



## Práctica 6: Integración de un Chatbot en ROS 2

Robótica de Servicios  
Máster en Robótica e Inteligencia Artificial

---

### Resumen

Esta práctica tiene como objetivo que los estudiantes integren y configuren un chatbot en un entorno ROS 2, utilizando el repositorio `chatbot_ros` disponible en GitHub. Los objetivos específicos son:

- Comprender la estructura y dependencias de un paquete ROS 2 complejo.
- Instalar y configurar las dependencias necesarias para el funcionamiento del chatbot.
- Ejecutar y probar el chatbot en un entorno ROS 2.
- Personalizar y extender las funcionalidades del chatbot según requerimientos específicos.

### Requisitos

Para llevar a cabo esta práctica, se requiere el siguiente entorno de trabajo:

- Sistema operativo: Ubuntu 22.04
- ROS 2 Humble Hawksbill instalado y configurado.
- Acceso a Internet para clonar repositorios y descargar dependencias.

### Instrucciones

#### 1. Clonar el repositorio `chatbot_ros`:

- Navega al directorio de trabajo de tu espacio de trabajo de ROS 2:

```
1 cd ~/ros2_ws/src
2
```

- Clona el repositorio:

```
1 git clone https://github.com/mgonzsl3/chatbot_ros
2
```

#### 2. Instalar las dependencias:

- Importa las dependencias utilizando `vcstool`:

```
1 vcs import < chatbot_ros/dependencies.repos
2
```

- Instala las dependencias de Python listadas en `requirements.txt`:

```
1 pip3 install -r requirements.txt
2
```

- Navega al directorio raíz del espacio de trabajo:

```
1 cd ~/ros2_ws
2
```

- Instala las dependencias de ROS 2:

```
1 rosdep install --from-paths src --ignore-src -r -y
2
```

### 3. Compilar el paquete:

- Compila el espacio de trabajo utilizando colcon:

```
1 colcon build --cmake-args -DGGML_CUDA=ON
2
```

- Fuente el entorno de trabajo:

```
1 source install/setup.bash
2
```

### 4. Ejecutar el chatbot:

- Lanza el chatbot utilizando el siguiente comando:

```
1 ros2 launch chatbot_bringup chatbot.launch.py
2
```

### 5. Personalizar el chatbot:

- Explora los archivos de configuración y scripts en el paquete `chatbot_ros` para entender su funcionamiento.
- Modifica las respuestas o comportamientos del chatbot a tu gusto. Por ejemplo, ¿Cómo realizarías un juego de pregunta-respuesta?

## Entrega

Al finalizar la práctica, cada estudiante deberá entregar un informe que incluya:

- Descripción del proceso de instalación y configuración realizado.
- Capturas de pantalla o registros que demuestren la ejecución exitosa del chatbot.
- Reflexión sobre los desafíos encontrados y cómo fueron superados.

## Recursos Adicionales

Para obtener más información sobre el paquete `chatbot_ros` y sus dependencias, se recomienda revisar la documentación oficial en el repositorio de GitHub: [https://github.com/mgonzs13/chatbot\\_ros](https://github.com/mgonzs13/chatbot_ros).

Además, la documentación de ROS 2 y los tutoriales disponibles en <https://docs.ros.org/en/humble/index.html> pueden ser de ayuda para comprender conceptos y comandos utilizados en esta práctica.

RASA es una plataforma de código abierto para desarrollar asistentes conversacionales basados en inteligencia artificial, que incluye capacidades de procesamiento de lenguaje natural (NLP) y manejo de diálogos. En el contexto de la práctica de `chatbot_ros` en ROS 2, RASA puede usarse como motor del chatbot, proporcionando el sistema de interpretación del lenguaje natural (NLU) y la lógica de diálogo (core) necesarios para entender e interactuar de manera efectiva con los usuarios. <https://github.com/RasaHQ/rasa>

## Referencias