

# Proyecto *Rachel*: bitácora

## 22 septiembre 2019

Se hizo una prueba con el documento `eyetracker.cpp`, llegando al mismo resultado en el que la ventana de la webcam se cerraba.

El código viene de la [esta liga](#).

En la cuál, en la función `main` existía un error dentro en la cual impedía que la ventana permaneciera abierta. No había un control para evitar su salida y cierre.

Se busco una alternativa para leer la imagen desde la cámara web, y se encontro en el [este enlace](#). La solución propuesta funciona para dos cámaras conectadas, **lo que podría ser útil más adelante**.

Se reviso este código usando un archivo en `/Prueba 2/eyedetector.cpp`, que logro abrir la ventana con la cámara web. Se adapto de la función `main` de `eyedetector.cpp` a la misma función en `eyetracker.cpp`. Esta prueba tuvo éxito después de realizar algunos ajustes de las variables.

Se hizo la prueba con `eyetracker.cpp` y finalmente la ventana con la imagen en la cámara pudo abrirse y funciona sin contratiempo con el resto del código. No pude notar si el cursor se movía, puesto que funciona sin gafas (y no veo la pantalla).

Por los resultados mostrados en la terminal se toman demasiados datos a la vez, un objetivo es corregir a una muestra de datos aceptable para evitar posible ruido o torpezas durante la captura.

Falta por lo tanto documentar correctamente usando los comentarios, indentar el código y hacerlo más fluido a la vista.

## 06 de octubre 2019

Se creo el repositorio en GitHub llamado [Proyecto Rachael](#) para tener un control del proyecto. Se continuo modificando el código en `eyetracker.cpp`, en el cuál se incluyeron la opción de poder elegir el index de la cámara conectada, se corrigió que el ratón estuviera estático y además dentro de la función se calculo la mitad de la pantalla, punto de partida del cursor. También se añadió la posibilidad de guardar un archivo `.csv` con la etiqueta de tiempo y las coordenadas de la posición del puntero.

Entre los problemas a resolver es la posición del puntero, el cuál no llega a los extremos de la pantalla. Para esto se me ocurren tres soluciones posibles:

- Exagerar la posición del puntero usando un factor con el cuál multiplicar la posición.
- Subir la cantidad de cuadros por segundo que toma el programa en relación con la cámara.
- Tomar una x cantidad de cuadros por segundo e interpretarlo como fotografías.

Una vez resuelto el problema de la posición del puntero, podemos pasar a análisis y filtrado de datos y además de la calibración.