

INFORME DE PRACTICA PROFESIONAL SUPERVISADA



Institución: Universidad Nacional de Lomas de Zamora – Facultad de Ingeniería

Alumno: Flores, Enzo

Tutor institucional/académico: Lukaszewicz, Cristian

Firma

INDICE

Lugar en donde he realizado la PPS	3
Detalles del trabajo realizado	4
Resumen	4
Objetivo	4
Modo de uso	4
Características generales	5
Software	6
Visión artificial	6
Reconocimiento de voz	7
Cinemática	8
Principal	10
Componentes electrónicos	10
Conclusiones	11

Reservado a la facultad para evaluación

Lugar en donde he realizado la PPS

Esta practica fue realizada en la Facultad de Lomas de Zamora perteneciente a la Universidad Nacional de Lomas de Zamora. Esta institución se encuentra ubicada en la localidad de Lomas de Zamora en la intersección entre la avenida Juan XXIII y Camino de cintura (Ruta 4).

Este trabajo se llevó a cabo en el marco del proyecto final de carrera de Ingeniería Mecánica con orientación a la Mecatrónica. Para el desarrollo de este trabajo tuve a disposición los elementos brindados; tanto por la catedra de Proyecto Final, como los propios recursos tecnológicos, laboratorios y docentes/profesionales de la facultad.

El desarrollo de este trabajo, sumado a la confección del repositorio en Github, donde se encuentra la documentación del trabajo, conforman la totalidad de las 200hs de practica las cuales realicé en el primer cuatrimestre del 2025.

Detalles del trabajo realizado

Resumen

La practica consistió en el desarrollo de un brazo robótico de cuatro grados de libertad (4GL) el cual, utilizando visión artificial y reconocimiento de voz, es capaz tomar piezas distinguidas por forma y color para colocarlas en los depósitos ordenados. Dado que su principal funcionalidad es manejar las funciones de pick and place a través de comandos de voz, el proyecto recibe el nombre de **VoiceBot**.

El software utilizado para el control de este brazo, con todas sus funciones, fue desarrollado en lenguaje Python utilizando diferentes librerías y herramientas tecnológicas que explicarán en mayor detalle en el transcurso de este informe.

Objetivo

El objetivo general propuesto es desarrollar el prototipo funcional de un brazo robótico capaz de recibir órdenes dinámicamente y poder detectar diferentes piezas con visión artificial para manipularlas sin necesidad de supervisión de un humano. Si bien estas alternativas se pueden encontrar en el mercado, la idea que incentiva este trabajo es poder realizar un desarrollo de bajo de coste que pueda escalarse para integrarse en pequeñas y medianas industrias que no tengan gran capacidad de implementación de grandes maquinarias robóticas. Para ello se hará uso de herramientas de vanguardia tecnológica como OpenCV, OpenAI y Python las cuales son de gran ayuda para un desarrollo fiable y de bajo coste.

Modo de uso

Para hacer una idea general al lector de las características que se van a desarrollar en este informe, considero de utilidad ilustrar cual seria su experiencia a la hora de utilizar VoiceBot.

Preparativos:

1. Colocar las piezas en el área de detección
2. Verificar que los depósitos se encuentren en posición
3. Verificar conexión a fuentes de energía, Webcam, Arduino/USB y micrófono activo.

Funcionamiento:

1. Ejecutar programa principal
2. Dictar una o varias órdenes en lenguaje natural indicando que pieza/s mover (considerando forma y color) y a que deposito llevarla/s.
3. Aguardar proceso de pick and place

4. Realizar nueva orden

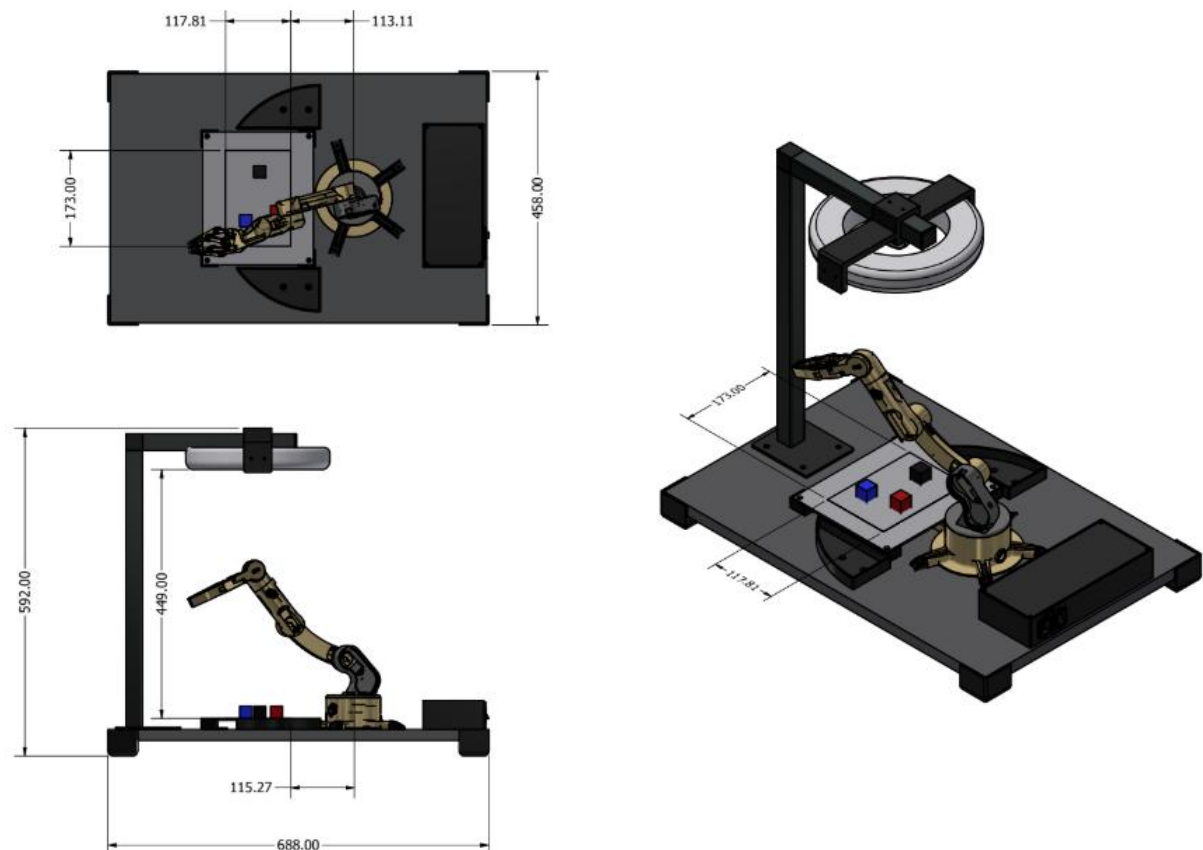
Una consideración importante en cuanto a la funcionalidad es que la orden dictada por voz **no** está limitada a frases estructuradas, sino que se puede indicar una frase extensa y el programa interpretará la orden, aunque no sea directa y concisa.

Características generales

El brazo robótico tiene una configuración de RRRR, cuenta con cuatro eslabones y cuatro ejes de giro que están controlados por servomotores a través de un Arduino UNO que actúa como procesador lógico que se alimentan con su respectiva fuente de potencia.

Los eslabones del brazo robótico fueron impresos en 3D con el material PLA. Las piezas, depósitos y apoyos también fueron fabricados con de la misma forma y con el mismo material.

Para completar la maqueta/prototipo; se encuentra un soporte metálico encargado de sostener la luz LED, necesaria para eliminar problemáticas relacionadas con la detección de imagen, y la Webcam encargada de tomar la imagen desde arriba. Todo esto se encuentra apoyado sobre una plataforma de madera de tipo melamina.

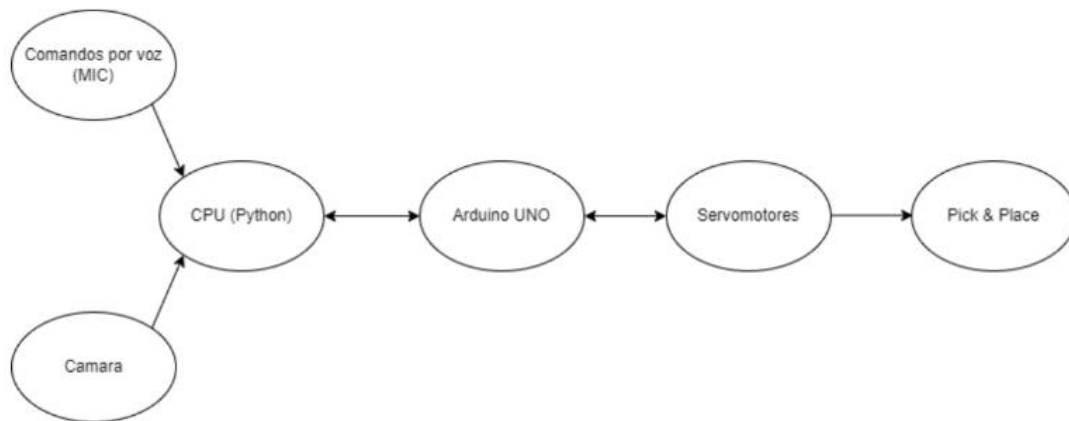


Esquema general del prototipo VoiceBot.

Software

Con excepción del control de los motores servos, que se realiza con un código de Arduino, el código del proyecto se encuentra realizado íntegramente con Python.

El diagrama de flujo utilizado para poder incorporar la interacción de las diferentes funciones del código es el siguiente:



La estrategia para el desarrollo del software se estableció con ayuda de la cátedra de Lenguajes de Programación de la Facultad de Ingeniería. El lineamiento principal fue resolver las diferentes partes en módulos según funcionalidad para luego integrarlas en un archivo principal que consulte a cada fuente por separado según necesidad.

En este informe se van a describir las características generales de código y cada uno de sus módulos. El código como tal, donde se podrán revisar todos los detalles técnicos, se encuentra en el repositorio de Github adjunto que complementa esta práctica.

Visión artificial

Para este código se utilizó la librería CV2 desarrollada por OpenCV. La delimitación del área de detección se realizó con marcadores aruco:



Estos marcadores son muy utilizados en el campo de la visión artificial dado que se pueden identificar fácilmente con su sistema de codificación y establecer los bordes para brindar un marco de referencia absoluto a la cámara.

En este modulo del software se realiza la identificación de cada una de las piezas situadas en el área de detección. Esta identificación se constituye en:

- **Forma:** el algoritmo utilizado identifica la forma del objeto según la cantidad de aristas observadas. Se brinda al código una serie de condicionales para las formas conocidas y se hace una verificación de cuál es la que se cumple.
- **Color:** esta basada en el modelo HSV que define un color según su matiz, saturación y brillo (valor). El código establece una serie de mascarar para determinar un rango predefinido para cada color identificable por nuestro software.
- **Coordenadas del centro de cada objeto:** dada la naturaleza regular de los objetos con los que se trabaja en el alcance de este proyecto, este se calcula de forma simple considerándolo inscripto en un rectángulo del cual se calcula su centro con $base/2$ para X y $altura/2$ para Y.

A nivel flujo, este módulo recibe como parámetro la imagen a analizar y debe devolver un vector con la información detectada. Es decir: forma, color y centro de cada objeto detectado.

Reconocimiento de voz

Las principales librerías utilizadas en este modulo son: Pyaudio y OpenAI.

Conceptualmente, esta sección del código realiza los siguientes pasos:

1. Recibe una orden por voz utilizando el micrófono. Esto será un audio en donde se encuentre la frase dictada por el usuario
2. La frase recibida será transcrita a texto haciendo uso del modelo "whisper-1" de OpenAI
3. El texto será procesado por un prompt predefinido que utilizará el modelo "gpt-3.5-turbo" y devuelve una orden concisa como cadena de caracteres con tres palabras: FORMA COLOR DEPOSITO. Por ejemplo: texto de entrada < "Quiero mover la pieza de color rojo y con forma cilíndrica al segundo deposito" > devolverá la instrucción < "CILINDRO ROJO 2" >

Esta sección del código, entonces, graba una orden mediante el micrófono y devuelve una cadena de caracteres de tres palabras FORMA COLOR DEPOSITO. El código principal utilizará esta cadena para conocer las

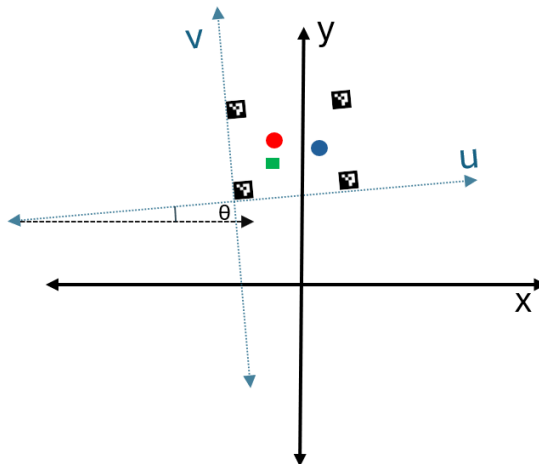
características del objeto que debe buscar dentro del vector de objetos que detecto la cámara en su reconocimiento.

Cinemática

Cuando el código encuentra la coincidencia entre el comando de voz y la detección por visión, este módulo se encarga de dirigir el efector del robot para tomar el objeto y llevarlo al depósito correspondiente.

Uno de los parámetros que recibe este código como entrada, es el centro del objeto que debe ir a tomar. El código realiza la conversión de ese centro, que esta referenciado a un vértice de la zona de detección, para transformarlo a una posición referenciada al origen del robot.

Esto se puede interpretar en el siguiente esquema donde el eje coordenado UV es el correspondiente a los objetos y el XY al robot, además de cómo se relacionan matemáticamente entre sí.



Donde:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

Obtenidas XY del robot, se ingresan como parámetro a las ecuaciones que resuelven el sistema cinemático y nos brindan las coordenadas articulares para girar los servomotores a sus respectivos ángulos.

La posición Z será definida por diseño dado que los objetos tienen una altura similar entre sí.

El método utilizado para la resolución de la cinemática del robot es el algoritmo de Denavit-Hartenberg.

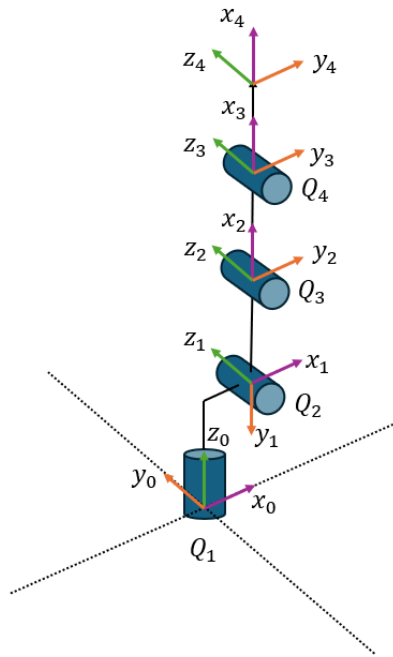


Tabla de parámetros Denavit-Hartenberg

GL	θ	d	a	α
1	q_1	L_{1z}	L_{1x}	$\frac{\pi}{2}$
2	$q_2 + \frac{\pi}{2}$	0	L_2	0
3	q_3	0	L_3	0
4	q_4	0	L_4	0

Las ecuaciones que determinan la posición en función de las coordenadas articulares resultan:

$$\begin{cases} Px = (-12000 * \sin(Q2) - 12286 * \sin(Q2 + Q3) - 13000 * \sin(Q2 + Q3 + Q4) + 835) * \frac{\cos(Q1)}{100} \\ Py = (-12000 * \sin(Q2) - 12286 * \sin(Q2 + Q3) - 13000 * \sin(Q2 + Q3 + Q4) + 835) * \frac{\sin(Q1)}{100} \\ Pz = 120 * \cos(Q2) + 6143 * \frac{\cos(Q2 + Q3)}{50} + 130 * \cos(Q2 + Q3 + Q4) + 603/10 \end{cases}$$

Aquí se agrega una cuarta ecuación para determinar la orientación del eje al aproximarse al objeto y completar el sistema 4x4.

$$\alpha = Q4 - Q2 + Q3$$

Donde α es el ángulo que conforma la orientación del último eslabón respecto al plano XY en la posición Z deseada. Este ángulo se determina por diseño en 30°.

Por último, esta sección del código traduce los valores de las coordenadas articulares a valores de giro para los servomotores. Los mismos se envían desde el script Python utilizando la librería Pyserial al Arduino conectado mediante USB.

Principal

El script principal main.py se encarga de orquestar cada entrada y salida que reciben los diferentes módulos que se describieron en este informe.

El diagrama de flujo del programa, entonces, queda de la siguiente forma:

1. Módulo deteccion.py recibe como parámetro la imagen a analizar y devuelve un vector que contiene la información de cada objeto detectado.
2. Módulo audio.py, se activa desde main.py y devuelve una cadena de tres palabras: FORMA COLOR DEPOSITO.
3. El código main.py realiza una búsqueda para cruzar la información de la orden con el vector de objetos detectados y devuelve una posición XY que es el centro del objeto que se debe tomar.
4. El modulo cinemática.py recibe esta coordenada XY y el número de depósito. Realiza las ecuaciones para el calculo de las variables Q1, Q2, Q3 y Q4 que llevan el efector a destino para tomar el objeto y depositarlo en el recipiente correcto.
5. Reinicio del bucle.

Componentes electrónicos

- Arduino UNO R3
- 4 servomotores MG995R
- 1 servomotores SG95
- Fuente de alimentación 5V 10A
- Capacitor electrolítico 3300 μ F 16V

Conclusiones

Dada la irrupción de la inteligencia artificial y los algoritmos de aprendizaje automático, considero que en los próximos años será fundamental que los ingenieros cultiven la capacidad de investigar y mantenerse actualizados tecnológicamente de forma continua. Esta práctica y sus funcionalidades fueron elegidas con la intención de aplicar conocimientos, pero también de adquirirlos.

Durante el desarrollo de esta actividad he utilizado herramientas matemáticas electrónicas, de robótica, de lenguajes de programación, entre otras, que son producto de mi aprendizaje en estos años de carrera. Sin embargo, más allá de estos elementos, considero que los métodos de trabajo y la manera de afrontar los distintos desafíos que surgieron son fruto de una matriz lógica adquirida en la carrera, que trasciende las cátedras o los conocimientos técnicos particulares.

La integración de tecnologías novedosas como la visión artificial y los modelos de inteligencia artificial aplicados al lenguaje resultó fundamental para aportar un componente de vanguardia a la práctica. Disfruté especialmente trabajar con estas herramientas. Considero un gran desafío pensar cómo integrar este tipo de tecnologías al programa curricular de la carrera de ingeniería, dada su creciente relevancia en el desarrollo tecnológico futuro.

Además del apoyo de otros docentes, la ayuda de mi tutor resultó clave para organizar las ideas, llevar a cabo la práctica y condensarlas en este informe.