

Proxy

Nombre:

Proxy

Clasificación del patrón:

Estructural

Intención:

Suministra un sustituto o un delegado para controlar el acceso de un objeto.

Otros nombres:

Surrogate, sustituto.

Motivación:

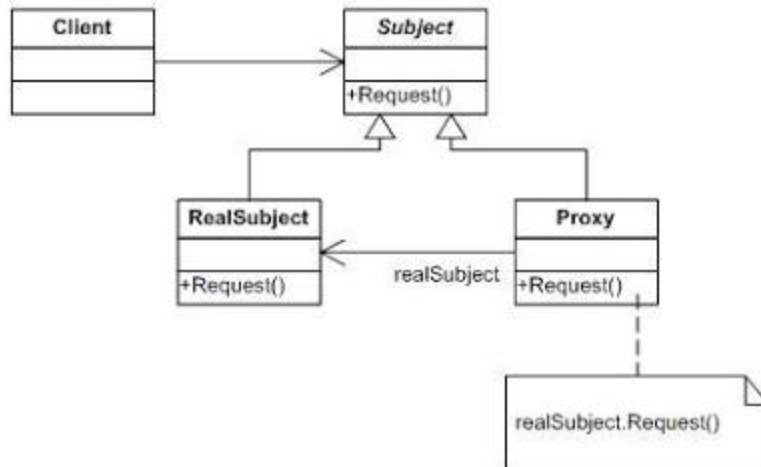
Entre las razones principales de no proporcionar el acceso directo a un objeto radica en reducir el costo de la creación y mantenimiento del mismo hasta que es realmente necesario. Un ejemplo sería algún software que implemente varias imágenes en diferentes instancias de la página, para que el sitio cargue de manera eficiente se puede restringir el acceso de las imágenes hasta que se muestren realmente.

Aplicabilidad:

Se debe usar cuando el acceso al objeto real puede no ser posible:

- Se debe utilizar un componente remoto.
- Un objeto costoso tiene que ser creado por demanda.
- El acceso al objeto original debe ser restringido.
- Solicitud frecuente de almacenamiento en caché para evitar llamadas de red.
- Gestión automática de recursos en el caso de C++.

Estructura:



Participantes:

- Sujeto: Define una interfaz común para el Proxy y el Sujeto real, de manera que se pueda usar Proxy donde espera un objeto real.
- SujetoReal: Define el objeto real representado.
- Proxy: Mantiene una referencia que le permite acceder al objeto real. Proporciona una interfaz idéntica al Sujeto de manera que se pueda aplicar la sustitución por el SujetoReal. Puede ser responsable de la creación y borrado de SujetoReal.

Colaboraciones:

Proxy hace las veces del SujetoReal cuando se considere apropiado.

Ventajas:

- Proxy introduce una indirección que puede ser útil.
- El proxy remoto oculta la ubicación del objeto real al cliente que lo está usando.
- Los proxies virtuales pueden crear objetos por demanda.
- La protección del proxy limita el acceso al objeto.
- Los puntos inteligentes ayudan a la gestión automática en C++.

Desventajas:

- Puede haber un acoplamiento “apretado” entre el proxy y el sujeto real.
 - En especial si el objeto inicia su creación por demanda.

Implementación:

- Proxy no siempre conoce el tipo de objeto de SujetoReal. En caso de que no se deben instanciar objetos del SujetoReal no ocurre nada, mientras que en otro caso sí.
- Sobrecarga de operadores de acceso.

Código de ejemplo:

- Interfaz del servidor de descargas:

```
1 public interface Server{
2
3     public void download(String url);
4
5 }
```

- Servidor real:

```
1 public class RealServer implements Server{
2
3
4
5     private int port;
6
7     private String host;
8
9
10
11     public RealServer(int port, String host){
12
13         this.port=port;
14
15         this.host=host;
16
17         System.out.println("Servidor iniciado...");
18     }
19
20
21
22
23     public void download(String url){
24
25         System.out.println("Descargando "+host+": "+port+"/"+url);
26     }
27
28
29 }
```

- Proxy:

```

1 public class ProxyServer implements Server{
2
3     RealServer srv;
4     private int port;
5     private String host;
6
7     public ProxyServer(int port, String host){
8         this.port=port;
9         this.host=host;
10        srv=null;
11        System.out.println("Proxy iniciado...");
12    }
13
14    public void download(String url){
15        if(!isRestricted(url))
16        {
17            if(srv == null)
18            {
19                srv = new RealServer(port,host);
20            }
21            srv.download(url);
22        }
23        else{
24            System.out.println("Actualmente se encuentra en un área que no permite la descarga del fichero.");
25        }
26    }
27
28    public boolean isRestricted(String fichero)
29    {
30        return !fichero.equals("/descarga/prohibida.avi");
31    }
32 }

```

- Principal:

```

1 public static void main(String [] args){
2
3     // Creamos el proxy a la página de descargas
4
5     Server srv = new ProxyServer(20,"http://paginadedescarga.as");
6
7
8
9     // Descargamos un archivo permitido
10
11    srv.download("/descarga/permitida.avi");
12
13
14
15    // En este punto será donde se cree el objeto RealServer
16
17
18
19    // Vamos a probar ahora con una descarga restringida
20
21    srv.download("/descarga/prohibida.avi");
22
23 }

```

Usos conocidos:

- NEXTSTEP para representar objetos en distribución, NXProxy.

Patrones relacionados:

- Adapter
- Decorator
- Iterator

Bibliografía:

No específico. (No específico). GoF Design Patterns (Versión 2.1.0) [Aplicación móvil].
Descargado de: <https://drive.google.com/file/d/0BywiVyFIlabXcVhGZIJBcnhWTKU/view>.

Patrones de Diseño (XIII): Patrones Estructurales - Proxy [Página web]. (s.f.). Ubicación
https://programacion.net/articulo/patrones_de_diseno_xiii_patrones_estructurales_proxy_1016.

O. Patrones de diseño. (Mayo de 2009). Proxy. Consultado en:
<http://dracohaste.blogspot.com/2009/05/patron-proxy.html>.

Junta de Andalucía. (s.f). Proxy. Marco de Desarrollo de la Junta de Andalucía.
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/200>.