

---

## Espacios de trabajo para OpenCV, TensorFlow y ROS2

---

### Objetivos

- Instalar y validar las herramientas de software a emplearse en el transcurso del semestre.
- Crear espacios de trabajo básicos para el desarrollo de aplicaciones con OpenCV, TensorFlow y ROS2.

### Procedimiento

En la práctica de esta semana usted instalará las herramientas de software principales (aparte de MATLAB) a emplear en el desarrollo del laboratorio durante el semestre. Luego, efectuará pruebas simples para validar las herramientas y creará espacios de trabajo básicos para el desarrollo de aplicaciones. Para lograr esto, realice lo siguiente:

1. Descargue la última versión de VirtualBox para Windows (o macOS si aplica) e instálela en su equipo. **NOTA: dado que se estarán desarrollando aplicaciones y corriendo simulaciones relativamente complejas, los requerimientos de la máquina virtual son altos. Si su computadora tiene menos de 16GB de RAM y/o un procesador muy antiguo entonces se recomienda hacer la instalación en una de las PCs del laboratorio bajo una cuenta de administrador (pida al catedrático que le de acceso).**
2. Copie el archivo de imagen, `ros2sandbox.ova`, suministrado por el catedrático de laboratorio. Importe la máquina virtual cargando el archivo de imagen anterior en VirtualBox, mediante la opción *Import Appliance* en el menú archivo. Espere a que finalice el proceso e inicie la máquina virtual haciendo doble clic en la misma o mediante el botón Start. Verifique que puede acceder al escritorio de Ubuntu, **la contraseña de administrador es mt3006.**
3. Si bien OpenCV y TensorFlow pueden instalarse en su máquina física, de este inciso en adelante se asumirá que todo se estará trabajando dentro de la máquina virtual con Ubuntu en la terminal (puede revisar los comandos más utilizados en la cheat sheet que se adjunta a este documento o bien en el siguiente video). Se recomienda trabajarlo de esta manera para tener un entorno de desarrollo aislado y estándar (entre todos los participantes del curso) que no presente problemas con respecto a dependencias, pero queda a su discreción si quiere instalar estas librerías sobre una versión de Python en su máquina física. ROS2 sí deberá instalarse sólo en la máquina virtual.

4. Abra una terminal desde el escritorio de Ubuntu (CTRL+ALT+T) y ejecute los comandos `sudo apt-get update` y `sudo apt-get upgrade` para actualizar los paquetes del sistema operativo. Luego, regrese al home directory (`cd`), cree una carpeta llamada `opencvws` (`mkdir opencvws`).
5. Lea la guía para crear y utilizar *virtual environments* (`python3 -m pip install --user ... virtualenv`) en Python. Los virtual environments se emplean para crear entornos de desarrollo locales de Python, en donde la instalación de paquetes no generará conflictos con la instalación global de Python. Ingrese a la carpeta que creó en el inciso anterior (`cd opencvws`) y cree un virtual environment (`python3 -m venv env`). Active el environment (`source ... env/bin/activate`) e instale OpenCV para Python (`pip install opencv-contrib-python`).
6. Dentro de la misma carpeta, cree un archivo de prueba (`touch test.py`) y edítelo ya sea mediante el editor de texto por defecto (`gedit test.py`) o mediante VSCode (`code ... test.py`, aunque primero debe instalarlo). Añada las líneas de código

```
1 import cv2
2 print(cv2.__version__)
```

guarde, cierre el archivo y desde la terminal ejecute el script mediante `python3 test.py`. El script deberá correr sin errores e imprimir la versión empleada de OpenCV. Desactive el virtual environment (`deactivate`).

7. **(Opcional)** Una desventaja de emplear virtual environments es que deben activarse y desactivarse antes de poder emplearlos y/o para movilizarse de un environment a otro. Esto puede resolverse añadiendo un *shebang* en la primera línea del script de Python que apunte a la versión de Python a emplear, es decir:

```
1 #!/home/ros2admin/opencvws/env/bin/python
2 import cv2
3 print(cv2.__version__)
```

Finalmente, otorgue permisos de ejecutable al script (`chmod +x test.py`, debe correrse una única vez por script) y córralo directamente desde la línea de comando (`./test.py`).

8. Regrese al home directory y repita los incisos 4, 5 y 6 pero para un nuevo directorio llamado `tfws`. Dentro de este, de nuevo cree y active un virtual environment e instale la versión CPU de TensorFlow (`pip install tensorflow-cpu`). Cree de nuevo un archivo de prueba (`test.py`) y coloque la línea de código

```
1 # De nuevo puede colocar (o no) el shebang pero cambiando el directorio de ...
   opencvws por el de tfws
2 import tensorflow as tf
```

Ejecute el script y este deberá correr sin errores (a lo sumo podrá imprimir alguna advertencia).

9. Siga la guía de instalación para ROS2 Humble e instale la versión *Desktop Install* (la que contiene ROS, RViz junto a los demos y tutoriales) junto con las *Development tools*. Luego que finalice el proceso, valide la instalación corriendo el ejemplo *Talker-listener*.

10. Durante la calificación, deberá mostrar al catedrático de laboratorio lo siguiente:

- La estructura de carpetas y archivos del home directory y de las carpetas `opencvws` y `tfws`.
- Los resultados de ejecutar los dos archivos de prueba `test.py` creados dentro de sus virtual environments respectivos.
- El resultado de correr el demo Talker-listener en ROS2.

Cuando esté satisfecho con sus resultados, presente los mismos al catedrático del laboratorio y responda a las preguntas que se le hagan. Recuerde que entregas tardías representan una penalización del 25 % por semana.