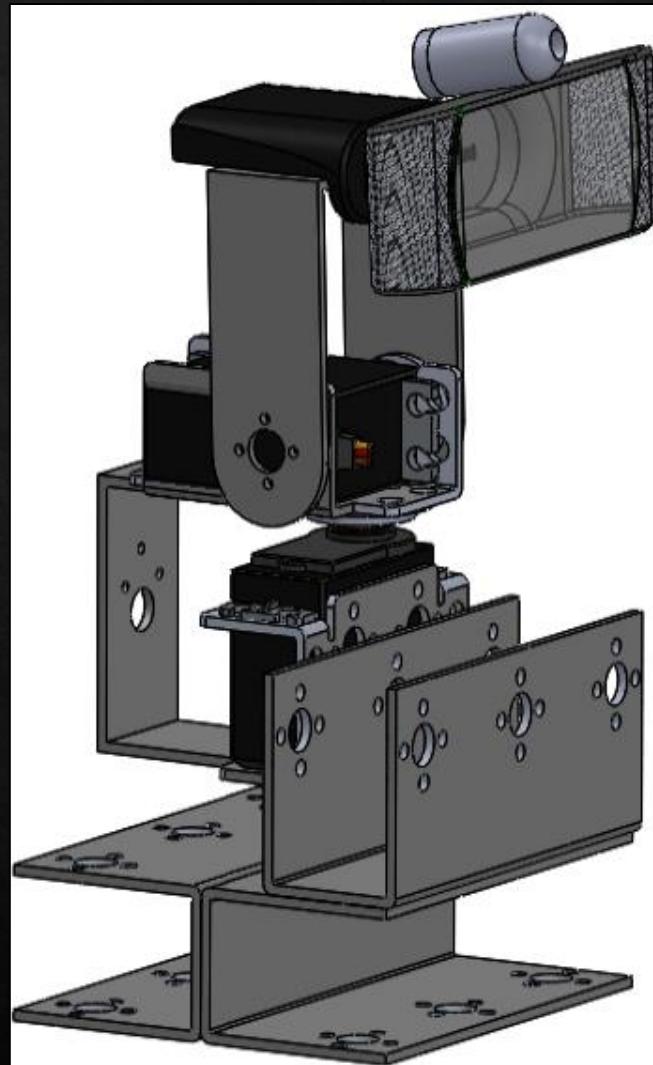


Sistema de servo-visión

- Martín Martínez - 1800000
- Maximiliano García - 180026
- Edgar García - 180032
- Fernando Durán - 180079
- Allan González - 180083

Introducción:

Se mostrarán los resultados del desarrollo de un algoritmo para apuntar al centro de un ArUco mediante un láser, así como la metodología empleada.



Planteamiento del problema:

- ❖ La precisión es una cualidad muy importante al momento de realizar tareas en la cuales el rango de error es muy pequeño, un ejemplo de este tipo de tareas se puede encontrar en el área médica. Al momento de realizar una operación de retina por medio de rayo láser, en donde la falta de precisión puede afectar de manera negativa el resultado.

Objetivo.

Desarrollar un algoritmo de servo visión para un robot de dos grados de libertad.

Objetivos particulares:

- ❖ Obtener parámetros de la cámara.
- ❖ Construir estructura del robot manipulador.
- ❖ Integrar cámara en el robot.
- ❖ Integrar láser.
- ❖ Obtener marco de referencia del láser.
- ❖ Desarrollar algoritmo de lector serial, decodificación de datos e escritura en Arduino.
- ❖ Desarrollar algoritmo de control PI en Python
- ❖ Comunicar Python y Arduino.
- ❖ Validar algoritmo.

Marco teórico:

Controlador PI.

Los controladores automáticos comparan el valor real de la salida de una planta con la entrada de referencia, es decir el valor deseado, determinan el error o desviación y producen una señal de control que reducir 'a el error de control a cero o a un valor pequeño. una de las representaciones para el controlador PI en función del tiempo esta dada por la ecuación

$$u(t) = K_p e(t) + \frac{1}{t_i} \int_t^0 e(t) dt$$

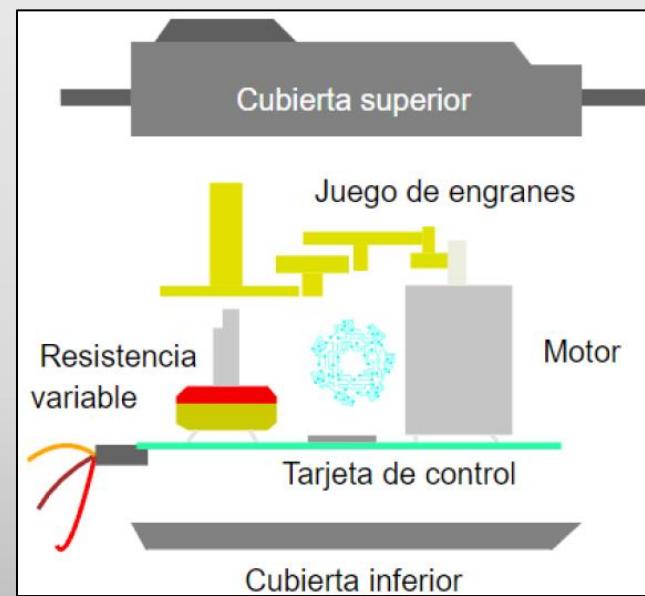
Cámara

Dispositivo encargado de tomar la imagen de seguimiento. Está diseñada para realizar el seguimiento de color y se puede utilizar para la localización de líneas de referencia, seguidor de líneas o el seguimiento de una luz en movimiento.

Servomotor.

Actuador rotativo o motor eléctrico, se define por una regulabilidad extrema y sus correspondientes posibilidades de control.

Permite el control preciso de posición de un ángulo, aceleración y velocidad. La estructura básica del servomotor incluye un sensor de posición del rotor.



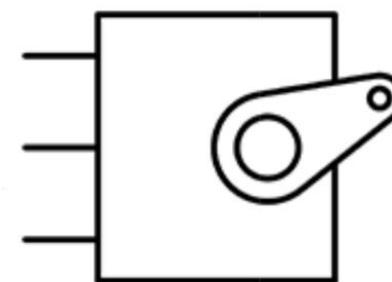
El servomotor cuenta con 3 terminales de conexión :

- VCC: Alimentación.
- GND: Tierra o negativo.
- Señal: Cable de señal de entrada PWM.

PWM=Orange (↑↑)

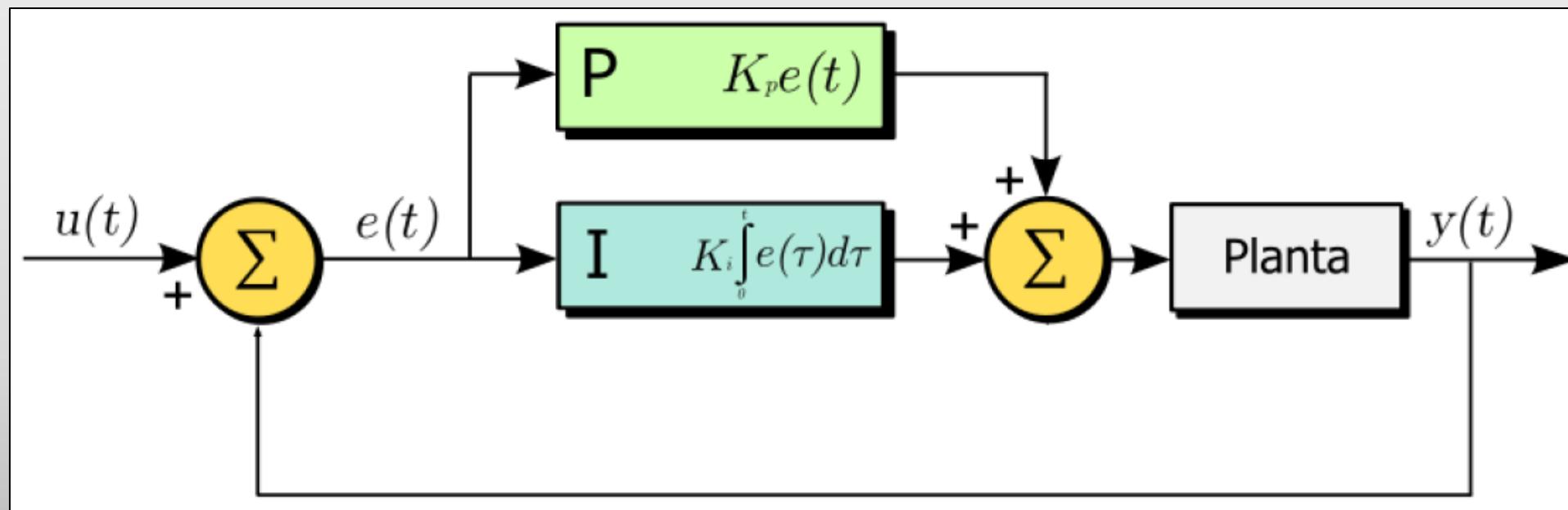
Vcc = Red (+)

Ground=Brown (-)



Sistema de control en lazo cerrado.

- ◆ Donde $u(t)$ es la variable de control y e es el error de control. El controlador PI. es la suma de tres términos donde tenemos que el término (P) que es proporcional al error, el término (I) que es proporcional a la integral del error. Los parámetros del controlador son ganancia proporcional K_p , tiempo integral representado por T_i .



Metodología.

Primero se tuvo que calibrar la cámara que se iba a utilizar, posteriormente se integro la cámara en el robot de 2 grados de libertad y también se le integro un laser. Una vez hecho esto, utilizamos un Arduino como driver para controlar los movimientos del robot y el estado del laser.

Establecemos una comunicación serial entre Python y el Arduino para poder enviar la señal de control que se calcula en Python y que el Arduino la reciba para así poder mover los servos y laser. Para calcular la señal de control usamos un control PI (Proporcional e Integral), obteniendo el error en relación de la distancia que se encuentre el centro del Aruco al centro de la imagen que captura la cámara.

Para obtener el error, debemos de calcular el error proporcional y el error integral. Para obtener el error proporcional, obtenemos la diferencia de pixeles entre el centro de la imagen y el centro del ArUco, con la siguiente ecuación (3.1).

$$E_p = E * k_p \quad (3.1)$$

Donde E es el error.

Una vez obtenido esto, calculamos el error integral, el cual es el área bajo la curva multiplicada por una ganancia integral. Para conseguir el área bajo la curva, lo hacemos con esta ecuación (3.2)

$$\text{AreaBC} = (\text{AreaBC} + E) h \quad (3.2)$$

Donde h es la base de tiempo de muestreo en segundos, Una vez obtenemos el error integral, lo multiplicamos por una ganancia integral (3.3)

$$Ei = \text{AreaBC} * ki \quad (3.3)$$

Después de calcular el error proporcional y el error integral, ya sacar la señal de control con la ecuación (3.4).

$$\text{Señal} = E_p + E_i \quad (3.4)$$

Diagrama de flujo de Python.

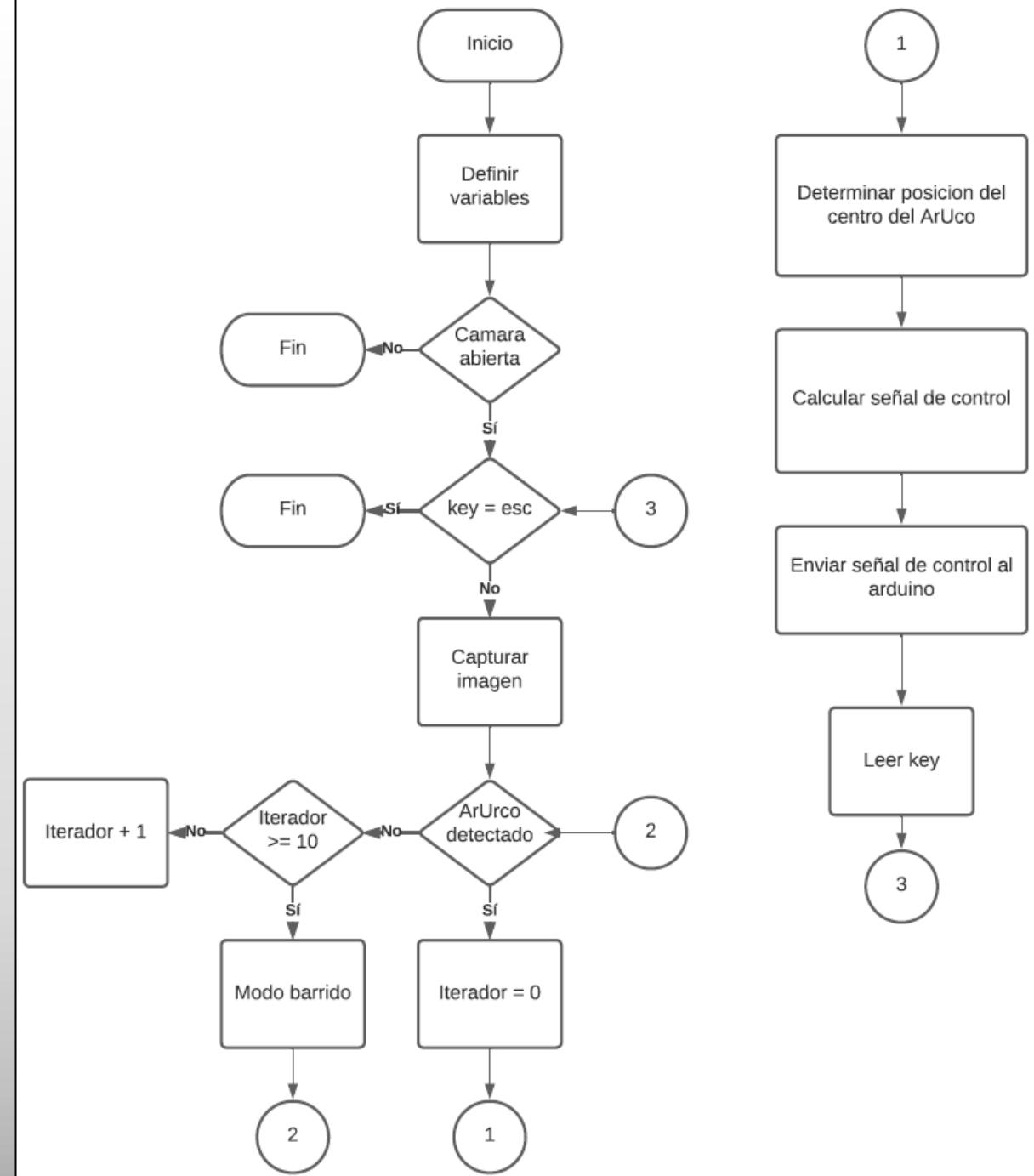
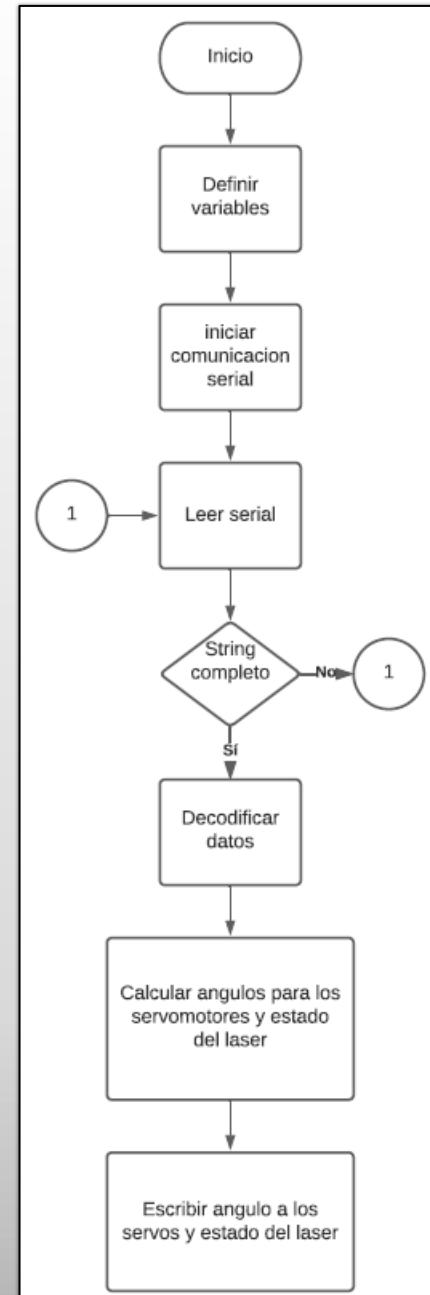
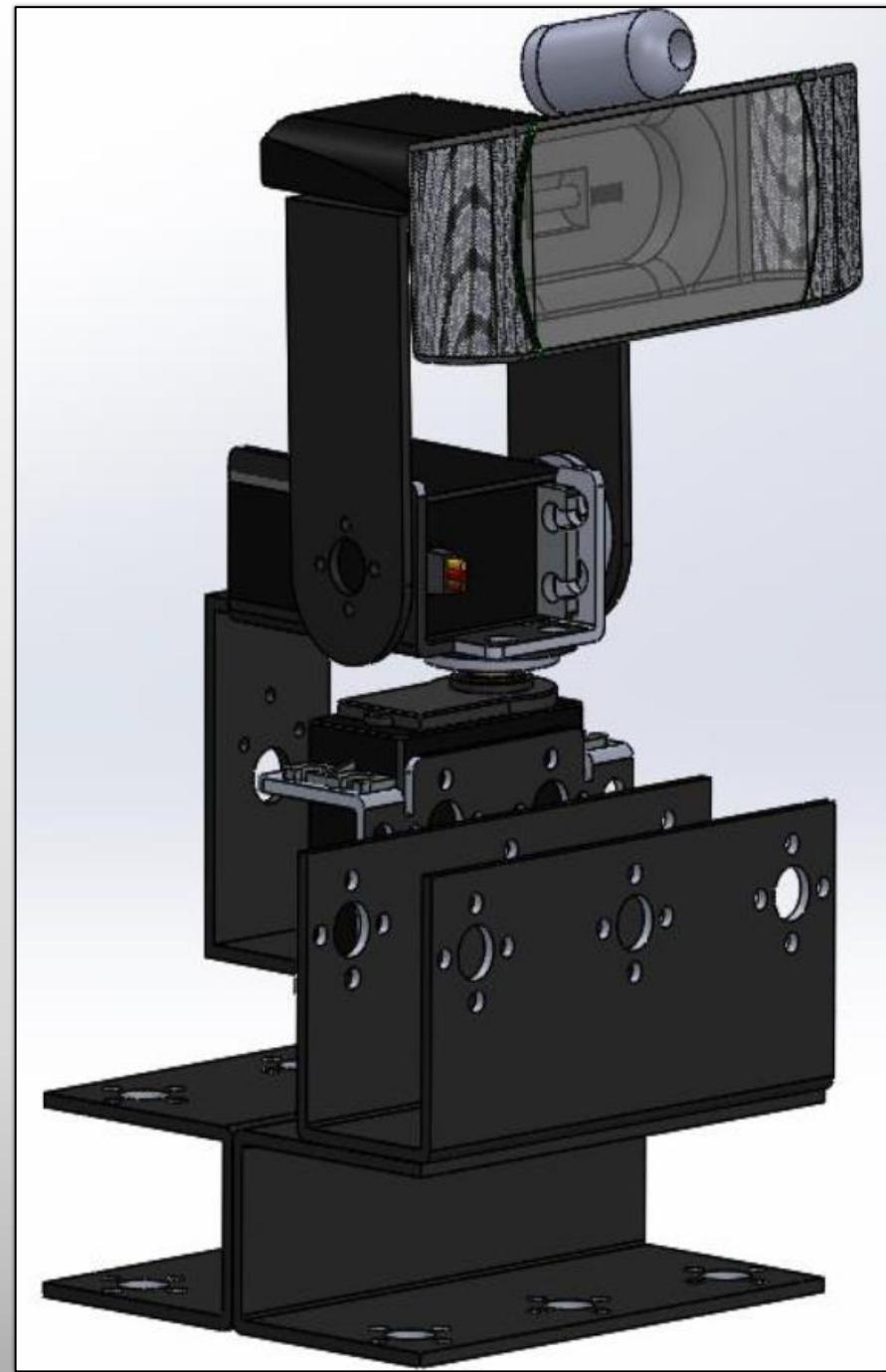


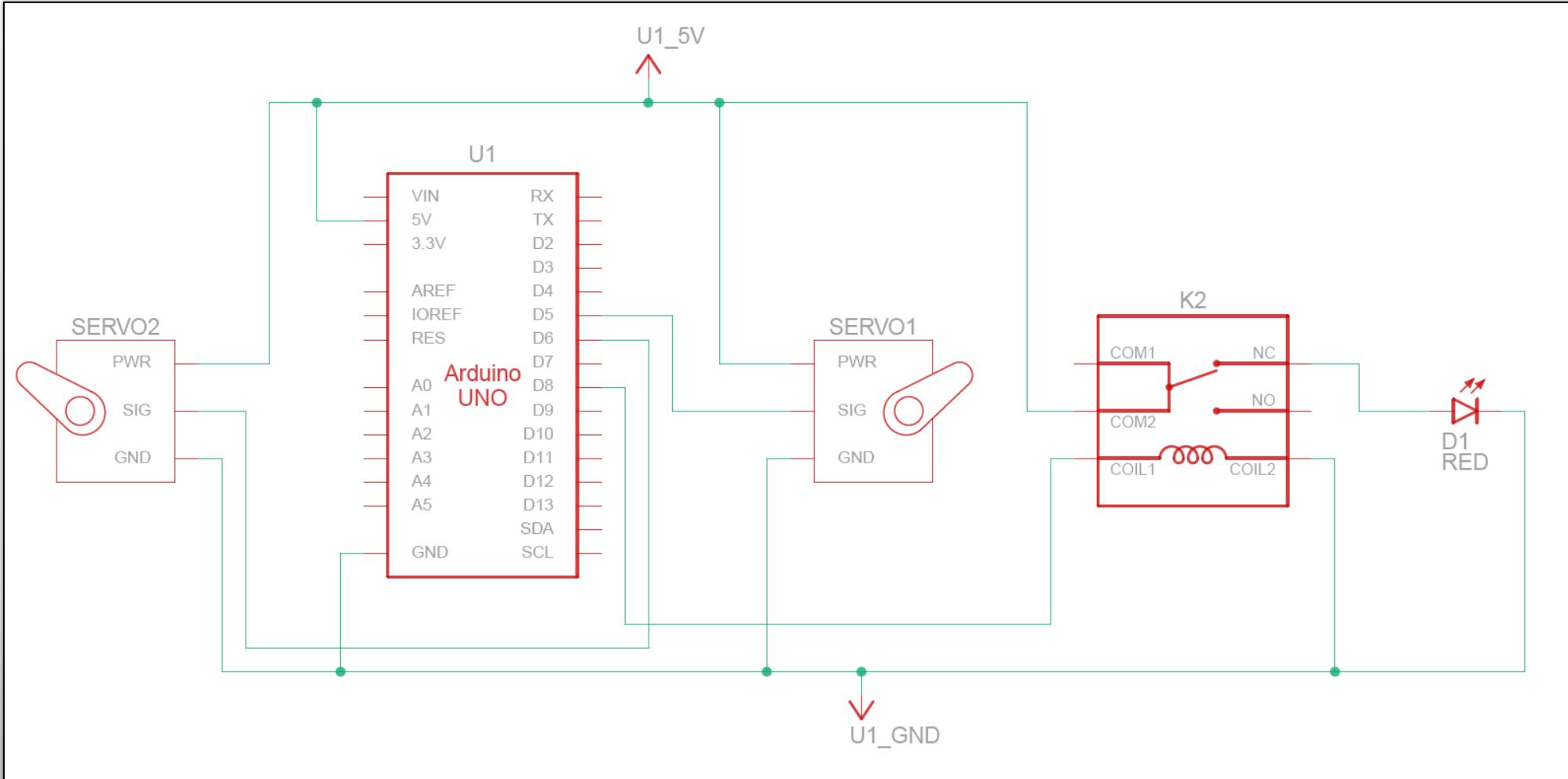
Diagrama de flujo de Arduino.



Modelo 3D.



Circuito Eléctrico.



Resultados.

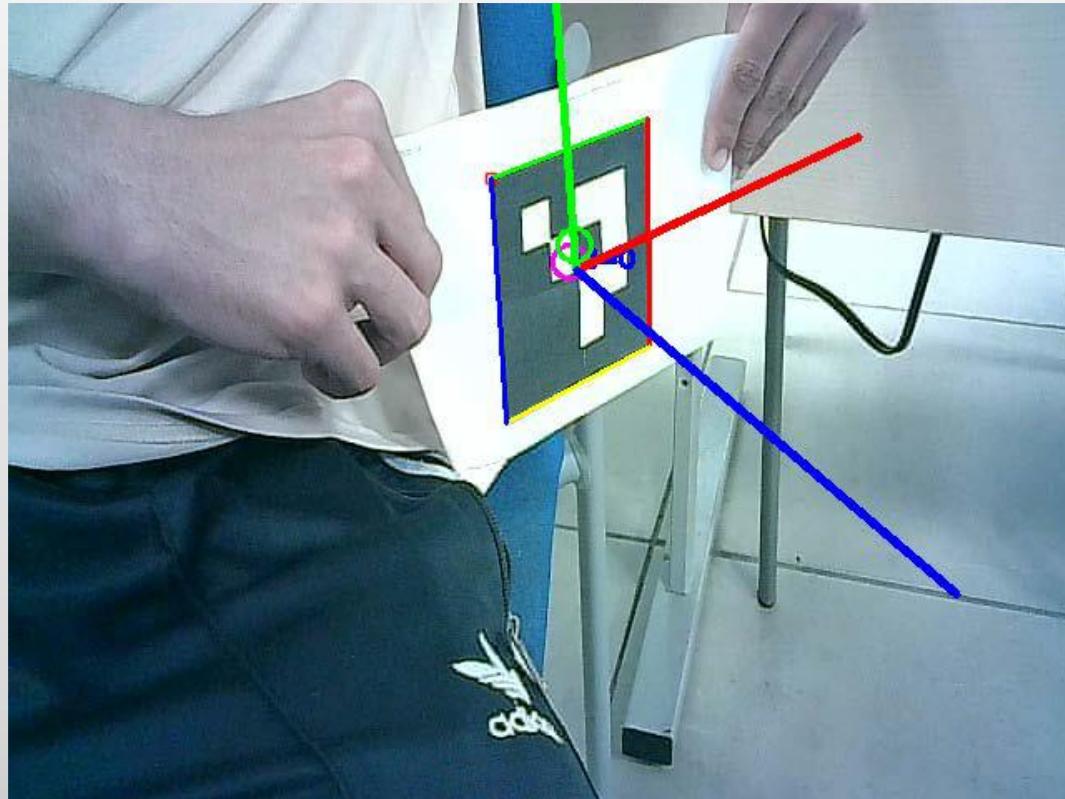
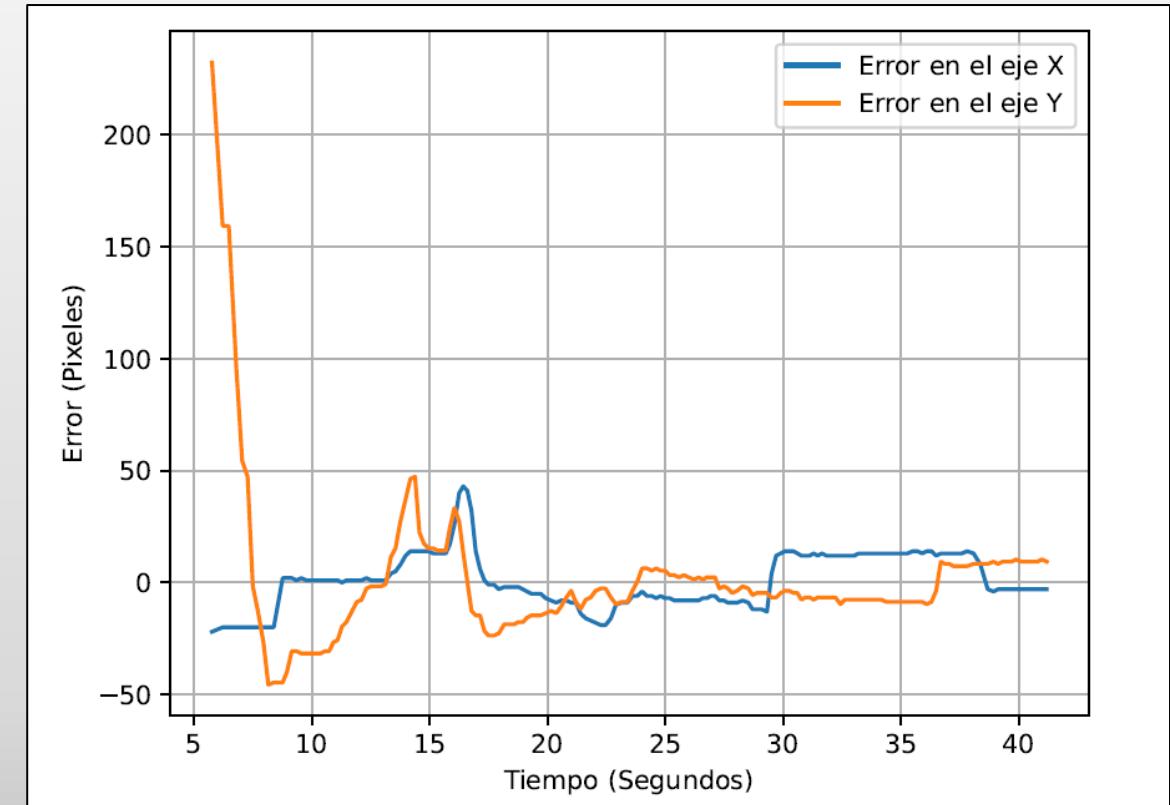


Imagen capturada y procesada por el sistema de servo-visión



Grafica que representa la corrección del error.

Conclusión:

El sistema cumple con la tarea que fue definida, pero la precisión con la que trabaja el sistema depende de muchos factores, como la iluminación, la calidad de la cámara, la posición en donde se encuentra el ArUco, la resolución de los servomotores, etc. Es por ello que un trabajo a futuro, se buscaría de trabajar en los factores que afectan al sistema para que ofrezca una mejor precisión y mejorar el sistema de control agregando el control derivativo.

Referencias:

- [1] "About-OpenCV". OpenCV (8 Oct 2021). [online], disponible:
<https://opencv.org/about/>
- [2] R. Jiménez, "Sistema de seguimiento de objetos usando OpenCv, ArUco y Filtro de Kalman extendido", tesis, Universidad de Sevilla, SE, España, 2018.
- [3] J. N. A. Rivero, «Robot Dirigido Por Voz,» Universidad de las Américas Puebla, Puebla, 2007.
- [4] E. A. Prieto y P. A. G. Quintero, «Diseño e implementación de un robot móvil autónomo con detección de color,» Universidad De San Buenaventura, Bogotá, 2006.
- [5] G. Viera-Maza, «Procesamiento de imágenes usando OpenCV aplicado en Raspberry PI para la clasificación del cacao,» Universidad De Piura, Piura, 2017.