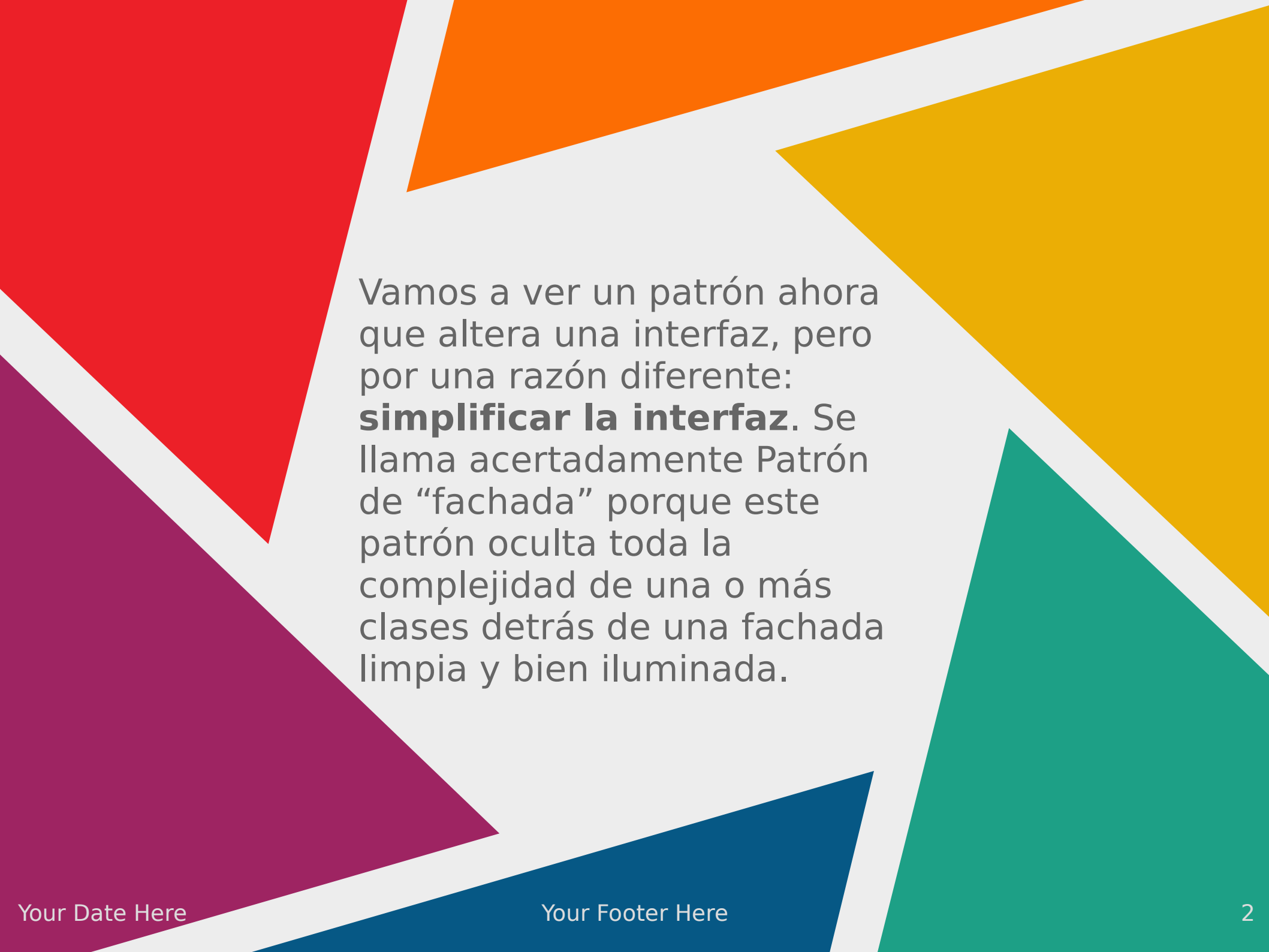




Facade



Vamos a ver un patrón ahora que altera una interfaz, pero por una razón diferente: **simplificar la interfaz**. Se llama acertadamente Patrón de “fachada” porque este patrón oculta toda la complejidad de una o más clases detrás de una fachada limpia y bien iluminada.

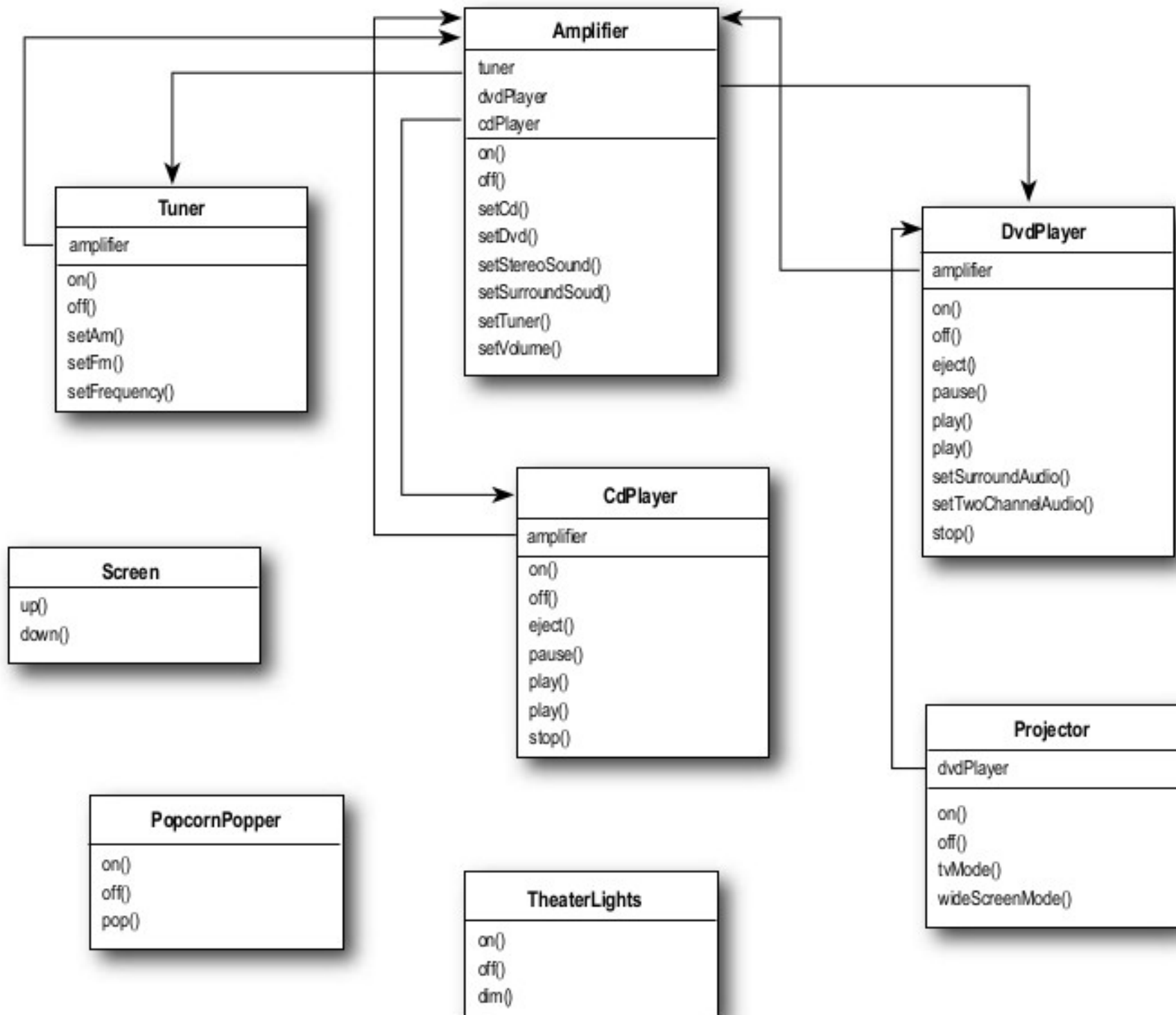
**FRONTEND**

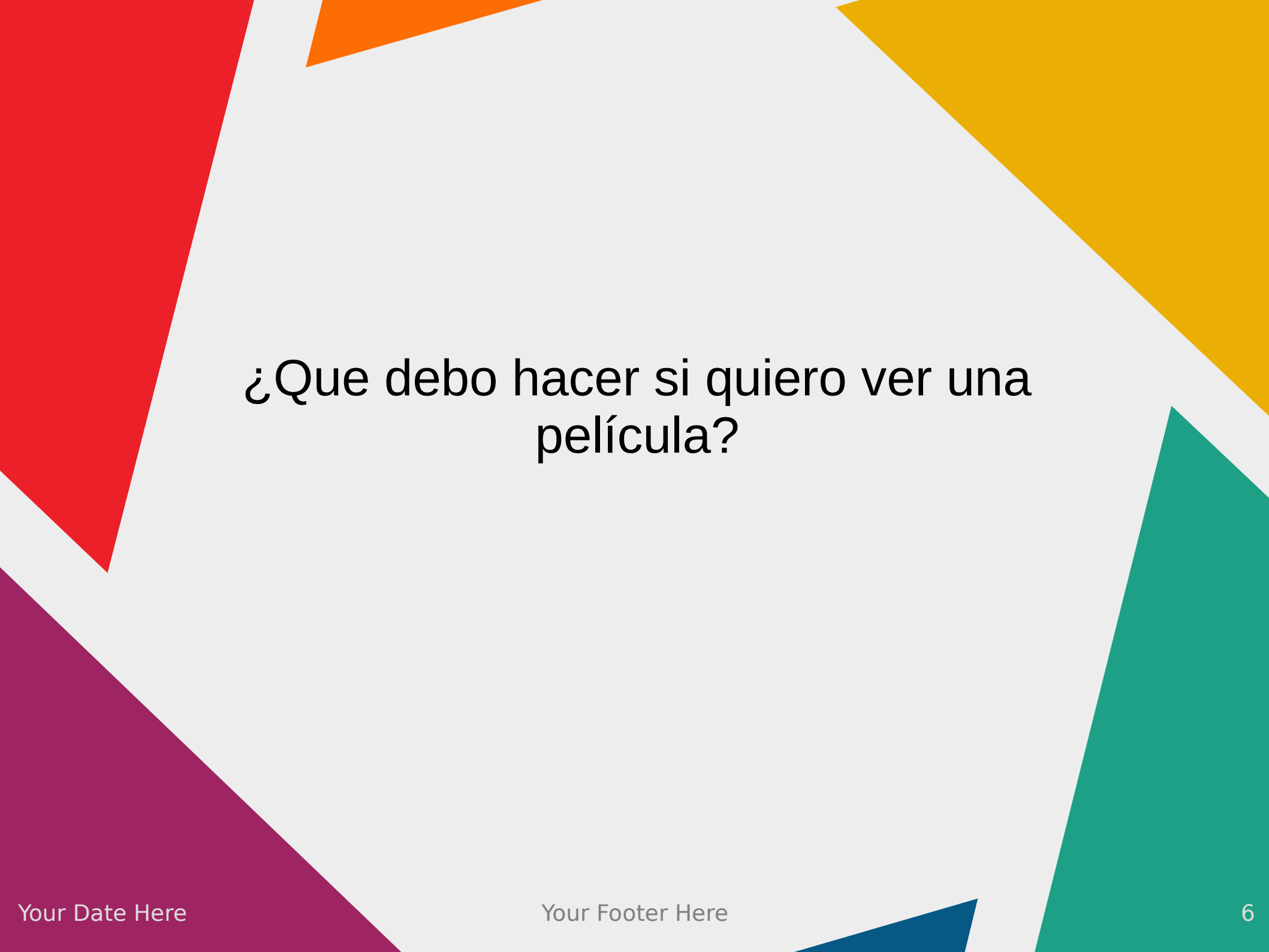


**BACKEND**



Vamos a la  
problemática del día





¿Que debo hacer si quiero ver una  
película?

- 1 Enciende el maquina de palomitas de maíz
- 2 Inicio entallamiento de palomitas
- 3 Atenúa las luces
- 4 Bajar la pantalla
- 5 Encienda el proyector
- 6 Configure la entrada del proyector en DVD

- 7 Ponga el proyector en modo de pantalla panorámica
- 8 Encienda el amplificador de sonido
- 9 Ajuste el amplificador a la entrada de DVD
- 10 Ajuste el amplificador a sonido envolvente
- 11 Ajuste el volumen del amplificador
- 12 Encienda el reproductor de DVD
- 13 Comience a reproducir la película el DVD



Six different classes involved!

```
popper.on();  
popper.pop();
```

Turn on the popcorn popper and start popping...

```
lights.dim(10);
```

Dim the lights to 10%...

```
screen.down();
```

Put the screen down...

```
projector.on();  
projector.setInput(dvd);  
projector.wideScreenMode();
```

Turn on the projector and put it in wide screen mode for the movie...

```
amp.on();  
amp.setDvd(dvd);  
amp.setSurroundSound();  
amp.setVolume(5);
```

Turn on the amp, set it to DVD, put it in surround sound mode and set the volume to 5...

```
dvd.on();  
dvd.play(movie);
```

Turn on the DVD player...  
and FINALLY, play the movie!

- Cuando la película termina, ¿cómo apagas todo?
- ¿No tendrías que hacer todo esto de nuevo, al revés?
- ¿Entonces lo que hay que hacer? ¡La complejidad de usar su sistema de cine en casa se está volviendo evidente!
- Veamos cómo el Patrón de fachada puede sacarnos de este lío para que podamos disfrutar de la película ...

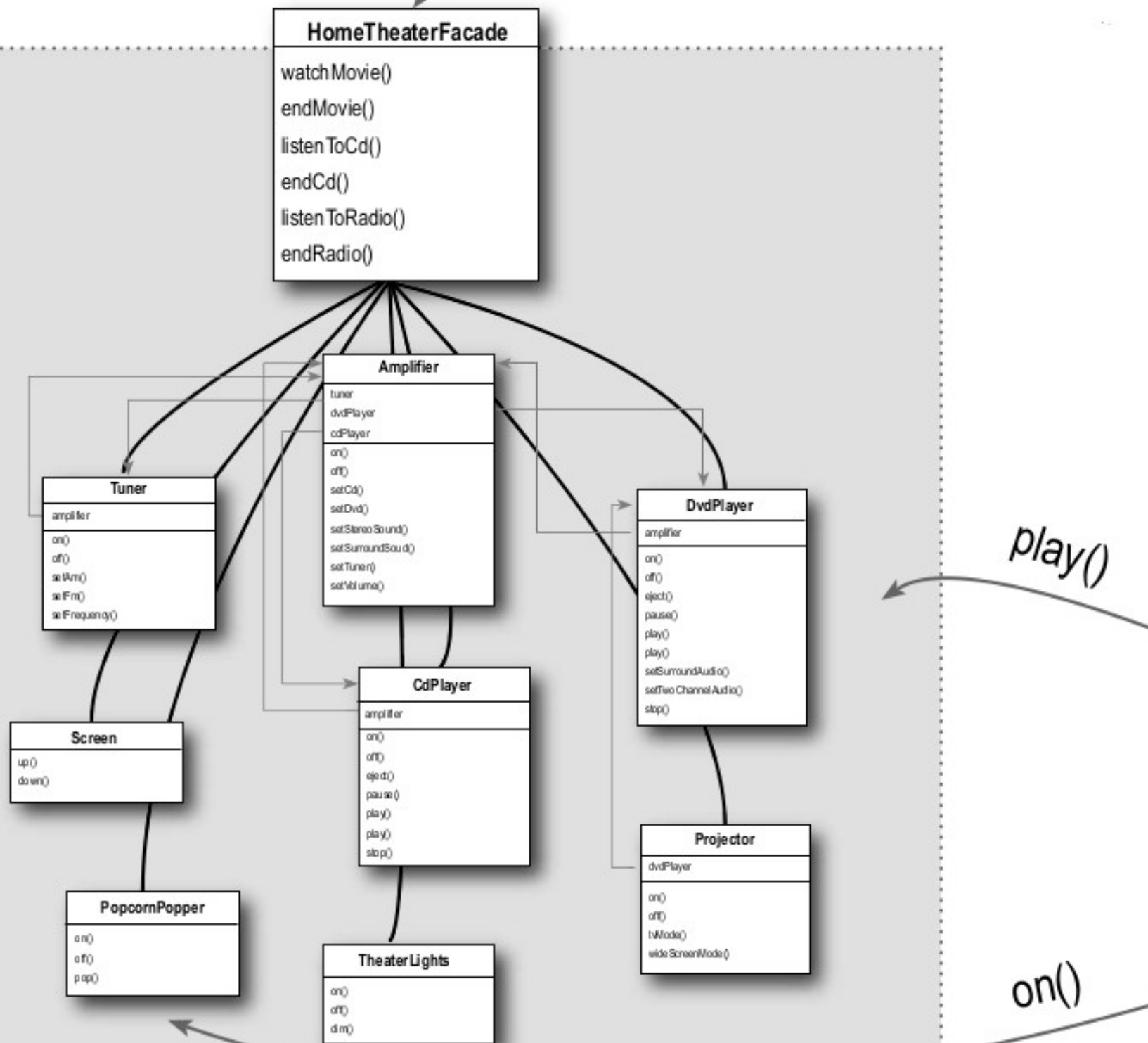
## Luces, cámara, fachada!

- El Patrón de fachada puede tomar un subsistema complejo y hacerlo más fácil de usar implementando una clase Fachada que proporciona una interfaz más razonable. No te preocupes si necesita la potencia del subsistema complejo, aún puede utilizarlo, pero si lo único que necesita es una interfaz sencilla, la Fachada está a su disposición.

- 1. Creamos una nueva clase, HomeTheaterFacade que expone algunos métodos simples como watchMovie().
- 2. La clase Facade trata los componentes del sistema de cine en casa como un subsistema y solicita al subsistema que implemente su método watchMovie().

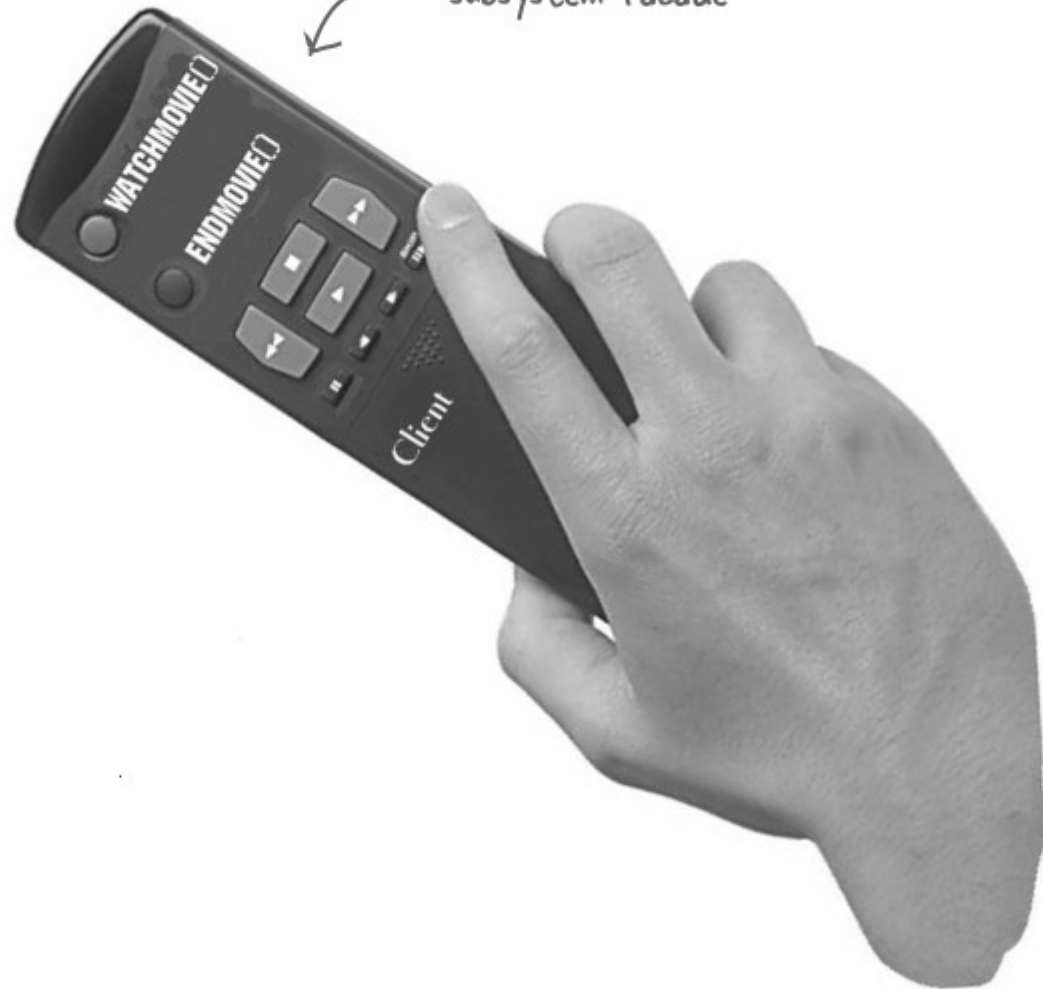
- 3. Su código de cliente ahora llama a métodos en la Fachada del cine en casa, no en el subsistema.
- Así que ahora para ver la película, simplemente llamamos a un método, `watchMovie()`; y se comunica con las luces, reproductor de DVD, proyector, amplificador, pantalla, y la maquina de palomitas de maíz por nosotros.

- 4. La fachada aún deja el subsistema accesible para ser utilizado directamente. Si tu necesita la funcionalidad avanzada de las clases del subsistema, están disponibles para su uso.



watchMovie()

A client of the  
~~subsystem~~ facade







## Control IR



Mi TV/Mi Box



TV



Decodificador



A/A



Ventilador



Caja inteligente



Receptor A/V



Reproductor de  
DVD



Proyector

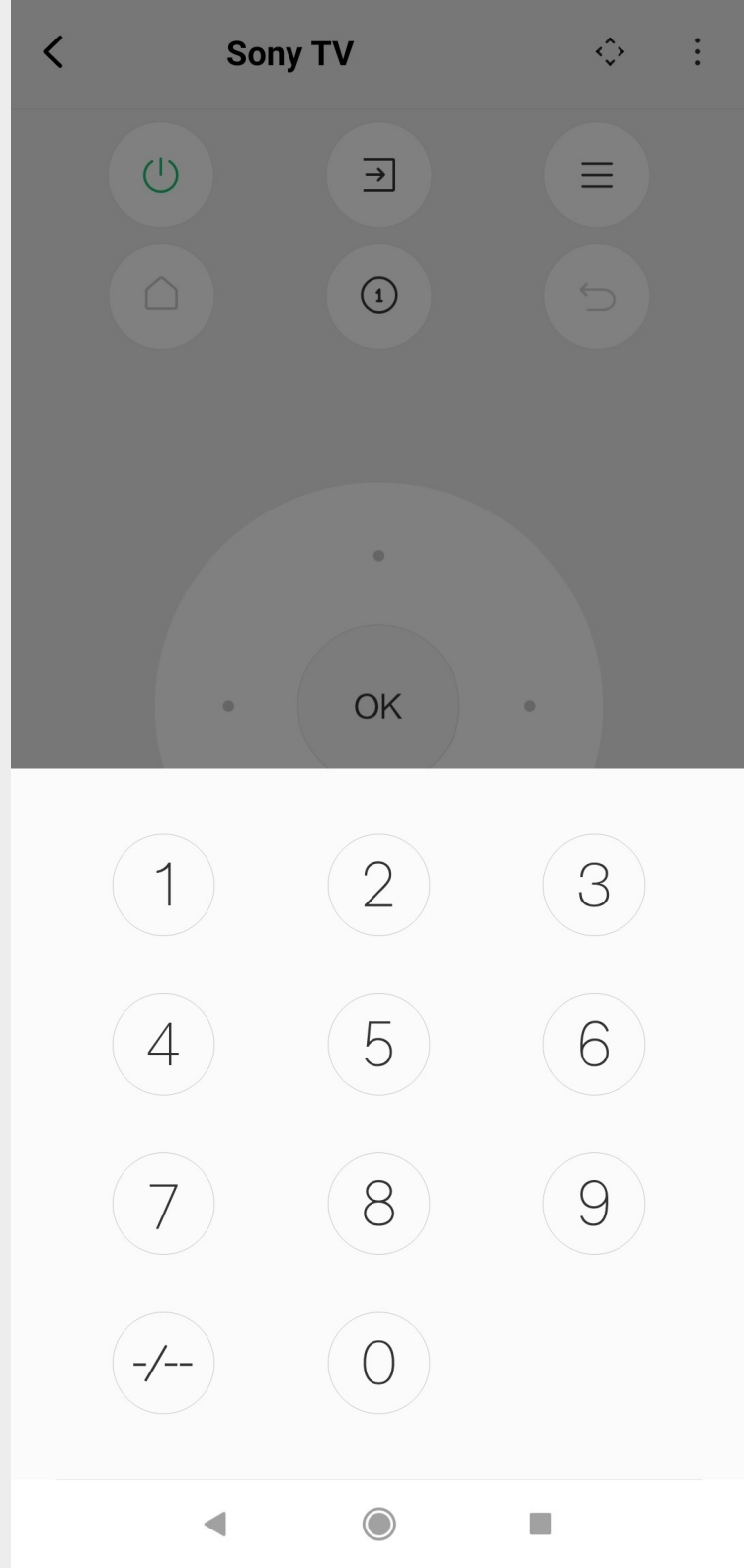
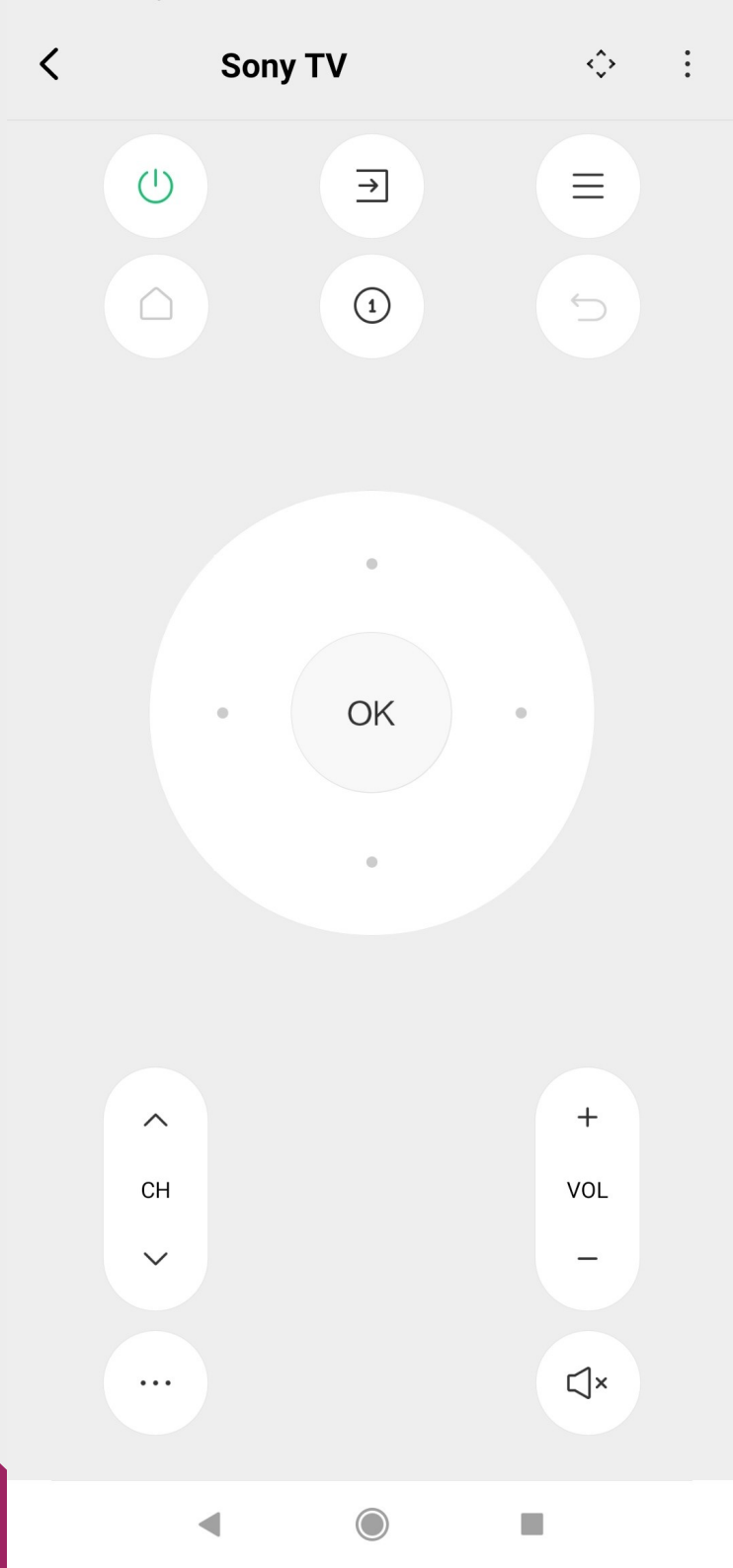


Televisión satelital  
china



Cámara






- Una fachada no solo simplifica una interfaz, sino que desacopla un cliente de un subsistema de componentes.
- Las fachadas y los adaptadores pueden envolver varias clases, pero la intención de una fachada es simplificar, mientras que un adapter es para convertir la interfaz a algo diferente.

- Avancemos en la construcción de HomeTheaterFacade: el primer paso es usar la composición para que la fachada tenga acceso a todos los componentes del subsistema:


```
public class HomeTheaterFacade {  
    Amplifier amp;  
    Tuner tuner;  
    DvdPlayer dvd;  
    CdPlayer cd;  
    Projector projector;  
    TheaterLights lights;  
    Screen screen;  
    PopcornPopper popper;
```

Here's the composition; these are all the components of the subsystem we are going to use.



```
    public HomeTheaterFacade(Amplifier amp,  
                             Tuner tuner,  
                             DvdPlayer dvd,  
                             CdPlayer cd,  
                             Projector projector,  
                             Screen screen,  
                             TheaterLights lights,  
                             PopcornPopper popper) {
```


The facade is passed a reference to each component of the subsystem in its constructor. The facade then assigns each to the corresponding instance variable.



```
        this.amp = amp;  
        this.tuner = tuner;  
        this.dvd = dvd;  
        this.cd = cd;  
        this.projector = projector;  
        this.screen = screen;  
        this.lights = lights;  
        this.popper = popper;  
    }
```

```
    // other methods here
```

We're just about to fill these in...





## Control IR



Mi TV/Mi Box



TV



Decodificador



A/A



Ventilador



Caja inteligente



Receptor A/V



Reproductor de  
DVD



Proyector



Televisión satelital  
china



Cámara



## Selección Proyector



Buscar marcas

Marcas populares

HP

BenQ

Epson

Acer

Panasonic

Hitachi

NEC

Todas

3M

Acer

Acto

Ampro

Artograph

ASK Proxima

Barco

BenQ

Boxlight

#

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W


X

Y


Z

- Implementemos la interfaz simplificada
- Ahora es el momento de combinar los componentes del subsistema en una interfaz unificada.

```
public void watchMovie(String movie) {  
    System.out.println("Get ready to watch a movie...");  
    popper.on();  
    popper.pop();  
    lights.dim(10);  
    screen.down();  
    projector.on();  
    projector.wideScreenMode();  
    amp.on();  
    amp.setDvd(dvd);  
    amp.setSurroundSound();  
    amp.setVolume(5);  
    dvd.on();  
    dvd.play(movie);  
}
```

 watchMovie() follows the same sequence we had to do by hand before, but wraps it up in a handy method that does all the work. Notice that for each task we are delegating the responsibility to the corresponding component in the subsystem.

```
public void endMovie() {  
    System.out.println("Shutting movie theater down...");  
    popper.off();  
    lights.on();  
    screen.up();  
    projector.off();  
    amp.off();  
    dvd.stop();  
    dvd.eject();  
    dvd.off();  
}
```

 And endMovie() takes care of shutting everything down for us. Again, each task is delegated to the appropriate component in the subsystem.



```
public class HomeTheaterTestDrive {  
    public static void main(String[] args) {  
        // instantiate components here  
  
        HomeTheaterFacade homeTheater =  
            new HomeTheaterFacade(amp, tuner, dvd, cd,  
                                   projector, screen, lights, popper);  
  
        homeTheater.watchMovie("Raiders of the Lost Ark");  
        homeTheater.endMovie();  
    }  
}
```

Here we're creating the components right in the test drive. Normally the client is given a facade, it doesn't have to construct one itself.

First you instantiate the Facade with all the components in the subsystem.

Use the simplified interface to first start the movie up, and then shut it down.

Here's the output.

Calling the Facade's  
watchMovie() does all  
this work for us...



...and here, we're done  
watching the movie, so  
calling endMovie() turns  
everything off.



File Edit Window Help SnakesWhy'dItHaveToBeSnakes?

```
%java HomeTheaterTestDrive
```

```
Get ready to watch a movie...
```

```
Popcorn Popper on
```

```
Popcorn Popper popping popcorn!
```

```
Theater Ceiling Lights dimming to 10%
```

```
Theater Screen going down
```

```
Top-O-Line Projector on
```

```
Top-O-Line Projector in widescreen mode (16x9 aspect ratio)
```

```
Top-O-Line Amplifier on
```

```
Top-O-Line Amplifier setting DVD player to Top-O-Line DVD Player
```

```
Top-O-Line Amplifier surround sound on (5 speakers, 1 subwoofer)
```

```
Top-O-Line Amplifier setting volume to 5
```

```
Top-O-Line DVD Player on
```

```
Top-O-Line DVD Player playing "Raiders of the Lost Ark"
```

```
Shutting movie theater down...
```

```
Popcorn Popper off
```

```
Theater Ceiling Lights on
```

```
Theater Screen going up
```

```
Top-O-Line Projector off
```

```
Top-O-Line Amplifier off
```

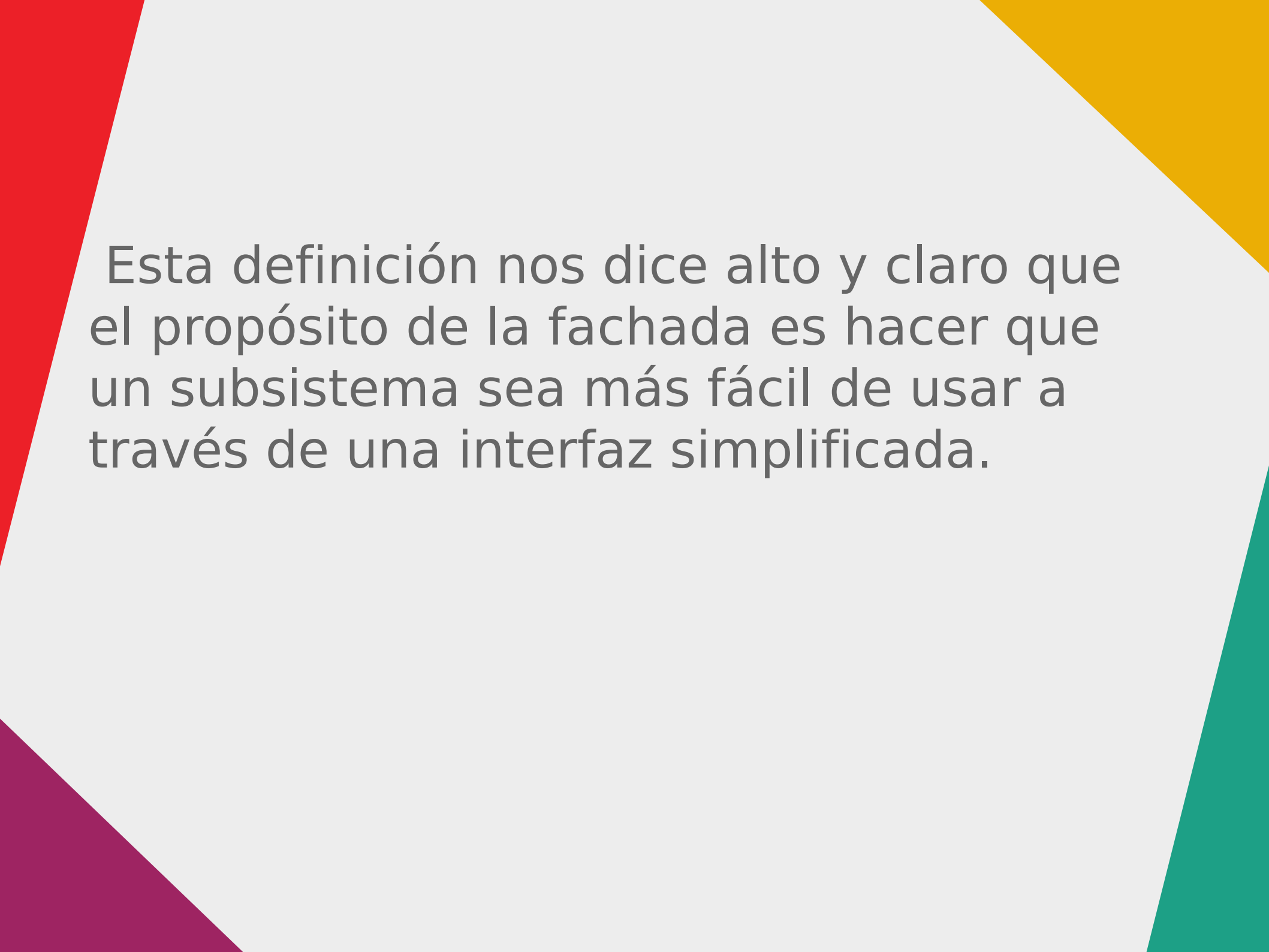
```
Top-O-Line DVD Player stopped "Raiders of the Lost Ark"
```

```
Top-O-Line DVD Player eject
```

```
Top-O-Line DVD Player off
```

```
%
```

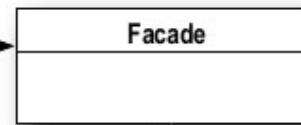
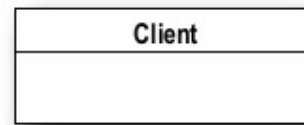
El patrón de Facade (fachada) proporciona una interfaz unificada para un conjunto de interfaces en un subsistema. Facade define una interfaz de nivel superior que hace que el subsistema sea más fácil de usar.



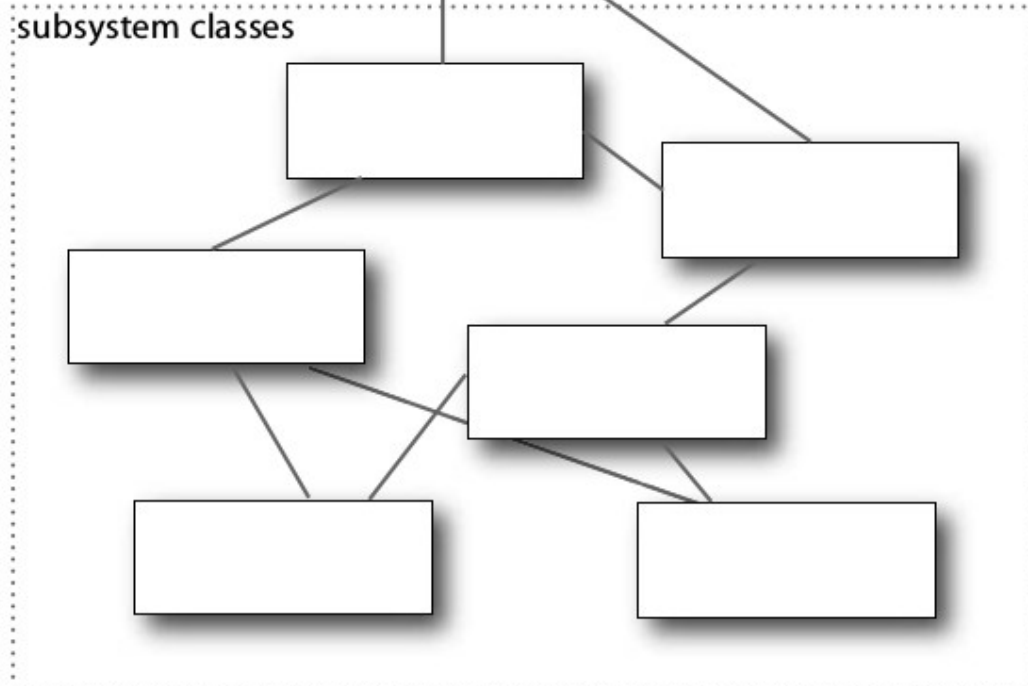
Esta definición nos dice alto y claro que el propósito de la fachada es hacer que un subsistema sea más fácil de usar a través de una interfaz simplificada.

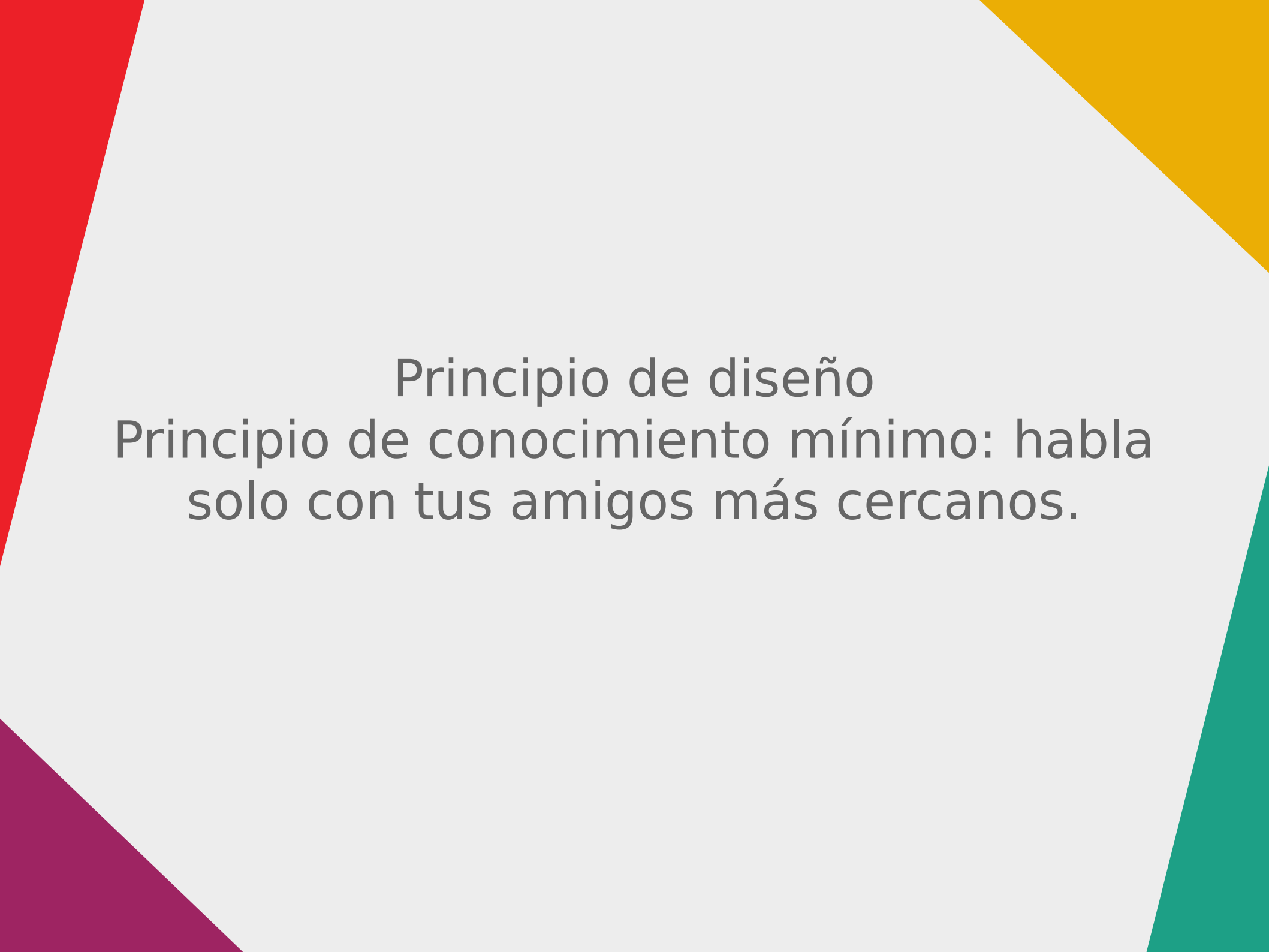
Happy client whose  
job just became  
easier because of  
the facade.

More complex subsystem.



Unified interface  
that is easier to use.





Principio de diseño  
Principio de conocimiento mínimo: habla  
solo con tus amigos más cercanos.

¿Que hace que?

A)Decorator

- \_\_ Convierte una interfaz en otra

B)Adapter

- \_\_ No altera una interfaz pero agrega responsabilidades

C)Facade

- \_\_ Hace una interfaz simple