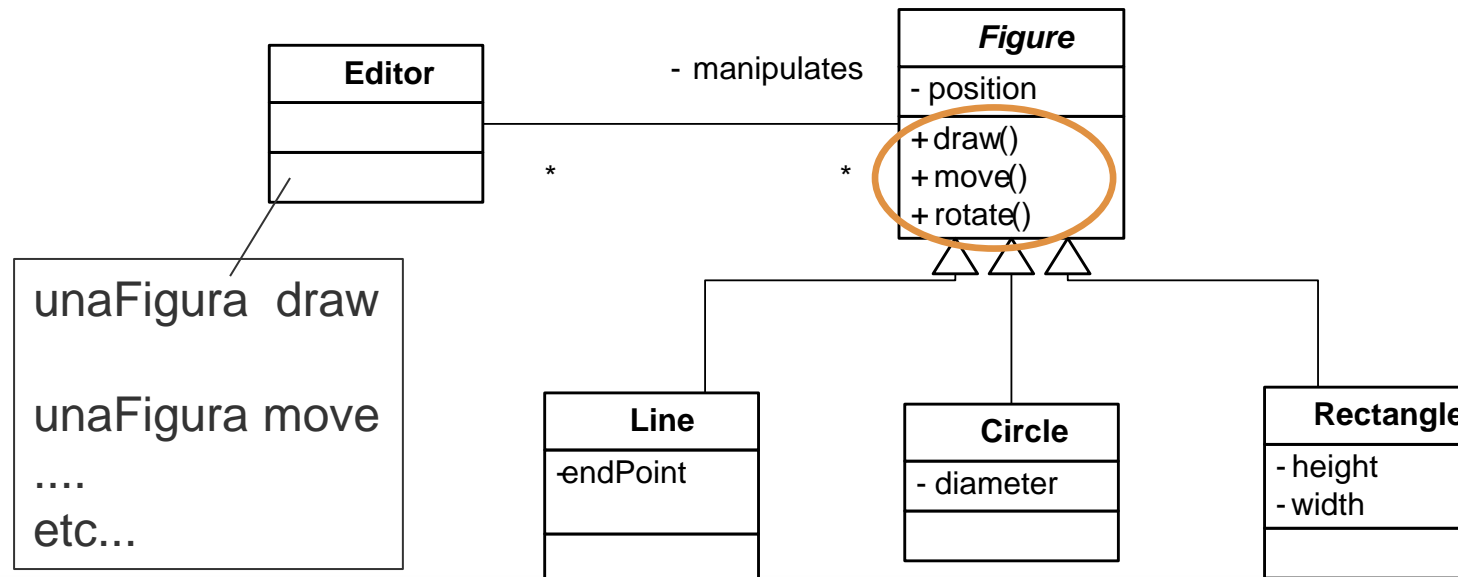
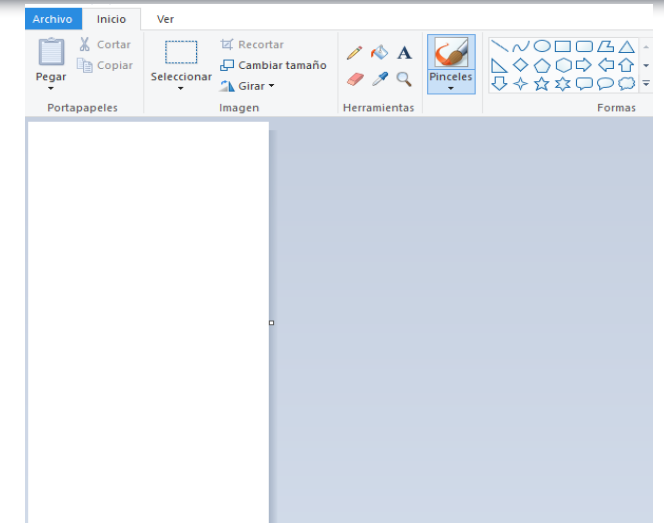


Como seguimos?

- ✓ Vamos a introducir varios Patrones
- ✓ Para cada uno de ellos, comenzamos con un ejemplo motivador y luego lo describimos con detalle

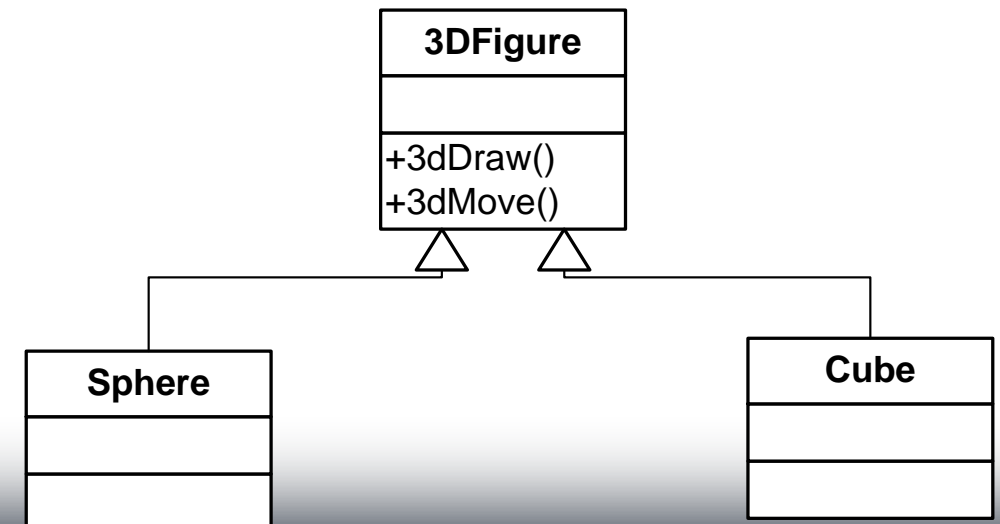
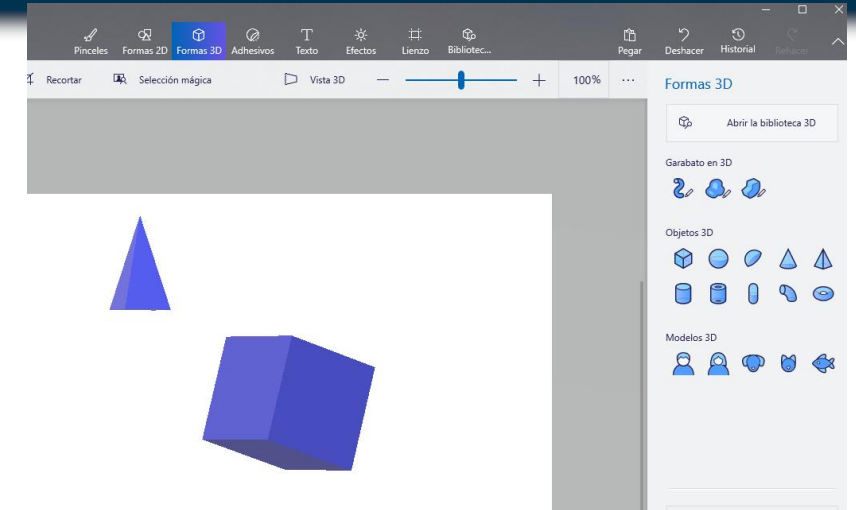


- ✓ Editor grafico para manejar figuras geometricas:
- ✓ Observen polimorfismo en la relacion Editor-Figure



Ejemplo..

- ✓ Queremos extender el editor a figuras 3D
- ✓ Programarlo nosotros llevaría mucho tiempo....
- ✓ Lo buscamos en Internet
- ✓ Encontramos un europeo....código abierto, el mismo lenguaje de programación
- ✓ Pero



Cual es el problema?

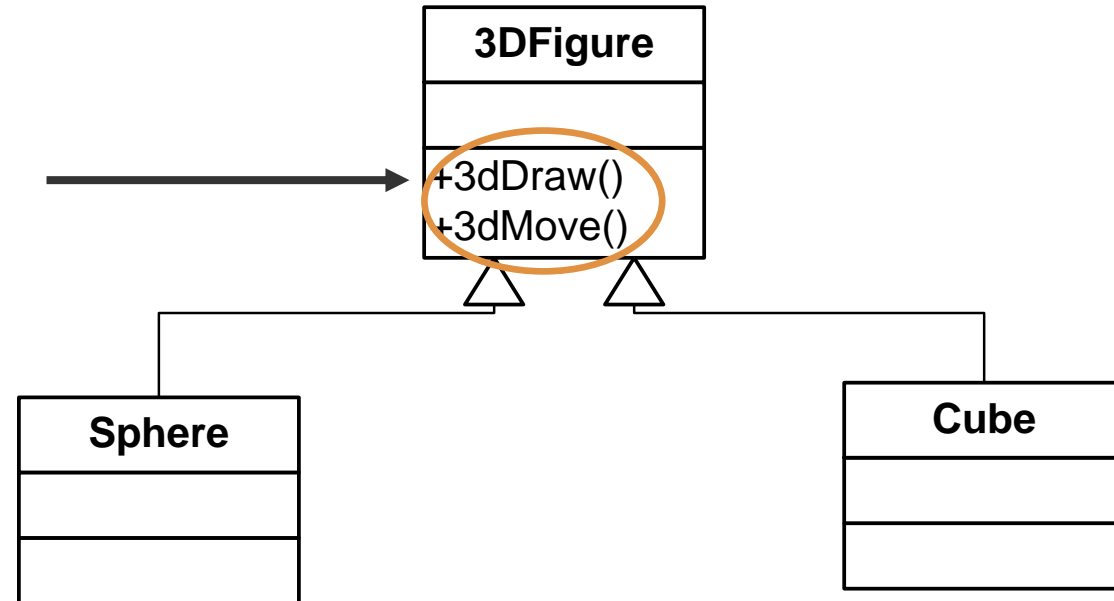
✓ Mientras el editor envía los mensajes:

unaFigura draw

unaFigura move

.....

✓ Esta jerarquía tiene un protocolo diferente

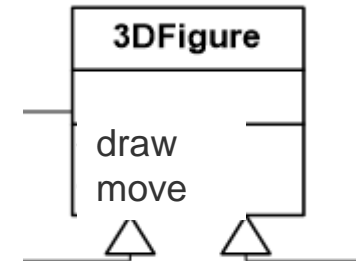


Problema

- ✓ Como integramos esta jerarquia?
- ✓ Que problemas tenemos?
- ✓ Sacrificamos polimorfismo? (con if en el editor)

```
If unaFigura es2D then: unaFigura draw  
else: unaFigura 3dDraw
```

- ✓ Editamos el codigo de la nueva jerarquia? (cambiando el nombre de los métodos)

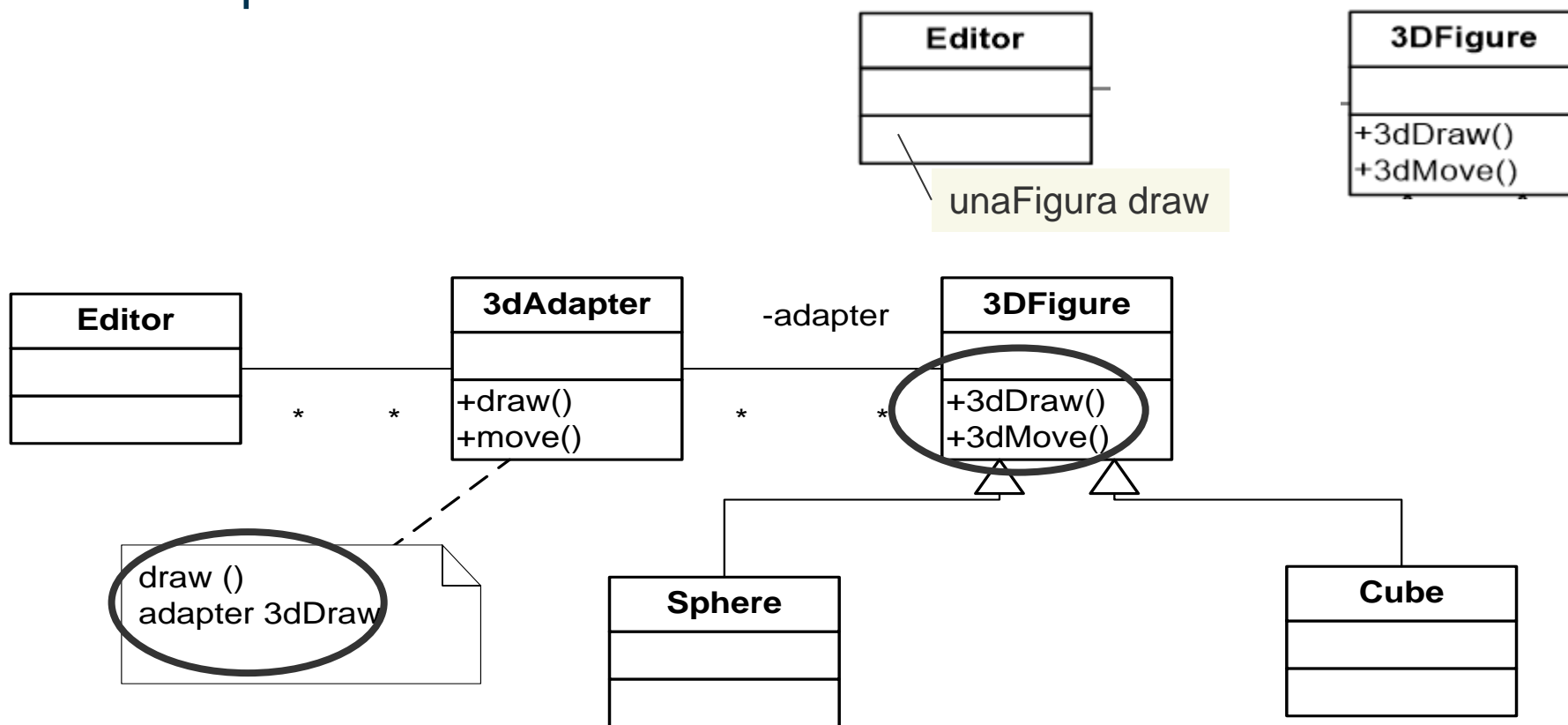


- ✓ Finalmente.....es el problema de la heladera, pero en software!!!
- ✓ Todo funciona correctamente, pero es “levemente” incompatible



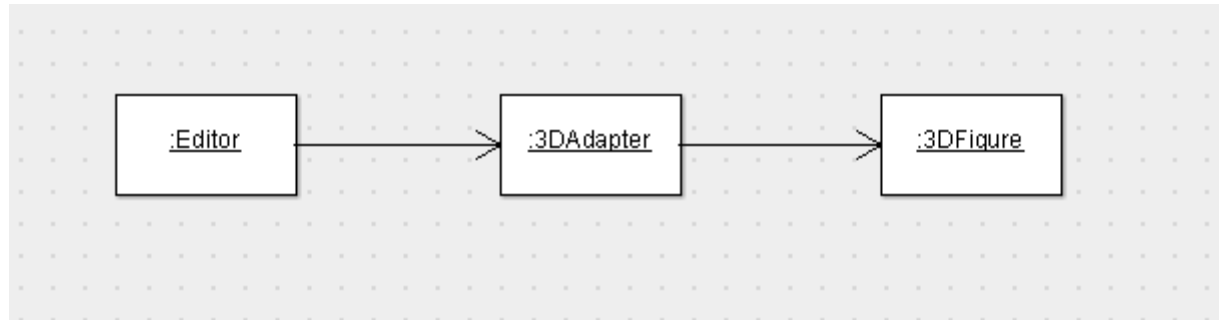
Solucion

- ✓ Cuando tratamos con interfaces incompatibles, intentar adaptarlas.



Que vemos a nivel de objetos?

- ✓ Cada vez que el editor crea una figura 3D, se crea su adaptador y funcionan “pegados”.



- ✓ Luego de la creación de ese objeto, para el editor, es exactamente lo mismo “hablar” con una figure o con una 3dFigure.
- ✓ Faltan detalles como: parámetros de los mensajes, creación de adapters para minimizar esfuerzo, cuantos adapters son necesarios...

✓ **Intencion:**

“Convertir” la interfaz de una clase en otra que el cliente espera. El Adapter permite que ciertas clases trabajen en conjunto cuando no podrian por tener interfaces incompatibles

✓ **Aplicabilidad:**

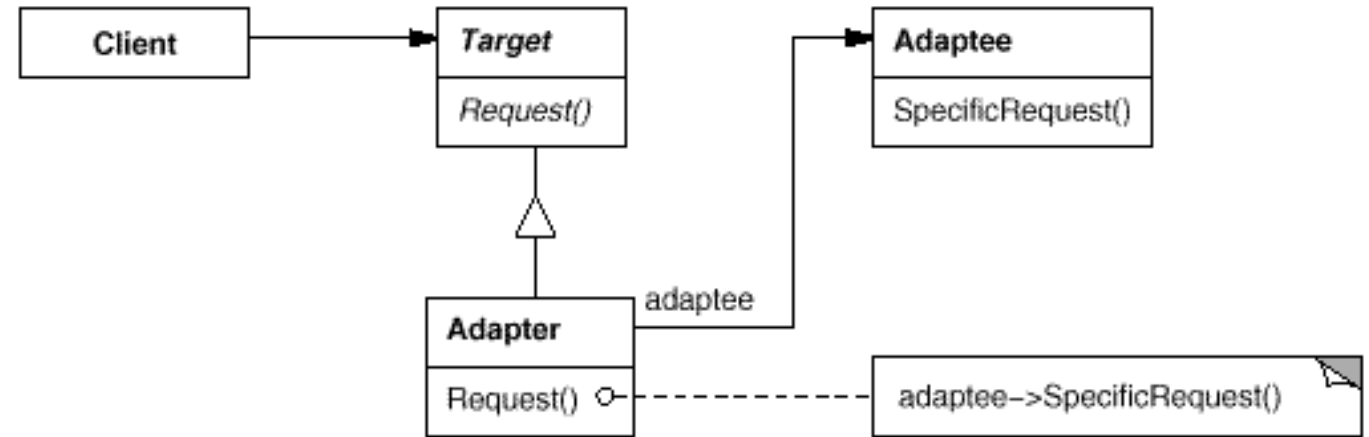
Use el adapter cuando:

Ud quiere usar una clase existente y su interfaz no es compatible con lo que precisa



Adapter

✓ Estructura



✓ Como se “lee” este diagrama?

✓ Que significa interpretar las clases como roles?

Adapter. Participants

✓ Participantes:

✓ Target (Figure)

- ✓ defines the domain-specific interface that Client uses.

✓ Client (Editor)

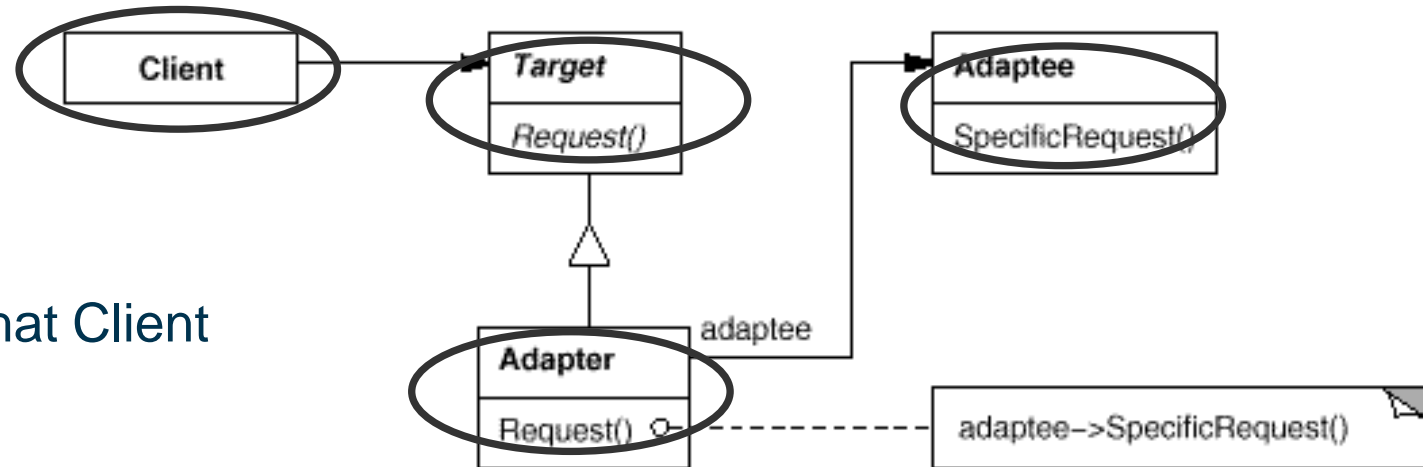
- ✓ collaborates with objects conforming to the Target interface.

✓ Adaptee (3DFigure)

- ✓ defines an existing interface that needs adapting.

✓ Adapter (3DAdapter)

- ✓ adapts the interface of Adaptee to the Target interface.



✓ *Collaborations*

El cliente colabora con objetos Adapter quienes a su vez lo hacen con instancias de Adaptee

✓ *Consequences*

✓ *Implementación*

✓ *Code*

✓ *Known Uses*