

# Manual de usuario de Nexos



## Contenido

Que es Nexos? .....	2
Consideraciones preliminares .....	2
Pasos para usar Nexos .....	4
Interfaz .....	7

## Que es Nexos?

Nexos es una aplicación diseñada para controlar el robot SCARA de la universidad EIA, basado en ROS2 y microROS. Su interfaz esta diseñada y programada con Tkinter, este es ejecutado dentro de un nodo de ROS2 y mediante la arquitectura de comunicación de microROS, se conecta al ESP32 del robot SCARA para controlarlo.

## Consideraciones preliminares

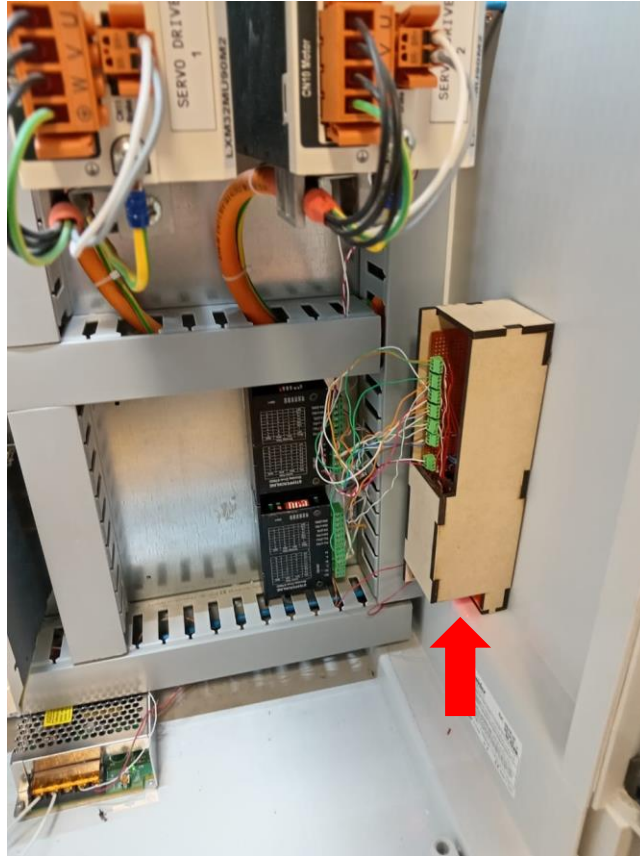
El sistema usa el computador portátil número 10 de la universidad que se puede pedir a los encargados de los laboratorios y un modem para poder funcionar, este último hace de puente entre el sistema de la computadora y el ESP32. Actualmente, el modem es algún celular que pueda crear un punto de acceso Wi-Fi, se recomiendan los Android dado que estos suelen mostrar la IP del dispositivo conectado.

Para el punto de acceso en el celular se debe primero asegurar que este no tiene ningún modo de ahorro de energía activado, dado que esto afecta la conexión con el ESP32. Una vez se tiene esto en orden, se puede proceder a la creación del punto de acceso con las siguientes características:

- Nombre de red: esp32scara.
- Contraseña: esp32scara.
- Banda de AP: 2.4 GHz.
- Seguridad: WPA2-Personal.

Tanto el ESP32 como el computador deben conectarse a la red. Si es la primera vez que se usa este celular para hacer la conexión, se debe revisar la IP del computador desde el celular y reprogramar el ESP32 con esta nueva IP de la siguiente manera:

1. Conectar el ESP32 al computador por cable USB, la parte baja del circuito tiene una abertura que permite conectarse al ESP32.



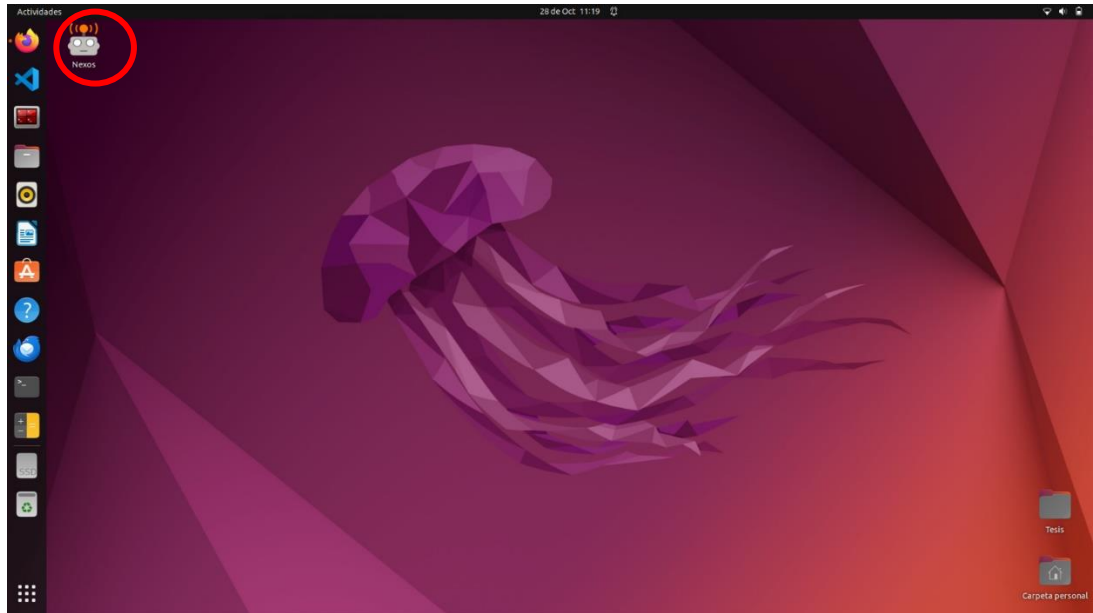
*Ilustración 1 Conexión al ESP32*

2. Para programar el microcontrolador, se debe entrar desde la terminal a la carpeta que contiene los códigos, es decir, '/home/eia/Nexos/esp32\_ws' y usar el comando 'sudo docker run -it --rm --user espidf --volume="/etc/timezone:/etc/timezone:ro" -v \$(pwd):/esp32\_ws -v /dev:/dev --privileged --workdir /esp32\_ws microros/esp-idf-microros:latest /bin/bash -c "cd esp32\_scara; idf.py menuconfig build flash monitor"'
3. Una vez se ejecuta el comando anterior y al salir el menú para su configuración, se debe ir a 'micro-ROS settings' y asegurarse de que el puerto es 8888 y que la IP del agente al que se va a conectar sea la que el modem (en este caso el celular) le asigne al computador.

En caso de que el computador no encuentre el puerto, se debe usar 'sudo chmod 666 /dev/ttyUSB0' para darle permisos de escritura y lectura al puerto USB donde está conectado el ESP32.

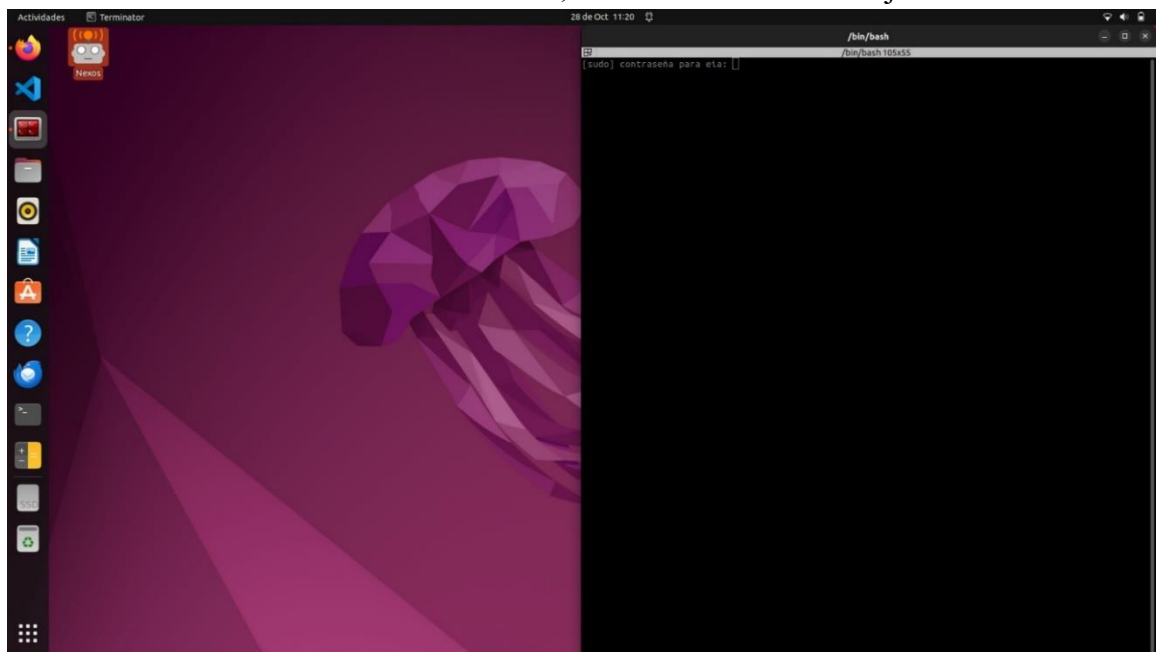
## Pasos para usar Nexos

1. Ejecutar el icono del escritorio del computador. Debemos asegurarnos de que el computador ya esté conectado a la red que se va a usar para la conexión, es decir la red a la que el ESP32 se va a conectar, en este caso 'esp32scara'.



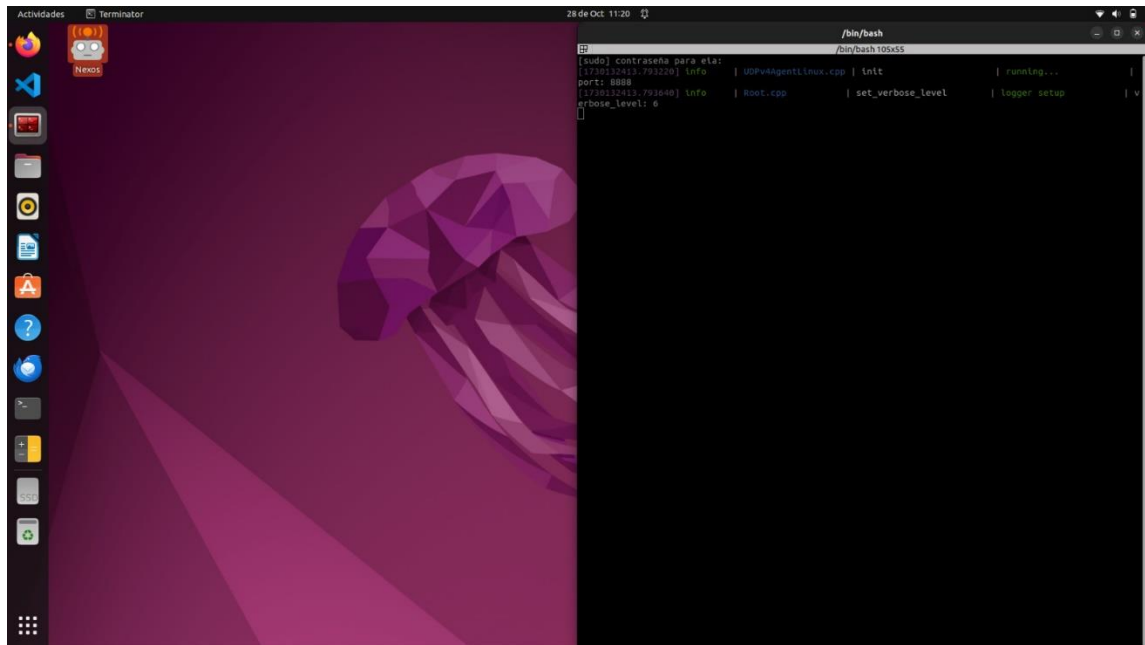
*Ilustración 2 Primer paso para usar Nexos*

2. Al abrirse la terminal escribir la contraseña, la cual es 'sistemaslomejor'.



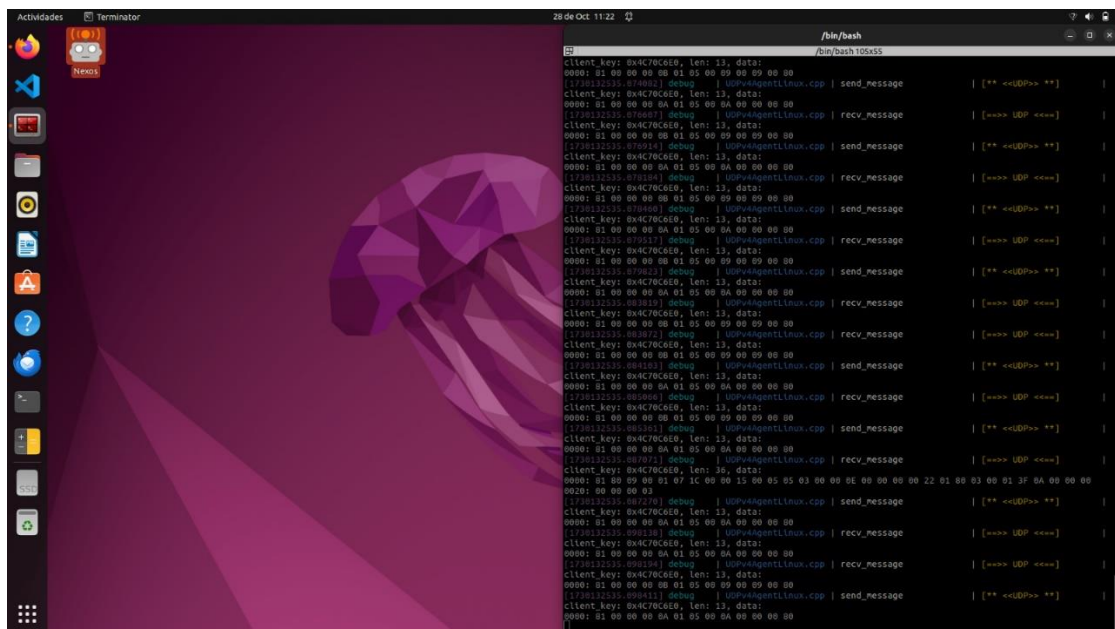
*Ilustración 3 Segundo paso para usar Nexos*

Una vez se introduce la contraseña, se ejecuta automáticamente el agente de microROS y se ve algo como sigue.



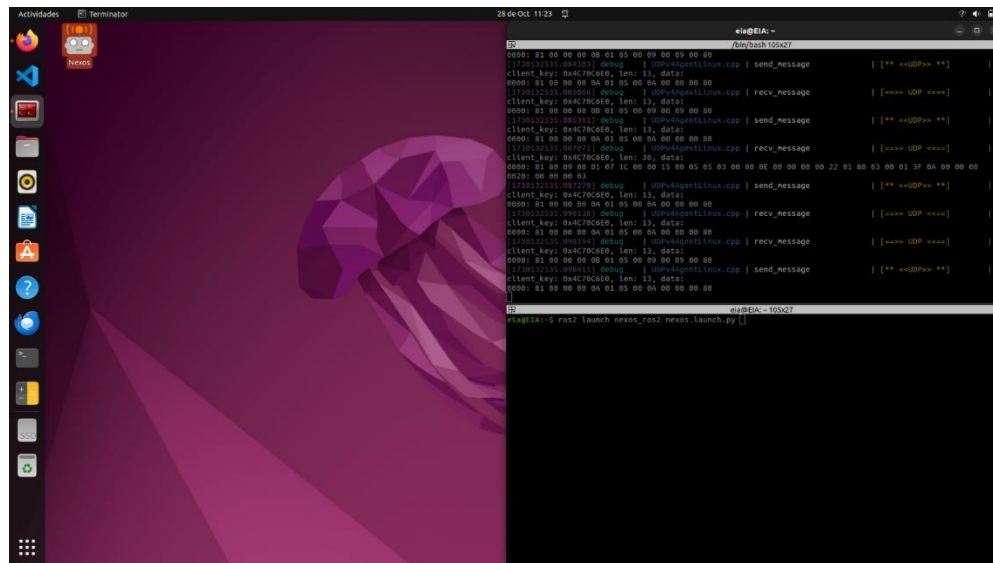
*Ilustración 4 Agente microROS iniciado*

Cuando el ESP32 logre conectar con la aplicación se debe ver algo como sigue.



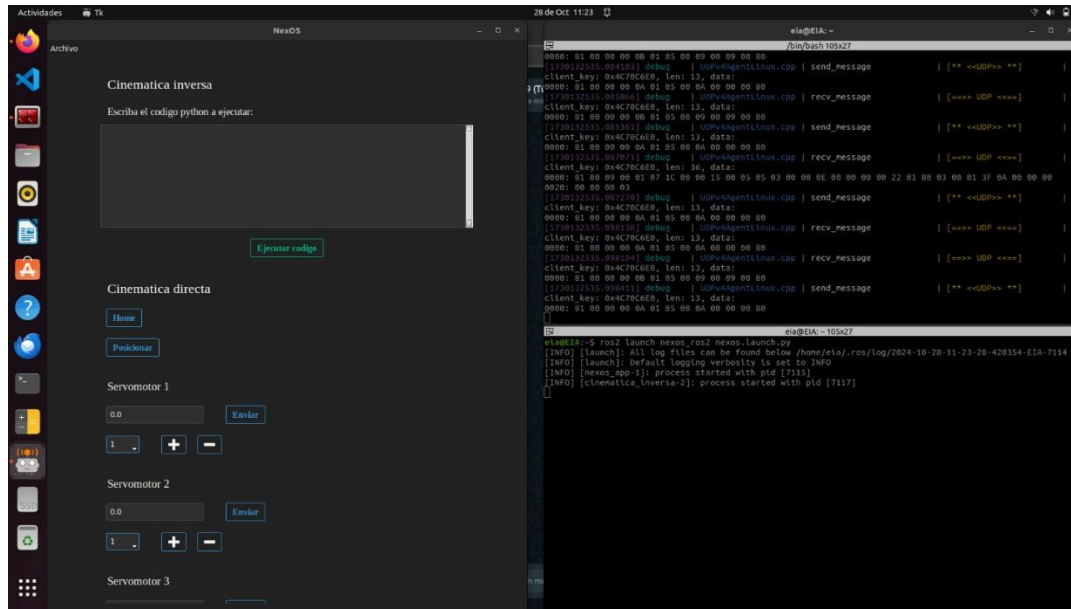
*Ilustración 5 ESP32 y Nexos conectados*

- Una vez se tiene lo anterior, se debe abrir una terminal nueva dentro de la misma ventana de Terminator, esto se hace usando el comando ‘ctrl + shift + o’. Cuando se abra la nueva terminal, se debe ejecutar un *launch* de ROS2 para abrir la aplicación y empezar a enviar comandos al robot. El comando a escribir es ‘*ros2 launch nexos\_ros2 nexos.launch.py*’, esto ejecutará dos nodos, uno de la aplicación hecha en Tkinter y otro que se encargará de calcular la cinemática inversa.



*Ilustración 6 Tercer paso para usar Nexos*

Cuando se abra la ventana de Nexos, se podrá empezar a enviar comandos, se recomienda hacer un ‘*home*’ para asegurarse de que el sistema está en la posición inicial cero. Tener en cuenta que las terminales permiten ver mensajes de error, o de información que pueden indicar cuando algo está fallando o cuando el usuario se está pasando de los límites permitidos.



*Ilustración 7 Nexos listo para usarse*

4. Cuando se finalice con el uso asegurarse de cerrar todas las terminales abiertas con ‘ctrl + c’ dado que si no se hace de esta forma la próxima vez que se quiera abrir Nexos éste no podrá usar Docker, siendo este mediante el cual se ejecuta el agente de microROS. Si lo anterior sucede, se debe usar el comando ‘sudo stop systemctl docker’ para poder volver a usar Docker con normalidad.

## Interfaz

1. Archivo: El menú archivo en la parte superior izquierda sirve para guardar o abrir archivos Python que contengan código ejecutable para la cinemática inversa del robot.
2. Cinemática inversa: Esta sección de la interfaz permite escribir código Python con el que se ejecutara la cinemática inversa del robot. Para indicarle a este una posición deseada para el efector final se utiliza la función `self.cinematica_inversa(x, y, z, z_or)` siendo ‘x’, ‘y’, ‘z’ las posiciones deseadas en centímetros y ‘z\_or’ la orientación deseada para ‘z’ en grados. Para entender mejor esto, puede usarse la demostración, que es un archivo Python con código listo para ejecutar encontrado dentro del paquete ‘nexos\_ros2/demo’ en este se puede ver que se importa la librería `time` para que, entre cada movimiento del robot, se dé un tiempo para que termine el movimiento actual antes de pedirle uno nuevo.

Una vez se tiene el código listo, simplemente hay que darle a ‘Ejecutar código’ para que el robot empiece los movimientos.

3. Cinemática directa: Esta sección se encarga del control de cada motor de forma que se pueda instruir una posición absoluta para cada uno de ellos. Cada motor tiene una entrada para escribir el valor deseado y luego al darle a 'Enviar' se ejecuta ese movimiento.

El signo '+' es para sumar a la posición actual el valor que este en la lista desplegable que tiene distintas opciones como 1, 90 o 180. De forma semejante pasa con el botón '-' que sirve para restar a la posición actual el valor en la lista desplegable. La idea de estos botones de '+' y '-' es mover el motor un poco más rápido sin tener que estar escribiendo y dando a 'Enviar' ya que una vez se presiona la suma o la resta, se envía automáticamente el nuevo valor.

Además, dentro de esta sección también se encuentran los botones 'Home' y 'Posicionar' los cuales sirven para reiniciar las posiciones de los dos motores Schneider, es decir, al darle a 'Posicionar' lo que se busca es que el robot se mueva unos pocos pasos en dirección negativa (según sus ejes de rotación) para los dos motores Schneider, esto con el fin de que los sensores TCRT5000 no detecten la cinta blanca que les indica la posición cero. Una vez hecho esto, al darle al botón 'Home' el robot comenzara a moverse en sentido positivo hasta que los sensores detecten la cinta blanca nuevamente, de forma que se configure la posición cero nuevamente.