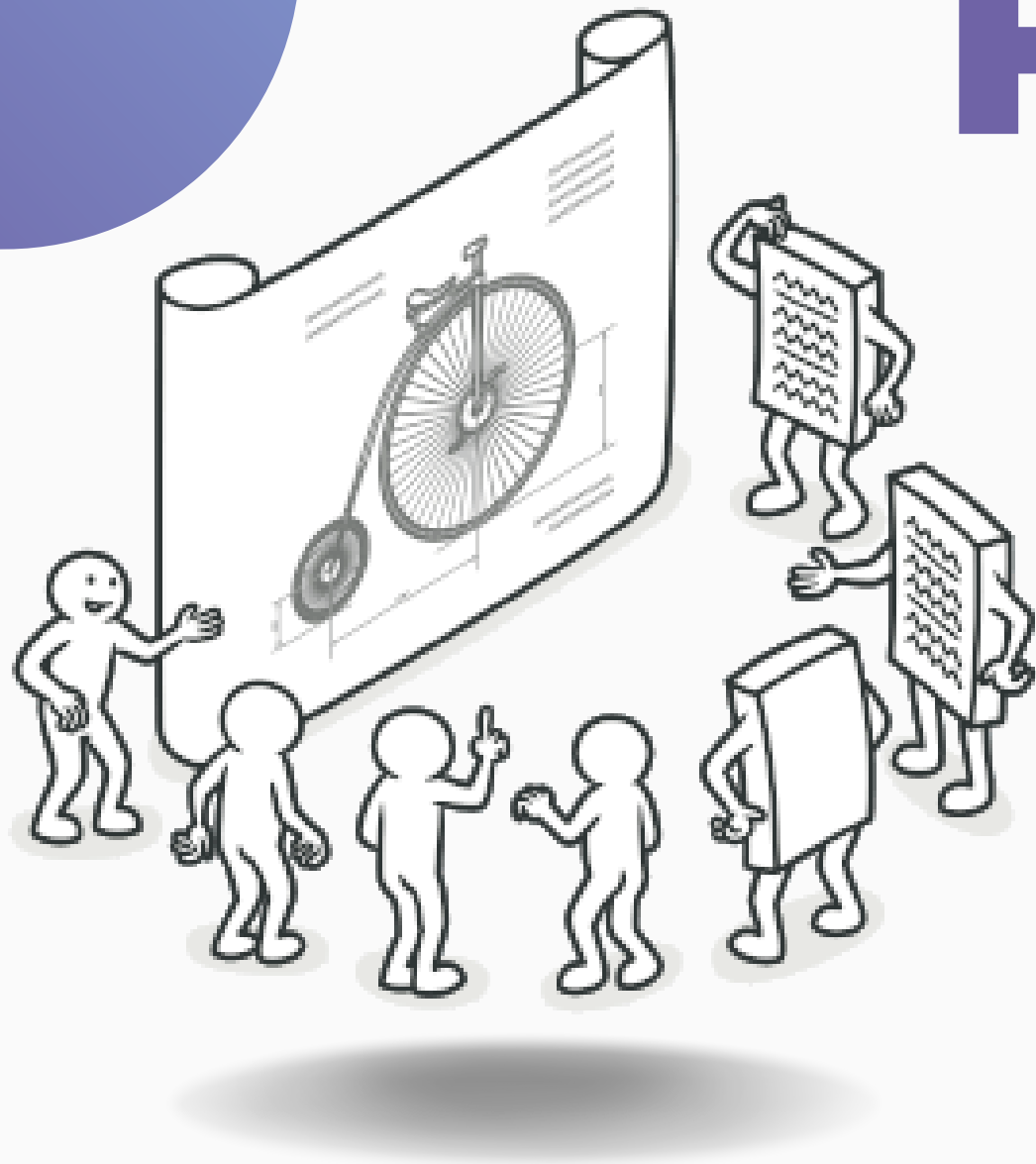


PATRON DE DISEÑO

PROXY

Felipe Rivas
Daniel Guzmán



PROXY

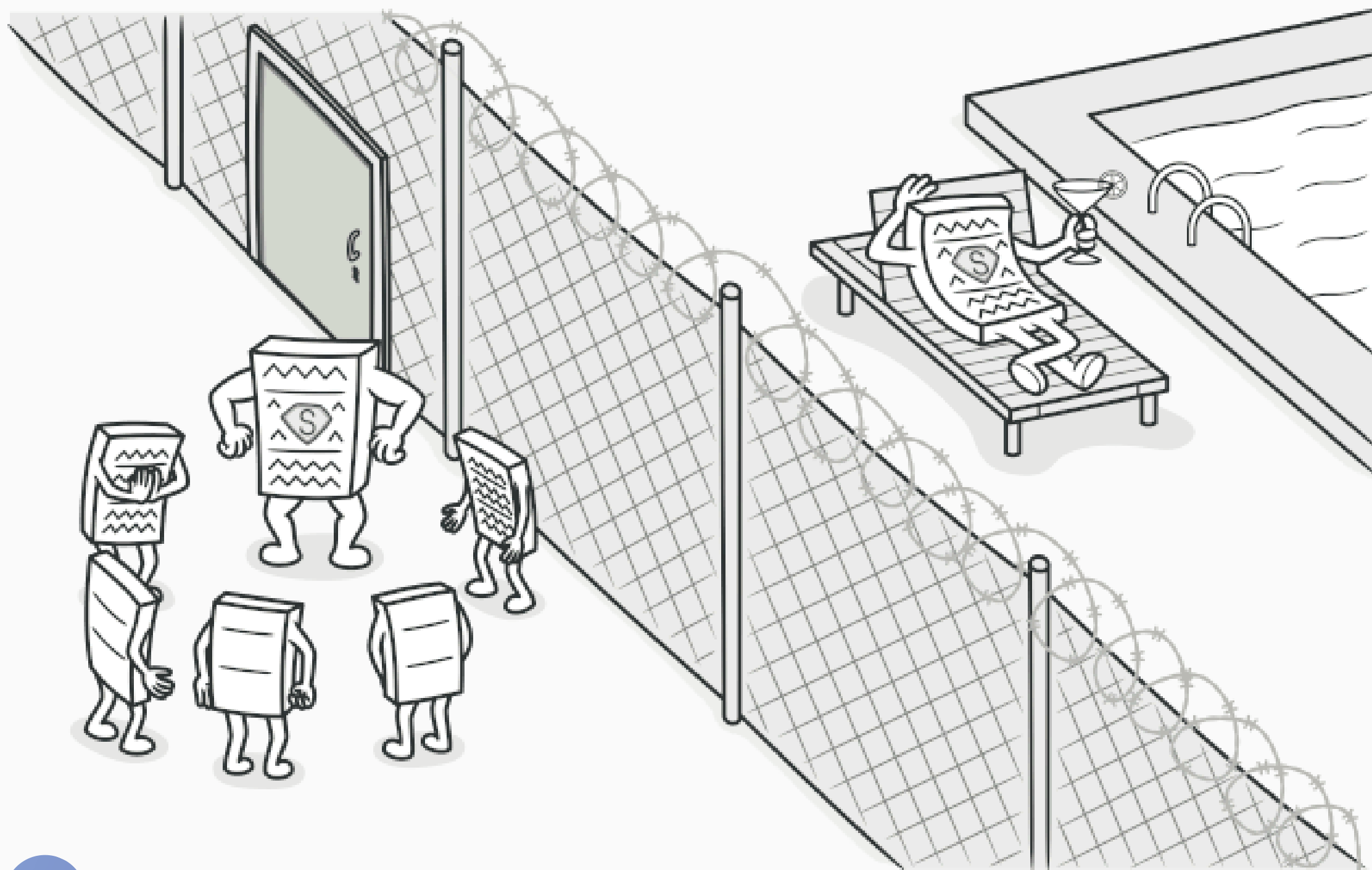
1. Propósito
2. Analogía con el mundo real
3. Problema de software
4. Solución del problema
5. Estructura o diagrama de clases
6. Implementación o código



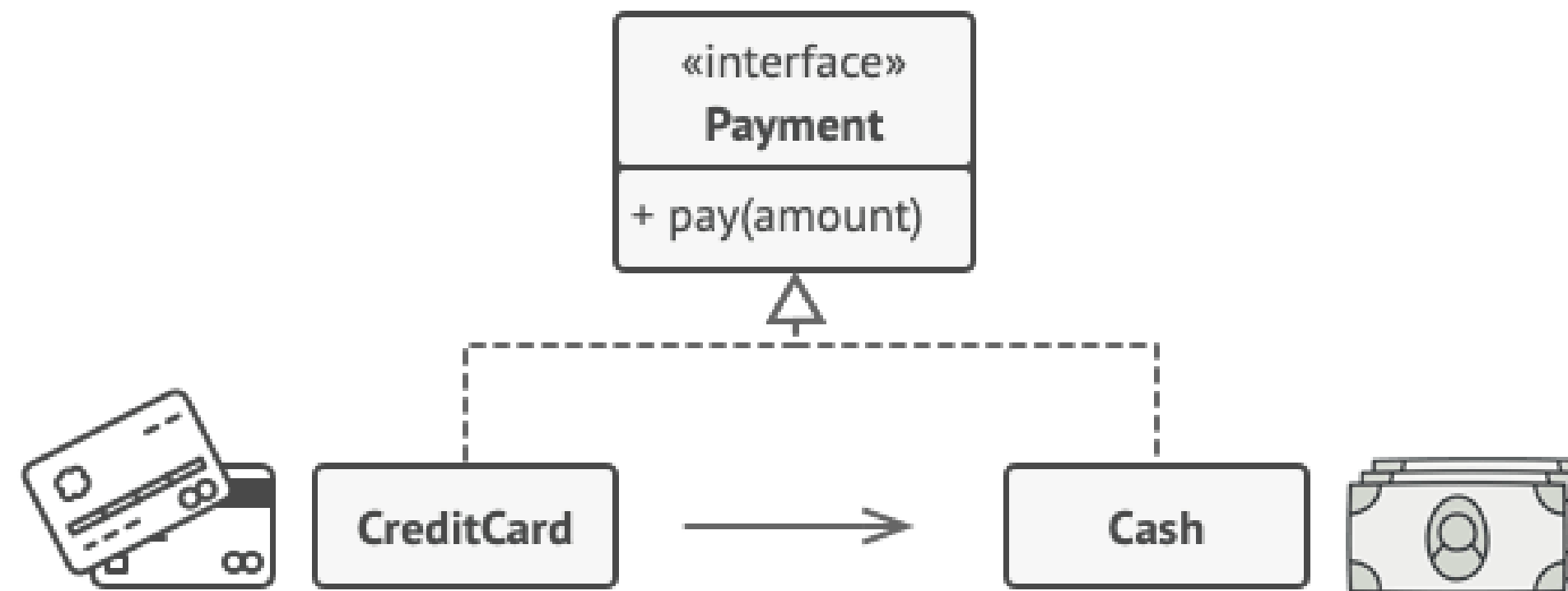
Propósito

Nos permite proporcionar un sustituto o marcador de posición para otro objeto. Un proxy controla el acceso al objeto original, permitiéndote hacer algo antes o después de que la solicitud llegue al objeto original.

Propósito

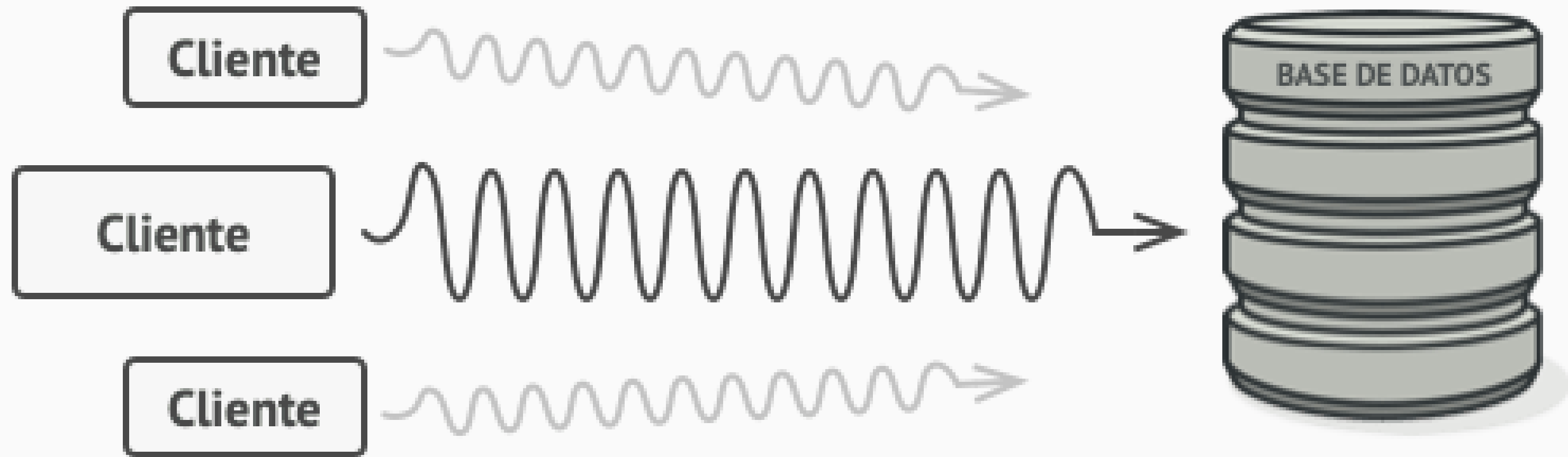


Analogía con el mundo real



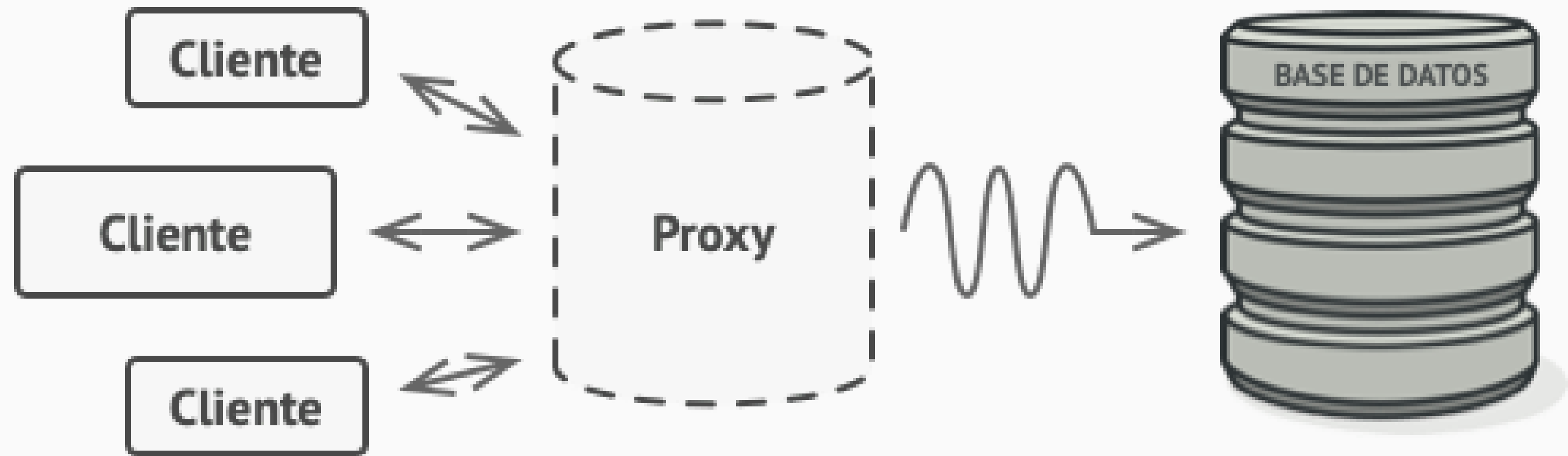
Tanto la tarjeta como los billetes sirven como método de pago, pero la tarjeta actúa como proxy en el sentido que tu plata esta mas segura y es mas privada en tu cuenta bancaria, haciendo que no puedan acceder a ella.

Problema



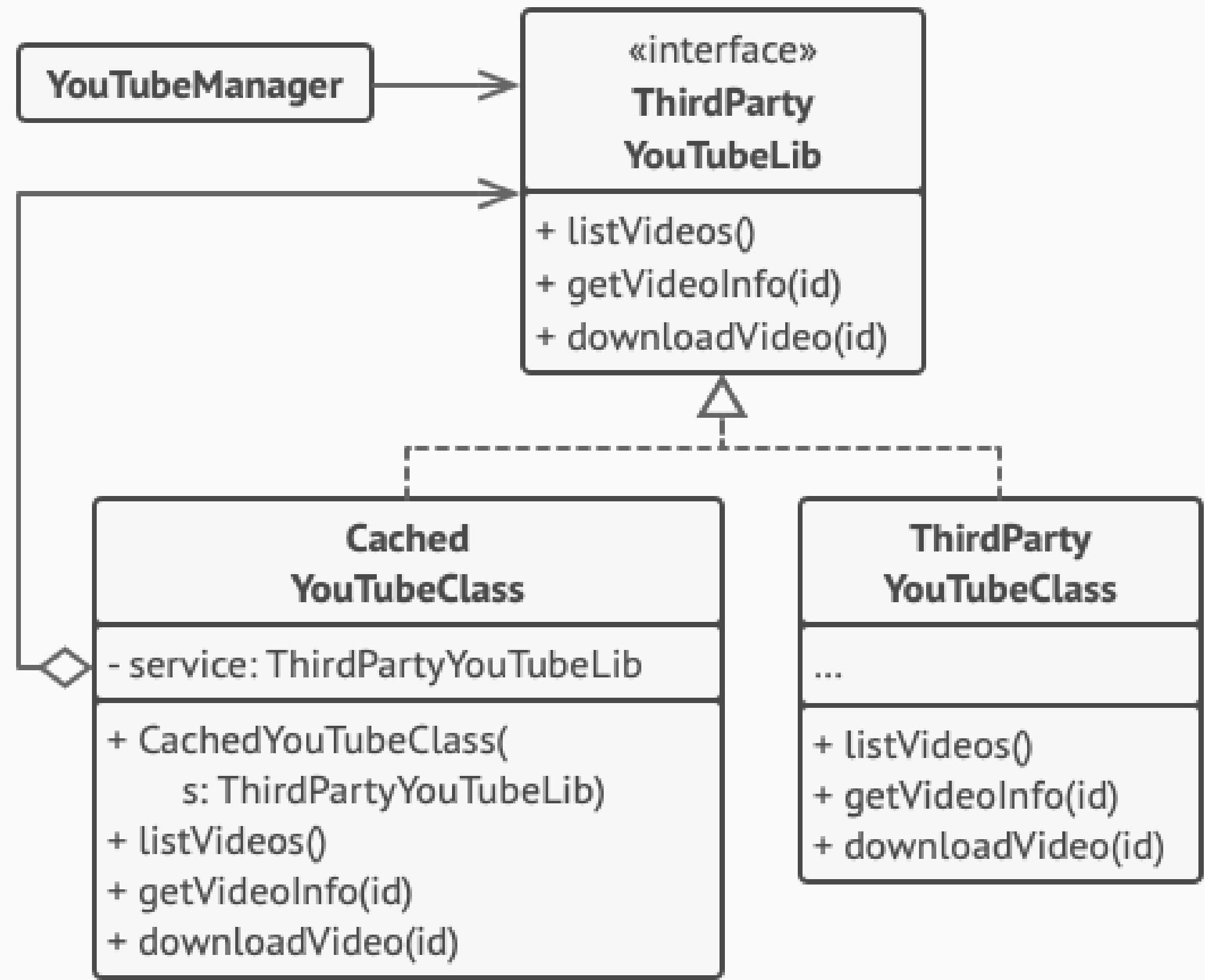
Las consultas a las bases de datos pueden ser muy lentas.

Solución



El proxy se camufla como objeto de la base de datos. Puede gestionar la inicialización diferida y el caché de resultados sin que el cliente o el objeto real de la base de datos lo sepan.

Estructura



Pseudocódigo

```
// La interfaz de un servicio remoto.
INTERFACE ThirdPartyYouTubeLib
    METHOD listVideos()
    METHOD getVideoInfo(id)
    METHOD downloadVideo(id)

// Implementación concreta del servicio de YouTube.
CLASS ThirdPartyYouTubeClass IMPLEMENTS ThirdPartyYouTubeLib
    METHOD listVideos()
        // Envía una solicitud API a YouTube.

    METHOD getVideoInfo(id)
        // Obtiene metadatos de algún video.

    METHOD downloadVideo(id)
        // Descarga un archivo de video de YouTube.
```

```
// Clase Proxy que añade almacenamiento en caché.
CLASS CachedYouTubeClass IMPLEMENTS ThirdPartyYouTubeLib
    PRIVATE ATRIBUTO service : ThirdPartyYouTubeLib
    PRIVATE ATRIBUTO listCache, videoCache
    ATRIBUTO needReset

    CONSTRUCTOR CachedYouTubeClass(service : ThirdPartyYouTubeLib)
        this.service = service

    METHOD listVideos()
        IF (listCache ES NULO O needReset)
            listCache = service.listVideos()
        RETURN listCache

    METHOD getVideoInfo(id)
        IF (videoCache ES NULO O needReset)
            videoCache = service.getVideoInfo(id)
        RETURN videoCache

    METHOD downloadVideo(id)
        IF (!downloadExists(id) O needReset)
            service.downloadVideo(id)
```

```
// Clase de manejo de la GUI.
CLASS YouTubeManager
    PROTEGIDO ATRIBUTO service : ThirdPartyYouTubeLib

    CONSTRUCTOR YouTubeManager(service : ThirdPartyYouTubeLib)
        this.service = service

    METHOD renderVideoPage(id)
        info = service.getVideoInfo(id)
        // Renderiza la página de video.

    METHOD renderListPanel()
        list = service.listVideos()
        // Renderiza la lista de miniaturas.

    METHOD reactOnUserInput()
        renderVideoPage()
        renderListPanel()
```

```
// Clase de inicialización de la aplicación.
CLASS Application
    METHOD init()
        aYouTubeService = NUEVO ThirdPartyYouTubeClass()
        aYouTubeProxy = NUEVO CachedYouTubeClass(aYouTubeService)
        manager = NUEVO YouTubeManager(aYouTubeProxy)
        manager.reactOnUserInput()
```

Diagrama de clases

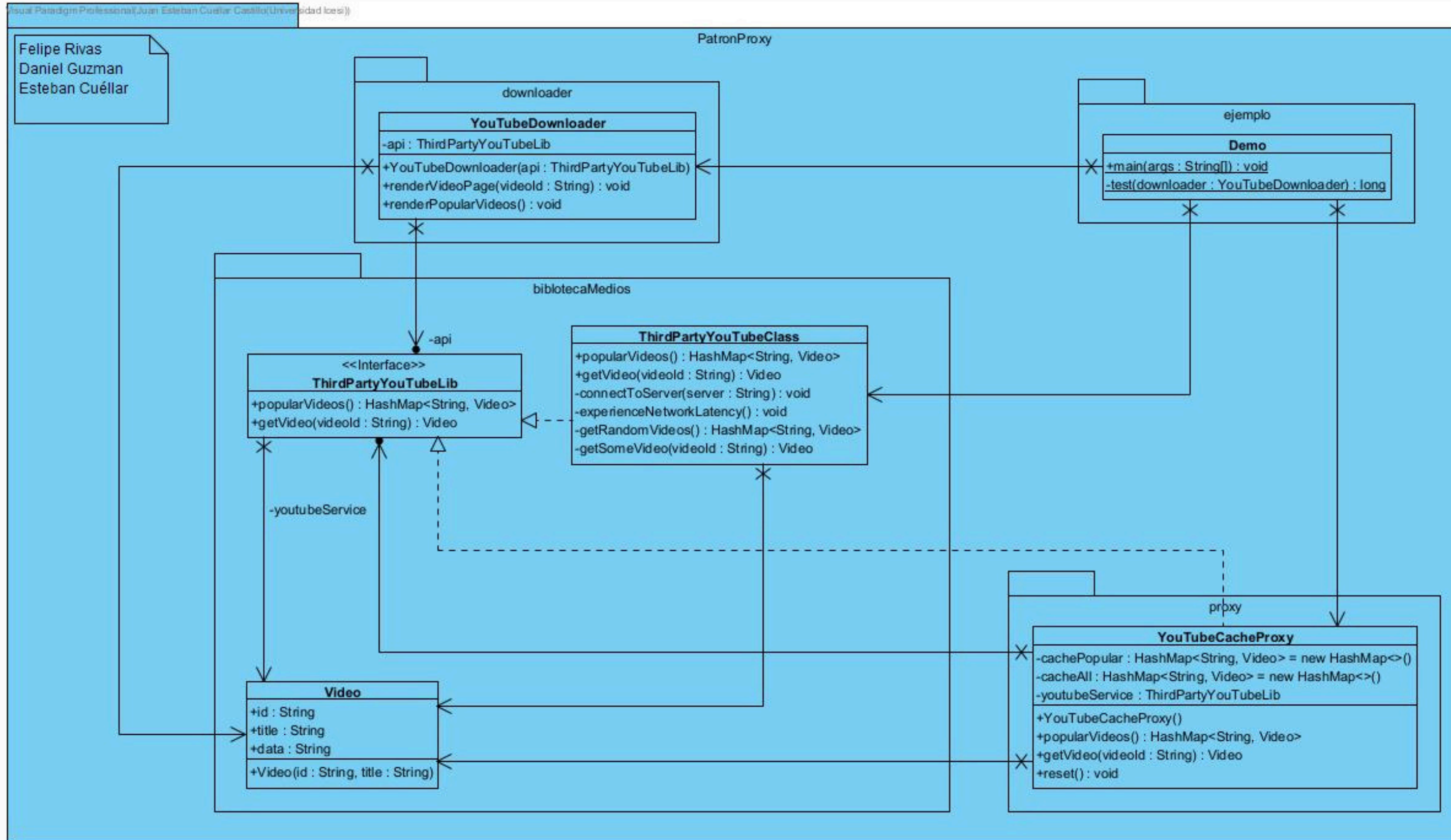
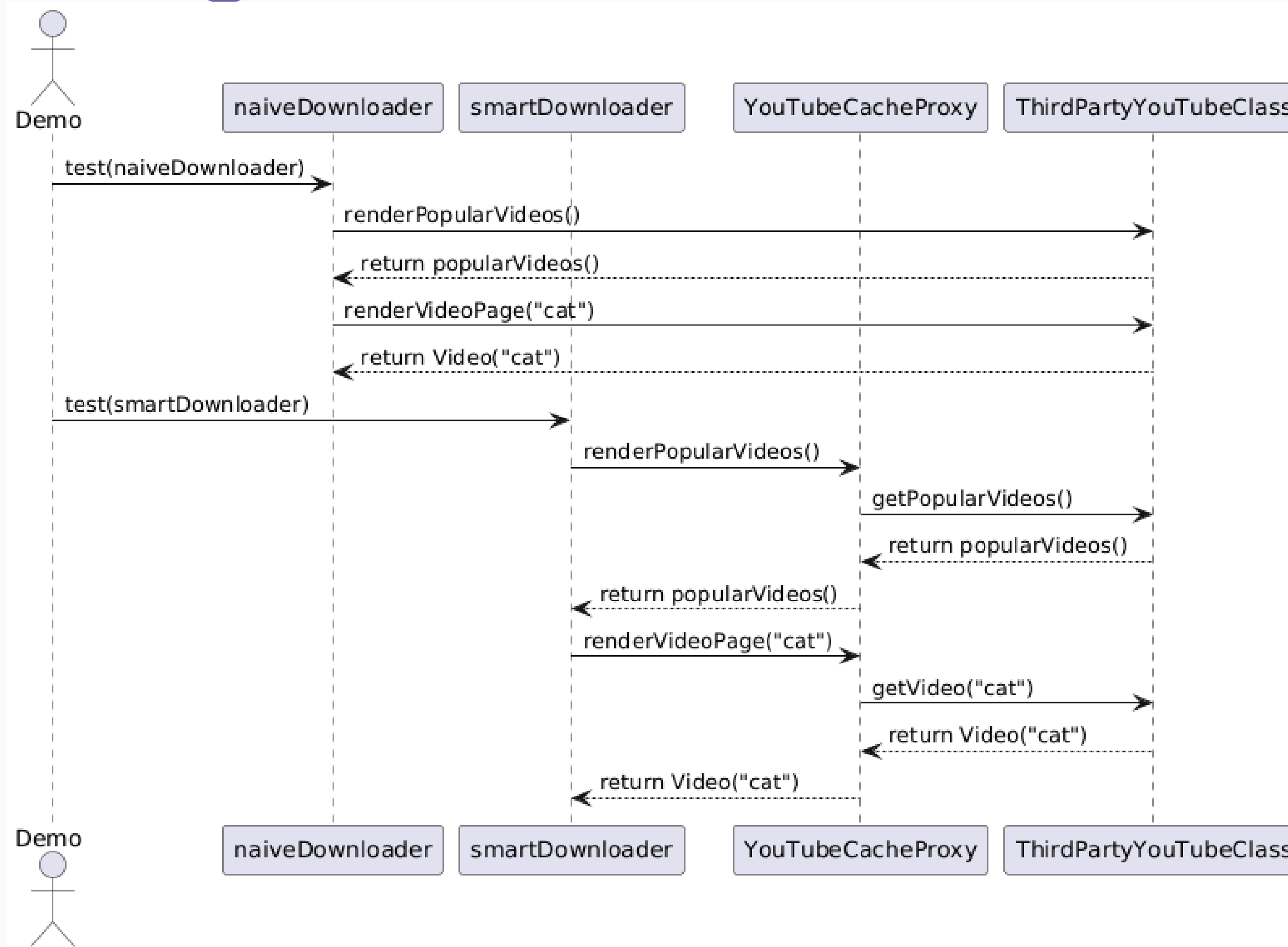


Diagrama de secuencia





Ponte en contacto con nosotros

Nuestras redes sociales

@felipe.rivass

Nuestra página web

<https://www.bmw.com.co/es/index.html>

Nuestro correo electrónico

fabiorivas69@gmail.com

Nuestro número de teléfono

(+57) 312 321 9120

