

Implementación de un sistema de visión por computadora integrable como módulo de ROS a una plataforma robótica móvil

Trabajo de graduación
María Alejandra Samayoa Gómez

Resumen

- Instalación de sistema operativo
- Comparación de cámaras y selección de módulo
- Pruebas con Microsoft Kinect
- Adaptación a ROS
- Características de la cámara
- Integración a la plataforma robótica móvil

Objetivos

Objetivo General

Implementar un sistema de visión por computadora que pueda integrarse como módulo de ROS a una plataforma robótica móvil.

Objetivos Específicos

1. Evaluar y comparar diferentes módulos de cámara para definir el más adecuado para la plataforma robótica móvil.
2. Adaptar el sistema de visión por computadora con la cámara seleccionada como módulo de ROS, para su integración a la plataforma robótica móvil.

Objetivos Específicos

3. Adaptar la cámara Kinect como módulo de ROS, para su integración a la plataforma robótica móvil.
4. Realizar pruebas básicas de algoritmos de visión por computadora para validar el sistema implementado.

Introducción

Visión por Computadora



Campo de la inteligencia artificial que es un compendio de algoritmos que interpretan los datos de un módulo de visión física. [1]

Detección de Módulos ArUco

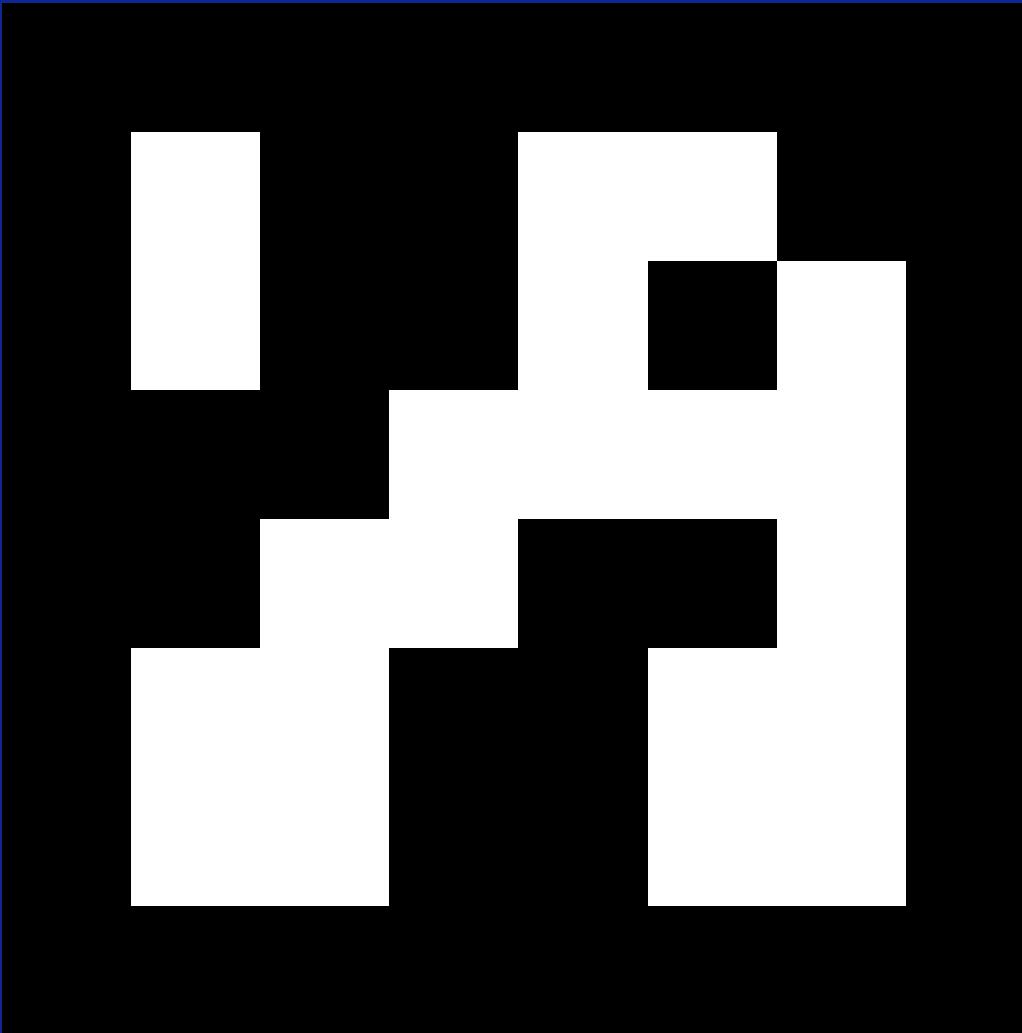
¿Qué son los marcadores?

- Marcadores binarios fiduciales.
- Bordes negros y matriz interior que define identificación.

Proceso de detección

- Thresholding adaptivo.
- Descartar bordes convexos/ figuras no cuadradas.
- Transformación para obtener matriz interna.
- Se determina si la identificación pertenece a un diccionario.

[2]



Módulos de cámaras

Raspberry Pi Camera

- Módulo oficial de cámara para Raspberry Pi.
- 8 Megapixeles de resolución.
- Habilidad de tomar fotos o videos de alta definición.
- Compatibilidad con librerías de Open CV, Python, Raspbian, etc. [3]

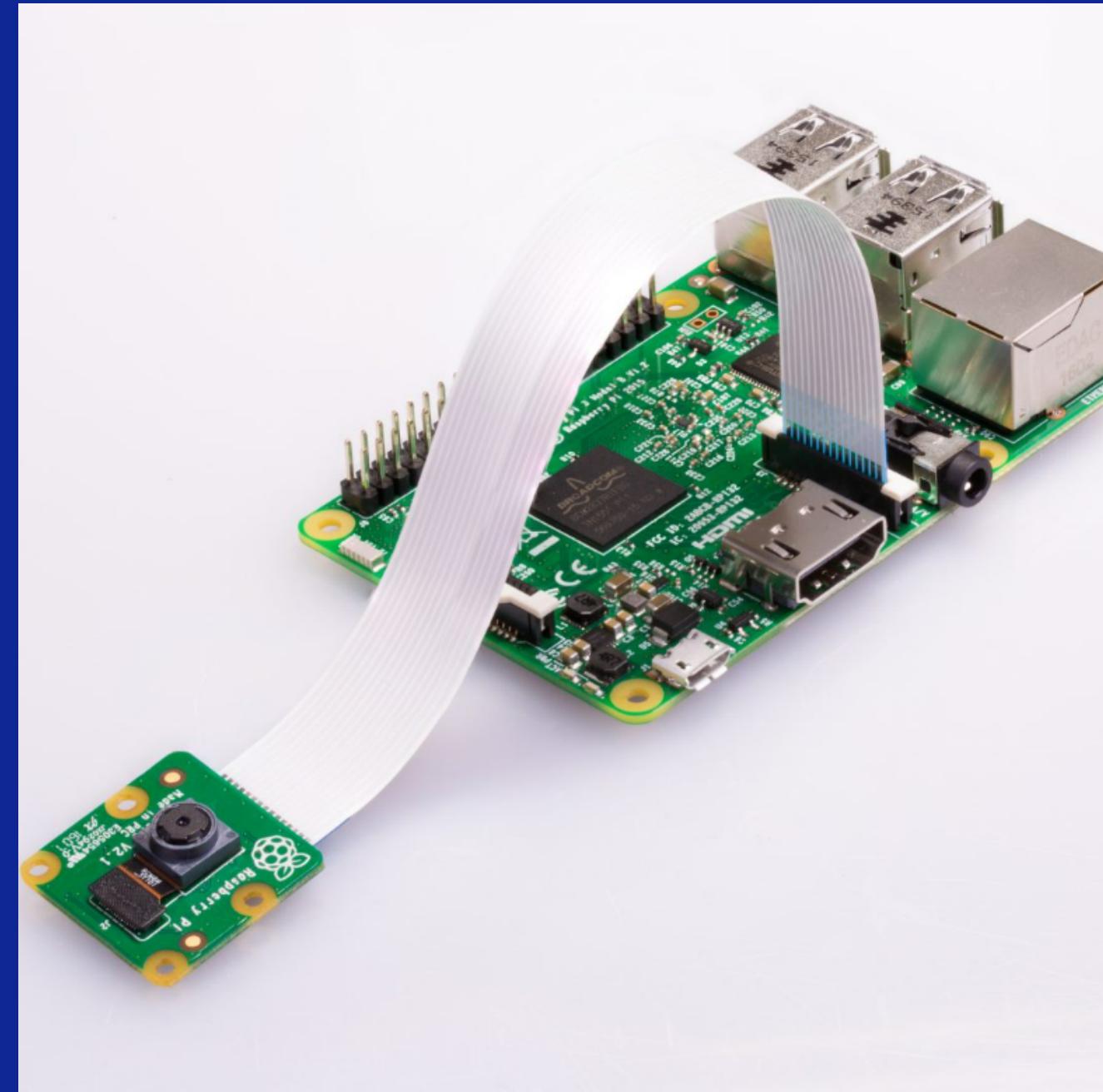


Figura: Cámara Raspberry Pi.

JeVois Smart Vision CameraA33



- CPU de 4 núcleos capaz de correr una instalación ligera de Linux.
- Cuenta con varios módulos de visión por computadora preinstalados.
- Fotos y videos con alta definición.

[4]

Figura: Cámara JeVois.

Microsoft Kinect para Windows

- Módulo de detección de movimiento de Microsoft.
- Cuenta con sensores de cámaras de profundidad, un sensor de cámara RGB, micrófonos para captura de audio y motores para movimiento del sensor.
- Ha aumentado su uso como módulo de cámara para visión por computadora.
- Perdió soporte oficial para los drivers a partir de Windows 8, ahora se usa Open Kinect. [5]



Figura: Microsoft Kinect.

Antecedentes

- Visión por computadora

Visión por computadora aplicada para robótica de enjambre sobre plataforma para identificación de agentes. Se utilizaron las librerías de OpenCV y programas de Python y Matlab. [6]

Estudio comparativo entre diferentes módulos de visión para un sistema embebido. Se compararon las cámaras ESP32 y la JeVois Smart Vision Camera. [7]

Antecedentes

- Plataforma robótica móvil
Control remoto por bluetooth, uso de sensores proprioceptivos y servidor de visión con la Raspberry Pi Cam.

[8]



Figura: plataforma robótica realizada por [8].

Metodología

Experimento

- Identificación de estaciones por medio de códigos ArUco

Plataforma

- Raspberry Pi 4
- ROS 2 Foxy Fitzroy

Instalación del sistema operativo

Consideraciones

- Sistema ROS 2 FoxyFitzRoy
- Linux Ubuntu 20.04 por el sistema de ROS
- Raspberry Pi 3/4.

Ubuntu Mate

- Para la Raspberry Pi 3.
- Sobre Ubuntu Server se descargó la interfaz gráfica de Ubuntu Mate.
- Interfaz ligera y rápida.
- Instalación de ROS2, Python, editores de texto, etc.



Figura: Pantalla principal de la instalación Ubuntu Mate.

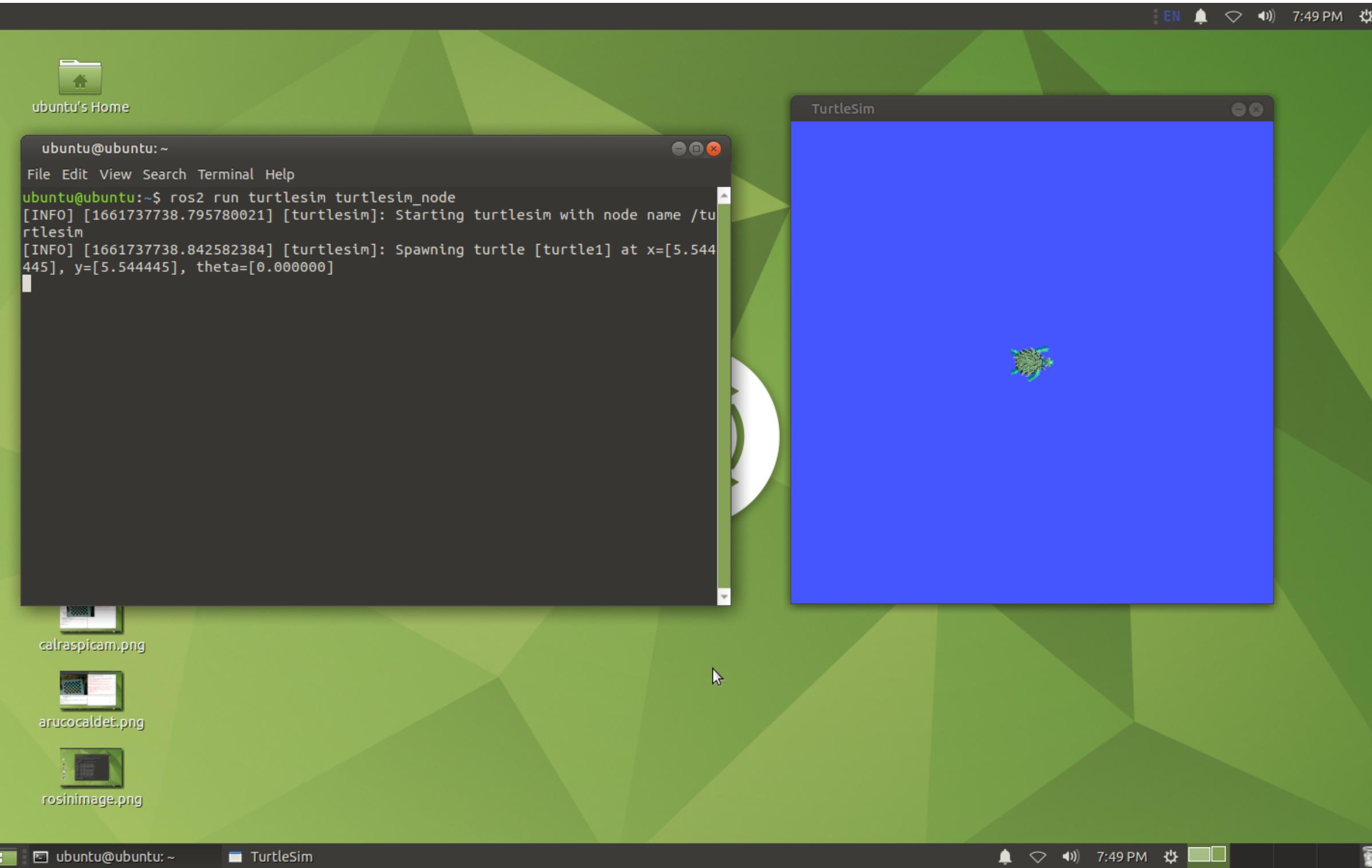


Figura: Ejemplo de ROS corriendo en Ubuntu Mate.

Ubuntu Desktop

- Para la Raspberry Pi 4.
- Descarga directa de la instalación.
- Interfaz con definición de alta calidad, más intuitiva y conocida para el usuario.
- Instalación de ROS2, Python, editores de texto, etc.

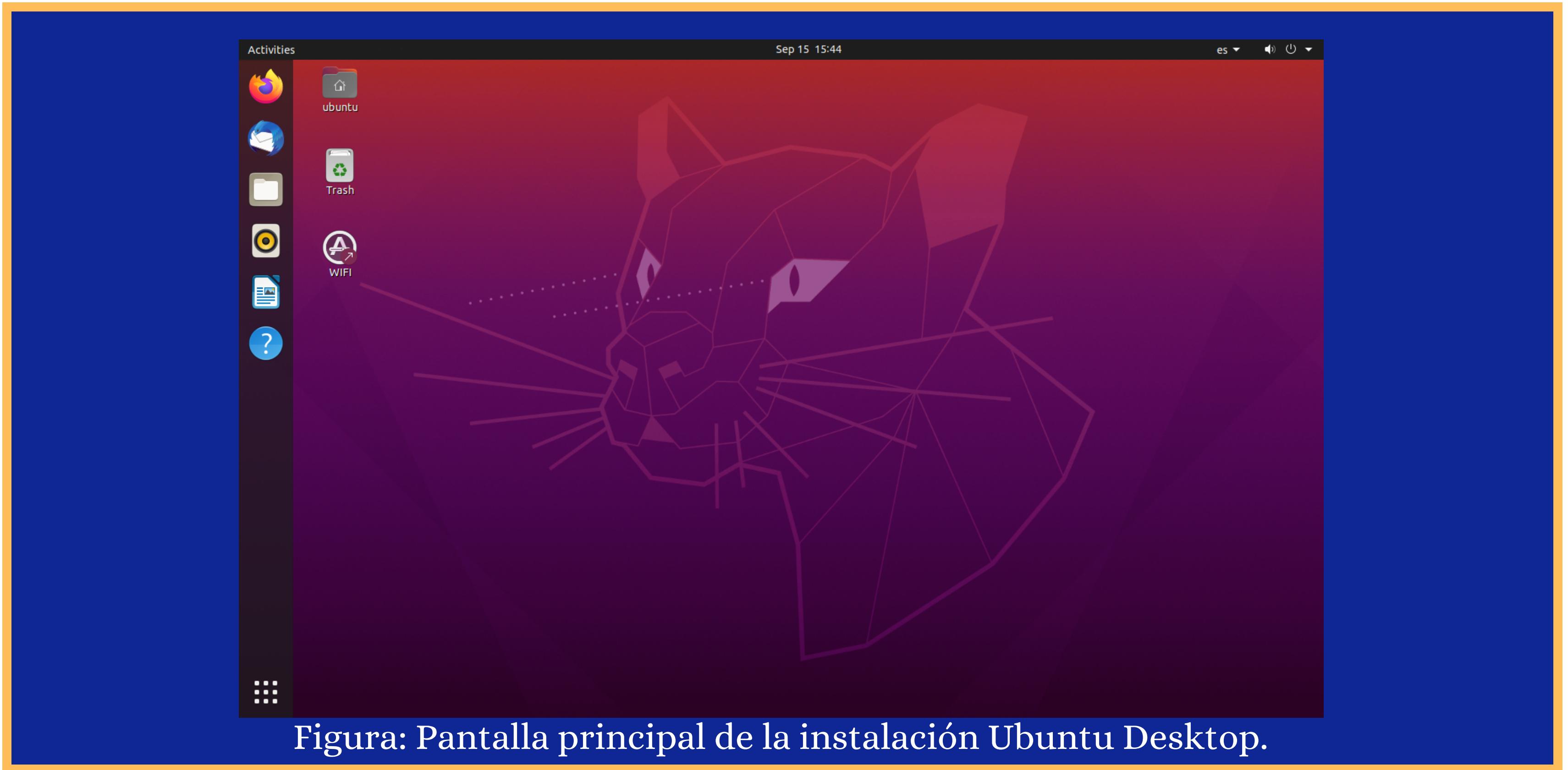


Figura: Pantalla principal de la instalación Ubuntu Desktop.

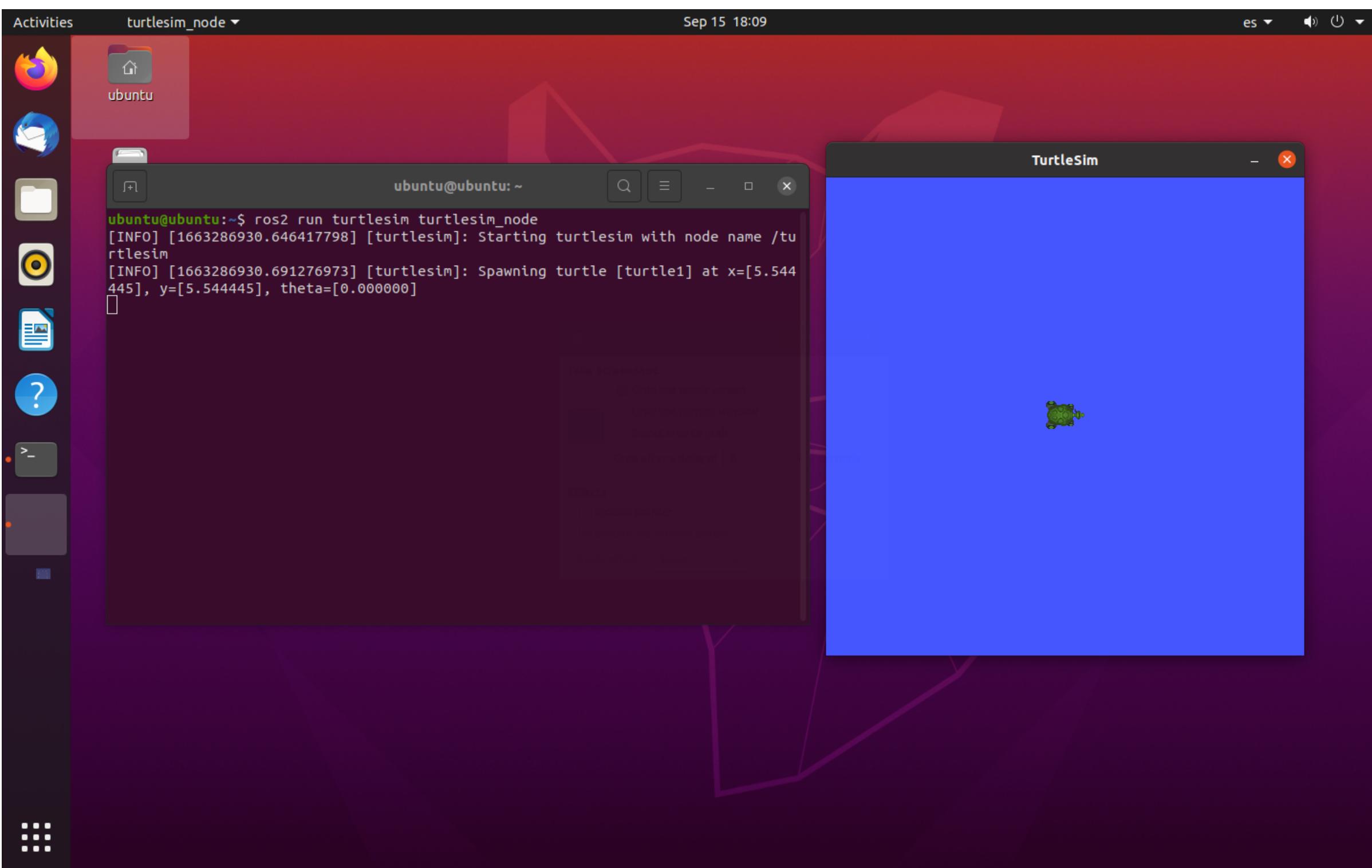
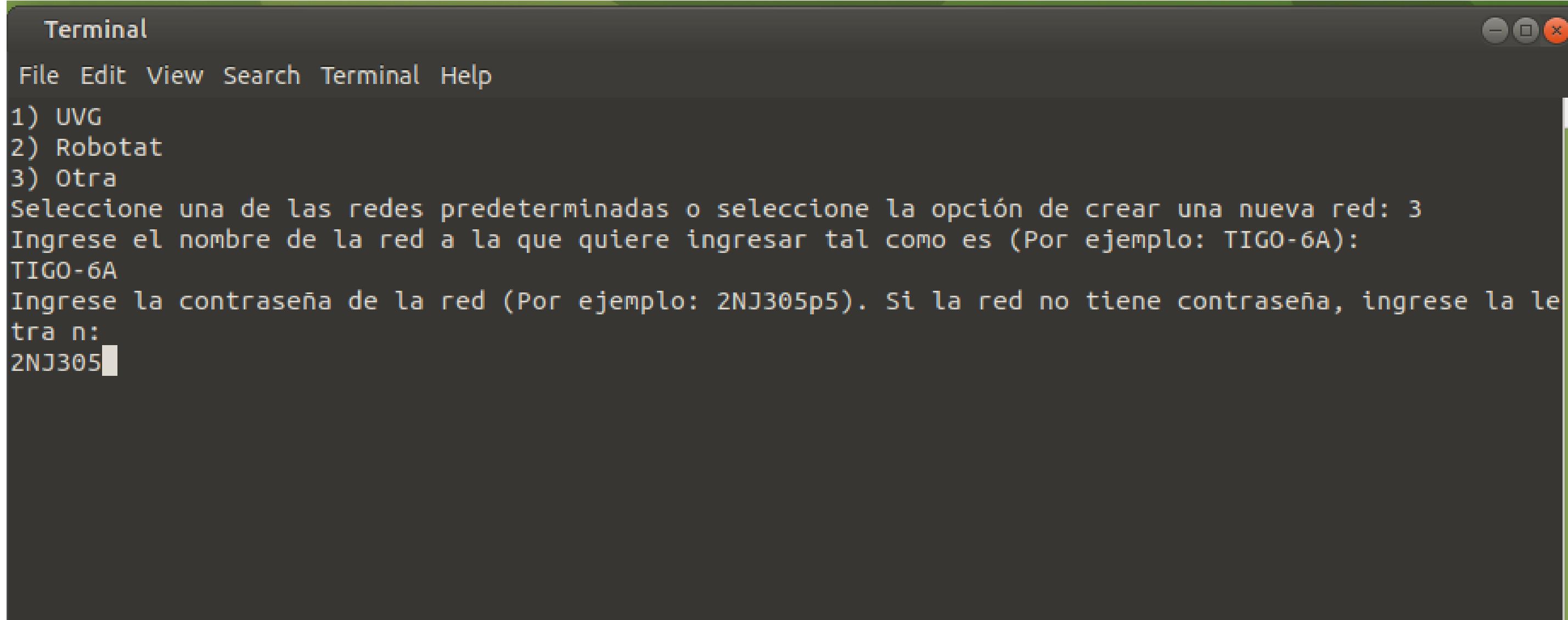


Figura: Ejemplo de ROS corriendo en Ubuntu Desktop.

Conexión a Internet

- Se encontraron problemas con la conexión directa a la red de Internet desde la interfaz.
- Se realizó conexión a la red por medio de modificaciones a un archivo de Netplan.
- Creación de *bash script* para que el usuario pudiera conectarse con facilidad.



The screenshot shows a terminal window with a dark gray background and a green title bar. The title bar has the word "Terminal" in white. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal contains the following text:

```
1) UVG
2) Robotat
3) Otra
Seleccione una de las redes predeterminadas o seleccione la opción de crear una nueva red: 3
Ingrese el nombre de la red a la que quiere ingresar tal como es (Por ejemplo: TIGO-6A):
TIGO-6A
Ingrese la contraseña de la red (Por ejemplo: 2NJ305p5). Si la red no tiene contraseña, ingrese la letra n:
2NJ305
```

Figura: *Bash script* para conexión a WiFi.

```
network:
  eternets:
    eth0:
      optional: true
      dhcp4: true
  version: 2

  wifis:
    wlan0:
      optional: true
      access-points:
        "TIGO-61AA":
          password: "2NJ555305178"
      dhcp4: true
```

Figura: Archivo que se modifica para realizar conexión a internet.

Guías

Se elaboraron diferentes guías detallando las instalaciones realizadas, las instalaciones del sistema operativo sobre una Raspberry Pi y los pasos a seguir para trabajar con estas.

Guía de instalación imagen de Ubuntu para ROS2 para Raspberry Pi 3 y 4

Imagen de Ubuntu Server y primeros pasos

Se puede encontrar un tutorial general de cómo realizar el proceso de cargar Ubuntu Server a una Raspberry en el siguiente enlace: <https://ubuntu.com/tutorials/how-to-install-ubuntu-on-your-raspberry-pi#1-overview>

Para la instalación del programa ROS2, es necesario tener en la Raspberry una versión de Ubuntu, no de Raspbian como vienen programadas de parte de la universidad. Dado que se está trabajando con Ros2: FoxyFitzRoy, la versión óptima de Ubuntu es la 20.04.

Las opciones de imágenes disponibles para la instalación de Ubuntu para las Raspberry Pi y las compatibilidades de cada una se pueden encontrar en el siguiente sitio: <https://ubuntu.com/download/raspberry-pi>

Como indica el sitio, la versión de Desktop de Ubuntu no es recomendada para Raspberries debajo de la Raspi 4, dado el RAM de esta. Además, para Raspberries menores a la Raspi 3, es recomendable instalar la versión de 32 bits en lugar de la de 64 bits. La versión de Ubuntu 20.04 de 32 bits ha sido descontinuada. Por lo tanto, se estará flasheado la tarjeta SD con Ubuntu Server 20.04.4 LTS de 64-bits. Para hacer esa operación se recomienda realizarla por medio del Raspberry Pi Imager que se puede descargar en el siguiente link: <https://www.raspberrypi.com/software/>

Para flashear la imagen solo es necesario tener una tarjeta microSD de 32 GB. En la aplicación del imager, se presiona el botón de Choose OS. El menú ofrece varias versiones de Linux que puede

Guía rápida de ingreso y conexión a internet

A continuación, se presenta una guía para el uso de la Raspberry Pi con la imagen de Ubuntu Mate o Ubuntu Desktop para la Raspberry Pi 3 o 4 creada por Alejandra Samayo en 2022.

Una vez se haya realizado la instalación de la imagen correctamente, la página principal pide las credenciales para ingresar. Los datos son:

Usuario: ubuntu

Contraseña: IEuvg!

Puede que tarde un poco en cargarse la página inicial de la interfaz gráfica. El siguiente paso es conectar la computadora a una red de internet WIFI. En ninguna de las imágenes creadas funciona el icono para conectarse desde la interfaz, por lo que es necesario hacer la conexión por otro medio. Para verificar la conexión de la Raspberry al internet solo basta con abrir una terminal (desde las opciones de la interfaz o presionando Ctrl+Alt+t) e ingresar la instrucción:

`ifconfig`

Esto debería de listar la dirección IP de la Raspberry en su conexión de wifi. Si en caso no estuviera conectada a la red, no sacará este valor para la conexión de wifi.

En la ventana principal de la interfaz se pueden ver los programas. Se debe buscar el ícono llamado WIFI.

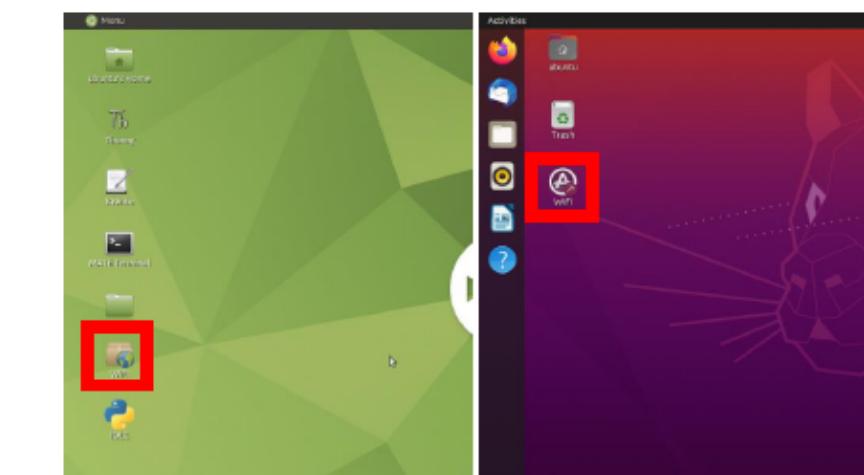


Figura: Guías realizadas para la instalación del sistema operativo.

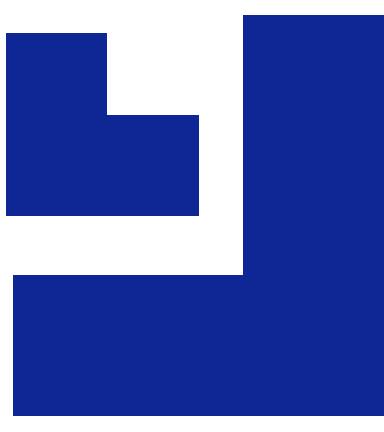
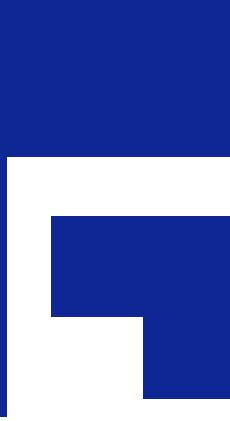
Adaptación de cámaras

Raspberry Pi Cam

- La habilitación de la cámara no se realizó de manera directa sino que se modificó el archivo de *Firmware*.
- La librería de Python de *picamera* fue compatible con la instalación de Ubuntu.
- Se utilizó OpenCV para realizar los módulos.

Jevois

- Librería interna de la cámara para modificar los módulos de visión por computadora no es compatible con la arquitectura ARM de la Raspberry Pi.
- Se llamaron los módulos al cambiar la configuración de la cámara por medio de comunicación serial.



Módulo de visión por computadora

Raspberry Pi Camera

Se utilizó OpenCV para realizar los módulos.

- Algoritmo de calibración

Utiliza un tablero de calibración y guarda la matriz de calibración de la cámara.

- Detección de marcadores

Realiza un proceso de *thresholding* y luego busca la formación detectada en el diccionario de marcadores de OpenCV.

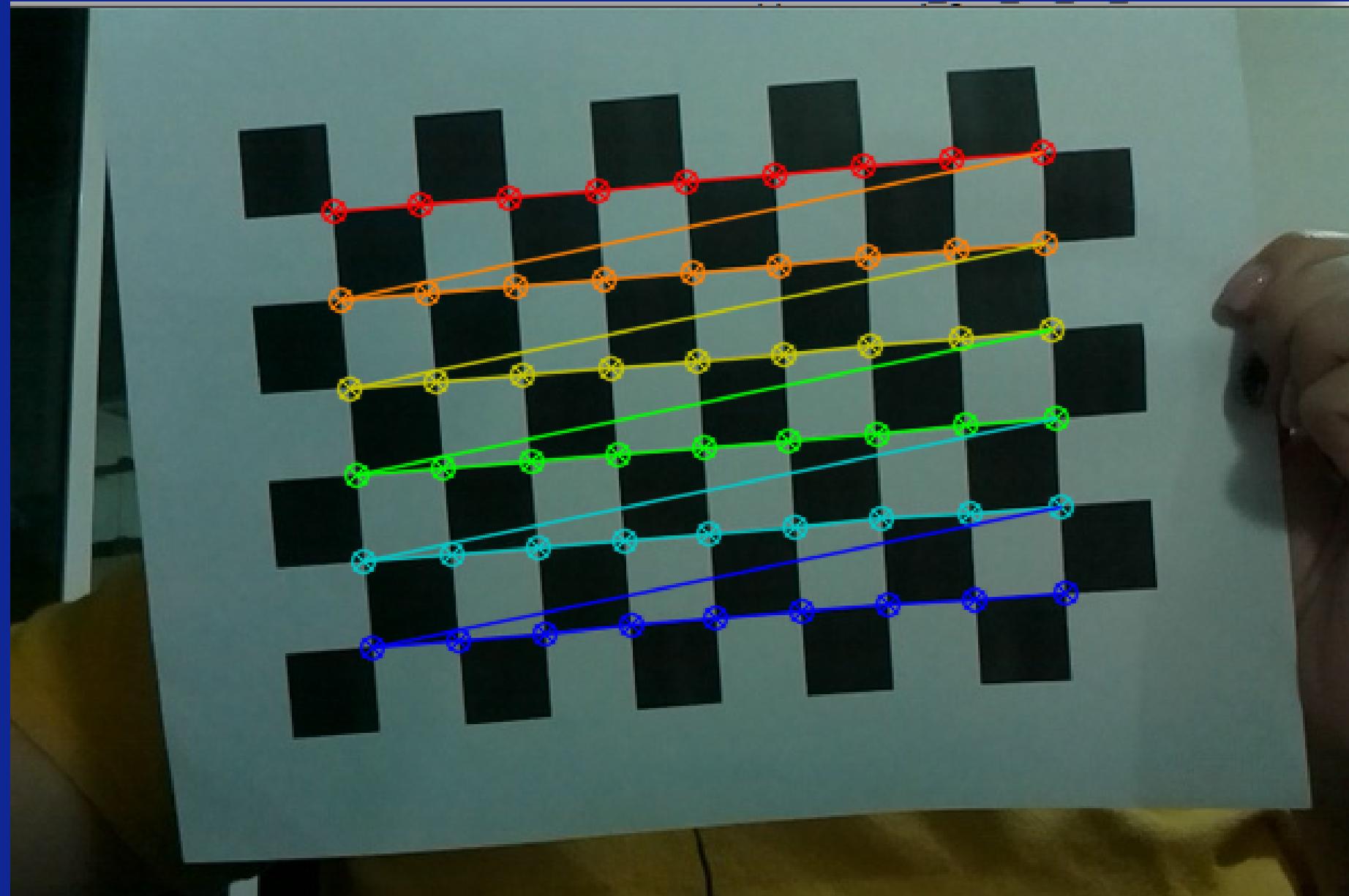


Figura: Algoritmo de identificación de esquinas sobre tablero de calibración.

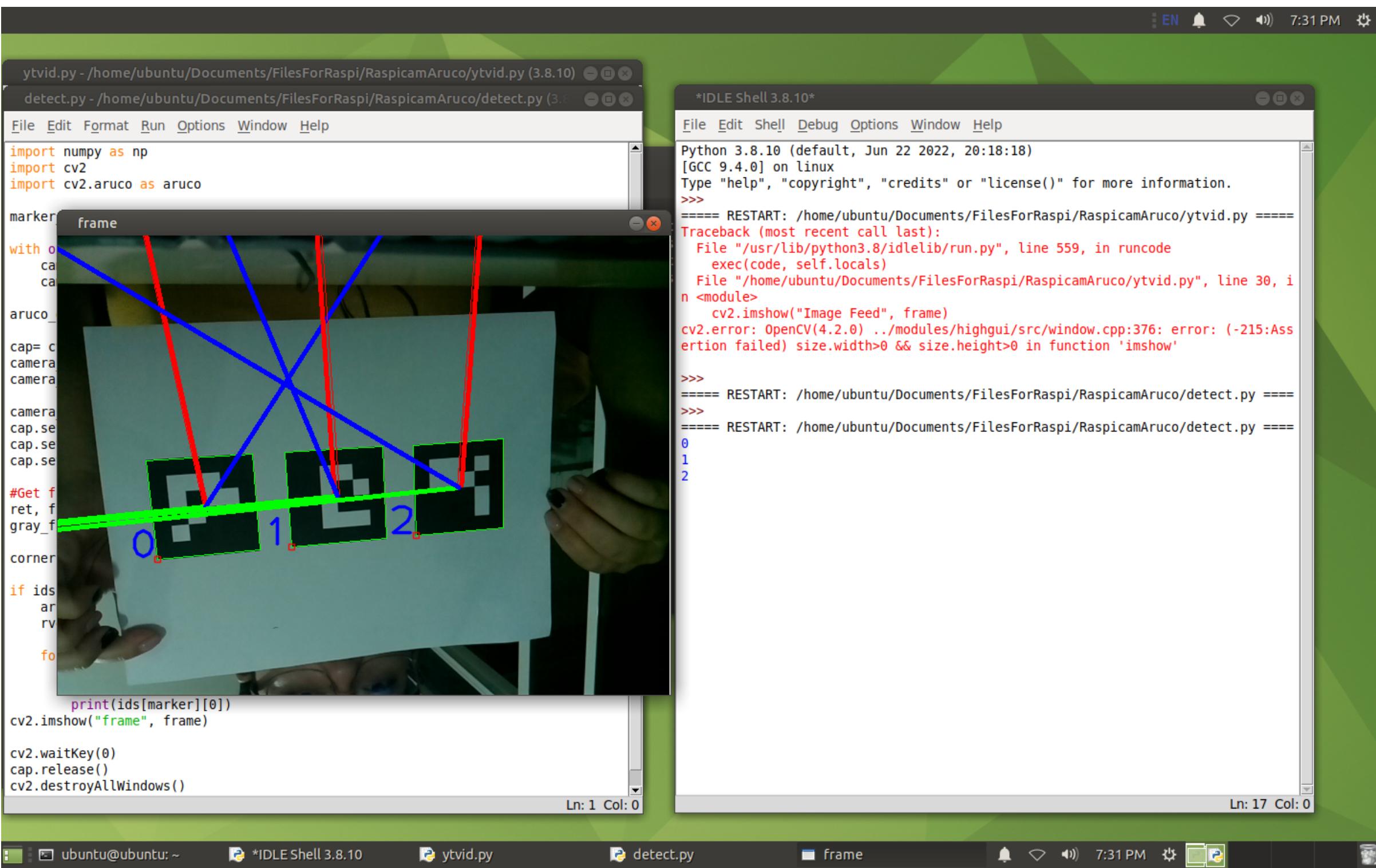
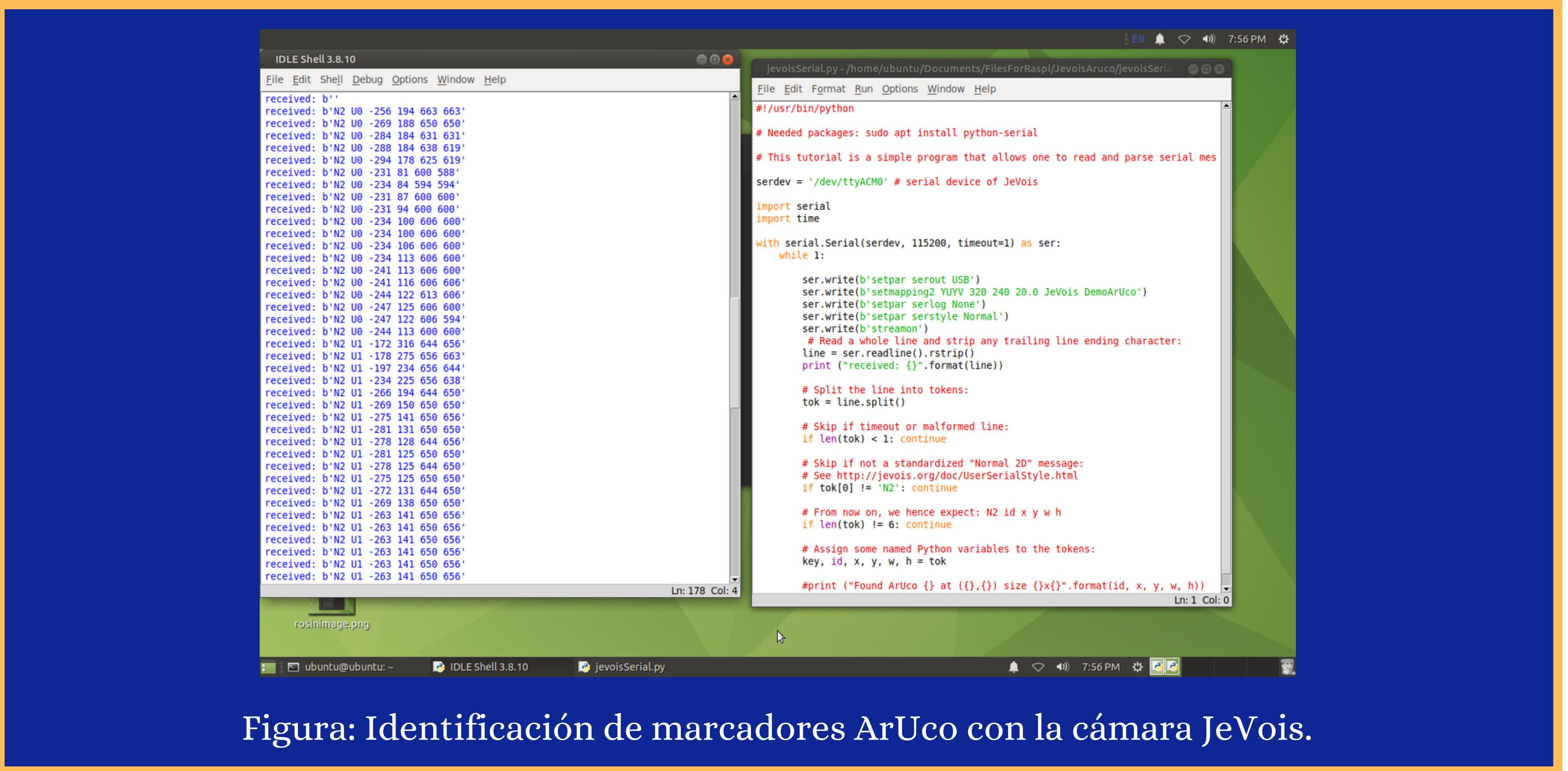


Figura: Identificación de marcadores ArUco con la Raspberry Pi Camera.

Jevois

- Archivo bash para llamar la configuración necesaria de la cámara.
- Se detalla el módulo y su configuración, mensajes en 3D, información detallada (cuaterniones).
- Se manda esta información por medio del puerto serial.



Selección de cámara

Consideraciones

- Detección de ArUco.
- Compatibilidad con ROS y Ubuntu.
- Otros módulos sobre la plataforma robótica.
- Implementaciones futuras.

Desventajas

Ventajas

Raspberry Pi Camera



- Se pueden crear otros algoritmos de visión por computadora.
- Compatibilidad con OpenCV.

- Procesamiento sobre computadora.
- No es compatible con otras computadoras sin adaptador (cable FFC).
- Para obtener información detallada de la ubicación y orientación del marcador es necesario hacer cambios a la programación.

JeVois

Ventajas

Desventajas



- Procesamiento en la cámara.
- Comunicación por puerto serial.
- Facilidad de alterar cómo se recibe la información.
- Facilidad de implementar otros algoritmos de visión por computadora.
- Acceso a información detallada, incluyendo la rotación del marcador.

- No hay acceso a imagen desde Python.
- Alimentación por USB.
- No es fácil crear algoritmos de visión por computadora.

Resumen de comparación de cámaras

• Ventajas

Raspberry Pi Cam

- Se pueden crear otros algoritmos de visión por computadora.
- Compatibilidad con OpenCV.

• Desventajas

- Procesamiento sobre computadora.
- No es compatible con otras computadoras
- Para obtener información detallada de la ubicación y orientación del marcador es necesario hacer cambios a la programación.

JeVois

- Procesamiento en la cámara.
- Comunicación por puerto serial.
- Facilidad de alterar cómo se recibe la información.
- Facilidad de implementar otros algoritmos de visión por computadora.
- Acceso a información detallada, incluyendo la rotación del marcador.

Selección de módulo

- Cámara JeVois
 - Procesamiento en la cámara, no sobre la Raspberry Pi.
 - Adaptabilidad de otros módulos de visión por computadora.
 - Facilidad de obtener datos deseados y cambiarlos de ser necesario.

Objetivo 1

Kinect

Problemas de adaptación

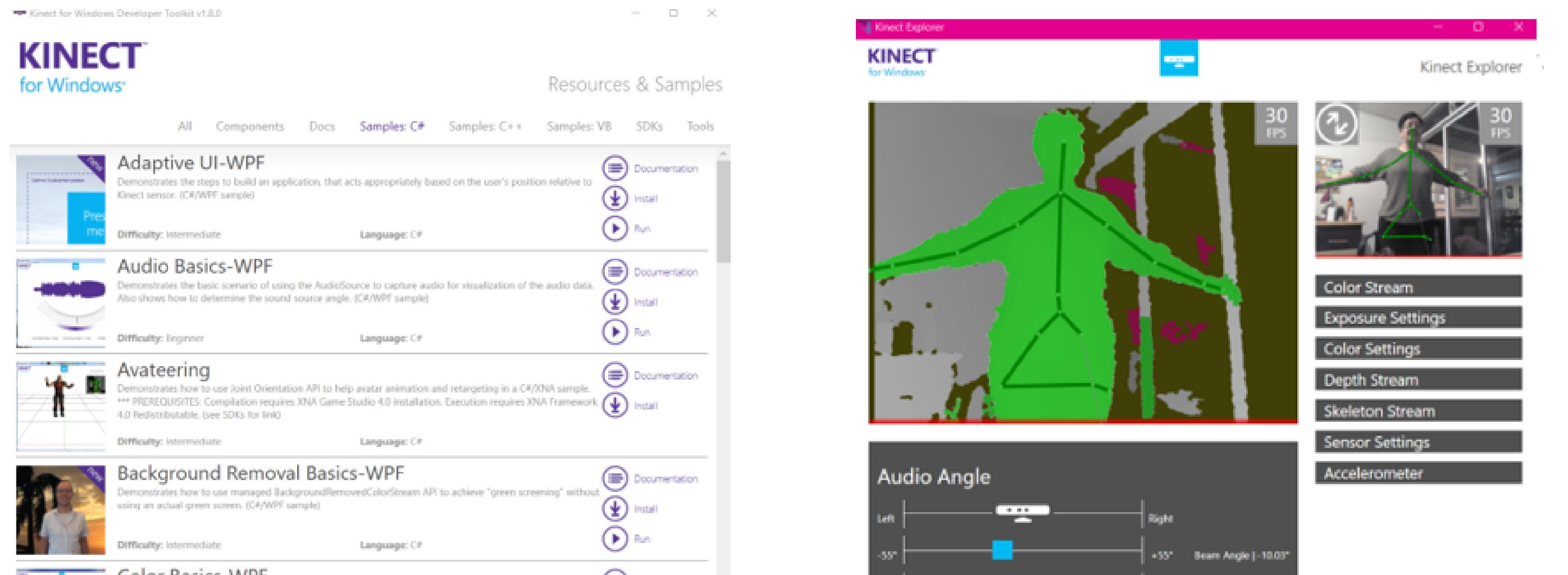
- Pérdida de soporte oficial en sistemas de Windows.
- Librerías *open source* desactualizadas.
- Necesita entradas USB3 (presentes solo en módulo 4).
- Se utilizó una versión de Ubuntu muy avanzada.



Pruebas en Windows 11

Se necesitó instalar:

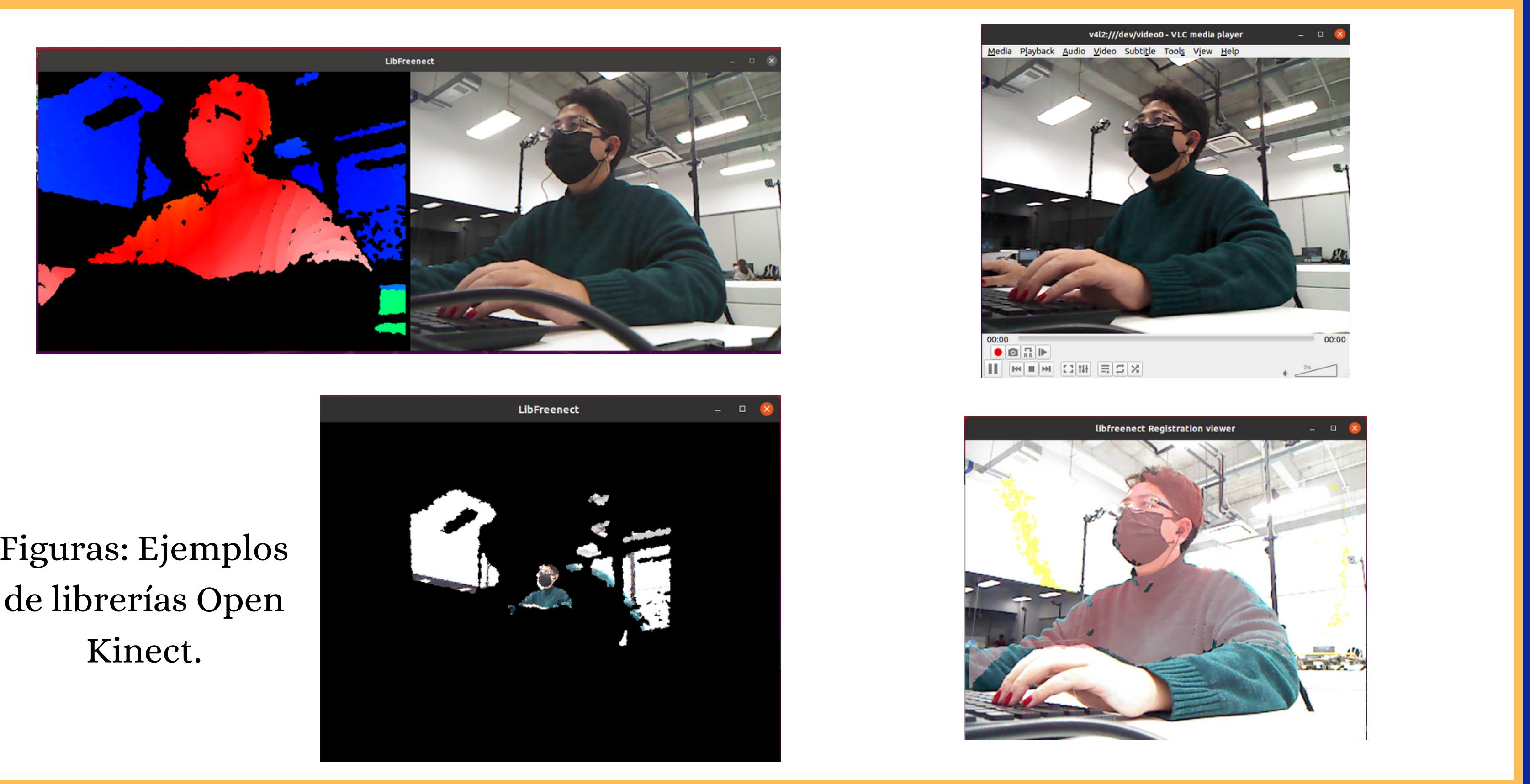
- Kinect for Windows SDK 1.8
- Kinect for Windows Developer Toolkit v1.8



Figuras: Windows Developer Toolkit (Izq.) y un ejemplo de módulo corriendo (Der.).

Adaptación a Raspberry Pi

- Descarga directa de las librerías de Open Kinect.
- Modificación archivo de permisos.
- Cambios a archivos con números actualizados de licencias.
- Permiso con instrucción *modprobe*.
- No se logró acceso a micrófonos ni a control de motores del módulo.



Módulo de detección de marcadores ArUco

Usando la librería de OpenCV y el mismo procedimiento de la Raspberry Pi Cam.

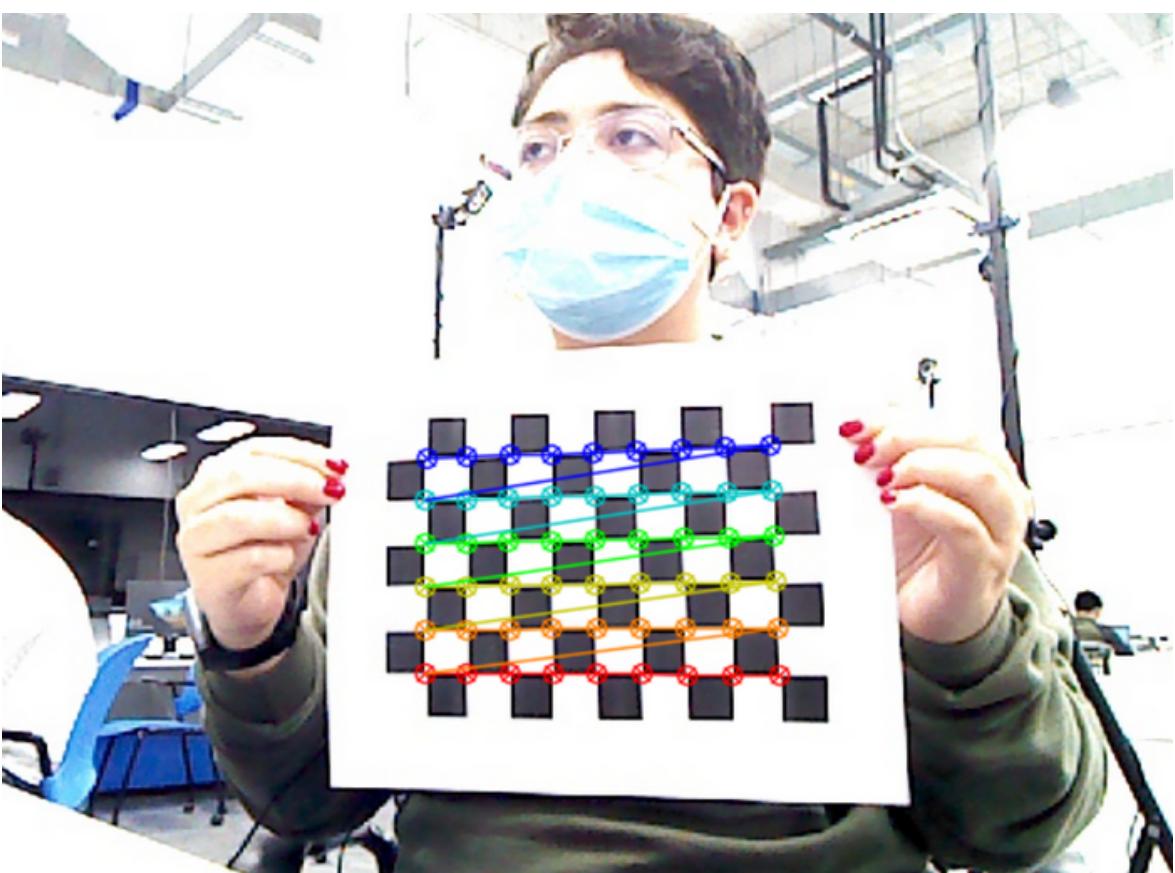


Figura: Algoritmo de calibración.

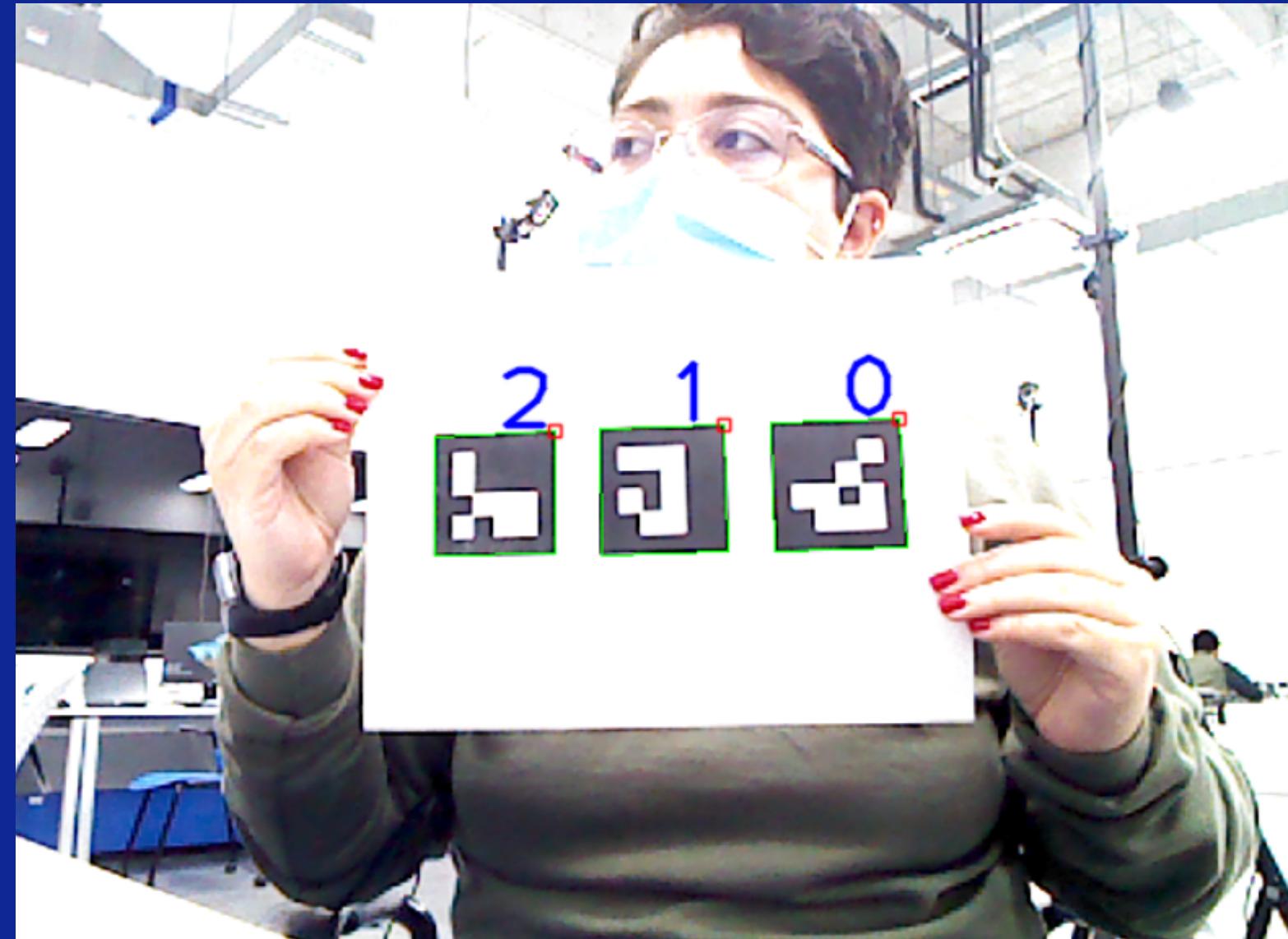
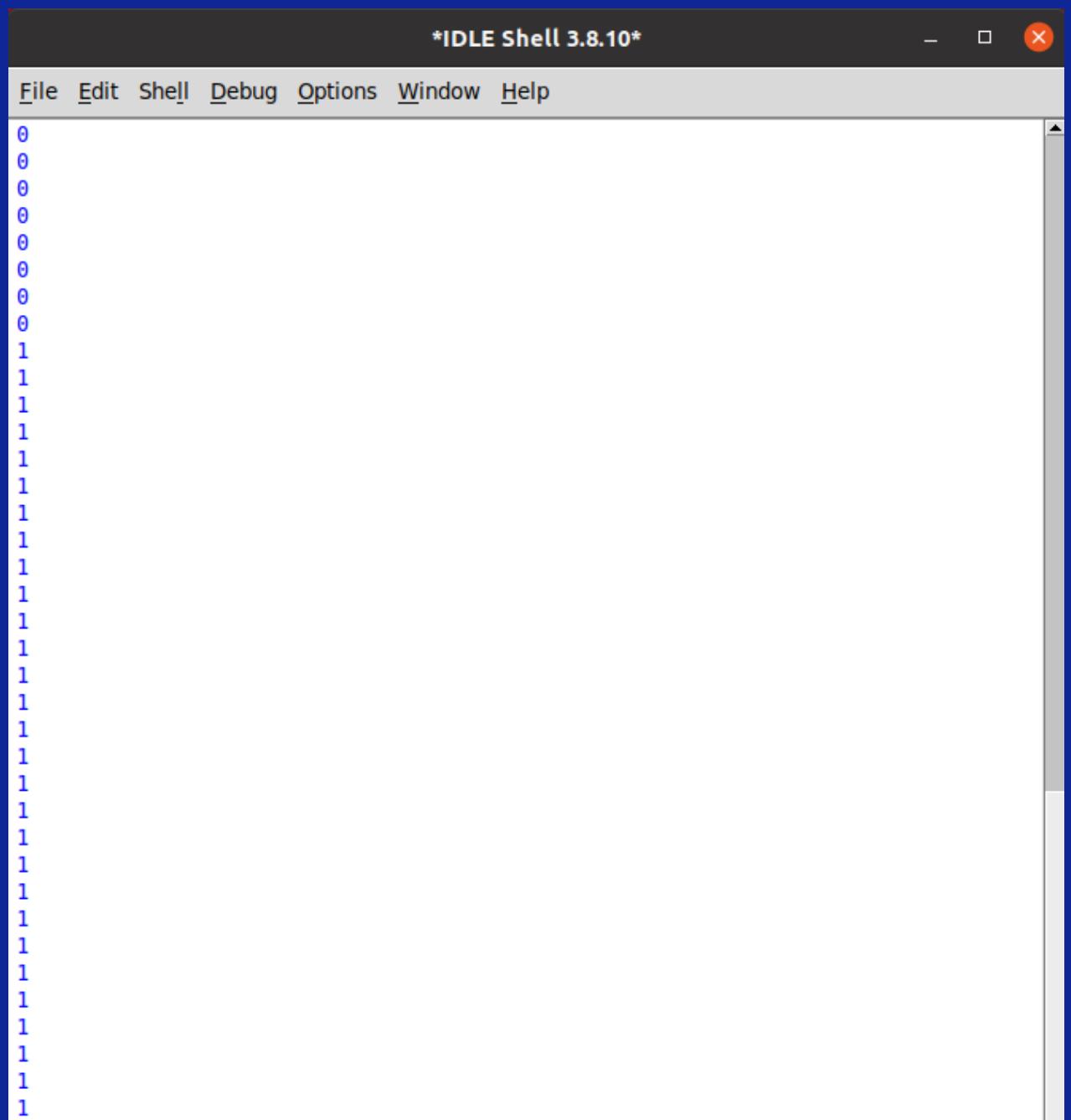


Figura: Detección de marcadores ArUco.

Conclusión de la cámara Kinect

A pesar de que se tuvo acceso a la cámara y se pudo realizar el módulo de detección de marcadores ArUco, se decidió no implementar la cámara al proyecto.

Requiere bastante carga computacional para el procesamiento y no se obtuvo acceso completo a la cámara sobre el sistema operativo de Ubuntu.

Objetivo 3

Guía para uso del Microsoft Kinect for Windows V1

Sobre el producto

El Microsoft Kinect for Windows es un producto que salió al mercado en el año 2012. El apoyo en cuanto a *drivers* y *software* (Kinect for Windows SDK) que sacó la compañía fue contaba con soporte para el sistema operativo Windows 7. En el 2014, Microsoft sacó un segundo modelo, para el cuál había bastantes cambios en cuanto al soporte. Con el tiempo, se fue perdiendo el soporte oficial para el aparato y no todas las librerías son compatibles con los nuevos sistemas operativos. Por lo mismo, surgió un esfuerzo colaborativo para generar el soporte necesario a través de unas librerías *open source* conocidas como Open Kinect, que buscan adaptar el aparato a diferentes sistemas operativos, como Linux y versiones más recientes de Windows. A través de diferentes pruebas que se estuvieron realizando con el aparato, surge esta guía para la instalación de las librerías con las que se pudo llamar el aparato.

Uso en Windows 11

La versión original de Kinect for Windows SDK 1.0 no es compatible con sistemas operativos de Windows a partir de Windows 8. Además, a partir de versiones de las librerías 2.0, el aparato no es compatible ya que fueron diseñadas para el modelo 2.

Después de varias instalaciones fallidas, con las que el sistema detectaba la cámara, pero sacaba un mensaje de error en los *drivers* al mostrar el aparato en el Administrador de Dispositivos, se encontró el sistema compatible con Windows 11: la versión de Kinect for Windows SDK 1.8.

Este se puede encontrar en el siguiente enlace: <https://www.microsoft.com/en-us/download/details.aspx?id=40278>

Para su uso se recomienda también instalar la interfaz gráfica de Kinect for Windows Developer Toolkit v1.9, ya que cuenta con varios módulos instalados y permite una visualización rápida de

Figura: Guía realizada para uso
del Mircosoft Kinect.

Integración a ROS2

Integración a ROS

- Módulo de comunicación de *topics*.
- Utilizando librerías de comunicación de odometría con *nav_msg* del paquete *geometry_msg*.
- Llama configuración de la cámara que se realiza por *bash script*, recibe información del puerto serial, verifica longitud del mensaje y publica la información deseada.

Objetivo 3

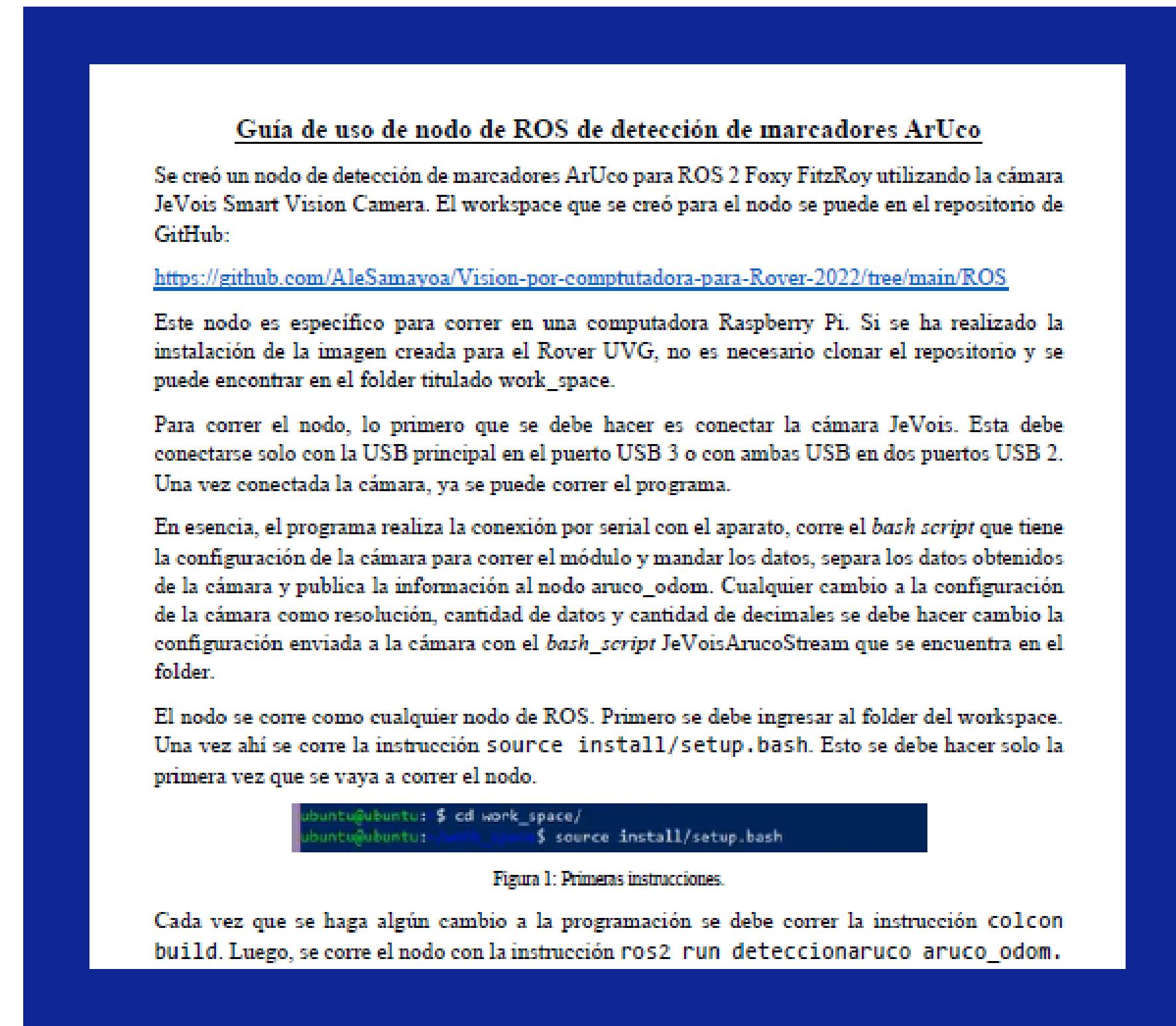
```
ubuntu@ubuntu: ~/work_space
File "/usr/lib/python3/dist-packages/serial/serialposix.py", line 483, in read
    ready, _, _ = select.select([self.fd, self.pipe_abort_read_r], [], [], timeout.time_left())
KeyboardInterrupt
ubuntu@ubuntu:~/work_space$ colcon build
Starting >>> deteccionaruco
Finished <<< deteccionaruco [4.72s]

Summary: 1 package finished [5.63s]
ubuntu@ubuntu:~/work_space$ ros2 run deteccionaruco aruco_odom
ArUco b'U1' en (b'-43.21',b'-5.03',b'420.69') (b'0.03', b'2.13', b'-2.21', b'-0
.12')
ArUco b'U1' en (b'-42.90',b'-5.18',b'418.27') (b'0.04', b'2.12', b'-2.21', b'-0
.09')
ArUco b'U1' en (b'-42.70',b'-5.16',b'418.58') (b'0.03', b'2.13', b'-2.21', b'-0
.09')
ArUco b'U1' en (b'-42.32',b'-5.23',b'416.57') (b'0.04', b'2.12', b'-2.21', b'-0
.06')
ArUco b'U1' en (b'-42.36',b'-5.20',b'416.80') (b'0.04', b'2.12', b'-2.21', b'-0
.06')
ArUco b'U1' en (b'-42.24',b'-5.19',b'416.96') (b'0.04', b'2.13', b'-2.21', b'-0
.06')
ArUco b'U1' en (b'-42.05',b'-5.20',b'416.87') (b'0.04', b'2.12', b'-2.21', b'-0
.07')
```

```
- 0.0
---
header:
  stamp:
    sec: 0
    nanosec: 0
  frame_id: odom_aruco
child_frame_id: b'U1'
pose:
  pose:
    position:
      x: -42.05
      y: -5.2
      z: 416.87
    orientation:
      x: 0.04
      y: 2.12
      z: -2.21
      w: -0.07
covariance:
- 0.0
- 0.0
- 0.0
- 0.0
```

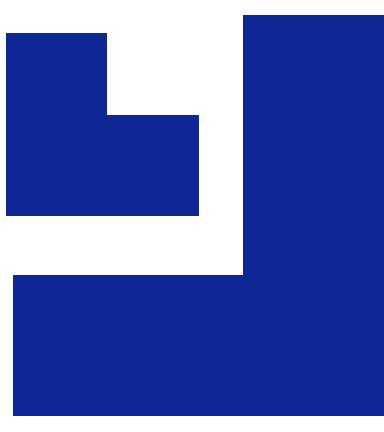
Figuras: Nodo corriendo (Izq.) y *echo* al tópico (Der.).

Figura: Guía realizada para uso del nodo creado.





Características del módulo de visión



Características de la cámara

- Mensajes 3D no han sido tan comprobados por la empresa.
- Pruebas para observar confiabilidad de los datos obtenidos y desviación estándar de los datos para matriz de covarianza de Filtro de Kalman en la integración de todo.
- Pruebas de profundidad, rango y dominio de la cámara.



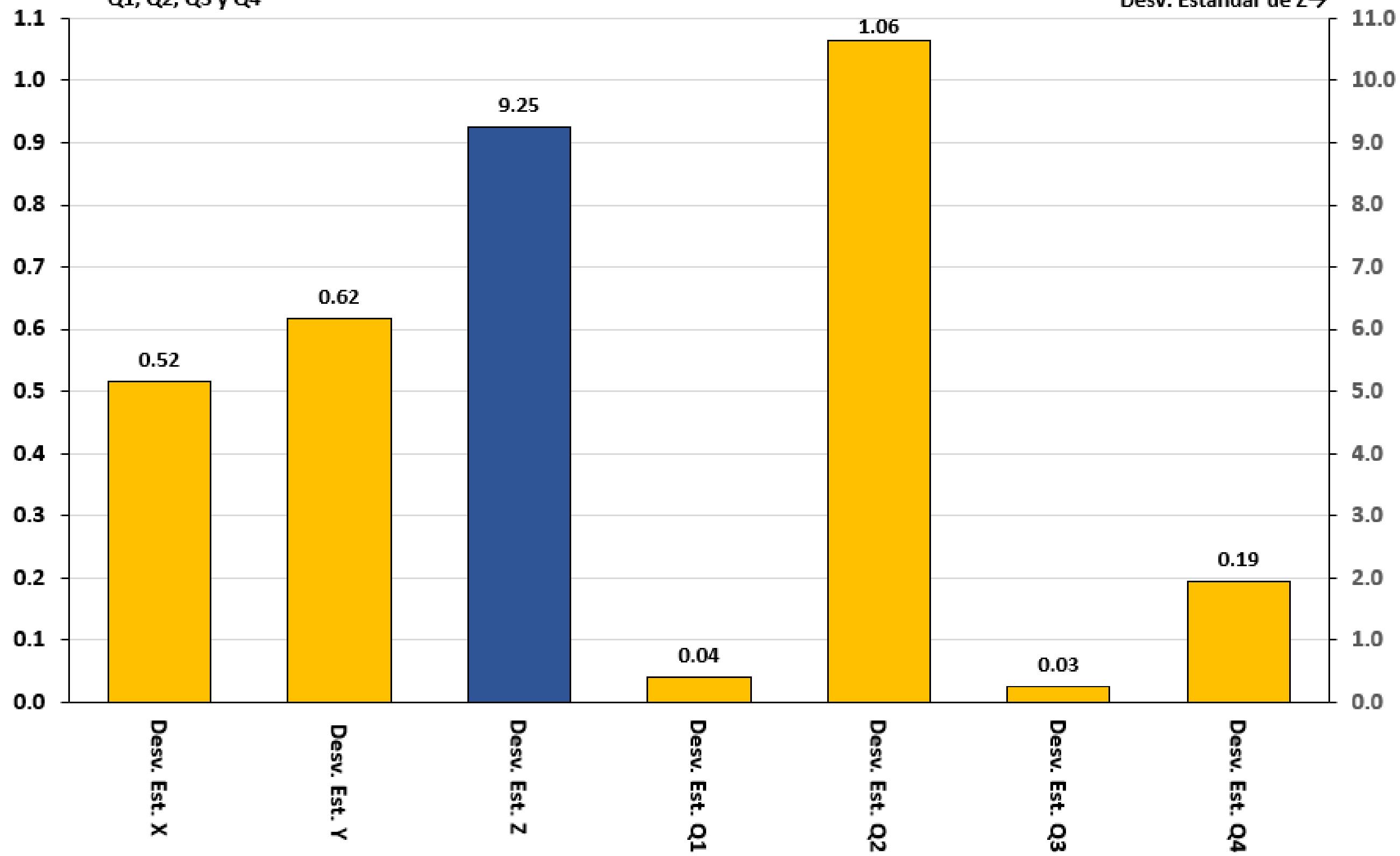
Figura: Prueba de la medida de profundidad de la cámara.

Media de la Desviación Estándar

-Milímetros-

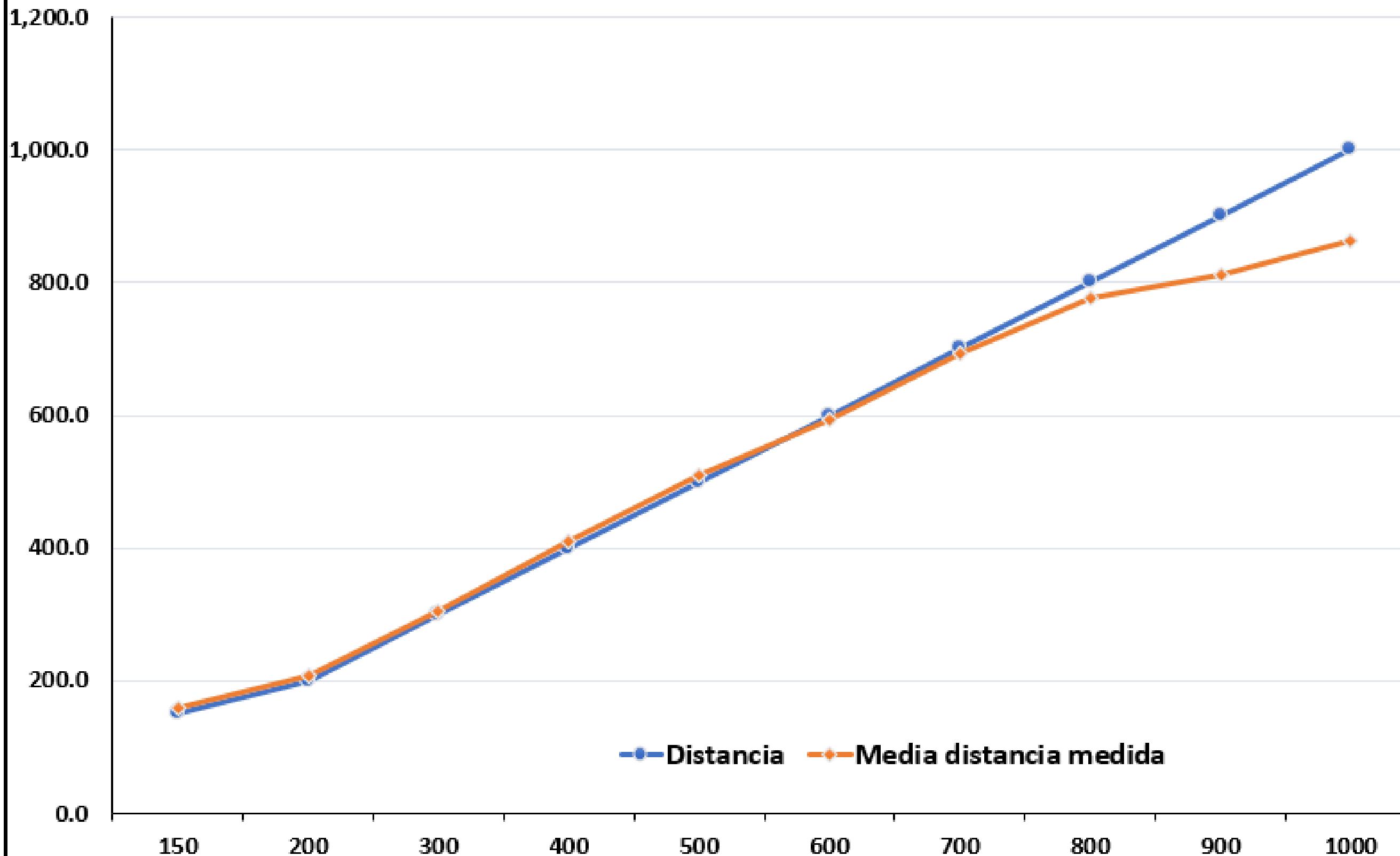
←Desv. Estándar de X, Y,
Q1, Q2, Q3 y Q4

Desv. Estándar de Z→

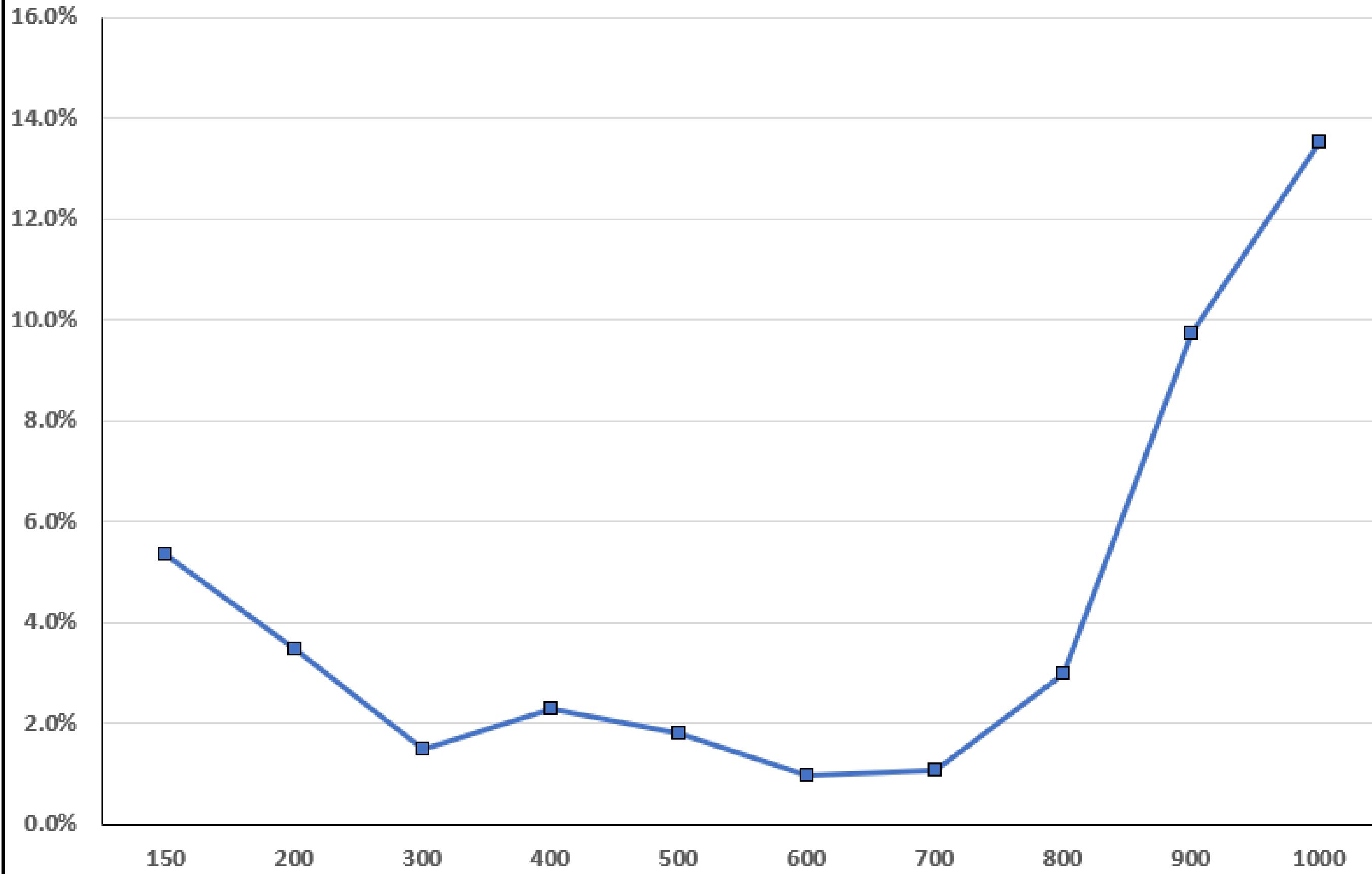


Comparación medida de profundidad

-Milimetros-

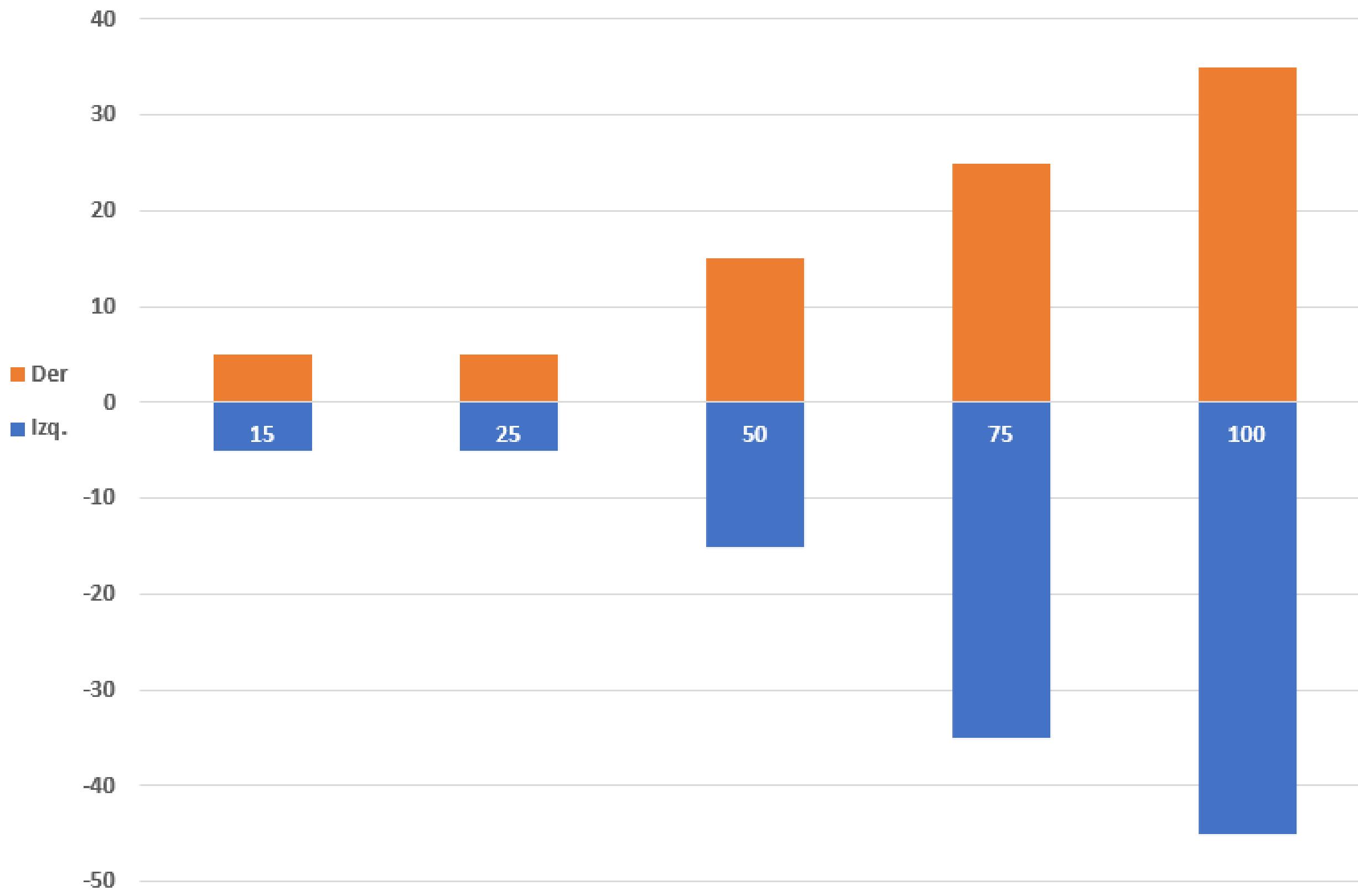


Error Experimental de Z -Milímetros-



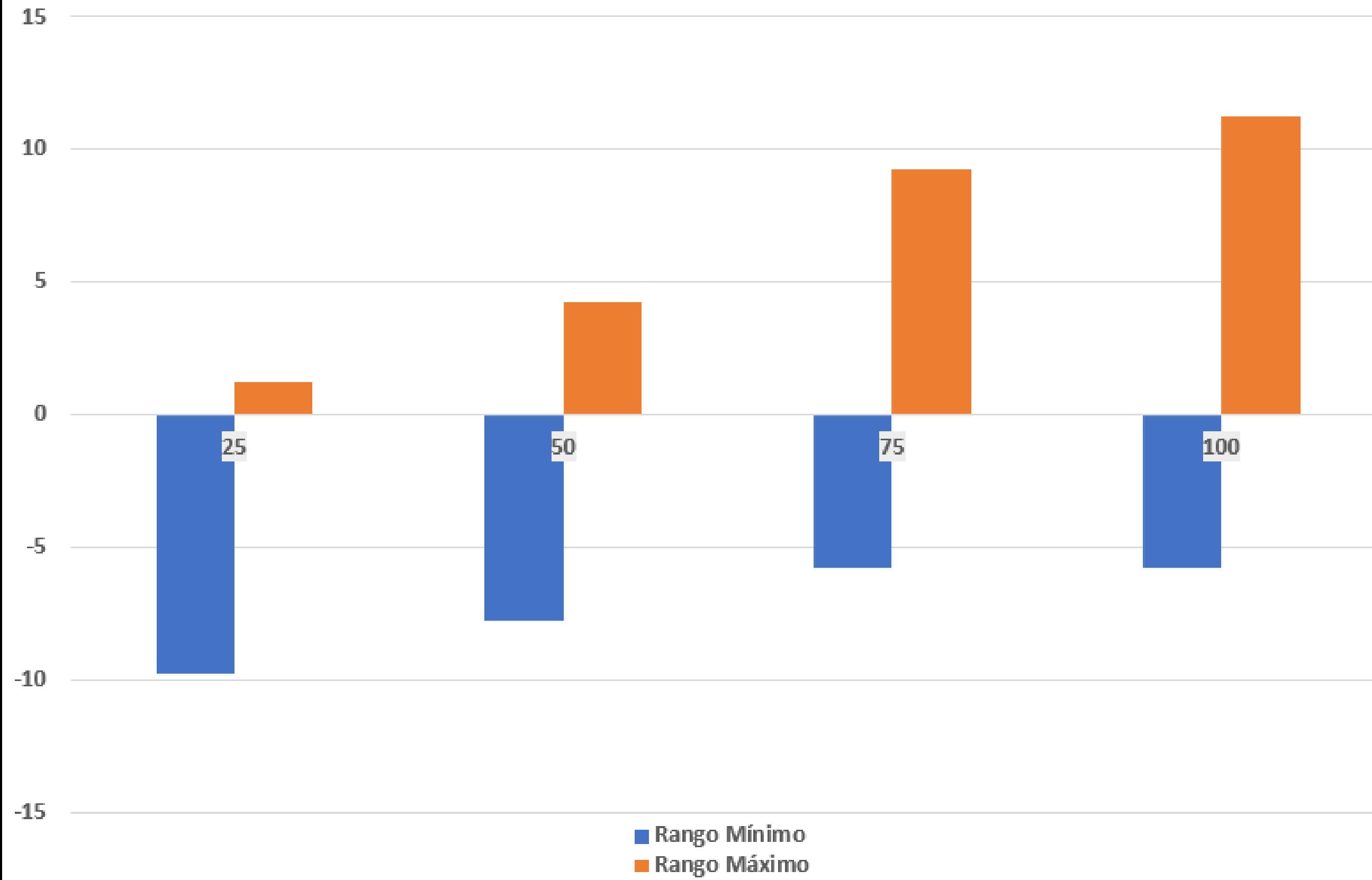
Dominio de la cámara (Eje horizontal)

-Centímetros-



Rango de la cámara (Eje vertical)

-Centímetros-



Características del Módulo

- Mayor desviación estándar es la medida de profundidad de 9.25 mm.
- Bastante presición de medida de profundidad hasta 1 metro. El mayor porcentaje de error se encuentra a 1 metro: 14% de error.

Características del Módulo

- El dominio de la cámara llega hasta 30-45 cm de ambos lados.
- El rango llega hasta 10 cm de alto y 10 cm por abajo.
- Funcionalidad de comunicación UART por pines TX y RT de la Raspberry Pi.

Integración a Rover

Integración

Se creo una función de mapeo en RViz con un nodo suscrito al que se había creado anteriormente.

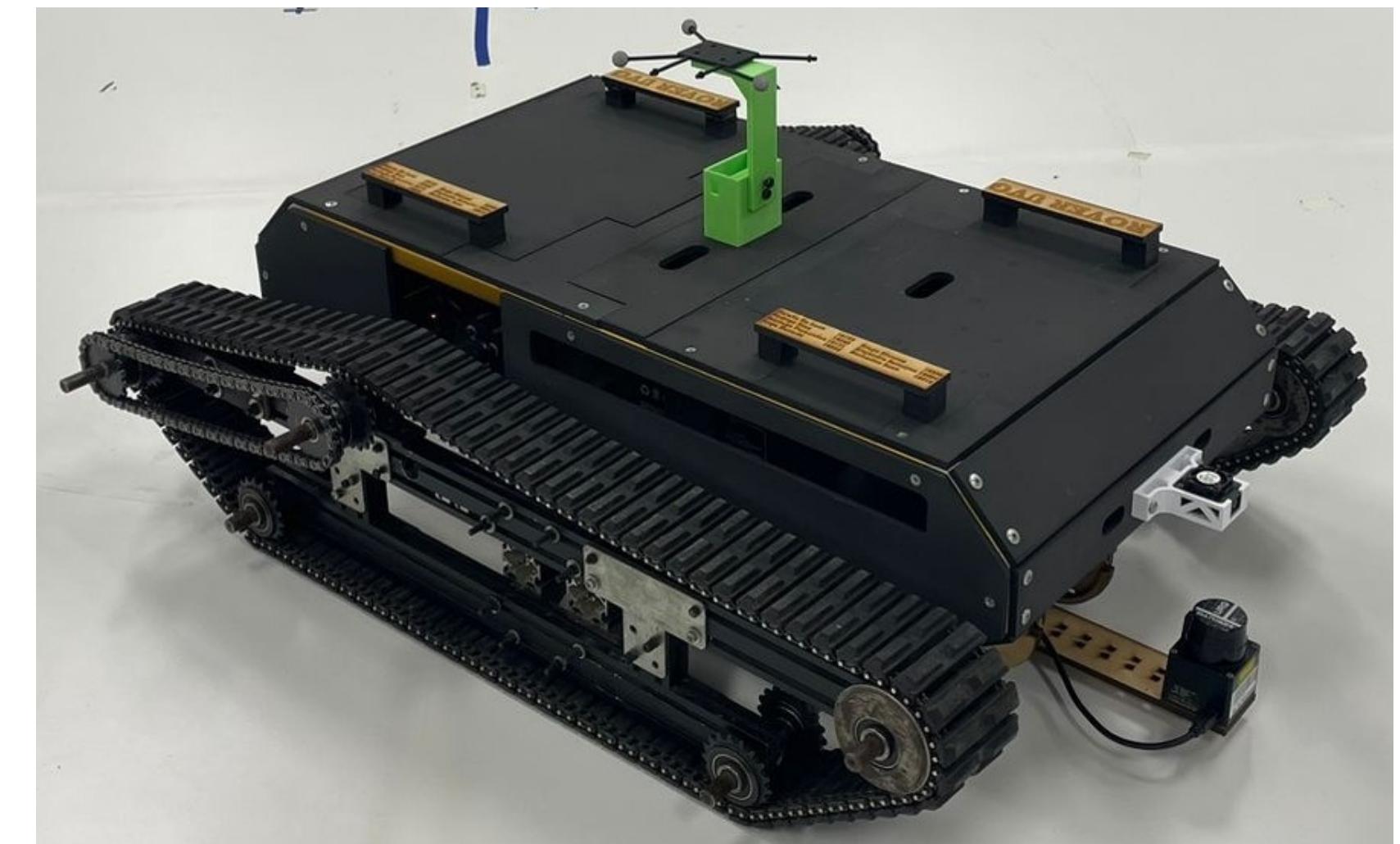


Figura: Plataforma robótica móvil

Conclusiones

Conclusiones

- Se logró la implementación de un módulo de visión por computadora de detección de marcadores ArUco a la plataforma robótica por medio de un nodo de ROS para una aplicación de mapeo.

Objetivo Principal

Conclusiones

- Fue necesario determinar un sistema operativo y realizar una instalación que contara con el sistema operativo de Linux Ubuntu 20.04 y ROS 2 FoxyFitzRoy.

Conclusiones

- Se logró un módulo de detección de marcadores ArUco en ambas cámaras en el sistema operativo que se estuvo utilizando sobre la plataforma robótica.

Objetivo 1

Conclusiones

- Se decidió que la cámara más adecuada para la visión por computadora es la JeVois Smart Vision Camera.

Objetivo 1

Conclusiones

- Se logró la implementación de este módulo de visión por computadora a la plataforma robótica por medio de un nodo de ROS.

Objetivo 2

Conclusiones

- Se realizaron varias pruebas con el Microsoft Kinect para Windows y se logró implementar el módulo de detección de marcadores ArUco.

Objetivo 3

Conclusiones

- Se realizaron varias pruebas para determinar las propiedades de la JeVois Smart Machine Vision Camera y confirmar la confiabilidad de los datos obtenidos de la cámara.

Objetivo 4

Recomendaciones

Recomendaciones

- Se recomienda realizar más investigación con la cámara de Microsoft Kinect para Windows. Desde Windows, donde tiene más soporte, se pueden realizar una gran variedad de aplicaciones con programas como Visual Studio.

Recomendaciones

- En otra iteración del proyecto se pudiera utilizar la visión por computadora y el control por visión para reemplazar módulos como el de ubicación en el entorno de Robotat que restringió el área de trabajo y le puede dar al robot un sentido de mayor autonomía.

Referencias

- [1] D. Forsyth y J. Ponce, Computer Vision: A Modern Approach. (Second edition). Prentice Hall, nov. de 2011, pág. 792. dirección: <https://hal.inria.fr/hal-01063327>.
- [2] OpenCV, Detection of ARUCO markers. dirección: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.
- [3] Raspberry Pi, Raspberry pi documentation, 2019. dirección: <https://www.raspberrypi.com/documentation/accessories/camera.html>.
- [4] JeVois, About JeVois, 2019. dirección: <https://www.jevoisinc.com/pages/whatis-jevois>.
- [5] Hickey, A. Buck y M. Obajemu, Kinect for windows - windows apps, 2022. dirección: <https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows>
- [6] J. I. Ramírez, “Herramienta de Software de Visión por Computadora para Aplicaciones de Robótica de Enjambre en una Mesa de Prueba - Fase III,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2021

Referencias

- [7] H. A. Klée, “Desarrollo e implementación de algoritmos de visión por computadora clásicos empleando OpenCV en sistemas embebidos.,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2021.
- [8] H. J. Sagastume, “Diseño Mecánico, Selección de Motores e Implementación de Sensores para un Robot Explorador Modular,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2021.

¡Gracias por su atención!