## Principio S.O.L.I.D: Principio de segregación de interfaces.

El principio se segregación de interfaces nos permite separar funciones que tengan algo que ver dentro de una interfaz, para poder crear clases mas concretas que implementen un numero de interfaces mayor, asi concretando mas las funciones que puede hacer cada clase evitando que se generen métodos adicionales que se queden vacíos.

Caso de uso:

Los teclados sabemos que tienen diferentes funciones aparte de las de escribir, tenemos funciones de luz y de control de música, por ello tenemos que diferenciar en diferentes interfaces las funciones que estén disponibles para cada tipo de uso. En este ejemplo vamos a crear un ordenador completo que tiene 8 esquemas de luz, y un ordenador barato que no es compatible con los controles de música y únicamente dispone de 3 esquemas de luz.

### Mala resolución del problema:

#### Interface teclado:

Reproducir/pausar

Siguiente cancion

Anterior cancion

Escribir letranumerosigno

ToogleMayus

Esquemaluzsiguiente

Esquemaluzanterior

ColorPersonalizado

### Con la segregación de interfaces:

SuperClase: TecladoCompleto implements:

#### Interface Escritura:

Escribir letranumerosigno

Tooglemayus.

#### Interface ControlMusica:

Reproducir/pausar

Siguiente cancion

Anterior canción

#### Interface ControlLuz:

Esquemaluzsiguiente

Esquemaluzanterior

Color personalizado

## UML

