# Diseño del Sistema

## DISEÑO FINAL

La parte física de la aplicación consiste básicamente en capturar imágenes, para ello se dispone de un artilugio similar a un trípode y una webcam, el montaje conlleva montar la webcam en el trípode y enfocarla en vertical hacia abajo donde habrá una mesa o una superficie lisa y uniforme, esta superficie será la utilizada por el usuario para mover las manos e interactuar con la aplicación.

Respecto al software utilizado se basa en el framework Processing, la integración con el hardware se realiza a través de la librería “Video” y se ajusta la entrada a una resolución soportada por la webcam, respecto a la generación de audio se utiliza la librería “Minim”. El software se basa principalmente en la librería “Blobscanner” que provee un mecanismo para detectar siluetas en las imágenes, como esqueleto de la propia aplicación se utiliza un ejemplo de la propia librería en el que dada una entrada de video el programa devuelve el número de dedos que detecta y proyecta la silueta de los dedos.

A nivel de implementación del proyecto y basándonos en las bases mencionadas, decidimos colocar en una capa inferior a la proyección de las siluetas el dibujo de una mesa de DJ, con esto el usuario verá sus manos por encima de la mesa y tendrá mas facilidad para interactuar con la misma.

A la hora de implementar la interacción con el usuario se decidió colocar unas figuras como rectángulos o circunferencias entre la capa de las siluetas y el dibujo de la mesa para que actúen como zonas de interacción cuando el usuario pone un dedo sobre estos una acción se ejecutará, para esto se utiliza un patrón “Observer”, en el bucle principal se notifica a todos los botones(Clase que encapsula las figuras previamente mencionadas) el estado actual del detector de siluetas, y son estos botones los que estiman si una silueta se encuentra en su zona de interacción y actúan en consecuencia. Botón es una interfaz, y existen varias implementaciones distintas en función a la acción requerida (‘Play’, ’Pause’, ‘Trigger Sound’, …).

## OTROS DISEÑOS/PROTOTIPOS

La idea de utilizar un patrón “Observer” para los botones siempre estuvo presente, sin embargo su implementación concreta era bastante distinta a la final, inicialmente se diseño pensando utilizar una única clase Botón con un patrón “Strategy” para la acción a realizar, sin embargo finalmente se descarto dado que no teníamos realmente una necesidad de cambiar dinámicamente el comportamiento del botón con lo que sería complicar la implementación sin necesidad, solución: Utilizar polimorfismo.

Otro idea que se descartó, también relacionada con la implementación del patrón “Observer” es una con la cual pretendíamos optimizar el rendimiento y mejorar la detección de pulsación. La aplicación base inicial utiliza un detector para la totalidad de la entrada de imagen, nuestra idea fue que cada Botón incluyese un detector propio que se ocupase de analizar específicamente cada zona de interacción y así ahorrarnos el tiempo de procesar partes de la imagen que no nos interesan, además de esta forma le podríamos pedir al detector el número de siluetas en esa zona y utilizarla por ejemplo para que solo se pulsase un botón con dos dedos por encima, cosa que no se puede hacer con un detector global puesto que la implementación de la librería no permite pedir el número de siluetas en una zona específica del detector sino solo en su totalidad. Cuando nos pusimos a implementar esto resultó que el sobrecoste producido por la generación de los detectores específicos (3 por cada botón pues los detectores trabajan por capas primero siluetas -> siluetas de manos -> siluetas de dedos) y la forma de procesar los datos de estos (En cada iteración teníamos que fraccionar la imagen principal + pasar el procesado de los 3 detectores y repetir por cada botón) llevó al efecto contrario al deseado puesto que el rendimiento se vio enormemente afectado con lo cual descartamos esto y volvimos a la implementación básica en el botón le pregunta al detector si hay una silueta en el centro de su figura y si se da el caso procede a realizar la acción.

https://github.com/robdanet/blobscanner