

Cosechas autónomas de fresas: Reporte técnico

Sistemas ciber-físicos, Equipo YEA

07 de diciembre del 2025

Emiliano González González
Ingeniería en Mecatrónica
Tecnológico de Monterrey
Querétaro, MX
A01705488@tec.mx

Yael Cortés Rubio
Ingeniería en Tecnologías
Computacionales
Tecnológico de Monterrey
Querétaro, MX
A01275893@tec.mx

Aldrick Tadeo Arellano
Ingeniería en Robótica y Sistemas
Digitales
Tecnológico de Monterrey
Querétaro, MX
A01710105@tec.mx

Abstract—Este reporte describe el diseño e integración de un sistema ciber-físico para la cosecha autónoma de fresas en un ambiente de cultivo controlado. El objetivo fue implementar, en 16 semanas, una celda funcional capaz de identificar y recolectar fresas maduras con una efectividad superior al 75%. La metodología integra cuatro subsistemas: (1) visión computacional mediante una cámara RealSense y un modelo YOLO11-S entrenado con un conjunto amplio de imágenes públicas y técnicas de aumento de datos para clasificar frutos maduros y no maduros, complementado con estimación de pose y localización espacial mediante un enfoque PnP para obtener la posición del fruto en coordenadas útiles para el robot; (2) planeación y ejecución de movimiento de un robot colaborativo UR3e apoyada por simulaciones en RoboDK para validar cinemática y evitar colisiones; (3) un gripper suave neumático diseñado para manipulación delicada, respaldado por análisis y consideraciones de manufactura de actuadores blandos; y (4) un sistema IoT de monitoreo del cultivo basado en ESP32 y Raspberry Pi para registrar temperatura, humedad ambiental, humedad del sustrato y nivel de luz, con visualización en dashboard y alertas. Los resultados muestran una integración funcional de monitoreo, percepción y manipulación, cumpliendo la meta de desempeño planteada. En conjunto, el trabajo demuestra la viabilidad de una solución de bajo costo y alta adaptabilidad, y establece líneas claras de mejora para aumentar la robustez en escenarios agrícolas reales.

Keywords—Cosecha autónoma; fresa; robótica colaborativa; visión por computadora; YOLO; estimación de pose; PnP; cámara RealSense; soft robotics; gripper neumático; IoT agrícola; RoboDK; agricultura de precisión.

I. INTRODUCCIÓN

La producción de fresa es una actividad intensiva en mano de obra que, en el esquema actual de cultivo y cosecha, depende la mayoría de veces de recolección manual. Esta labor requiere trabajadores hábiles para seleccionar y manipular los frutos delicados sin dañarlos; además, las demandas de fuerza de trabajo se concentran en ventanas cortas de maduración, lo cual conlleva a problemas de disponibilidad, variabilidad en la calidad de la cosecha y aumento de costos operativos. La escasez de la mano de obra agrícola, ha motivado la búsqueda de soluciones tecnológicas que reduzcan la dependencia del trabajo manual y mejoren la continuidad del suministro. (Estabrook, 2020).

La fresa es un fruto complicado de automatizar por el tamaño, su forma irregular y la facilidad con la que se daña, cualquier sistema automatizado debe combinar detección confiable de frutos, localización tridimensional precisa, un manipulador con movimiento suficientemente rápido y repetible, y una herramienta de agarre que ofrezca suavidad

y control de fuerza. Aunque la investigación en robots cosechadores, visión artificial, manipuladores y grippers blandos ha avanzado bastante, aunque todavía existen retos en velocidad, robustez en entorno real y hacerlo todo a un costo que sea viable para usar comercialmente.

La problemática a resolver presentada por nuestros socios formadores (Tecnológico de Monterrey, CAETEC) se describe como la búsqueda de una solución de cosecha automatizada de fresas. Entre los sistemas que consisten esta solución se encuentra, un sistema de visión responsable de detectar las fresas en el espacio de cultivo y determinar la posición de la misma en el espacio. En este trabajo se describen las preliminares teóricas que fundamentan las herramientas a utilizar.

Movimiento del brazo robótico a la posición determinada por el sistema de visión, para llegar a esta meta se está trabajando progresivamente con un ambiente simulado para ir estableciendo los fundamentos conceptuales.

Gripper con tecnología de *soft robotics*, incluyendo el diseño de este desde los actuadores suaves a todo su sistema de control. Se propone en este reporte un diseño inicial.

Monitoreo del cultivo a través de un sistema IoT, incluyendo la captura de datos, transmisión, almacenamiento y visualización.

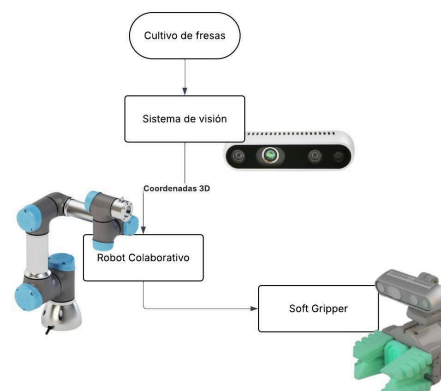


Figura 1. Diagrama de conexión

El sistema propuesto para la recolección automatizada de fresas se compone de tres elementos principales: el sistema de visión encargado de la percepción del entorno, el robot colaborativo responsable del posicionamiento y

movimiento, y el soft gripper diseñado para realizar la toma del fruto mediante un mecanismo de actuación neumática. Estos subsistemas trabajan de manera integrada para permitir una recolección precisa, delicada y segura en entornos reales de cultivo.

En el sistema de visión, que básicamente es la vista automática de todo el proyecto. Con una cámara RealSense, este módulo captura imágenes y usa algoritmos de visión basados en redes neuronales, para detectar qué fresas se encuentran maduras. A partir de esto, el sistema calcula la posición 3D del fruto usando la información de profundidad y determina el punto de agarre adecuado. El sistema también genera la orientación del objetivo y un nivel de confianza en la detección, enviando esta información como un conjunto de coordenadas estructuradas hacia el controlador del robot. Esta etapa es esencial, ya que define la precisión con la que el robot podrá aproximarse al fruto sin colisionar con algún elemento del entorno.

Cuando ya se cuenta con las coordenadas tridimensionales, entra el robot colaborativo, en este caso el UR3. Este robot recibe la información generada por el sistema de visión y la utiliza para planear su movimiento hacia el fruto. Los algoritmos que se emplean para la planeación de la trayectoria, serán validados en un entorno simulado por RoboDK con el fin de verificar colisiones, resolver la cinemática del brazo y confirmar que la fresa es accesible desde la orientación requerida.

Con estos datos, el UR3 calcula la trayectoria óptima para posicionarse en el punto objetivo, tomando en cuenta, sus propias limitaciones mecánicas, el espacio disponible entre las plantas y la orientación necesaria para que el agarre funcione correctamente. En conjunto, esta integración entre visión y movimiento del robot es lo que permite transformar una detección en un movimiento preciso y confiable.

El gripper trabaja en conjunto con los demás sistemas (visión y movimiento de robot), su funcionamiento depende de un sistema de actuación neumática que será integrado por microtubos, electroválvula, bomba neumática, sensor de presión, entre otros componentes, para administrar la presión necesaria.

Cuando el robot alcanza la posición indicada por el sistema de visión, envía una señal mediante MQTT a la ESP32 para activar el mecanismo de neumática y con ello los dedos del gripper comienzan a inflarse de forma controlada.

Una vez el robot completa la extracción del fruto, el sistema desactiva el sistema de neumática. Este ciclo se repite hasta que el sistema de visión no detecta más fresas.

II. ADMINISTRACIÓN DEL PROYECTO

A. Project Charter

El project charter es un documento que autoriza el inicio del proyecto, y establece el propósito, los objetivos, el alcance, y los elementos fundamentales de su ejecución. Este documento sirve como referencia para todas las decisiones de gestión y define las responsabilidades principales del equipo y de los patrocinadores.

1) Propósito del proyecto:

Impulsar el desarrollo de un sistema automatizado de recolección de fresas en un ambiente controlado, combinando tecnologías de visión computacional, robótica

colaborativa, actuadores blandos e IoT para investigación aplicada en agricultura de precisión.

2) Objetivo SMART:

En un periodo de 16 semanas, implementar un sistema funcional capaz de identificar y recolectar fresas maduras con una efectividad superior al 75%, asegurando la integración completa de la celda robótica, el modelo de visión y del sistema IoT de cultivo.

3) Alcance del proyecto:

El alcance del proyecto incluye:

- a) Implementación de un invernadero experimental.
- b) Diseño y manufactura de soft gripper para manipulación delicada.
- c) Integración de robot, cámara, algoritmo IA y sistema de control.
- d) Entrenamiento del modelo de visión computacional.
- e) Sistema IoT de monitoreo del ambiente de cultivo.

Excluye:

- Operación en ambientes no controlados o exteriores
- Adaptación del sistema a múltiples tipos de fruta.

4) Entregables Principales:

- a) Modelo de visión computacional entrenado.
- b) Simulaciones y control del brazo robótico.
- c) Soft Gripper funcional.
- d) Sistema IoT operativo.
- e) Código fuente (visión, control, decisiones).
- f) Propuesta inicial y reporte final.

5) Beneficios y justificación:

El proyecto permitirá evaluar la viabilidad de sistemas de cosecha autónoma, reduciendo tiempos operativos, minimizando daño al fruto y generando evidencia para futuras aplicaciones agrícolas avanzadas.

B. Roles y responsabilidades

Project Sponsor: CAETEC

- Autoriza y valida el inicio del proyecto
- Proporciona dirección estratégica y define el propósito general.
- Aprueba entregables clave y cambios mayores de alcance

Project Lead: Emiliano González González

- Supervisa el avance general del proyecto.
- Facilita la colaboración entre los miembros del equipo.
- Fomenta la mejora continua del proceso
- Da seguimiento al cumplimiento del plan, recursos y fechas.

Project Team - YEA:

- Emiliano González González
- Aldrick Victor Tadeo Arellano
- Yael Cortés Rubio

- Desarrollar los entregables técnicos.
- Planificar actividades y administrar tiempos.
- Asegurar calidad de integración, software y pruebas.
- Documentar avances y riesgos

Responsabilidades:

1. **Project Sponsor:** Encargada de aprobar el proyecto, define el propósito del proyecto.
2. **Project Lead:** Ayuda a promover la mejora continua del proceso y la colaboración.
3. **Project Team:** Desarrollar el proyecto, organización, responsabilidad, asegurar calidad.
4. **Stakeholders:** Apoyan o forman parte del proyecto.

```

graph TD
    A[1 Invernadero] --> B[2 Soft gripper]
    B --> C[3 Visión y data set]
    C --> D[4 Robot]
    D --> E[5 Cierre]
  
```

1 Invernadero

- Diseño invernadero
- Diseño del sistema de monitoreo IoT
- Obtención de componentes para invernadero
- Construcción y programación de invernadero

2 Soft gripper

- Diseño de soft gripper
- Pruebas en software
- Manufactura de gripper
- Pruebas físicas
- Implementación de gripper

3 Visión y data set

- Determinar modelo de visión y cámaras a utilizar
- Investigación de cámara RAGB
- Aprendizaje de uso de cámara
- Entrenamiento de modelo de visión
- Pruebas de resultados de registro de datos y visión computacional
- Mostrar datos del dataset en Dashboard

4 Robot

- Determinar cuáles a recibir desde el sistema de visión
- Definir rutinas de acercamiento del robot hacia las fresas
- Pruebas de rutina de acercamiento
- Pruebas de corte con gripper implementado

5 Cierre

- Entregables
- Presentación
- Videos

La Figura 2 muestra el WBS realizado por el equipo YEA, su elaboración fue fundamental para la planificación del proyecto, ya que nos permitió desglosar el trabajo total en tareas más pequeñas, manejables y claramente definidas. Este proceso facilitó identificar todas las actividades necesarias, establecer responsabilidades y comprender la magnitud real del proyecto desde etapas tempranas. Gracias al WBS fue posible visualizar el alcance completo y evitar omisiones que pudieran generar retrasos y reprocesos.

Así mismo ayudó a mejorar la comunicación y organización dentro del equipo, ya que cada integrante pudo entender las tareas que forman parte del proyecto y cómo es que se realizan.

[illegible]

En la Figura 3 se muestra el diagrama de Gantt del equipo YEA, nos ayudó en la organización y determinar la complejidad de cada tarea, sin embargo al no tener una base sólida del WBS, el diagrama de Gantt fue muy complicado distribuir correctamente las tareas y determinar el tiempo de cada una, algunas nos llevaron más tiempo que otras, debido, tuvimos que reducir el tiempo de otras tareas lo cual complicó una parte del proyecto.

Se desarrolló un sistema de monitoreo para observar el crecimiento y las condiciones de las plantas de fresa. Este sistema permite registrar y visualizar los datos de temperatura, humedad ambiental, humedad del sustrato y porcentaje de luz, con el fin de mantener un historial durante todo el proyecto. Además, los valores se muestran en un dashboard para facilitar el seguimiento en tiempo real del estado del cultivo.

The diagram illustrates a Raspberry Pi acting as a central hub for an IoT system. It is connected to an MQTT Broker, which manages the communication between the Pi and various sensors and actuators. The connected components include:

- Sensors:** Temperature sensor, Humidity sensor, Air quality sensor, and Motion sensor.
- Actuators/Outputs:** Webcam, Telegram bot, and HTML5 node.js.

La Figura 4 muestra la arquitectura general del sistema de monitoreo, el cual se divide en tres partes principales:

1. **ESP32 y sensores:** La ESP32 se encarga de obtener los datos de los sensores: un DHT22 para medir la temperatura y humedad ambiental, un LDR para el nivel de luz y tres sensores capacitivos para la humedad del sustrato. Los valores se convierten en porcentajes, y en el caso de la humedad del sustrato se saca un promedio de los 3 sensores. Una vez calculados, los datos se envían por medio de MQTT a sus respectivos tópicos para que la Raspberry PI 5 los reciba.
2. **Raspberry PI 5:** Aquí se encuentra la base de datos, donde almacena los valores recibidos desde la ESP32, las imágenes capturadas por una cámara USB conectada a la misma Raspberry y las alertas generadas en caso de detectar algún valor fuera de rango. Además, la Raspberry ejecuta 3 códigos principales:
 - a) **MQTT.py:** Recibe los datos de los sensores a través de los tópicos MQTT y los guarda en la base de datos. Envía una solicitud cada 10 minutos exactos (por ejemplo si la primera solicitud se envió a las 13:10:00 y se recibieron los datos, la siguiente será a las 13:20:00) así mismo, el código ayuda a que los tiempos sean exactos es decir que si se ejecuta a las 13:23:03 calcula el tiempo para que la solicitud la mande a los 10 minutos exactos, es decir, 13:30:00 y así sucesivamente. En caso de no recibir respuesta después de 30 segundos de haber mandado la solicitud, manda una alerta indicando que existe un error al recuperar datos.
 - b) **Camera.py:** Controla la cámara USB para capturar imágenes de las plantas de fresa y guardarlas en la base de datos en formato binario. Las fotos se toman cada hora exacta (por ejemplo, si inicia el programa a las 13:23:00, la siguiente la será a las 14:00:00), al igual que el código MQTT.py calcula el tiempo pero esta vez para la siguiente hora.
 - c) **Telegram.py:** Revisa la base de datos para leer el último valor guardado y, si detecta datos fuera de los rangos normales, envía un mensaje de alerta por Telegram. Además, cada 10 minutos manda un resumen con los datos actuales, y cada 6 horas revisa si hay nuevas alertas que notificar. Esto ayuda a mantener un control constante y evitar pérdida de información.
3. **Dashboard y Telegram:** Ambas herramientas permiten visualizar datos. El dashboard muestra los valores históricos, las alertas, las imágenes capturadas, mientras que Telegram se usa principalmente para recibir alertas rápidas y datos en tiempo real.

2. Diseño de interfaz

Vista principal

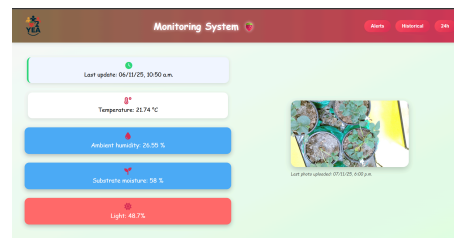


Figura 5. Vista principal dashboard

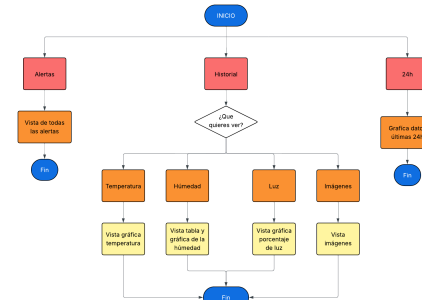


Figura 6. Diagrama de flujo general “Dashboard”

La Figura 2 muestra la vista principal del dashboard, en la cual se muestran los últimos datos guardados de los sensores de la base de datos, además de mostrar la última foto capturada por la cámara, tanto los datos como la imagen muestran la fecha y hora. En el header se encuentra los botones para acceder a las otras vistas “Alerts” “Historical” y “24h”

Vista “Alerts”

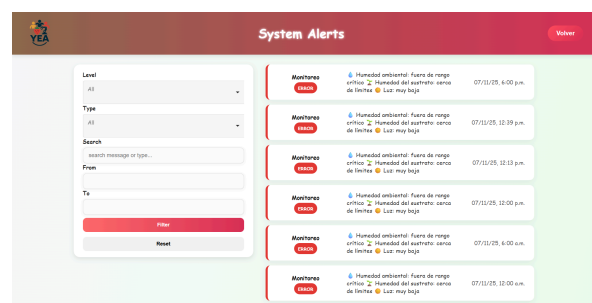


Figura 7. Vista “Alerts”

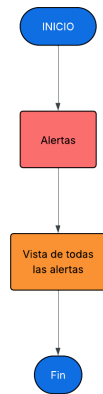


Figura 8. Diagrama de flujo Alerts

Al entrar a la vista de alertas mostrará el historial de todas las alertas que se han guardado en la base de datos, además puedes filtrar los, por nivel, tipo, mensaje o palabra, por último por fecha.

Vista “24h”

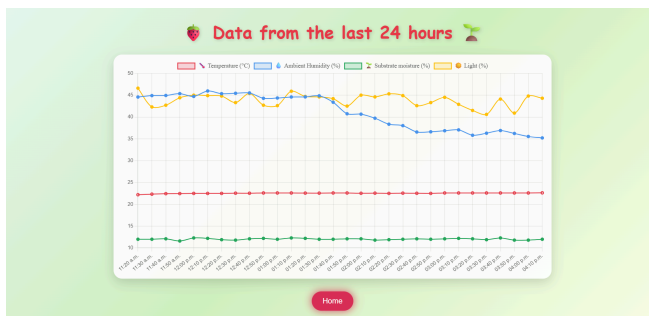


Figura 9. Vista “24h”

La figura 9 muestra la gráfica de los datos de temperatura, humedad ambiental, humedad del sustrato y porcentaje de luz de las últimas 24 horas.

Vista “Menu Historical”

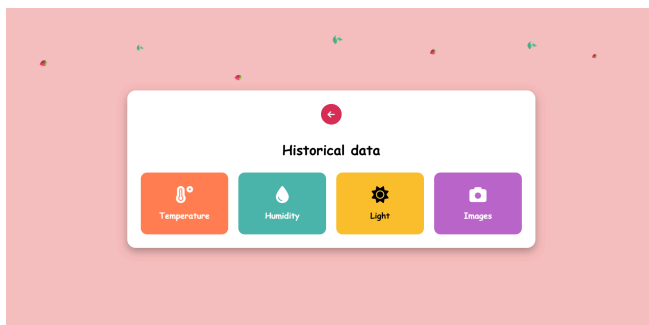


Figura 10. Vista “Historical”

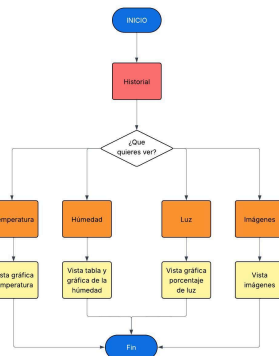


Figura 11. Diagrama de flujo

La Figura 10 muestra la vista de las opciones a ver del registro de datos, el usuario podrá elegir qué es lo que desea ver, puede ver todos los datos que se han registrado de cada sensor.

Vista “Temperature”

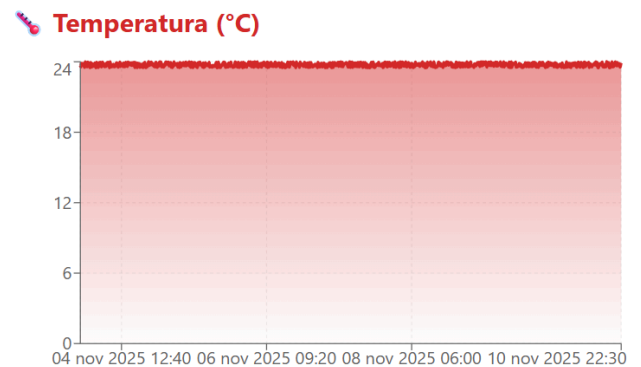


Figura 12. Vista “Temperature”

Muestra la gráfica de los datos de temperatura, filtrando un rango de fechas, así como se muestra en la Figura 12, sin embargo en caso de colocar mal las fechas o no existan datos en ese rango de fechas saldrá un mensaje “No existen datos en ese rango de fechas”

Vista “Humedad”

Date	Ambient humidity (%)	Substrate moisture (%)
24/09/25, 11:22 a.m.	32	32
25/09/25, 12:42 p.m.	25	25
25/09/25, 5:09 p.m.	32	33
26/09/25, 5:07 p.m.	65	78
27/09/25, 6:11 p.m.	23	54
29/09/25, 11:49 a.m.	43	67
29/09/25, 11:49 a.m.	13	47
29/09/25, 11:53 a.m.	39	56

Figura 13. Tabla que contiene

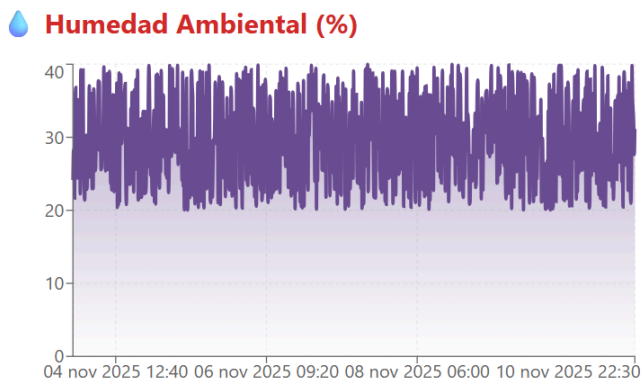


Figura 14. Gráfica “Humedad Ambiental (%)”

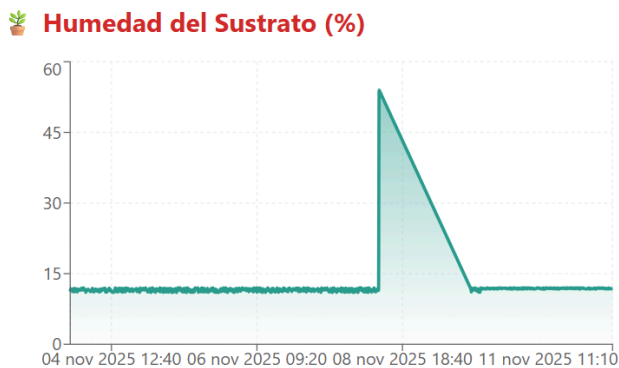


Figura 15. Gráfica “Humedad del sustrato (%)”

En la Figura 14 muestra una tabla de las humedades registradas, tanto del sensor DHT22 y los sensores capacitivos, al igual que la vista de la temperatura, se filtran colocando rangos de fechas, en caso de no colocar fechas correctamente “No existen datos en ese rango de fechas”. Así mismo se observan las gráficas de datos.

Vista “Luz”

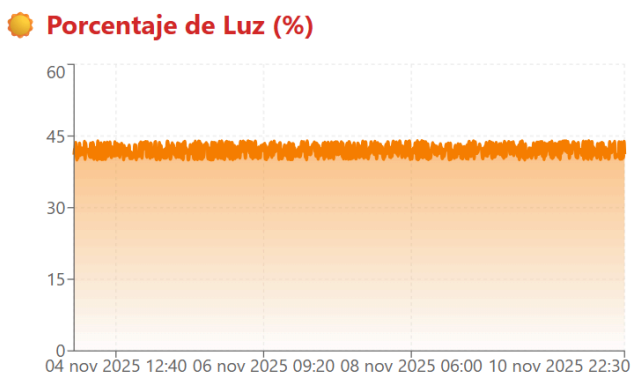


Figura 16. Vista “Percentage of light”

Al igual que la vista de temperatura mostrará una gráfica con los datos guardados por el sensor LDR, así como se muestra en la Figura 16.

Vista “Images”

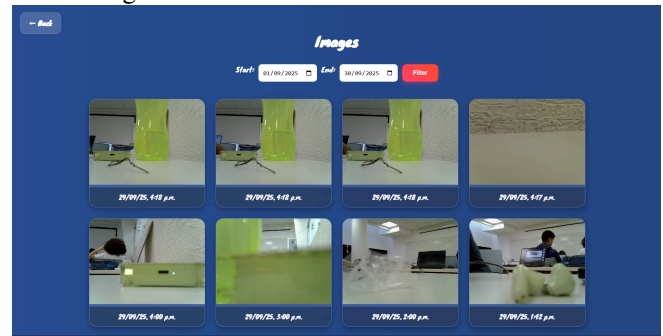


Figura 17. Vista “Images”

Al colocar un rango de fechas mostrará las últimas 10 imágenes en ese rango de fechas, tal cual se muestra en la Figura 17.

3. Problemas presentados

Dentro de los problemas que se nos presentaron durante esta parte del proyecto, fue la calibración de sensores, debido a que en ciertas ocasiones a pesar de estar desconectados los sensores de la ESP32 mandaba datos de igual forma, otro problema presentado fue la falla de la Raspberry, se desconoce el motivo de la falla, sin embargo señalaba problemas para encender, hasta que dejó de funcionar, y con ello la pérdida de datos de los sensores de la ESP32, además de las fotos tomadas por la cámara, ya que toda esa información se almacenó dentro de la Raspberry, fue el problema más significativo dentro de esta parte del proyecto.

B. Clasificación de las fresas

Se utilizó una red neuronal convolucional para la tarea de detección, clasificación y ubicación en la imagen de las fresas. Elegimos la red YOLO11 (You Only Look Once v11) desarrollada por Ultralytics. En búsqueda de un balance de desempeño y velocidad, se utilizó la escala *S* que contiene 9.4 millones de parámetros, adicional a esto, se utilizó la arquitectura base sin pre-entrenamiento [3].

El *dataset* utilizado para entrenar nuestro modelo para la tarea mencionada consistió en una recopilación de múltiples *datasets* públicos de libre acceso, sumando casi 18,000 imágenes en una variedad de escenarios, con un total de 62,213 anotaciones divididas entre las clases. Para preparar estas imágenes se estandarizó la resolución a 640 x 640 píxeles, con un procesamiento de *Fit to frame* para los que no entraban en ese criterio. El *dataset* base no tuvo más modificaciones aparte de esto, con el resto del trabajo realizado por la opción de *data augmentation* disponible en el método de entrenamiento de Ultralytics [3]. El *dataset* consiste de imágenes de fresas en múltiples escenarios y de diferentes fuentes, con todas las fresas anotadas con un *bounding box* y etiquetada en una de tres clases: [“flower”, “ripe”, “unripe”] (flor, maduro, no maduro, respectivamente). De estas tres clases mencionadas, solo dos fueron utilizadas en el sistema final: [“ripe”, “unripe”]. [6]

Para el entrenamiento del modelo se utilizó la librería de Ultralytics para modificar las variables de entrenamiento y hiperparámetros. Como se mencionó anteriormente se utilizó las herramientas de *data augmentation* que proporciona Ultralytics para aumentar la cantidad de

variaciones entre las imágenes, así reduciendo las probabilidades de *overfitting* y reduciendo los casos imprevistos, algo que ayudó para su utilización directa en los escenarios de prueba ya que no se tenía imágenes provenientes del laboratorio en nuestro *dataset*. La red se entrenó en un total de 74 epochs, este siendo un número arbitrario basado en los resultados de las validaciones generadas por epoch y por el progreso en el *loss landscape* analizando las estadísticas de pérdida de la red durante el entrenamiento. En el aspecto de *data augmentation* se introdujeron modificaciones geométricas para diversificar la capacidad de reconocimiento en un mayor número de poses de caso extremo, además de agregar modificaciones de color para acoplarse a las distintas posibilidades de iluminación que se puede enfrentar el sistema, de los tres factores modificables de color en el espacio de color HSV (Hue, Saturation, Value. Matiz, Saturación y Valor en español) La matiz fue la única que no se activó la modificación. [3][6]

La validación del modelo proporcionó resultados importantes para poder diagnosticar posibles problemas, fallos y puntos débiles del modelo. Como se observa en la Figura 18, hay unos aspectos del modelo que no son ideales, especialmente en el alto índice de confusión entre la clase “unripe” y el fondo o nulo. Esta alta inestabilidad puede suceder por dos casos primarios, uno siendo resultante de las características del *dataset*, el cual tiene varias imágenes con fondos de hojas verdes con fresas no maduras de color verde sobrepuestas como se ve en la Figura 25, otro factor que genera esta inestabilidad es el sesgo inherente que tiene las imágenes utilizadas para la validación que al momento de hacer la validación, estos resultados no representan exactamente las condiciones utilizadas en el escenario de pruebas. Sumando a esto, por la naturaleza del *dataset* de consistir de imágenes públicas, existen imágenes que no tienen todas las fresas anotadas, las cuales el modelo si logra detectar como se observa si se compara la Figura 25 con la Figura 26. Por el otro lado, la clase de “ripe” que se puede considerar como las más importante proporciona una respuesta excelente en la matriz de confusión, con un índice casi nulo de falsos positivos contrastando “ripe” con “unripe”, lo cual resalta que la probabilidad de procesar (detectar y cosechar) una fresa no madura es extremadamente bajo. Aunque se pueda decir esto, no es la historia completa, ya que el índice de confusión cuando se contrasta “ripe” con el fondo provee un puntaje por encima del rango nominal (<0.20).

Continuando con la validación de las fresas, la curva de medida F1 valida que el límite de confianza para obtener el mejor balance de precisión y exhaustividad/sensibilidad (recall en inglés) es de 0.634, con una medida F1 para todas las clases de 0.8, observando la Figura 19 podemos ver que el desempeño de la clase de “ripe” es significativamente mejor que las otras, dando buena señal. Aunque la curva F1 proporciona estos resultados, es importante analizar el comportamiento aislado de la curva de precisión-confianza (Figura 20) y exhaustividad-confianza (Figura 21). Enfocando en la curva de precisión-confianza vista en la Figura 20, se puede observar que la precisión se ve altamente afectada en el límite de confianza que sugiere la curva F1. Extrapolando un valor después de una comparación entre las curvas, se determinó un valor de 0.65 como el límite mínimo de confianza para filtrar las detecciones.

Para comprobar los resultados de la validación se realizaron pruebas de casos extremos. En el escenario de pruebas, con el límite de confianza mencionada anteriormente de 0.65 no se logró generar algún falso positivo pero se pudo generar un falso negativo al obstruir la fresa parcialmente con una hoja como se ve en la Figura 28. Para inducir casos de falsos positivos y así analizar los posibles errores que pueden suceder se redujo el límite mínimo de confianza a 0.25 lo cual generó un caso donde el modelo detectó el botón de paro de emergencia del robot UR3e como se nota en la Figura 27. Los resultados de estas pruebas en el escenario experimental proporcionan una buena visualización de la eficacia del modelo en un ambiente distinto al que se entrenó.

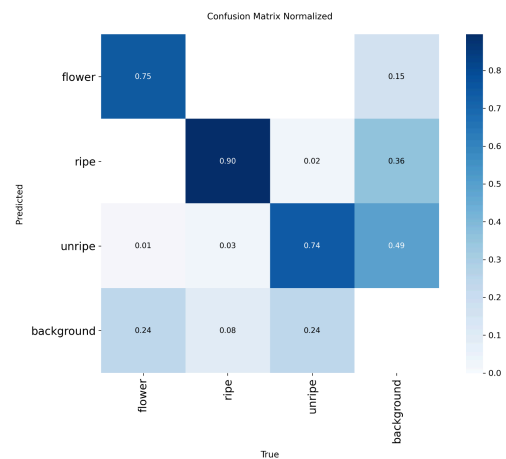


Figura 18. Matriz de confusión del modelo, generado durante la validación final del modelo.

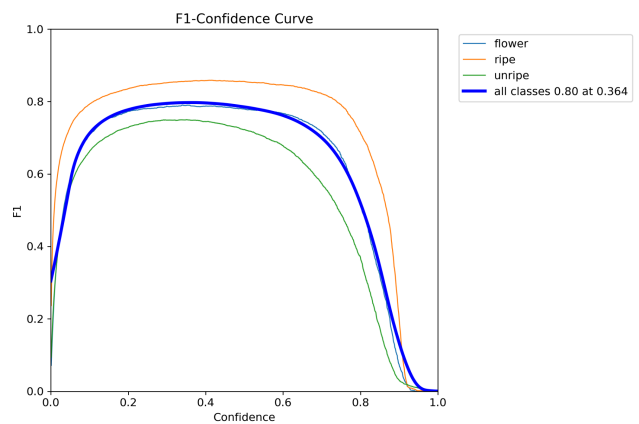


Figura 19. Curva de relación entre *F1 Score* y confianza del modelo.

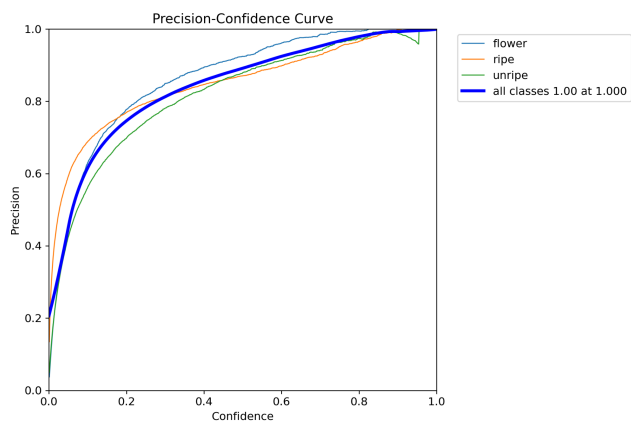


Figura 20. Relación entre Precisión y confianza.

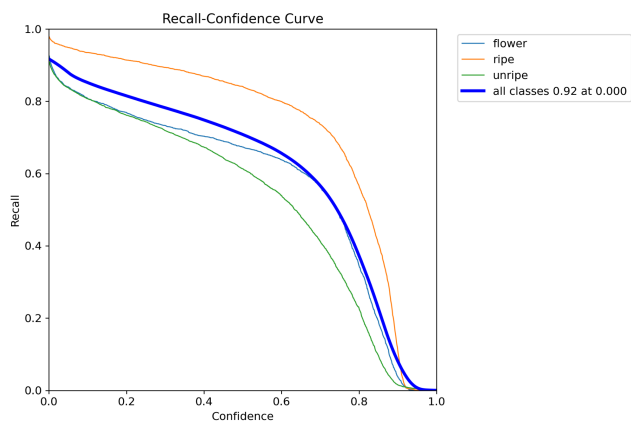


Figura 21. Relación entre Exhaustividad y confianza.

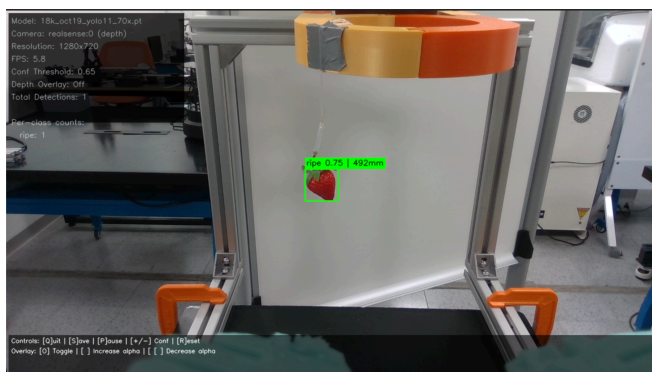


Figura 22. Prueba en condiciones ideales, una sola fresa, escenario de pruebas estable.



Figura 23. Prueba en condiciones de detección múltiple, mostrando que el modelo detecta ambas clases correctamente pero se filtra las “unripe” como no válida para cosechar.



Figura 24. Prueba con obstrucción parcial en múltiples fresas, con diferentes niveles de obstrucción. Mostrando la eficacia del modelo en estas condiciones. De izquierda a derecha: >50%, <20%, ~30%.

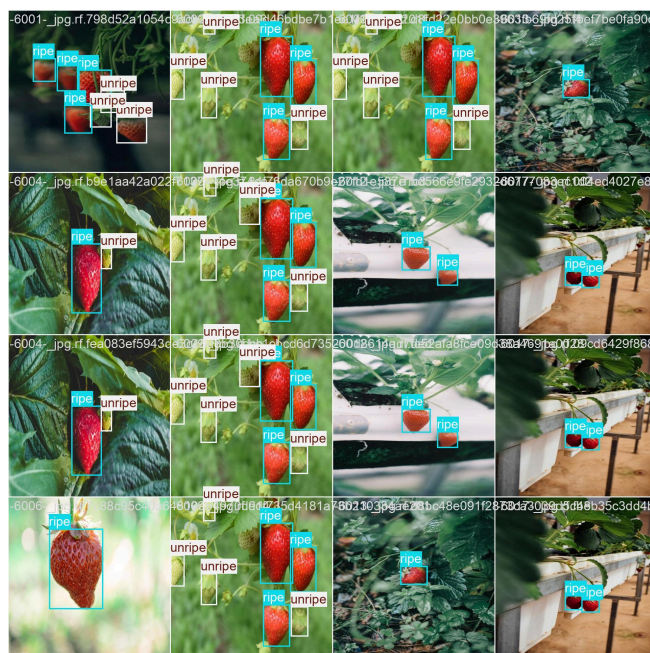


Figura 25. Matriz de imágenes de validación con sus etiquetas

originales.



Figura 26. Matriz de imágenes de validación con el resultado de la inferencia. Se observa que el modelo detectó fresas que no estaban etiquetadas, generando una desviación en los resultados del proceso de validación.

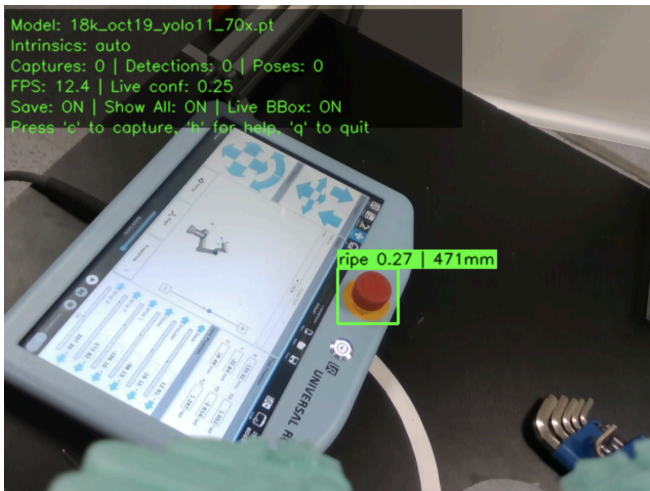


Figura 27. Falso positivo producido con un límite de confianza de 0.25 ± 0.02 .



Figura 28. Falso negativo producido en un caso de múltiples frutos, oclusión de ~20% total.

IV. DISEÑO Y SISTEMA DE ACTUACIÓN DEL GRIPPER SUAVE

A. Conceptualización del gripper a partir de la revisión de literatura

El concepto general del gripper se enmarca en la línea de los grippers blandos neumáticos desarrollados para manipulación delicada, donde la compliance del elastómero y la actuación por redes de cámaras permiten adaptar la forma del efector a geometrías irregulares con bajas presiones de contacto. Esta aproximación está ampliamente sustentada por revisiones de grippers blandos y actuadores neumáticos, que describen cómo arquitecturas tipo PneuNet generan curvatura mediante la presurización de cavidades internas y cómo parámetros geométricos (espesor de paredes, número/tamaño de cámaras y capa restrictiva) gobiernan la amplitud y estabilidad del movimiento [13][14].

B. Diseño del actuador base y componentes del gripper

1. Diseño del actuador base

Los tres actuadores blandos incorporados en el diseño de la herramienta de cosecha están formados por una geometría base en forma de punta de flecha bifurcada en la punta. La superficie de los actuadores tiene una forma en curva con elementos sobresalientes formando un patrón para aumentar la fricción del actuador blando y por ende la capacidad de agarre.

El método de actuación de los grippers es a través de una serie de cámaras neumáticas. Cada actuador tiene dos filas de cámaras cuyos ejes son intersecantes, este formato y posicionamiento de las cámaras asegura un accionamiento de dos formas, una curvatura en una dirección principal y otra curvatura perpendicular a esta. Este método proporciona un aseguramiento de la fruta más confiable con menos actuadores individuales. El ángulo de los ejes de las cámaras respecto al centro del actuador fue determinado a través de la construcción de la geometría base, resultando un un ángulo de incidencia de 8.53° , se observará el resultado de esta geometría en la sección del análisis de los resultados del FEA (Finite Element Analysis).

En la Figura 29 se puede observar la geometría del actuador a detalle a través de una serie de vistas acotadas. Muchas de las medidas fueron resultantes de acondicionamientos realizados al modelo base (operaciones de chaflán y fillet) por lo que resultan no enteros. El actuador tiene una longitud efectiva de 47.82mm, medida que no es afectada con el diseño embebido del colchón. Tomando como referencia el final del penúltimo par de cámaras contando desde la base (entrada de aire), se tiene un ancho de 30.08mm. Estas medidas representan una tolerancia aceptable para la medida de las fresas con las que se van a trabajar, la compresibilidad del “colchón” proporciona aproximadamente 5-7mm adicionales en el caso de una fresa de mayores dimensiones (altura de la fresa). En términos del diámetro de la fruta, la forma curvada de los actuadores (arco con radio de 61.27mm, sin incluir el patrón de protuberancias de 1.5mm) permite una mejor adaptabilidad a la curvatura de la fresa en contraste con un diseño plano.

Para tener una actuación adecuada en un rango de presiones aceptable y estable se decidió en 2 series de 5 cámaras a lo

largo del actuador, cada cámara con dimensiones internas de 5mm de largo, 2.5mm de ancho y 12mm de alto. Las dimensiones externas se establecen por el grosor de 2mm a todas las paredes. Adicionalmente, se tiene un grosor máximo de 13.8mm en la base del actuador, con un grosor mínimo de 6.5mm donde se requiere una mayor permisibilidad.

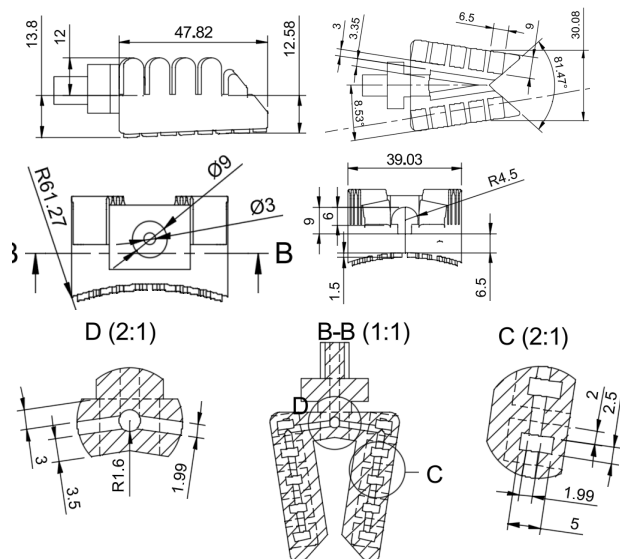


Figura 29. (De izq. a der., 1ra fila : A, B; 2da fila: C, D; 3ra fila: E, F. G): Se observan las vistas Lateral (A), Superior (B), Trasero (C), Frontal (D), Superior con corte transversal (F), Sup. enfoque interno base (E), Sup. enfoque interno cámaras (G). Todas las vistas acotadas. Para el dibujo técnico completo, vea el Anexo A6.

Se calcula un volumen ocupado por elastómero de 21757.268 mm^3 . Tomando en cuenta la densidad del elastómero (Mold Star 15 SLOW) que se va ocupar para la fabricación de los actuadores de 0.0011 g/mm^3 . Multiplicando estos valores resulta en una masa de 25.67g de material para cada uno de los actuadores.

2. Componentes para montaje en el robot UR

Considerando los tres actuadores blandos neumáticos que se utilizan en la herramienta manipuladora, se requieren algunos componentes adicionales para ensamblar. Una de las características principales de la herramienta es la posibilidad de ajustar la distancia respecto al eje central de cada uno de los actuadores sobre su propio riel, para contemplar esto, se tiene un tipo de carruaje para cada actuador que lo guía sobre el riel. Adicional a esto, se tiene la placa principal la cual contiene los rieles mencionados, contiene el colchón, el punto de anclaje para el soporte de la cámara y las piezas de anclaje al acoplador universal. Se puede observar el ensamble de estas piezas en la Figura 30. Finalizando, el ensamble de la herramienta manipuladora se asegura al acoplador universal con cuatro tornillos de máquina DIN 912 de medida M3 x 16mm, ya que solo se utilizan dos pilares estructurales espejados, con dos tornillos por lado se ocupan 4 tornillos en total.

Como elemento complementario se diseñó un “colchón” denominado *Center Pad* que se posiciona en el centro de los grippers de forma perpendicular, habilitando que todas las superficies en posible contacto con la fresa sean blandas, aumentando la seguridad de la fruta y reduciendo el riesgo de mallugar la fresa. Este elemento es visible en la Figura

31 que es la vista frontal del ensamble presentado en la Figura 30.

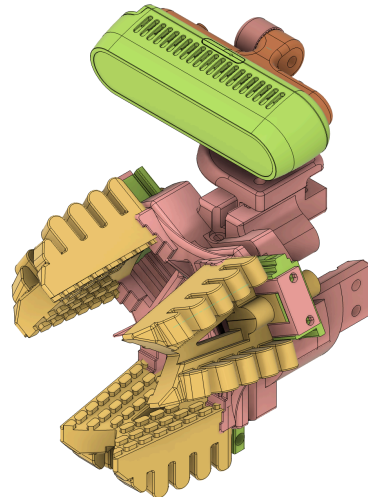


Figura 30. Vista isométrica normal del ensamble de la herramienta manipuladora completa, con los componentes resaltados en diferentes colores. No incluye el acoplador al efecto final.

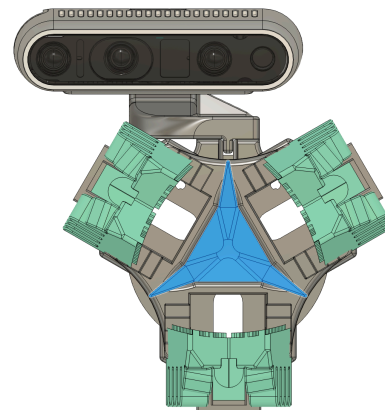


Figura 31. Vista Frontal del ensamble de la herramienta manipuladora completa, con los componentes resaltados en diferentes colores. No incluye el acoplador al efecto final. Se percibe el “colchón” diseñado resaltado en azul.

3. Acoplador entre herramienta de cosecha y efector final del robot

Considerando que el escenario de pruebas es compartido entre múltiples *grippers* por otros equipos de desarrollo fue necesario el desarrollo de un método de acoplamiento al robot UR3e que fue utilizado, por esto mismo diseñamos un acoplador universal de rápido ensamble y desensamble para reducir tiempos de intercambio de grippers. Este acoplador se queda anclado de forma permanente a la brida del robot y expone cuatro puntos de anclaje concéntricos perpendiculares a la brida, facilitando el acceso a los tornillos de anclaje entre el gripper y el acoplador. La Figura X muestra la geometría de esta pieza, la cual por su diseño compacto y método de anclaje reduce las dimensiones y extensión máxima de la herramienta. El acoplador es asegurado al robot con 4 tornillos de máquina DIN 912 de medida M6 con una longitud de 12 mm.



Figura 32. Vista isométrica rotada del acoplador compartido diseñado por YEA para el robot UR3e compartido, se muestran los tornillos M6 utilizados para anclar al robot.

4. Diseño de moldes para actuadores blandos

Los moldes utilizados en los actuadores fueron diseñados en base a métodos existentes pero con algunas mejoras experimentales.

La fabricación de los actuadores neumáticos está diseñada para realizarse en dos partes, un molde para la sección de las cámaras neumáticas y otro molde para la base. En una sección más adelante se divulgará a detalle el proceso de manufactura.

El molde de la sección de cámaras está diseñado para ser ensamblado en cuatro partes para facilitar el desmolde de la pieza. Se tienen tres secciones principales como se observa en la Figura 33, las cuales consisten en una base, tapa con la forma interior de las cámaras y un tapón que forma la boquilla del actuador, este tapón funciona como la primera pieza a remover para liberar la presión negativa que se genera al separar las piezas. La “tapa” de este molde incluye orificios de escape de aire para evitar que se atrapen burbujas, algo que es altamente probable por la geometría negativa que forma la tapa. Todas las piezas contienen elementos para facilitar su agarre al momento de desmoldar, al igual que puntos en donde se puede insertar una herramienta para hacer palanca si es necesario. Adicionalmente, el molde se ensambla con una serie de tornillos de máquina M3 de longitud variable, esto ayuda a que el molde esté encapsulado y alineado correctamente, asegurando que todos los actuadores manufacturados tengan mínimo error entre uno y otro. Este diseño también ayuda a que se pueda mover el molde armado y durante la etapa de curado sin mayores problemas.

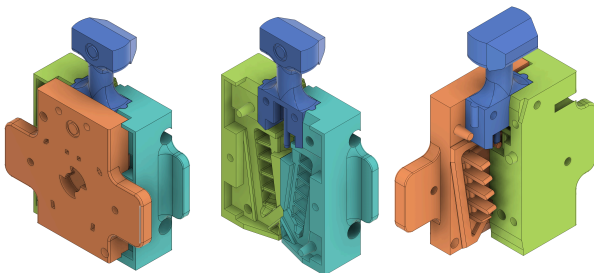


Figura 33. Vistas isométricas del molde de sección de cámaras en diferentes orientaciones y visibilidad de piezas, se resaltan las piezas por colores distintos.

El molde de la sección de la base consiste de igual forma de cuatro piezas de similar función, con dos formando la base, una tapa con mínima geometría, y un tapón que genera una geometría negativa y ayuda para purgar la presión formada al momento de desmoldar. Al igual que el molde de las cámaras, este contiene elementos para facilitar el desmolde con la diferencia principal siendo la posición neutra en la cual se coloca el molde en la etapa de curado, como se observa en la Figura 34, este molde se mantiene en una posición “parada” o perpendicular a la posición neutra del molde de la sección de cámaras.

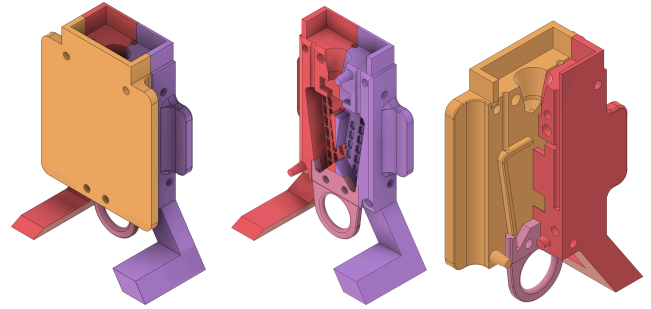


Figura 34. Vistas isométricas del molde de sección de base en diferentes orientaciones y visibilidad de piezas, se resaltan las piezas por colores distintos.

C. Simulaciones FEM

El modelo FEM es una herramienta que ayuda a visualizar cómo se comportará el objeto a estudiar, en este caso cuanto se deformará el gripper dependiendo de la presión utilizada y de la geometría del mismo.

El primer paso para lograr esto es definir el material junto con sus características dentro del software de ANSYS utilizando la herramienta de “Static Structural”. Para este proyecto se utilizará un elastómero hiper elástico conocido como “Mold Star 15 SLOW” cuya densidad es de 1180 kg/m^3 . Se especifica que el material se caracterizó realizando diferentes pruebas mecánicas con las cuales se encontró un modelo de Yeoh de tercer orden, dicha caracterización fue realizada por Cruz-Santos (2019) [1]. Las propiedades encontradas para este material fueron las siguientes:

$$C10 = 66728 \text{ Pa}, C20 = 7993.5 \text{ Pa}, C30 = 0 \quad (1)$$

$$D1 = 0 \text{ Pa}^{-1}, D2 = 0 \text{ Pa}^{-1}, D3 = 0 \text{ Pa}^{-1}, \quad (2)$$

Una vez definido el material junto con sus características se importa la geometría a analizar dentro del software. Se abre la ventana de “Mechanical” en donde se especifica el material de cada uno de los elementos importados. También se añade un soporte fijo en la base de cada uno de los dedos y se establece que la deformación se verá afectada por la gravedad. El mallado utilizado para este análisis es el que da ANSYS por default. La figura 35 muestra las principales características dentro del software a usar.

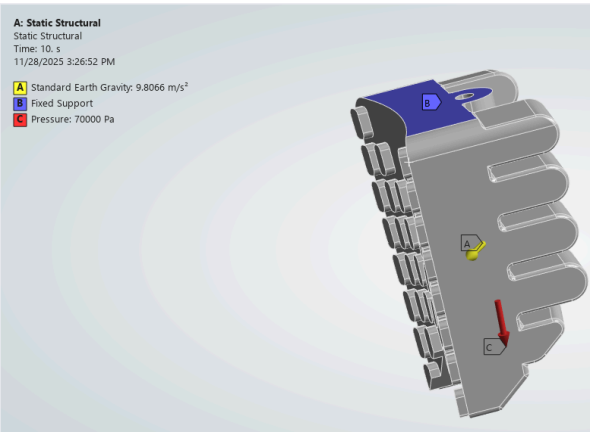


Figura 35. Condiciones establecidas en ANSYS

El siguiente paso es establecer que el objeto se verá deformado por presión neumática. Se seleccionan todas las caras internas del objeto que tendrán contacto con el aire. Asimismo se establece que la presión entrará al actuador con una señal tipo rampa para que el análisis sea lo más apegado a la realidad. Por último, se establece el nivel de presión neumática máxima que se le aplicará al gripper y se deja que el software resuelva el modelo matemático y demuestra cómo se deformará dicho gripper como se muestra en la figura 36. Asimismo en la tabla 1 se muestra como fue cambiando la presión suministrada al actuador con respecto al tiempo.

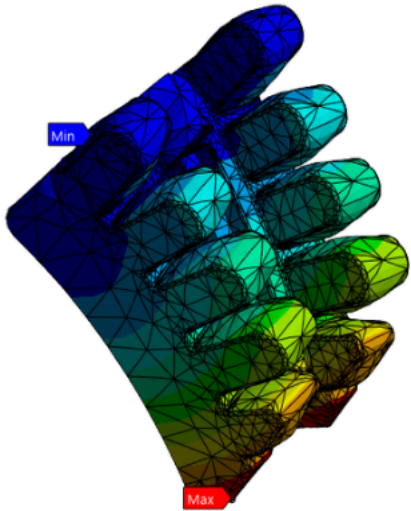


Figura 36. Resultado de gripper deformado

TABLA I. PRESIÓN VS TIEMPO

Presión vs tiempo	
Tiempo (s)	Presión (Pa)
0	0
1	0
2	10,000

Presión vs tiempo	
Tiempo (s)	Presión (Pa)
3	20,000
4	30,000
5	40,000
6	50,000
7	60,000
8	70,000

a. Tabla explicando la relación entre tiempo y presión

Para comprobar que el gripper diseñado cumpliera con los objetivos deseados se realiza un análisis *PROBE*, el cual consiste en analizar el comportamiento de diferentes puntos durante la activación del sistema neumático.

En las siguientes gráficas se muestra el comportamiento de dos puntos distintos en el eje x y en el eje y del diseño propuesto con anterioridad.

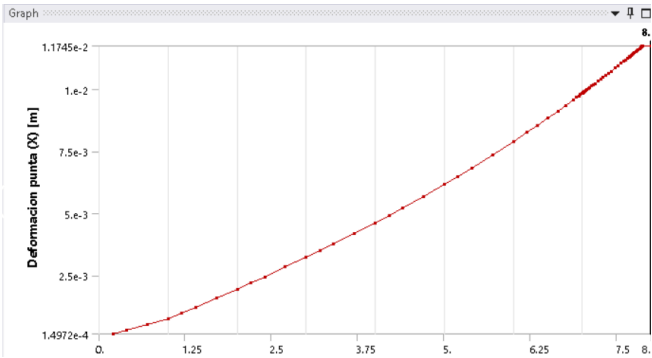


Figura 37. Comportamiento de la punta de cada dedo en el eje x al aplicar 70 kPa

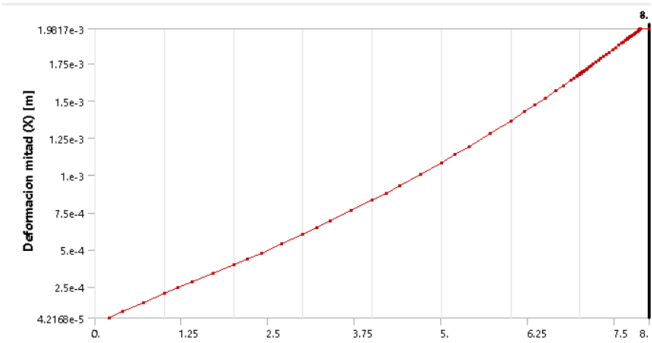


Figura 38. Comportamiento de un punto medio de cada dedo en el eje x al aplicar 70 kPa

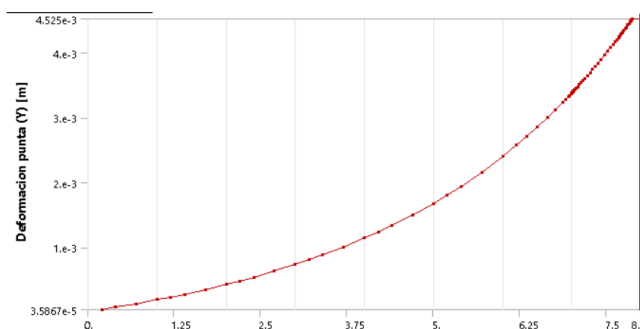


Figura 39. Comportamiento de la punta de cada dedo en el eje y al aplicar 70 kPa

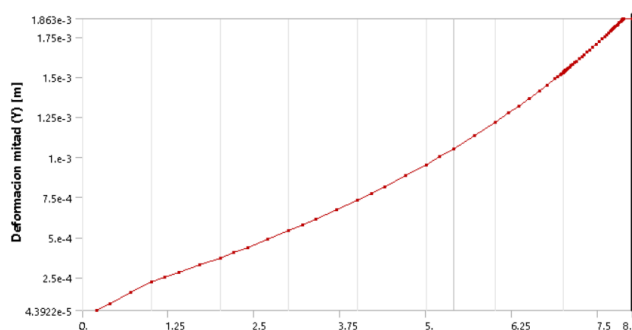


Figura 40. Comportamiento de un punto medio de cada dedo en el eje y al aplicar 70 kPa

Solo se analizó la punta y la parte media del dedo ya que la base es fija y como consecuencia no se ve afectada por deformación.

Una vez obtenidos los datos de deformación en cada eje se analizará la relación que existe entre la deformación (posición en el espacio) y la presión. Esto se realizó al obtener la magnitud del análisis *PROBE* realizado anteriormente. En la figura 41 se muestra la gráfica que representa dicha relación.

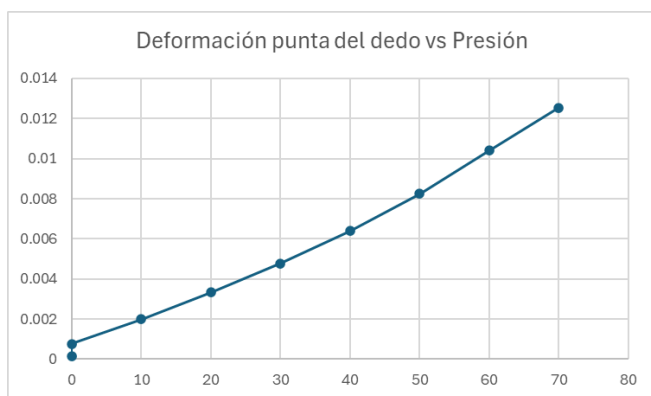


Figura 41. Gráfica deformación vs presión de la punta del dedo

D. Proceso de manufactura

Una vez se realizó el diseño y simulaciones FEM asegurando que el gripper soportaría las presiones de trabajo previstas y generaría el movimiento necesario para sujetar el fruto sin dañarlo, se comienza con la manufactura del Gripper.

Iniciamos calculando la masa de cada parte del gripper esto

con el fin de obtener la cantidad de elastómero a usar, esto para evitar desperdiciar material.

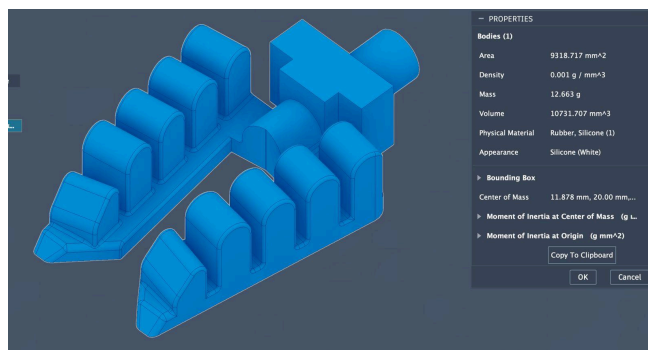


Figura 42. Análisis de propiedades “Cámaras”

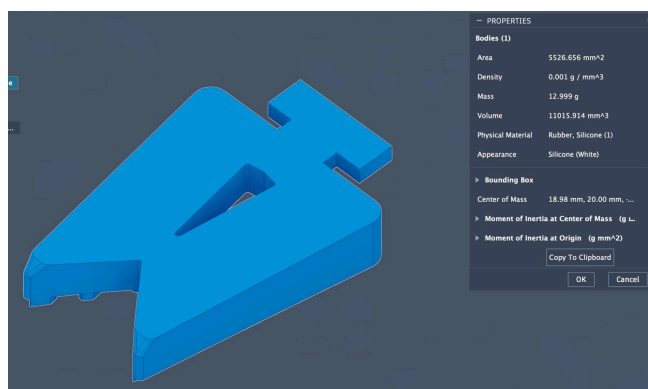


Figura 43. Análisis de propiedades “Capa”

La Figura 42 muestra las propiedades de las cámaras del gripper, con ello la cantidad a usar de elastómero para cubrir el molde, en la Figura 43 de igual forma muestra las propiedades de la capa del Gripper, para las cámaras se usará es con exactitud “12.663 g” y para la capa “12.999 g” con esto redondeamos a 13 gramos por molde con un total de 26 g por dedo, más 5 g extra para pegar ambas partes.

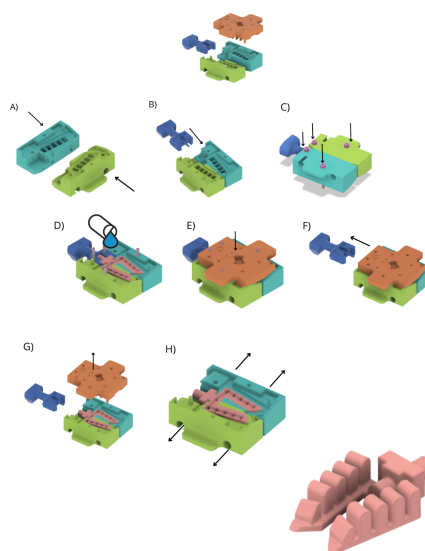


Figura 44. Diagrama de fabricación “Cámaras”

Una vez finalizada la impresión del molde y determinado el volumen del elastómero necesario, procedemos al proceso de vaciado. Este paso es crucial para garantizar la correcta formación de las cavidades internas y la integridad

estructural del actuador.

La Figura 44 ilustra de manera secuencial el procedimiento (incisos A - E), comenzando por el ensamble del molde. En esta etapa se verifica que las superficies de contacto estén limpias y libres de residuos para asegurar un sellado adecuado, además de colocar desmoldante, esto para facilitar el desmoldeo al momento de sacar la pieza del molde. Posteriormente se prepara la mezcla de elastómero siguiendo las proporciones ya calculadas, cuidando la eliminación de burbujas.

Con el molde correctamente alineado y sellado, se realiza el vaciado del elastómero. Durante el proceso se recomienda verter de manera continua y a baja altura para minimizar la formación de burbujas.

El tiempo de curado, una vez completamente sellado el molde, es de aproximadamente 5 horas, aunque se recomienda que sea 1 día, durante el cual debe permanecer en reposo sin ser manipulado para evitar deformaciones.

Transcurrido el tiempo, se procede al desmoldeo (incisos F - H). En esta fase se retiran los tornillos y molde y se separan las piezas con el fin de extraer la pieza del actuador. Finalmente, se inspecciona la pieza resultante para verificar la correcta formación de las cámaras y la ausencia de imperfecciones superficiales que puedan comprometer su funcionalidad.

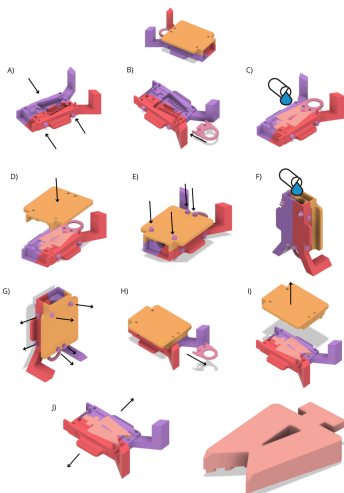


Figura 45. Diagrama de fabricación "Capa"

Para la fabricación de la capa exterior del Gripper, se siguió un procedimiento similar al utilizado previamente en la elaboración de las cámaras neumáticas. Una vez impreso el molde y determinada la cantidad de elastómero necesaria, se procede al ensamble del molde y el vaciado, tal cual como se muestra en la Figura 45.

En este caso en particular, se recomienda realizar el vaciado en dos etapas. Primero, se vierte una cantidad inicial de elastómero antes de colocar la tapa y sellar el molde, esta aplicación es necesaria debido al diámetro reducido del orificio superior el cual limita la velocidad de llenado y provocar que el elastómero tarde más llenar el molde. Posteriormente, se levanta se levanta el molde y se coloca en posición vertical para realizar un segundo vaciado a través del orificio superior.

El tiempo de curado del elastómero es de aproximadamente

5 horas, sin embargo, para asegurar una solidificación completa, se recomienda dejar reposar el molde durante un día completo antes de manipularlo. Una vez concluido el tiempo de curado, se lleva a cabo el desmoldeo (incisos G - J), retirando con cuidado cada sección del molde. Finalmente, se inspecciona la pieza para verificar que el espesor sea uniforme y que no existan burbujas o irregularidades que puedan afectar el desempeño del gripper.

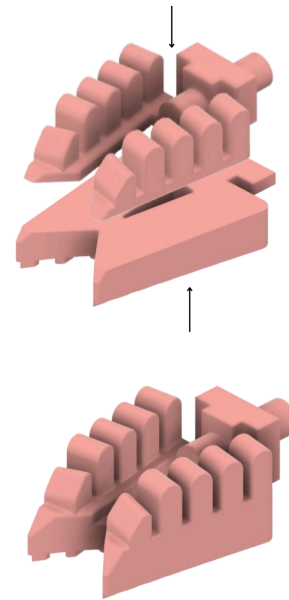


Figura 46. Diagrama de fabricación "Actuador"

Una vez tengamos ambas piezas, procederemos a unir las, para ello usaremos los 5 gramos extras que añadimos.

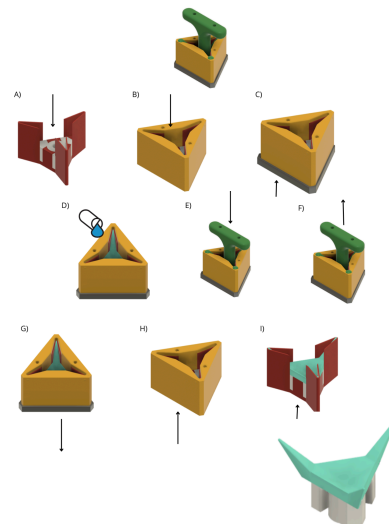


Figura 47. Diagrama de fabricación, colchón del centro del Gripper

Como complemento de nuestro Gripper se diseñó una base con el mismo material de la fabricación del actuador, esto con la finalidad de proporcionar un mejor colchón para evitar que la fresa resbale de los dedos. La Figura 47 muestra el proceso de manufactura de la base, así como el desmoldeo.

E. Ensamble del gripper

Una vez fabricamos los dedos correspondientes y necesarios para nuestro Gripper además de la base procederemos al armado del Gripper.

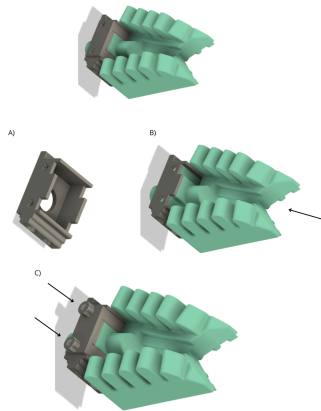


Figura 48. Diagrama de ensamblaje de los actuadores

Antes de proceder con el ensamblaje final del gripper, es necesario preparar cada uno de los dedos de forma individual. Este preensamblaje asegura que todas las piezas encajen correctamente y que, al integrarlos en la estructura principal, el mecanismo funcione de manera uniforme. En la Figura 48 se muestra el proceso detallado de ensamblaje de un dedo, el cual debe repetirse para los tres dedos necesarios para conformar el gripper completo.

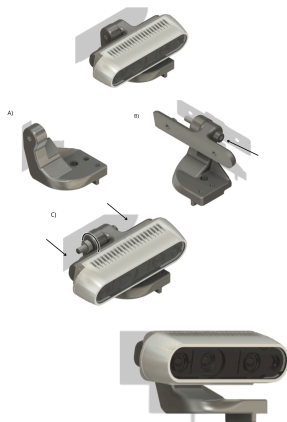


Figura 49. Diagrama de ensamblaje “Cámara”

De manera similar, el ensamblado del módulo de la cámara es un paso fundamental para garantizar su correcta integración con el gripper. La Figura 49 muestra el proceso de ensamblaje de la base diseñada para sostener la cámara, así como la forma en que esta se acopla a la estructura del gripper. Este procedimiento asegura una alineación adecuada del sistema de visión y contribuye al correcto funcionamiento del mecanismo de manipulación.

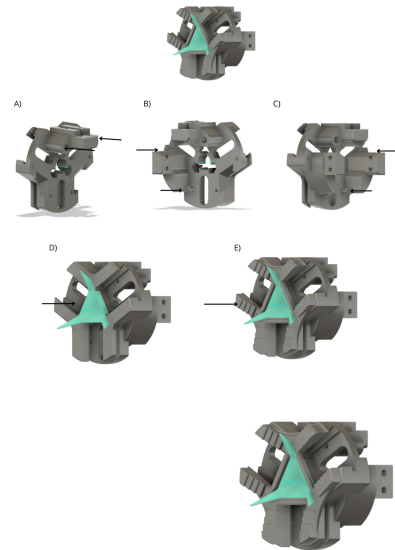


Figura 50. Diagrama de ensamblaje “Gripper”

Primeramente, es necesario ensamblar la estructura principal del gripper antes de integrar la cámara y los actuadores neumáticos. La Figura 50 muestra el proceso de ensamblaje del gripper, en el cual también se incorpora la base previamente diseñada para su posterior acoplamiento con el sistema de visión. Este orden de montaje garantiza un ajuste adecuado de los componentes y facilita la instalación final del sistema completo.

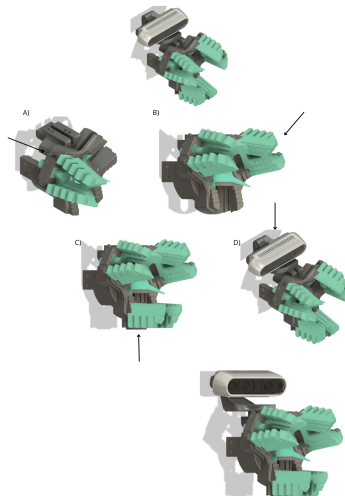


Figura 51. Diagrama de ensamblaje “Gripper completo”

La Figura 51 muestra el proceso completo de ensamblaje del gripper, incluyendo la instalación de los actuadores y el módulo de la cámara. Por su parte, la Figura 52 presenta el gripper totalmente armado una vez integrados todos los componentes.



Figura 52. Foto del Gripper completo

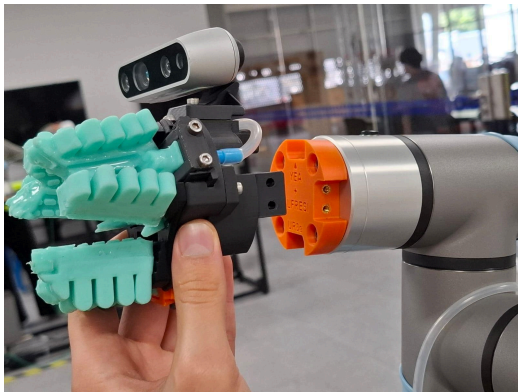


Figura 53. Foto del Gripper junto con el acoplador

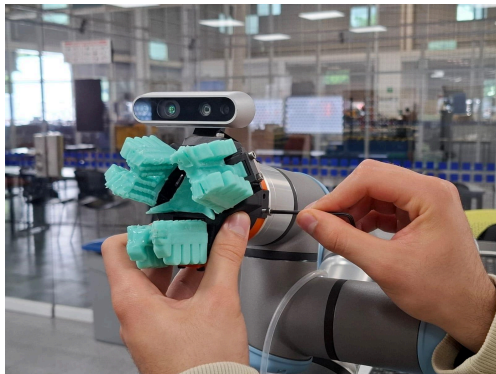


Figura 54. Foto del Gripper montado al acoplador

Las Figuras 53 y 54 presentan fotografías del gripper completamente ensamblado y posteriormente montado en el acoplador, mostrando su integración final con la estructura del sistema.

F. Problemas presentados

Los principales problemas surgieron durante la manufactura de los actuadores. Al inicio, la mezcla generaba una cantidad considerable de burbujas en zonas críticas, lo que afectaba directamente su funcionamiento. Además, en varios intentos se aplicó una cantidad insuficiente de desmoldante, lo que dificultó el desmoldeo y en algunos casos provocó que el molde se rompiera, obligando a imprimir nuevos moldes. Otros inconvenientes estuvieron relacionados con el sellado del actuador: al unir las cámaras con la capa superior, el elastómero no siempre se distribuía

de manera uniforme, generando pequeñas fugas de aire que impedían una deformación adecuada del actuador.

G. Sistema de actuación

En el sistema de actuación se utilizará un circuito electro-neumático con el cual se manipulará la presión neumática que se proporciona hacia el actuador. Algunos de los elementos principales para realizar este circuito fueron un microcontrolador, una bomba neumática, una electroválvula, un sensor de presión, un convertidor de nivel, un ADS y dos módulos MOSFET. En la tabla 2 se especifica el uso de cada componente y los componentes adicionales usados para su correcto funcionamiento.

TABLA II. COMPONENTES UTILIZADOS EN EL SISTEMA DE ACTUACIÓN

Componentes sistema de actuación	
Componente	Uso
ESP-32	Controlar la parte eléctrica del circuito
Bomba neumática	Proveer aire al sistema neumático
Electrovalvula 3/2 MAC 34B-AAA-GDNA-1 BA	Controlar el flujo de aire del sistema
Sensor de presión XGZP6847A (-100 a 100 kPa)	Obtener la presión que existe en el sistema
Convertidor de nivel	Adaptar la señal de 3.3 V a 5 V para el funcionamiento del ADS y el sensor de presión
ADS 1115A	Permitir una correcta lectura del sensor presión por parte de la ESP-32
Modulo MOSFET D4184	Controlar el voltaje suministrado mediante PWM de la bomba y la electroválvula
Regulador de voltaje 7805	Adaptar la señal de 12 V a 5 V para los componentes requeridos
Fuente 12 V	Fuente de alimentación para electroválvula y bomba neumática

b. Tabla explicando el objetivo de uso de los componentes

En la figura 55 se muestra un diagrama de las conexiones entre cada uno de los elementos mencionados anteriormente.

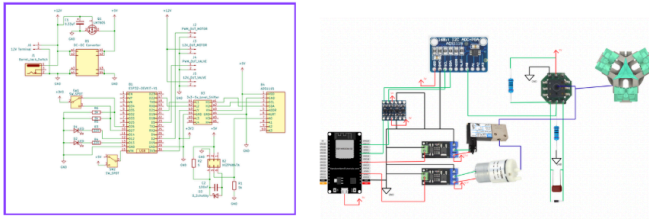


Figura 55. Diagrama de conexiones del sistema de actuación con sus representaciones reales.

En la figura 56 se muestra la implementación física del circuito diseñado y mostrado en la figura 55

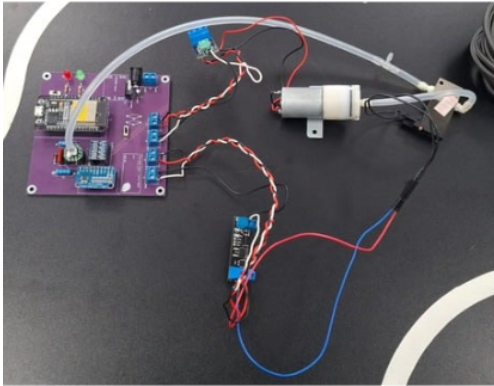


Figura 56. Implementación física del sistema de actuación

El funcionamiento de este circuito será el siguiente:

1. Inicio de proceso de movimiento de robot.
2. Cuando el robot se ubique en la posición para tomar la fresa se enviará el mensaje "Gripper encendido" mediante MQTT para iniciar el circuito.
3. Una vez alcanzada la presión deseada, con una tolerancia de ± 2 kPa, el circuito mandará el mensaje "inflado" mediante MQTT para que el robot pueda continuar con la rutina de movimiento.
4. Cuando el robot llegue a la posición para depositar la fresa se mandará el mensaje "Gripper apagado" mediante MQTT para desactivar el sistema y sacar el aire existente en el sistema.
5. Una vez que el aire haya salido del sistema y que el sensor de presión otorgue una lectura de 0 kPa, el controlador mandará el mensaje "desinflado" por MQTT para así poder terminar la rutina de movimiento o en su caso volver a empezar dicha rutina.

Se realizaron diferentes experimentos con el fin de relacionar la presión con el PWM suministrado a la bomba. Para hacer estos experimentos se diseñó un código que aumentaría el valor del PWM cada 5 segundos. El siguiente paso fue conectar la salida de la bomba directo al sensor de presión para poder obtener las lecturas de presión.

Se realizaron tres diferentes pruebas de caracterización para poder observar la relación mencionada anteriormente. Para obtener una lectura correcta sobre la presión presente en el sistema se utilizó la siguiente fórmula que explica la relación entre la presión obtenida por el sensor y el voltaje de salida suministrado por el mismo sensor hacia el

convertidor analógico, dicha conversión se obtiene gracias a los datos presentes en la datasheet [2]:

$$\text{Presión} = (50 * \text{Voltaje}) - 125 \quad (3)$$

El modelo de presión/PWM a utilizar en el sistema de control se definió después de analizar el coeficiente de determinación y el comportamiento de las ecuaciones y gráficas generadas. Para el caso del control de esta plataforma se utilizará la siguiente ecuación con su respectivo coeficiente de determinación:

$$\text{PWM} = (3.0922 * \text{presión}) + 19.497 \quad (4)$$

$$R^2 = 0.9791 \quad (5)$$

También se pueden observar los datos obtenidos por el sensor y la gráfica generada en la figura 57 y la tabla 3.

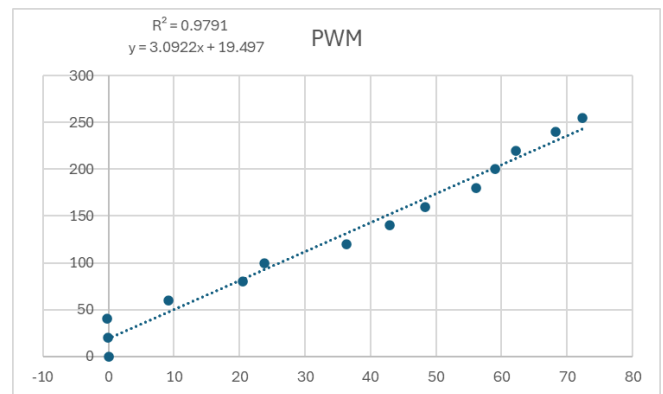


Figura 57. Gráfica de relación entre presión y PWM

TABLA III. DATOS OBTENIDOS POR PRUEBA DE CARACTERIZACIÓN

Datos	
Presión (kPa)	PWM
-0.08	20
-0.17	40
9.18	60
20.54	80
23.79	100
36.33	120
42.92	140

Datos	
Presión (kPa)	PWM
48.36	160
56.1	180
59.03	200
62.12	220
68.21	240
72.35	255

c. Datos obtenidos por prueba de caracterización

Para evitar un comportamiento no deseado y evitar la existencia de error alguno en el sistema de actuación se utilizará un control PI. Una vez definido el modelo de presión a utilizar, gracias a las pruebas de presión en estado estacionario, se realizaron distintos experimentos con el fin de observar el comportamiento del sistema con diferentes valores en las ganancias proporcional e integral.

La definición de estos valores finales se realizó en base al número de oscilaciones que se generaban en el PWM suministrado a la bomba, al calcular el error del sistema, y el tiempo que tardaba el sistema en establecerse para poder llegar a la presión establecida. Los valores obtenidos fueron los siguientes:

$$Kp = 7, Ki = 2 \quad (6)$$

Para complementar el desarrollo del circuito de control para la actuación neumática del gripper, se diseñó el circuito en una PCB de dos capas, este PCB fue diseñado para ser utilizado universalmente para grippers blandos neumáticos de esta escala. Como se observa en el esquemático de la Figura 55, sigue el mismo circuito base neumático. Esta PCB integra los sistemas responsables de la medición del sensor de presión y generación de la señal PWM, al igual de ser fuente de poder para los sistemas 12V y 5V. La integración completa se observa en la Figura 56. Los archivos del diseño se encuentran en el Anexo A7.

V. IDENTIFICACIÓN DE POSE DE LOS FRUTOS MEDIANTE VISIÓN POR COMPUTADORA

En la figura 58 se muestra el diagrama cinemático del setup a implementar. En este diagrama se pueden notar los principales elementos que conforman la estación, cobot UR3e, soft gripper con soporte para cámara Realsense D435 y la ubicación de las fresas. Asimismo se puede observar la orientación de los sistemas de referencia que se utilizarán durante la rutina de movimiento. En este caso los sistemas referenciales a usar son el del cobot, la cámara y el objeto a manipular.

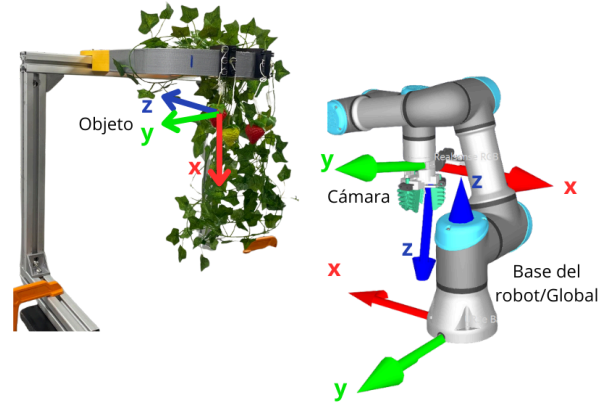


Figura 58. Diagrama cinemático del setup implementado, el tamaño y posición de elementos no está a escala.

Las referencias del sistema serán definidas con la siguiente nomenclatura para simplificar su representación:

Marcos de Referencia:

Base del robot: B

Brida del robot: F

TCP del gripper: G

TCP de la cámara: C

Fresa: S , obtenido por PnP como ${}^C T_S = {}^{camara} T_{strawberry}$

De los cuales la notación ${}^C T_S$ representa una transformación homogénea 4x4.

A. Pick and place

El proceso de recolección inicia desde la pose fija de Foto, donde se procesa la captura tomada por la cámara Realsense D435, esta pose es un *target* estático definido en RoboDK definido por ${}^B T_{Foto}$ es en este punto se invoca el servicio de visión que calcula PnP que da como resultado la matriz ${}^C T_S$. Continuando la secuencia se tiene *prepick_plane* que es otra pose estática que orienta el robot a la configuración apta para mover a las poses consecuentes, la transformada de *prepick_plane* se define como ${}^B T_{prepick_plane}$. Siguiendo esto se tiene la pose dinámica de *prepick* ${}^B T_{prepick}$, la cual se define en base de *pick* con un offset vertical (eje y si se toma el marco de referencia de la fresa $\{S\}$) predeterminado, este offset se aplica en base de los resultados del vector tvec, donde el offset se determina por el vector:

$$prepick_offset_mm = [0.0, 100.0, 0.0]$$

Posteriormente se tiene la pose dinámica de *pick*, determinada como ${}^B T_{pick}$ que es el equivalente de ${}^C T_S$. La forma en que está creado el sistema permite modificar los valores de offset en traslación y rotación para ambos *prepick* y *pick*. Esta herramienta se utiliza para reorientar el gripper para que quede en paralelo el eje X de $\{G\}$ con el eje X de $\{S\}$.

La sección de la secuencia de *post-pick* se divide en dos segmentos principales, el primero *post-pick* que se encarga

de maniobras para separar la fresa de la planta de forma segura y el segundo segmento encargado de colocar la fresa en su lugar deseado llamado *place*. Iniciando con *post-pick*, que consiste en dos maniobras definidas como movimientos articulares específicos que toma de base la configuración actual de las articulaciones $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ y se le suma el ángulo predefinido a las articulaciones específicas. Continuando con el segmento de *place*, el robot se mueve a ${}^B T_{home}$ y posteriormente a ${}^B T_{place_final}$ siendo *targets* específicos y es donde se termina el ciclo. El ciclo luego reinicia moviéndose a Foto. A continuación, en la Figura 59 se describe la secuencia general del Pick and Place en forma de pseudocódigo.

None

```
funcion ciclo_pick_and_place():
  inicializar_RoboDK_y_robot()
  inicializar_camara_RGBD_y_servicio_de_vision()
  inicializar_gripper_MQTT()
  mientras modo_operacion_continua_activo:
    mover_a_target("Foto") # T_B_Foto
    captura_datos_vision <- capturar_y_ejecutar_inferencia()
    para cada deteccion_fruit en datos_vision:
      # 1) Transformaciones y pre-pick
      T_B_S <- calcular_T_base_fruit(fruit, pose_TCP_en_Foto)
      T_B_prepick, T_B_pick <-
        crear_targets_prepick_y_pick(T_B_S)
      ejecutar_funcion_pre_pick(T_B_prepick)
      # 2) Pick propiamente dicho
      ejecutar_funcion_pick(T_B_prepick, T_B_pick)
      # 3) Post-pick y place
      ejecutar_funcion_post_pick(T_B_pick)

      # 4) Verificacion global tras colocar las fresas
      if verificacion_habilitada:
        exito <- verificar_pick_por_vision()
        actualizar_estado(exito)
```

Figura 59. Pseudocódigo representando la secuencia general del Pick and Place.

B. Función de Pre-pick

Entrando a detalle con el proceso de recolección se debe obtener la pose de la fresa, definida como una pose de 6-DOF, esta pose es obtenida con respecto al marco de referencia de la cámara $\{C\}$, para poder ser utilizado en las operaciones de cinemática inversa es necesario realizar la transformada de la fresa con respecto a la base del robot. Este proceso se describe a continuación.

Para obtener ${}^B T_S$ (transformación base-fresa) es necesario realizar la siguiente cadena de transformaciones:

1. Transformación fresa-cámara a partir de PnP: ${}^C T_S$
2. Transformación cámara-brid: ${}^F T_C$
3. Transformación brida-base: ${}^B T_F$
4. Transformación fresa-base:

$${}^B T_S = {}^B T_C \cdot {}^C T_S \quad (7)$$

El servicio de visión devuelve, para cada fresa detectada, la matriz homogénea ${}^C T_S$ los vectores de rotación y traslación *rvec* y *tvec* obtenidos por un algoritmo de PnP (Perspective-n-Point).

Además se incluyen los parámetros intrínsecos de la cámara:

- La matriz intrínseca K , que consiste en $\begin{bmatrix} f_x' & f_y' & c_x' & c_y' \end{bmatrix}$
- Coeficientes de distorsión $(k_1, k_2, p_1, p_2, k_3)$

Que por ser intrínsecos tomados de fábrica, los coeficientes de distorsión se marcan en ceros [7].

La relación de proyección utilizada por OpenCV para solvePnP se puede escribir como:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R_{CS} & t_{CS} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} \quad (8)$$

- (X_s, Y_s, Z_s) son puntos conocidos en el marco de la fresa $\{S\}$.
- (u, v) son sus proyecciones en píxeles dentro de la imagen.
- R_{CS} y t_{CS} representan la pose de la fresa respecto a la cámara.

En el proceso para resolver PnP, los puntos de imagen se obtienen a partir de la caja delimitadora *bbox_cxcywh* proporcionada por la red neuronal (YOLO11). Las esquinas del bounding box generan un conjunto de puntos 2D; los puntos 3D se definen en un modelo geométrico de la fresa (dimensiones conocidas) en el marco $\{S\}$. Con esto se manda a llamar al método de solvePnP para recuperar r_{CS} (*rvec*) y t_{CS} (*tvec*).

Como se menciona, la cámara RGB-D utilizada mide la profundidad creando un mapa de profundidades sampleadas por pixel. De esta medida se tienen varios métodos para acertar la profundidad de la fresa, con el procedimiento actual consistiendo en la reducción del ROI de medición al bounding box generado por sistema de detección. Procediendo luego con la subdivisión del ROI y la aplicación de una mascara de color a los pixeles de profundidad para evitar que la medida se tome de hojas o elementos que no sean la propia fresa, después de estos filtros, se toma la medida valida mas cercana al centro del bounding box y se aplica un offset calculado como el diametro de la fresa multiplicado por un factor (cercano al 50%) dando una medida de profundidad que esté cerca del centro de la fresa. Esto finalmente sirve para proyectar el centro del bounding box (u_c, v_c) hacia el espacio 3D usando el modelo de pinhole. Si la opción de *enforce_depth_constraint* está activada, la cual es el caso en los experimentos realizados, el sistema ancla el valor de la profundidad después de realizar PnP.

Como ejemplo, se tiene los siguientes resultados que se pueden encontrar en su formato completo en el Anexo A8:

media de profundidad: 0.314m

tvec_z ≈ 0.314m

T_cam_fruit [2, 3] = 0.33675m

Que refleja la combinación de profundidad directa y refinamiento geométrico por PnP.

Continuando con la transformación de la cámara a la brida del robot, este es determinado de forma fija y no obtenida


```

si no exito_mqtt:
    registrar_error("Fallo al cerrar gripper")
    abortar_ciclo_fresa()

```

Figura 61. Pseudocódigo describiendo el segmento de Pick.

D. Función de Post-pick

La función de post-pick incluye dos aspectos: maniobras de desprendimiento de la fresa de la planta seguido por el transporte y depósito de la fresa en el destino *place_final*, seguido de una verificación visual de éxito.

El destino de la fresa es un target fijo *place_final* en la estación RoboDK. Este target representa, por ejemplo, una charola o contenedor donde se colocan las fresas cosechadas.

La secuencia de movimientos posterior al pick (por fresa) es la siguiente:

1. Ejecutar la secuencia de post-pick generada automáticamente relativa a ${}^B T_{pick}$.
2. Movimiento *move_home_before_place* (joint) hacia *Home*: $\text{moveJ}({}^B T_{Home})$.
3. Movimiento *move_place* (linear) hacia *Home* (misma pose, usado como transición segura).
4. Movimiento *move_place_final* (linear) hacia *place_final*: $\text{moveL}({}^B T_{place_final})$.
5. Apertura del gripper en *place_final*.

La secuencia de *post pick* activa una secuencia de maniobras de desprendimiento tras cerrar el gripper. Estas maniobras son construidas por el método de *_execute_post_pick_sequence*. El cual toma como entrada la transformada ${}^B T_{pick}$, construyendo una secuencia de targets tomando esa pose de referencia, estos targets se construyen a partir de desplazamientos *offset_mm* y rotaciones *rotation_deg*, o bien incrementos de juntas *joint_deltas_deg*. Tomando estos dos modos:

Modo cartesiano (*offset_mm* / *rotation_deg*):

Para cada target k-ésimo:

1. Convertir offset (mm) a metros.
2. Dependiendo del modo de rotación:
 - a. En *absolute*: los ejes de rotación se definen en la base.
 - b. En *cumulative*: los ejes están en el marco actual del TCP (gripper).
3. Calcular la rotación incremental $R_{\Delta k}$ mediante Rodrigues.
4. Actualizar la pose actual $T_{current}$.

Modo espacio de juntas (*joint_deltas_deg*):

1. Se obtiene la configuración de juntas actual q_k desde RoboDK.
2. Se suma el vector de incrementos Δ_{q_k} (en grados).

3. Se manda un MoveJ en espacio articular hacia q_{k+1} ; la pose cartesiana resultante es ${}^B T_{post,k+1}$.

En ambos modos, la secuencia se ejecuta cuidando la evitación de colisiones según la configuración *collision_avoidance*.

Después de completar la secuencia de movimientos y de liberar la fresa en *place_final*, la verificación de éxito se realiza mediante visión en la función *_verify_pick_success()*:

1. El robot vuelve a Foto.
2. Se captura una imagen de verificación, y se llama al servicio de visión para obtener nuevas detecciones de fresas maduras.
3. Se cuentan las detecciones.

El criterio de éxito es:

$$\text{exito} \Leftrightarrow N_{ripe} \leq \max_ripe_berries = 0 \quad (12)$$

Si la verificación falla (es decir, quedan fresas maduras), el sistema permanece en modo continuo y reinicia el ciclo de captura desde Foto.

En la figura 62 se puede observar el condensado en estilo de pseudocódigo del segmento de *Post_pick* de la secuencia.

None

```

funcion ejecutar_funcion_post_pick(T_B_pick):
    # 1) Secuencia de desprendimiento
    si post_pick.enabled:
        T_current <- T_B_pick
        para k en 1..num_post_pick_targets:
            cfg <- post_pick.targets[k]
            si cfg.tiene_joint_deltas:
                q_current <- leer_juntas_robot()
                q_next <- q_current + cfg.joint_deltas_deg
                mover_juntas(q_next)
                T_current <- pose_desde_juntas(q_next)
            sino:
                o_k <- cfg.offset_mm / 1000
                R_delta <- Rodrigues(cfg.rotation_deg)
                (p_current, R_current) <- descomponer(T_current)

                # Aplicar offset y rotacion segun rotation_mode
                si rotation_mode == "absolute":
                    R_ref <- I
                sino:
                    R_ref <- R_current

                p_next <- p_current + R_ref * o_k
                R_next <- R_current * R_delta

                T_current <- [R_next, p_next]
                mover_a_pose(T_current, modo=cfg.move_type)

        # 2) Movimientos hacia el lugar de deposito
        mover_a_target("Home", modo="joint")
        mover_a_target("Home", modo="linear") # paso de transicion
        mover_a_target("place_final", modo="linear")

        # 3) Apertura de gripper con confirmacion MQTT
        exito_apertura <- activar_gripper("open", confirmar_usuario=true,
                                         esperar_confirmacion=true)
        si no exito_apertura:
            registrar_warning("Apertura de gripper no confirmada por MQTT")

        # 4) Verificacion visual de pick
        si verificacion.enabled:
            mover_a_target("Foto")
            exito_verif <- verificar_pick_por_vision()
            actualizar_estado(exito_verif)

```

Figura 62. Pseudocódigo describiendo el segmento de Post-pick.

E. Experimentos

Utilizando el Setup Experimental descrito a continuación en la sección VII, se realizaron todos los experimentos relacionados al sistema, con su gemelo digital construido en RoboDK para realizar simulaciones y realizar todos los movimientos de antemano para detectar y evitar posibles

colisiones. A lo largo de la secuencia de Figuras numeradas en serie desde la Figura 63 a la Figura 76, se puede observar cada paso de la secuencia descrita en esta sección tanto en el robot real como en el simulador, al igual de las imágenes visualizando lo que ve el sistema en el servicio de visión (Figura 65, 66, 75, 76). A lo largo del desarrollo del sistema esta secuencia pasó por varias iteraciones, eventualmente llegando a la observada, con pasos adicionales de seguridad y especialmente una nueva sección de *post_pick* calibrado específicamente en las capacidades del gripper blando neumático diseñado. El escenario aplicado en este experimento se considera uno de los más complejos dentro de los criterios de evaluación mencionados en la sección III-B, con extra complejidad en la abundancia de hojas presentes en espacio de movimiento del robot, aumentando el riesgo de enredarse y de oclusión de la fresa. Con esto se comprueba que la funcionalidad del sistema alcanza el estándar deseado y lo sobrepasa ligeramente.

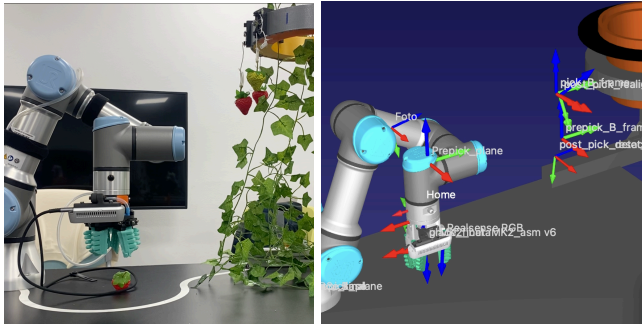


Figura 63. Paso 0 de la secuencia, pose de Home ${}^B T_{Home}$

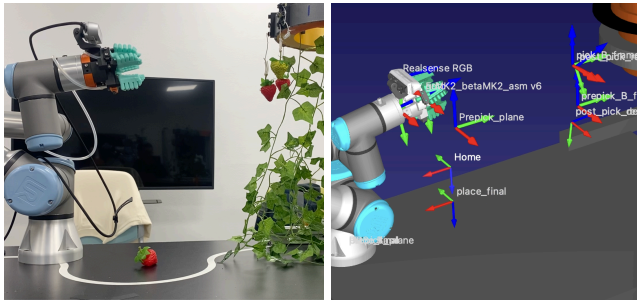


Figura 64. Paso 1, movimiento a pose de Foto ${}^B T_{Foto}$ (Standby).



Figura 65. Paso 2, llamado a servicio de visión: Captura, detección y cálculo de PnP.



Figura 66. Paso 2, con filtro de color en el mapa de profundidad.

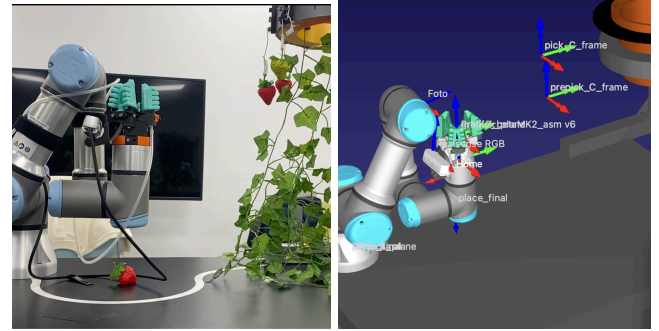


Figura 67. Paso 3, movimiento a *prepick_plane* ${}^B T_{prepick_plane}$

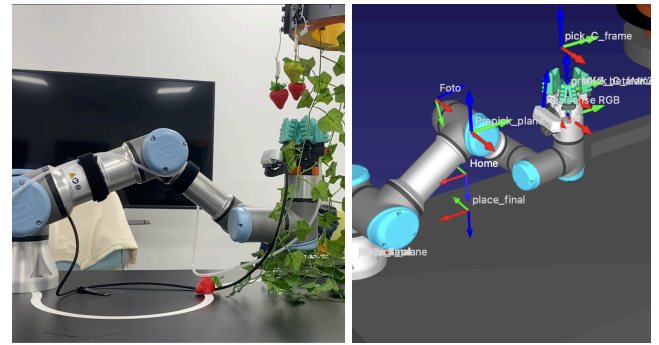


Figura 68. Paso 4, movimiento a *Pre-pick* ${}^B T_{prepick}$

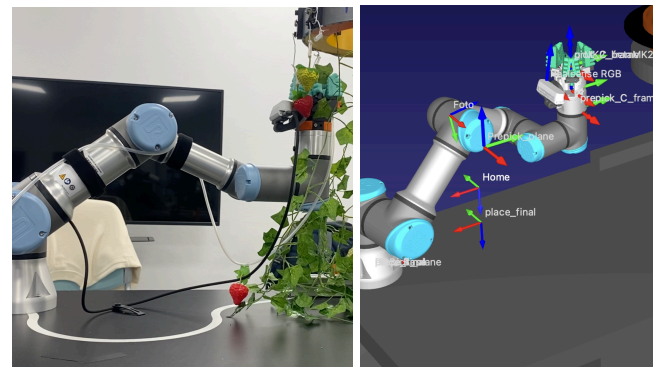


Figura 69. Paso 5, movimiento a *Pick* ${}^B T_{pick} = {}^B T_s$

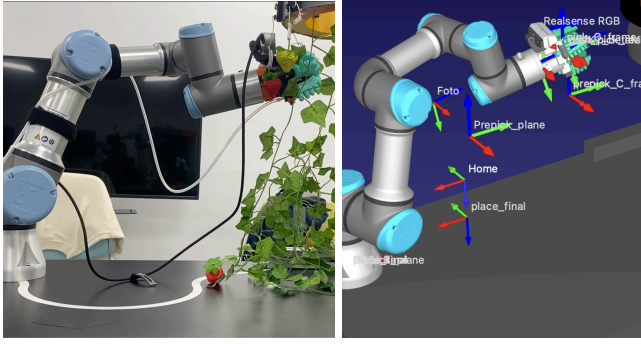


Figura 70. Paso 6, maniobra de *Post-pick* 1 (reorientación).

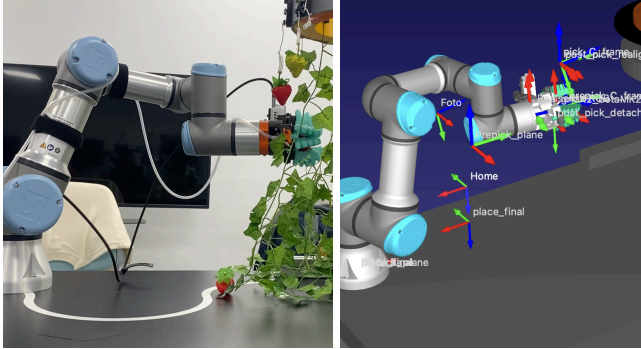


Figura 71. Paso 7, maniobra de *Post-pick* 2 (giro).

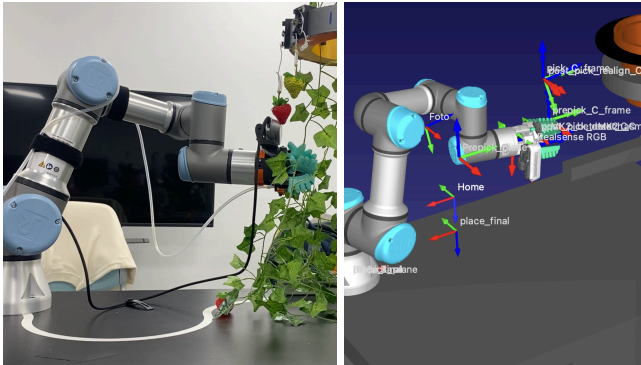


Figura 72. Paso 8, maniobra de *Post-pick* 3 (anti-giro).

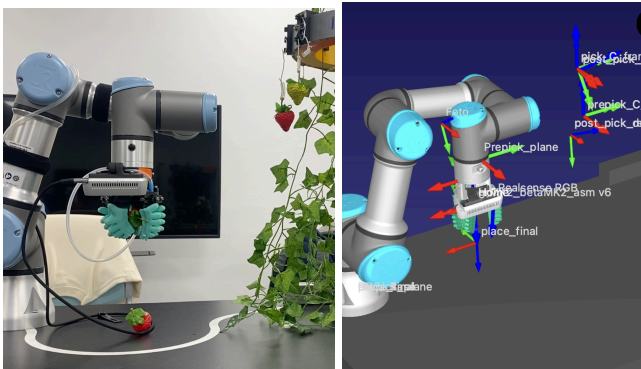


Figura 73. Paso 9, regreso a $Home$ ${}^{B}T_{Home}$ (*pre_place*).

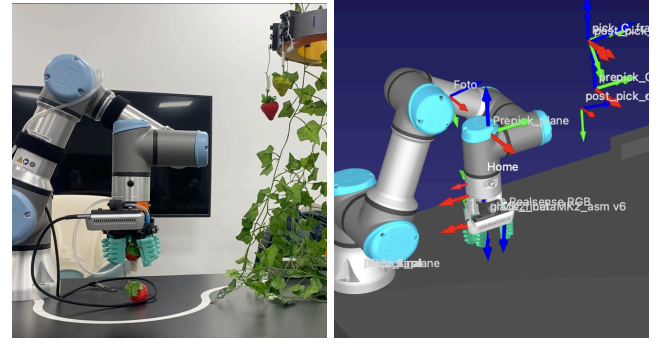


Figura 74. Paso 10, movimiento a $Place_final$ ${}^{B}T_{place_final}$.

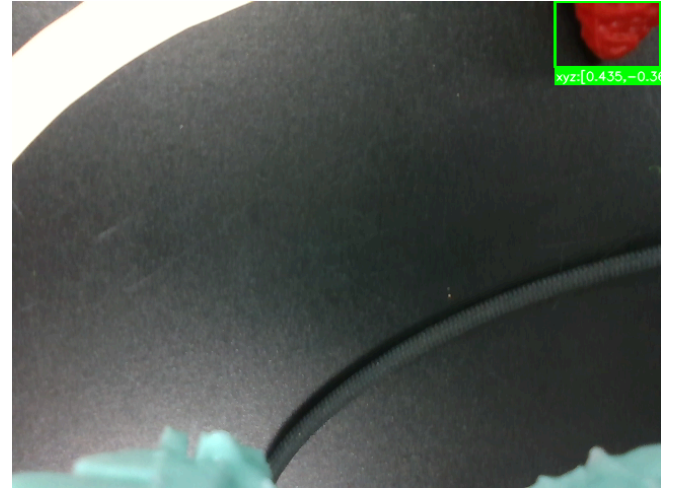


Figura 75. Paso 11, captura y detección para el primer paso de validación.



Figura 76. Paso 12, captura y detección para el segundo paso de validación.

VI. PLANEACIÓN DE TRAYECTORIAS CON EL ROBOT UR3

A. Parámetros de configuración del UR3

Varias de las configuraciones del UR3e fueron definidas dentro del archivo de configuraciones YAML del sistema principal del robot para la experimentación rápida y facilidad de configuración, además de centralizar lo más posible estos elementos. Estas configuraciones se aplican tanto en RoboDK como el robot UR3:

- TCP: $tvec = [0.0, 0.0, 57]$ con respecto a la brida del robot.
- Velocidad máxima:
 - Linear: 80% del máximo.
 - Joint: 50% del máximo.
 - Aceleración: 50% del máximo.
- Límites de los ejes:

- $\theta_1 = -360^\circ$ a 360°
- $\theta_2 = -180^\circ$ a 20°
- $\theta_3 = -360^\circ$ a 360°
- $\theta_4 = -360^\circ$ a 360°
- $\theta_5 = -360^\circ$ a 360°
- $\theta_6 = -2160^\circ$ a 2160°
- Con el paro de emergencia siendo activable con el Teach Pendant del UR3 o con Ctrl+C al programa controlando el robot.
- El payload es configurable solamente a través del Teach Pendant del UR3, configurado a 0.350kg que es la masa de nuestra herramienta manipuladora incluyendo la cámara.

B. Planeación de trayectorias basada en visión

El procesamiento y consecuente generación de trayectoria se describe a detalle en la Sección V. A continuación se observa una tabla condensando cada paso de la trayectoria como se describe en la Sección V. Las velocidades y aceleraciones para cada movimiento son los valores máximos definidos en el punto VI.A. De parte del blend, este parámetro no fue utilizado en ningún paso de esta trayectoria.

TABLA IV. PLANEACIÓN DE TRAYECTORIAS

Pasos de la trayectoria planeada del robot			
WP	Tipo de movimiento	Posición referencial	Descripción
Home	MoveL	Base	Posición segura cercana a la base del robot. Se considera este punto como OFF.
Foto	MoveJ	Base	Posición fija para la captura de foto.
Prepick_plane	MoveJ	Base	Posición fija orientando el robot al "plano" estandar de movimientos Prepick.
Prepick	MoveL	Frame del objeto	Posición con un offset de 100mm en el eje Y respecto a Pick.
Pick	MoveL	Frame del objeto	Pose de la fresa obtenida por PnP.
Postpick_k_Orient	Joint Specific Offset	~	Re-orienta la herramienta para un mejor agarre de la fresa.
Postpick_k_rotate	Joint Specific Offset	~	Giro para despegar la fresa.

Pasos de la trayectoria planeada del robot			
WP	Tipo de movimiento	Posición referencial	Descripción
Postpick_k_unrotate	Joint Specific Offset	~	Giro para revertir el giro previo.
Place_final	MoveL	Home	Movimiento para colocar la fresa

C. Análisis de seguridad conforme a la norma ISO 10218-2:2025

Al analizar la estación y tomando en cuenta lo mencionado en la norma ISO 10218-2:2025 se alcanzó a observar que pueden existir riesgos mecánicos ya que se pueden producir movimientos inesperados y colisiones. Esto se trata de evitar al primero realizar la simulación del movimiento del robot en RoboDK. Una vez confirmado que en el movimiento a realizar no existe alguna colisión, el robot físico realiza este movimiento.

También se encontró que debido a la vibración de la bomba neumática existían riesgos por vibraciones ya que tanto la mesa como el setup se veían afectados. Estos riesgos por vibraciones fueron reducidos al conservar todos los componentes del sistema de actuación dentro de una caja de MDF. Las vibraciones aún existen, sin embargo, ya no se considera que sea un riesgo ya que la mesa y el setup no se ven afectados al guardar el circuito en la caja.

Lo mencionado anteriormente se visualiza en la tabla 4 en donde se describe el riesgo, la situación y las medidas de mitigación aplicadas.

TABLA V. NORMA ISO 10218-2:2025

Riesgos en el sistema			
Riesgo Identificado	Descripción	Medida de mitigación	Referencia ISO 10218-2:2025
Movimientos inesperados Colisiones con setup	Debido a que se utiliza la cinemática inversa evaluada por Robo DK para llegar a las poses de prepick y pick el robot puede realizar movimiento no deseados, incluso llegando a colisionar con el setup	Se evalúa el movimiento realizando una simulación en Robo DK. Una vez que se verifica que no existen colisiones el robot físico realiza este mismo movimiento.	5.4.7.1 Limiting motion GENERAL
Movimiento de la estación por vibraciones	Debido a la vibración ocasionada por la bomba del sistema electroneumático, se genera	Almacenar el circuito dentro de una caja protectora, evitando así el contacto directo entre la	4.3.2 Risk assessment for contacts between moving parts of the robot

Riesgos en el sistema			
Riesgo Identificado	Descripción	Medida de mitigación	Referencia ISO 10218-2:2025
	un movimiento que afecta la estabilidad de la mesa en donde se encuentra el sistema, setup y cobot.	bomba y la mesa reduciendo las vibraciones existentes.	application and operator(s)

VII. INTEGRACIÓN DEL SISTEMA AUTOMATIZADO DE COSECHA DE FRUTOS DELICADOS

En la Figura 77 se puede observar un diagrama que ilustra la arquitectura del sistema, entrando a detalle de los módulos, el denominado “Ciclo de Control” es el encargado de controlar cada paso del robot y de los módulos auxiliares (denominados servicios), este módulo central se enlaza directamente con RoboDK para controlar el robot en simulación, donde el RoboDK igualmente sirve de intermediario con el robot real. Para todas las pruebas realizadas, se utilizó un *Setup Experimental* el cual se menciona en la Sección VI. Este *Setup* fue originalmente diseñado para ser utilizado con una maceta de fresas por ende tiene un diseño de anillo, para poder realizar experimentos repetitivos, se modificó el Setup para utilizar fresas falsas emulando el posicionamiento, forma y resistencia de las reales. En todo el desarrollo y experimentaciones se utilizó esta alternativa. Dentro del Setup se establecieron tres escenarios de pruebas de complejidad variable como se listan a continuación:

1. Fresa individual lista para cosechar, totalmente visible y sin ruido cercano.
2. Fresa lista para cosechar con una fresa no madura en contacto, con no más del 20% de oclusión.
3. Fresa madura tapada por lo menos 40% por hojas.

Estos tres escenarios fueron comprobados y experimentados de forma simultánea en la gran mayoría de las pruebas como se describe en la Sección V.E, observado en las Figuras 63 al 76. Para todos los experimentos se define como exitosa la prueba al cosechar la fresa y depositarla en el lugar definido sin que se escape durante los movimientos y sin causar daños al resto de las fresas y escenario, se contempló que daños mínimos a las hojas es inevitable por la alta densidad que se decidió colocar.

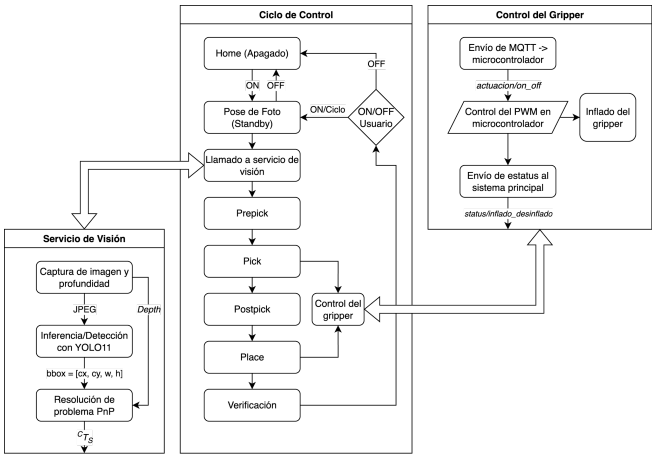


Figura 77. Diagrama ilustrando la arquitectura del sistema automatizado de cosecha.

VIII. CONCLUSIONES

El proyecto logró integrar de manera funcional un sistema completo que combina monitoreo ambiental, visión artificial y un gripper neumático suave, demostrando que es posible automatizar tanto la supervisión del cultivo como la manipulación delicada de fresas mediante herramientas de bajo costo y alta adaptabilidad. A pesar de estos avances, surgieron limitaciones relevantes, como fallas en la calibración de sensores, pérdida de datos por problemas en la Raspberry Pi, errores del modelo de visión ante oclusiones y fondos similares al fruto, así como variaciones en la manufactura del gripper que afectaron su sellado y consistencia mecánica. Estos hallazgos evidencian áreas de mejora que orientan el trabajo futuro, el cual debe enfocarse en reforzar la robustez del hardware y los mecanismos de respaldo, ampliar y depurar el dataset de entrenamiento, optimizar los procesos de fabricación del actuador blando e implementar técnicas de control más precisas y adaptativas para garantizar un desempeño confiable en escenarios reales de operación agrícola.

Al inicio del proyecto establecimos el siguiente objetivo: “En 16 semanas instalar un sistema automatizado y funcional de recolección de fresas, con un porcentaje de éxito mayor al 75% en la identificación y recolección de frutos en un estado específico de maduración”. Con el paso del tiempo, el trabajo realizado y el esfuerzo invertido, logramos cumplir dicho objetivo. Gracias a que el propósito estuvo bien definido desde el principio, no fue necesario realizar ajustes significativos al plan del proyecto. Si bien las herramientas de gestión empleadas fueron fundamentales para distribuir actividades y administrar de manera ordenada el avance del proyecto, reconocemos que fue necesario dedicar más atención a esta área. Un mayor énfasis en la gestión y en el uso adecuado de estas herramientas habría permitido optimizar la organización interna y mejorar la comunicación dentro del equipo.

Continuando con el gripper, aunque la idea inicial se pudo transmitir al diseño final, se noto que no fue un diseño ideal o por lo menos no era uno completo. Durante las múltiples rondas de experimentos se pudo identificar puntos débiles del diseño, varios que se resolvieron al rediseñar y otros que se resolvieron con otros elementos del proyecto, específicamente con la rutina modificada del robot para acoplar las debilidades del gripper. Al igual, se identificó la

falta de fundamento y experiencia con esta clase de grippers blandos neumáticos ya que varios de los problemas surgieron desde la raíz del diseño. Aunque se menciona todo esto, resultó funcional el gripper pero con muchas áreas de mejora, se pudo observar el potencial que tiene al refinarlo más.

Al igual del gripper, en la implementación del sistema de visión nos desviamos del camino regular, experimentando con diferentes poses de la cámara que dio altos beneficios en contraste con la competencia aunque por el otro lado se tuvo que considerar de otra forma elementos como la medición de profundidad. Por lo general, el resultado del sistema implementado fue positivo al tener un rango de eficacia más grande y de forma más simple. Además el sistema totalmente configurable que se desarrolló dio lugar a mejores experimentaciones y facilidad en la búsqueda del mejor resultado.

Por otro lado, el sistema de detección y clasificación de fresas dio resultados promedios, con áreas amplias de oportunidad que se conocían desde el inicio del desarrollo. La versión utilizada del modelo entrenado se identificó como débil en varios aspectos pero suficientemente robusto para la aplicación dada. Por el método utilizado de haber entrenado el modelo con fotos no específicas del escenario utilizado, se supo de la debilidad que tenía el modelo, aunque por el otro lado, el tiempo ahorrado al utilizar este método fue valioso y comprobó que no es totalmente necesario generar un dataset totalmente personalizado.

Finalizando con el robot UR3e y el Setup Experimental, en muchos aspectos no es lo ideal, ya que la combinación de cómo fueron integrados estos dos no reflejan su utilidad en la industria, dejando los resultados como prueba de concepto. A grandes rasgos, una mejora para esto sería el cambio del Setup a replicar aquellos vistos en cultivos modernos controlados, con el robot UR3 movilizado en el espacio. Enfocando en el robot UR3, este da buenos resultados al tener 6-DOF pero en términos de alcance se vió altamente limitado.

IX. CONTRIBUCIÓN INDIVIDUAL AL PROYECTO

1. Aldrick Tadeo Arellano

Dentro del equipo de trabajo estuve encargado en varios elementos diversos del proyecto, siendo el responsable de los siguientes aspectos:

1. Diseño del gripper neumático blando, al igual del resto de la herramienta manipuladora. *Ver Sección IV.B.1, IV.B.2, Figura 29, Figura 30, Figura 31 y Anexo A7.*
2. Diseño del acoplador universal para acoplar las herramientas manipuladoras y el UR3e, *visto en Sección IV.B.3, Figura 32, Anexo A12.*
3. Diseño de los moldes para la manufactura de los grippers/actuadores neumáticos blandos. *Ver Sección IV.B.4, Figura 33, Figura 34, Anexo A13.*
4. Implementación del sistema de detección de fresas, incluyendo recopilación del Dataset, entrenamiento del modelo, *fine-tuning*, análisis e implementación como módulo del servicio de visión. *Ver Sección III.B.*
5. Creación del sistema de visión (denominado servicio de visión) incluyendo detección de fresas (ver entrada anterior (IX.1.4)), módulo de medición

de profundidad calibrado y filtrado y módulo de resolución de problema de PnP. *Ver Sección V.*

6. Desarrollo de la rutina completa del robot, incluyendo enlace con gemelo digital en RoboDK. *Ver Sección V, VI, VII y Figura 62 a Figura 75.*
7. Implementación de sistema de prevención de colisiones, con *pre-check* de cada movimiento en simulación (RoboDK). *Ver Sección V, Sección VI.*
8. Verificador de colecta correcta de fresas, incluyendo registro persistente de intentos fallidos e intentos exitosos. *Ver Sección V.*
9. Diseño de arquitectura del sistema de cosecha automatizado e integración de módulos. *Ver Sección VII.*
10. Programación del Firmware utilizado en el sistema de monitoreo al igual de integración adecuada y recalibración de sensores. *Ver Sección III.A.*
11. Diseño y ensamble del circuito de control neumático en formato de PCB. *Ver Sección IV.G, Figura 55 y Figura 56.*

A lo largo de las responsabilidades y tareas que completé tuve la oportunidad de desarrollar mis habilidades en una gran variedad de áreas y reforzar otras al mismo tiempo. De los retos y por ende aprendizajes más grandes que tuve fue el desarrollo del sistema de visión integrado con el sistema del robot, la unión de tantos sistemas en paralelo y que estén en función dio lugar a aprender a realizar una planeación de gran escala, implementando todo tipo de subsistema que ayudaban en proceso de desarrollo y de experimentación. Por las limitaciones inherentes de RoboDK me quede con mucho por querer experimentar y desarrollar con el robot y el resto del sistema en otro tipo de frameworks como es ROS2. En general fue una buena experiencia poder tener la oportunidad de desarrollar tantos elementos e integrarlos de forma satisfactoria pero faltó tiempo para poder profundizar en algunos aspectos, tiempo que se desperdició en implementaciones de sistemas muy alejados del *scope* de valor del proyecto como fue el cuidado y monitoreo de los cultivos de fresas.

2. Yael Cortés Rubio

A lo largo del proyecto, fui responsable de diversas tareas que contribuyeron al cumplimiento de los objetivos generales. Entre las actividades realizadas son:

1. Administración del proyecto, backlog, diagrama de Gantt.
2. Programación de la Raspberry PI 5, creación de la base de datos para almacenar los datos de los sensores que monitorean las fresas, así mismo el código para la toma de fotos de las plantas y envío de alertas vía telegram. Así mismo la creación de un dashboard para la visualización de datos en gráficas y en rango de fechas, alertas, entre otras cosas.
3. Manufactura de actuadores.
4. Pruebas de actuadores.
5. Ensamblado de Gripper.
6. Dashboard para visualizar datos del sistema de visión junto con los movimientos del robot (Logs, secuencia y estatus del robot, imagen en tiempo real tomada por la cámara).
7. Base de datos para almacenar imágenes.

Durante el desarrollo del proyecto llevamos a cabo diversas actividades. Entre ellas destacaron el uso de RoboDK, la implementación de la red neuronal YOLO 11 y el aprendizaje relacionado con componentes electrónicos, además del estudio teórico necesario para ejecutar las tareas técnicas, como la calibración de la cámara. Cada una de estas actividades contribuyó de manera significativa al funcionamiento integral del sistema automatizado. Lo que más me motivó y disfruté a lo largo del proyecto fueron las habilidades que logré adquirir y el nuevo conocimiento que desarrollé. Además, el acompañamiento constante tanto de los profesores como de mis compañeros de equipo fue fundamental para superar los distintos retos que se presentaron. Este apoyo colectivo fortaleció mi proceso de aprendizaje y mejoró la calidad del trabajo realizado. Entre los conocimientos más valiosos que obtuve se encuentran la comprensión del funcionamiento de un sistema neumático y la integración de sus componentes, el manejo y comportamiento de los dispositivos electrónicos utilizados, los principios de manufactura aplicados a la creación de *soft grippers*, las conexiones mediante el protocolo MQTT y el entendimiento general del funcionamiento de las redes neuronales. Finalmente, me hubiera gustado profundizar aún más en la teoría relacionada con el movimiento del robot, mejorar mi dominio del uso de RoboDK y ampliar mis conocimientos sobre los procesos de entrenamiento en sistemas de visión artificial. Considero que fortalecer estas áreas habría potenciado aún más mi desempeño dentro del proyecto.

3. Emiliano González González

Durante la realización del proyecto estuve involucrado en distintas tareas tomando diferentes roles, asumiendo roles de liderazgo y apoyo con el objetivo de cumplir con las metas establecidas dentro del equipo.

Actividades de las cuales fui responsable:

1. Simulaciones FEM: Realización de simulaciones para verificar el comportamiento del actuador. Así como el análisis e interpretación de los resultados obtenidos.
2. Caracterización, implementación física y programación del circuito electroneumático para lograr la activación y control deseado en el actuador suave.
3. Caracterización e implementación de sensores para el sistema de monitoreo de las fresas.
4. Diseño e implementación del setup en donde se encontraron las fresas para su recolección.
5. Creación del gemelo digital, estableciendo poses de inicio y final.
6. Implementación de comunicación MQTT entre circuito electroneumático y gemelo digital para lograr la rutina de movimiento.
7. Verificación y análisis de riesgos existentes según la norma ISO 10218-2:2025

Algunas actividades en las que di apoyo fueron:

1. Apoyo en definición de poses para el desarrollo de la rutina del robot para la recolección de fresas.
2. Apoyo en la comunicación entre el gemelo digital y el sistema físico.

Un reto que enfrente durante estas semanas fue la realización de simulaciones FEM, ya que ANSYS al ser un software muy específico fue un poco difícil aprender a manejarlo y la mayoría de las veces no funcionaba de la manera correcta en mi propia computadora por lo que algunos días debía estar horas en los salones del CIMA ya que también el software es un poco tardado. Sin embargo, el lograr estas simulaciones así como el análisis de todos los resultados obtenidos también fue una parte gratificante en este proyecto. Otro aspecto que me resultó muy gratificante al verlo completado fue la correcta comunicación y funcionamiento entre el gemelo digital y el sistema electroneumático ya que logre aplicar físicamente un control PI que antes solo había realizado su análisis de manera teórica. Otro aprendizaje que me llevo de este proyecto fue la aplicación y manufactura de *soft grippers*, se me hizo muy interesante todas las aplicaciones que se le puede dar a esta rama de la robótica y las diferentes maneras en las que se realizan. Por último, un área que me hubiera gustado aprender es la parte del entrenamiento de la red neuronal, ya que me parece muy interesante como con solo imágenes se puede entrenar un sistema para que tome decisiones en tiempo real, sin embargo, por mis conocimientos y el tiempo no pude involucrarme mucho en este aspecto.

AGRADECIMIENTOS

Queremos agradecer a los profesores que nos guiaron a lo largo del semestre:

Christopher Diego Cruz Ancona, Xóchitl Yamile Sandoval Castro, Alfonso Gómez Espinosa, Iván Fernando Hernández Araujo, Ivo Neftali Ayala García y Alejandro González de Alba.

Su guía y apoyo fue de alta importancia y un gran beneficio para el logro realizado en este proyecto. Muchas gracias.

REFERENCIAS

- [1] N. Cruz Santos, "Caracterización de elastómeros líquidos para análisis de deformaciones en actuadores inflables elásticos, a través de la evaluación de modelos hiperelásticos", Tesis de Maestría, Instituto Politécnico Nacional, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Unidad Querétaro, México, 2019. [En línea]. Disponible en: <https://tesis.ipn.mx/handle/123456789/30307>
- [2] CFSensor, "XGZP6847A Pressure Sensor (Data Sheet) V2.5," Calibrated Pressure Sensor Series, High-Tech Park, Wuhu P.R.C., Dec. 2022. Available: <https://empere.in/wp-content/uploads/2023/02/XGZP6847A-Pressure-Sensor-V2.5.pdf>
- [3] Ultralytics, "Ultralytics YOLO11", Ultralytics YOLO Docs, 23 de noviembre de 2025. <https://docs.ultralytics.com/models/yolo11/#performance-metrics>
- [4] B. Estabrook, "Farmers Can't Find Enough Workers to Harvest Crops-and Fruits and Vegetables Are Literally Rotting in Fields," *EatingWell*, Dec. 9, 2020. Accessed: Nov. 26, 2025. <https://www.eatingwell.com/article/291645/farmers-cant-find-enough-workers-to-harvest-crops-and-fruits-and-vegetables-are-literally-rotting-in-fields/>
- [5] T. Delbridge, "Robotic Strawberry Harvest Is Promising but Will Need Improved Technology and Higher Wages to Be Economically Viable," *California Agriculture*, vol. 75, no. 2, pp. 57–63, 2021. <https://doi.org/10.3733/ca.2021a0009>.
- [6] YEA, "yea_strawberry_class_2dbbox_aggr-1vnjy_dataset", yea_strawberry_class_2dbbox_aggr-1vnjy_dataset. Roboflow, octubre de 2025. [Roboflow Universe]. Disponible en: https://universe.roboflow.com/enceladus/yea_strawberry_class_2dbbox_aggr-1vnjy

- [7] Beyondmetis, “rs2_intrinsic coeffs[] all 0 by default”, GitHub, 25 de marzo de 2018.
<https://github.com/realsenseai/librealsense/issues/1430>
- [8] “OpenCV: Perspective-n-Point (PnP) pose computation”.
https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html
- [9] Wikipedia contributors, “Denavit–Hartenberg parameters”, Wikipedia, 8 de septiembre de 2025.
https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters
- [10] “RoboDK API for Python — RoboDK API Documentation”.
<https://robodk.com/doc/en/PythonAPI/index.html>
- [11] “OpenCV: Camera Calibration and 3D Reconstruction”.
https://docs.opencv.org/4.x/d9/d0c/group_calib3d.html
- [12] “Universal Robots - DH Parameters for calculations of kinematics and dynamics”.
<https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/>
- [13] M. S. Xavier et al., “Soft Pneumatic Actuators: A Review of Design, Fabrication, Modeling, Sensing, Control and Applications”, IEEE Access, vol. 10, pp. 59442-59485, ene. 2022, doi: 10.1109/access.2022.3179589.
- [14] J. Shintake, V. Cacucciolo, D. Floreano, y H. Shea, “Soft robotic grippers”, Advanced Materials, vol. 30, n.o 29, p. e1707035, may 2018, doi: 10.1002/adma.201707035.

ANEXOS

Todos los Anexos se localizan en la carpeta de Teams del proyecto ANEXOS-YEA. Al igual que en su respaldo en Drive y Github. En cada Anexo específico referenciado se proporciona su enlace específico y ubicación dentro de la carpeta de Teams. Si no hay acceso a Teams, dirigirse al respaldo público en Github.

Enlace a la carpeta principal en Teams:

<https://teams.microsoft.com/l/channel/19%3A9a08750f724542f9abe3a579d3d81efe%40thread.tacv2/YEA?groupId=dbb697b5-36a5-4dc0-a890-eacc4ea375c0&tenantId=c65a3ea6-0f7c-400b-8934-5a6dc1705645&ngc=true>

Enlace al repositorio de Github:

https://github.com/YaoCr003/pickafresa_yea

Enlace al la carpeta de respaldo en Drive:

<https://drive.google.com/drive/folders/1eUh4Sc1Ob7JIUvLUGUgkzJ8xAvZ-xoe-h?usp=sharing>

[A1] control_mqtt.ino: Código del controlador para circuito electroneumatico;
https://tecmx.sharepoint.com/u:/s/Course_mr3001c_101-gFDloj95AfLCq-YEA/EZXPRTEDOVFKuKJEgN3qZwgBzHD49fvqKZr_Z69I6-5mDO?e=gb47hX

[A2] Tabla reporte.xlsx: Excel en donde se analizó los resultados de deformación total del gripper con respecto al tiempo.

https://tecmx.sharepoint.com/x:/s/Course_mr3001c_101-gFDloj95AfLCq-YEA/EdJ3wcqKollMqPyY7_RkdFOB7iBCzqX6FWR7gV4TmtXjO?e=E2Kbn

[A3] Excel control.xlsm: Archivo excel en donde se registraron y obtuvieron los resultados de la caracterización de la bomba.
https://tecmx.sharepoint.com/x:/s/Course_mr3001c_101-gFDloj95AfLCq-YEA/EYux-EPTZMFJs_RDnHOuimEBX7ekpR0MSdWUKj35WpmbOg?e=FkRb2n

[A4] Tablas deformación.xlsx: Archivo en donde se registraron los resultados de deformación obtenidos por ANSYS.
https://tecmx.sharepoint.com/x:/s/Course_mr3001c_101-gFDloj95AfLCq-YEA/ESzKQgz01-NEiwcKexQFqIB9_sEWgKHUPfm7oVaiCsb9A?e=tlxvtI

[A5] FEM propuesta 6.wbpj: Archivo ANSYS de la ultima simulación FEM realizada.

https://tecmx.sharepoint.com/u:/s/Course_mr3001c_101-gFDloj95AfLCq-YEA/EfOnUjUHixdJOn0Y7eqGe9IsBtsiWB_d14NwMgl9S_nxCXg?e=b9HmbS

[A6] actuator_arrowhead_hug_drawing v5.pdf; Gripper Final: Dibujo Técnico;

https://tecmx.sharepoint.com/b:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/actuator_arrowhead_hug_drawing%20v5.pdf?csf=1&web=1&e=s6bfln

[A7] electropneu_r4.zip; Archivos del PCB;
https://tecmx.sharepoint.com/u:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/electropneu_r4.zip?csf=1&web=1&e=Ejf9TG

[A8] 20251203_165826_data.json; Vision Service JSON output example;
https://tecmx.sharepoint.com/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/lavouts/15/download.aspx?Uniqueld=11fee57113f2412a83c80cf45bfa1aff&e=fmoCd8

[A9] robot_pnp_config.yaml; robot_pnp_cli YAML config;
https://tecmx.sharepoint.com/u:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/pickafresa_robot/configs/robot_pnp_config.yaml?csf=1&web=1&e=zksIOz

[A10] grMK2_betaMK2_asm v6.step; Diseño del gripper MK2 ensamblado;

https://tecmx.sharepoint.com/u:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/Dise%C3%B1o%20Gripper/grMK2_betaMK2_asm%20v6.step?csf=1&web=1&e=cPJ9Gu

[A11] /Diseño Gripper/; Carpeta de diseños;
https://tecmx.sharepoint.com/f:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/Dise%C3%B1o%20Gripper?csf=1&web=1&e=aj9R0z

[A12] ur3e_tool_coupler_slim_multi.step; Diseño del acoplador universal;

https://tecmx.sharepoint.com/u:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/Dise%C3%B1o%20Gripper/ur3e_tool_coupler_slim_multi.step?csf=1&web=1&e=77h1Mz

[A13] actuator_arrowhead_hug_mold_mk1.f3z; Diseño de los moldes de gripper;

https://tecmx.sharepoint.com/u:/r/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/YEA/ANEXOS-YEA/Dise%C3%B1o%20Gripper/actuator_arrowhead_hug_mold_mk1.f3z?csf=1&web=1&e=K1h7hR

[A14] Imágenes diagramas fabricación de actuador y ensamble de Gripper completo

https://tecmx.sharepoint.com/sites/Course_mr3001c_101-gFDloj95AfLCq-YEA/Shared%20Documents/Forms/AllItems.aspx?csf=1&web=1&e=HnlRdg&CID=cb70dd08%2D2848%2D478e%2D81ab%2D849ca1510f75&Fol derCTID=0x01200052568ABFA22A894CB73FF165C0E5FBB9&id=%2F sites%2FCourse%5Fmr3001c%5F101%2DgFDloj95AfLCq%2DYEA%2F Shared%20Documents%2FYEA%2FANEXOS%2DYEA%2FManufatura %20Gripper