

## :: U4 ::

# Introducción a la programación robótica



## 3. Comunicación entre nodos

**Curso 2024-25**

# Comunicación entre nodos (I)



**CIPFP Mislata**  
Centre Integrat Públic  
Formació Professional Superior

1. ¿Qué es el “paso de mensajes”?
  - a. Concepto de nodo
  - b. Concepto patrón publicador-suscriptor
  - c. “Tópicos”
  - d. Mensajes (tipos de datos)
2. Desarrollo de un publicador-suscriptor simple
3. Turtlebot 3D
  - a. Introducción al simulador Gazebo

# Comunicación entre nodos (I)



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

1. ¿Qué es el “paso de mensajes”?
  - a. Concepto de nodo
  - b. Concepto patrón publicador-suscriptor
  - c. “Tópicos”
  - d. Mensajes (tipos de datos)
2. Desarrollo de un publicador-suscriptor simple
3. Turtlebot 3D
  - a. Introducción al simulador Gazebo

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

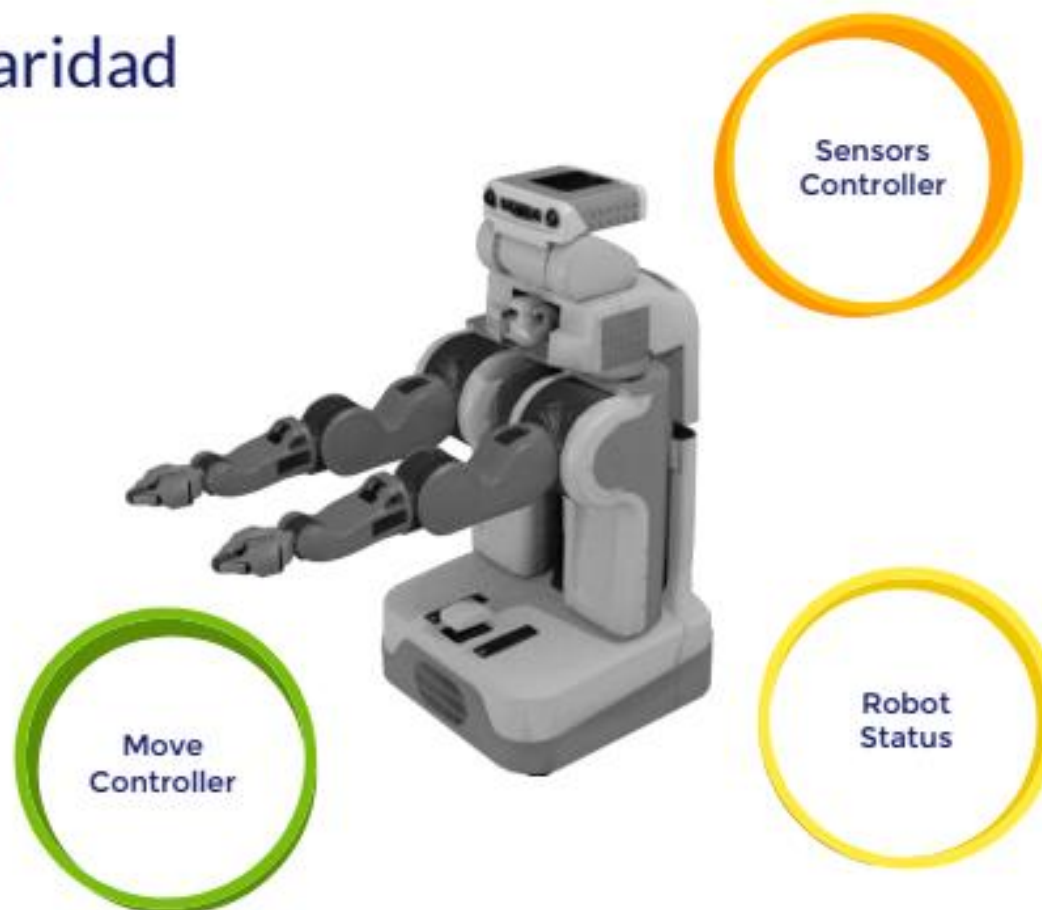
- El concepto de **nodo** es central en la comunicación de procesos.
- Un nodo en ROS es un proceso que realiza cualquier tipo de computación.
- El objetivo de dichos procesos es tener un nivel de **granularidad** lo más fina posible.

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

Granularidad  
**gruesa**



# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

Granularidad  
**fina**



# Comunicación entre nodos (I)



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

1. ¿Qué es el “paso de mensajes”?
  - a. Concepto de nodo
  - b. Concepto patrón publicador-suscriptor
  - c. “Tópicos”
  - d. Mensajes (tipos de datos)
1. Desarrollo de un publicador-suscriptor simple



# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

- **Concepto del patrón *Publish-Subscribe***
  - Los nodos productores de información se denominan **publicadores** (o fuentes).
  - Los nodos consumidores de información se denominan **suscriptores** (o sumideros)



# 1. ¿Qué es el “paso de mensajes”?



- **Concepto del patrón *Publish-Subscribe***

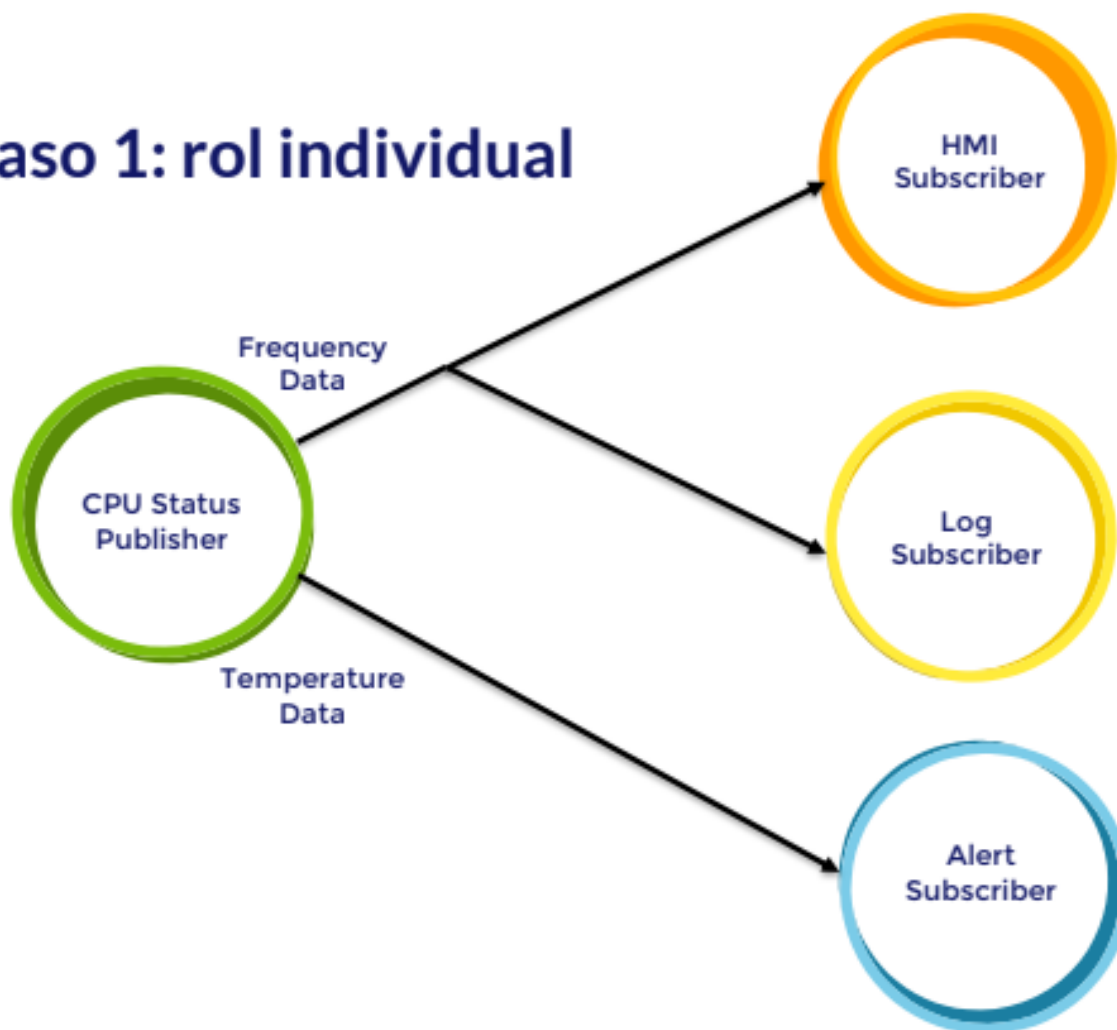
- Los nodos subscriptores se suscriben a datos de los nodos publicadores
- Un dato producido por un publicador le llega a todos y cada uno de sus subscriptores y no más.
- Un nodo puede actuar como publicador y suscriptor a la vez.

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## Caso 1: rol individual

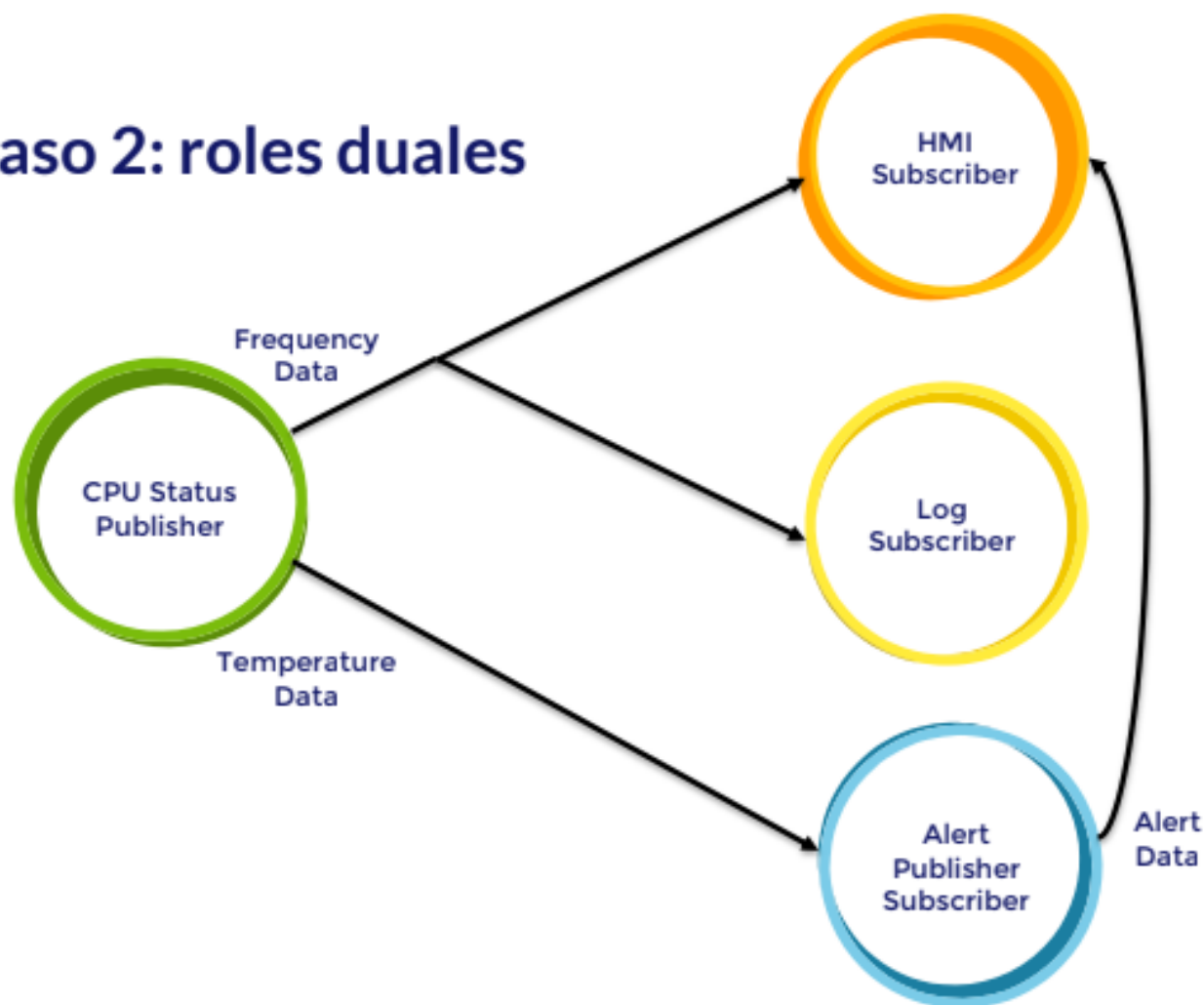


# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## Caso 2: roles duales



# Comunicación entre nodos (I)



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

1. ¿Qué es el “paso de mensajes”?
  - a. Concepto de nodo
  - b. Concepto patrón publicador-suscriptor
  - c. “Tópicos”
  - d. Mensajes (tipos de datos)
1. Desarrollo de un publicador-suscriptor simple

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## ¿Qué es un “tópico”?

- un tópico puede verse como un bus de datos sobre el que los nodos intercambian mensajes de un tipo.
- cada tópico recibe un nombre de forma unívoca.
- nodos interesados en un tipo de datos se suscriben al tópico correspondiente.

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## ¿Qué es un “tópico”?

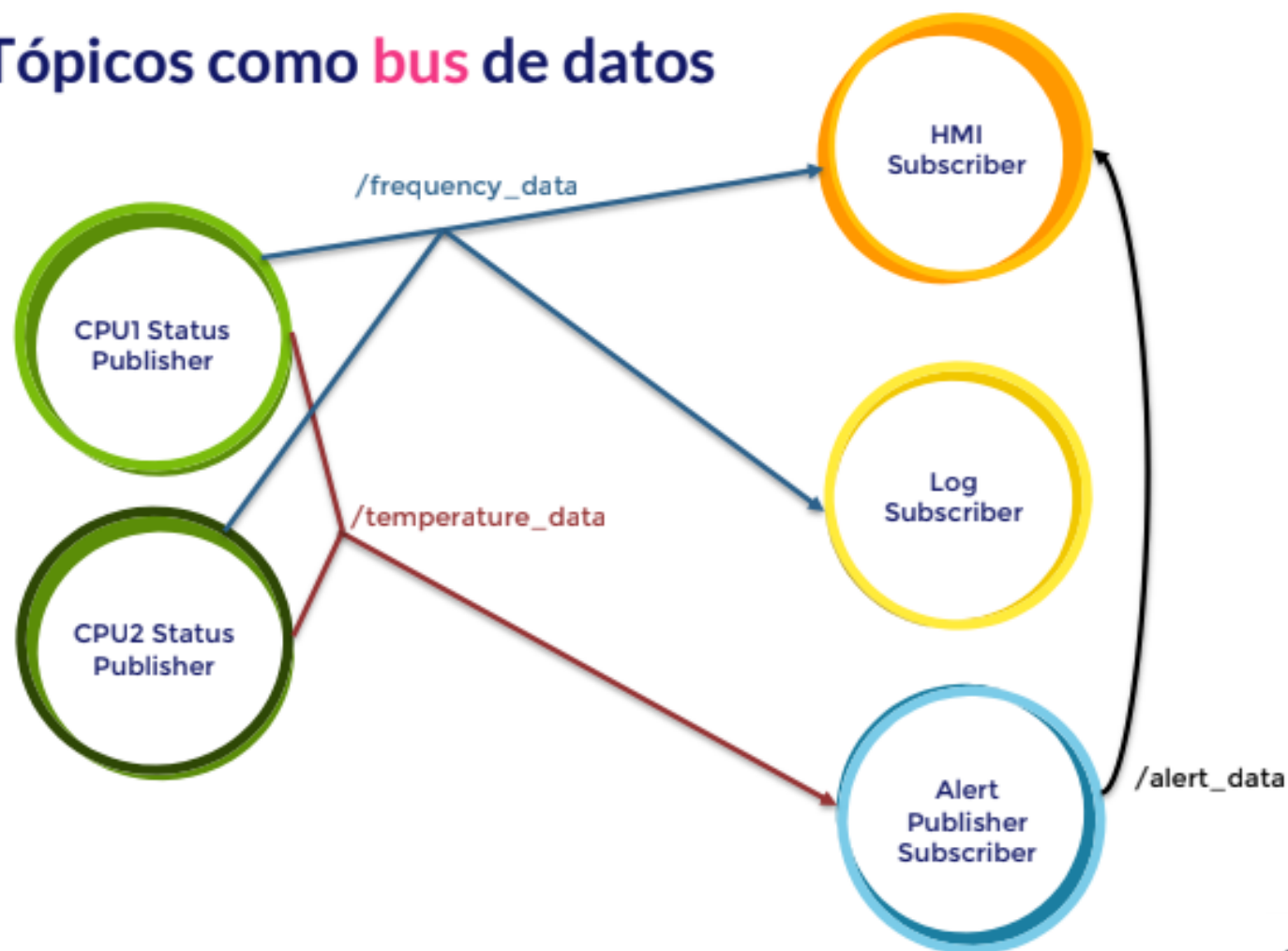
- nodos que generan datos publican por el tópico correspondiente.
- pueden existir varios publicadores/suscriptores sobre el mismo tópico.
- OJO! Los tópicos están destinados a una comunicación unidireccional!

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## Tópicos como **bus** de datos





# Comunicación entre nodos (I)



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

1. ¿Qué es el “paso de mensajes”?
  - a. Concepto de nodo
  - b. Concepto patrón publicador-suscriptor
  - c. “Tópicos”
  - d. Mensajes (tipos de datos)
1. Desarrollo de un publicador-suscriptor simple

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## ¿Qué es un mensaje?

- un mensaje en ROS es el dato que viaja a través de los “tópicos”.
- todo mensaje pertenece a un tipo de datos, que debe coincidir con el tipo de datos del “tópico” por el que viaja.
- el tipo de datos del tópico es definido por el publicador.

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## Ejemplo: paquete `std_msgs`

- contiene la definición de múltiples mensajes estándar de ROS, incluyendo mensajes que representan tipos de datos primitivos.
- consultar la página de referencia:  
[http://wiki.ros.org/std\\_msgs](http://wiki.ros.org/std_msgs).

# 1. ¿Qué es el “paso de mensajes”?



tipos  
básicos  
que se  
usan para  
definir los  
mensajes



**Primitive Type**

**Serialization**







Conversión  
de datos  
para  
transmisión

**C++**

**Python2**

**Python3**

bool (1)	unsigned 8-bit int	uint8_t (2)	bool	
int8	signed 8-bit int	int8_t	int	
uint8	unsigned 8-bit int	uint8_t	int (3)	
int16	signed 16-bit int	int16_t	int	
uint16	unsigned 16-bit int	uint16_t	int	
int32	signed 32-bit int	int32_t	int	
uint32	unsigned 32-bit int	uint32_t	int	
int64	signed 64-bit int	int64_t	long	int
uint64	unsigned 64-bit int	uint64_t	long	int
float32	32-bit IEEE float	float	float	
float64	64-bit IEEE float	double	float	
string	ascii string (4)	std::string	str	bytes
time	secs/nsecs unsigned 32-bit ints	 ros::Time	 rospy.Time	
duration	secs/nsecs signed 32-bit ints	 ros::Duration	 rospy.Duration	

Fuente: <http://wiki.ros.org/msg>

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## Ejemplo: paquete `std_msgs`

- cada mensaje viene definido en un fichero de texto de extensión `.msg`.
- el fichero contiene los campos de datos del mensaje, que pueden ser de tipo primitivo o incluso otros mensajes.
- podemos definir nuestros propios mensajes

# 1. ¿Qué es el “paso de mensajes”?



```
robot@alumnx-MIA: ~/catkin_ws
roscore http://alumnx-MIA:11311/ x robot@alumnx-MIA: ~/catkin_ws x
robot@alumnx-MIA:~/catkin_ws$ rosmmsg package std_msgs
std_msgs/Bool
std_msgs/Byte
std_msgs/ByteMultiArray
std_msgs/Char
std_msgs/ColorRGBA
std_msgs/Duration
std_msgs/Empty
std_msgs/Float32
std_msgs/Float32MultiArray
std_msgs/Float64
std_msgs/Float64MultiArray
std_msgs/Header
std_msgs/Int16
std_msgs/Int16MultiArray
std_msgs/Int32
std_msgs/Int32MultiArray
std_msgs/Int64
std_msgs/Int64MultiArray
std_msgs/Int8
std_msgs/Int8MultiArray
std_msgs/MultiArrayDimension
std_msgs/MultiArrayLayout
std_msgs/String
std_msgs/Time
std_msgs/UInt16
std_msgs/UInt16MultiArray
std_msgs/UInt32
std_msgs/UInt32MultiArray
std_msgs/UInt64
std_msgs/UInt64MultiArray
std_msgs/UInt8
```



Comando **rosmmsg**:  
con el argumento  
*package* se  
obtienen todos  
los mensajes  
definidos en el  
paquete.



# 1. ¿Qué es el “paso de mensajes”?



```
robot@alumnx-MIA:/opt/ros/indigo/share/std_msgs/msg$ ll
total 132
drwxr-xr-x 2 root root 4096 jul 12 2021 ./
drwxr-xr-x 4 root root 4096 jul 12 2021 ../
-rw-r--r-- 1 root root 9 mar 10 2016 Bool.msg
-rw-r--r-- 1 root root 10 mar 10 2016 Byte.msg
-rw-r--r-- 1 root root 210 mar 10 2016 ByteMultiArray.msg
-rw-r--r-- 1 root root 9 mar 10 2016 Char.msg
-rw-r--r-- 1 root root 40 mar 10 2016 ColorRGBA.msg
-rw-r--r-- 1 root root 14 mar 10 2016 Duration.msg
-rw-r--r-- 1 root root 0 mar 10 2016 Empty.msg
-rw-r--r-- 1 root root 12 mar 10 2016 Float32.msg
-rw-r--r-- 1 root root 210 mar 10 2016 Float32MultiArray.msg
-rw-r--r-- 1 root root 12 mar 10 2016 Float64.msg
-rw-r--r-- 1 root root 210 mar 10 2016 Float64MultiArray.msg
-rw-r--r-- 1 root root 583 mar 10 2016 Header.msg
-rw-r--r-- 1 root root 11 mar 10 2016 Int16.msg
-rw-r--r-- 1 root root 210 mar 10 2016 Int16MultiArray.msg
-rw-r--r-- 1 root root 10 mar 10 2016 Int32.msg
-rw-r--r-- 1 root root 210 mar 10 2016 Int32MultiArray.msg
-rw-r--r-- 1 root root 10 mar 10 2016 Int64.msg
-rw-r--r-- 1 root root 210 mar 10 2016 Int64MultiArray.msg
-rw-r--r-- 1 root root 10 mar 10 2016 Int8.msg
-rw-r--r-- 1 root root 210 mar 10 2016 Int8MultiArray.msg
-rw-r--r-- 1 root root 141 mar 10 2016 MultiArrayDimension.msg
-rw-r--r-- 1 root root 911 mar 10 2016 MultiArrayLayout.msg
-rw-r--r-- 1 root root 12 mar 10 2016 String.msg
-rw-r--r-- 1 root root 10 mar 10 2016 Time.msg
-rw-r--r-- 1 root root 12 mar 10 2016 UInt16.msg
```



Cada mensaje se define en un fichero de texto con extensión **.msg**.



# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

```
robot@alumnx-MIA: /opt/ros/indigo/share/std_msgs/msg
roscore http://alumnx-MIA:11311/ x robot@alumnx-MIA: /opt/ros/indigo/share/std_ms... x
robot@alumnx-MIA: /opt/ros/indigo/share/std_msgs/msg$ rosmmsg show std_msgs/Float64
float64 data
robot@alumnx-MIA: /opt/ros/indigo/share/std_msgs/msg$
```



Comando **rosmmsg**:  
con el argumento  
*show* se obtienen los  
campos de un  
mensaje.

El fichero correspondiente  
contiene la misma  
información

# 1. ¿Qué es el “paso de mensajes”?



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

## NOTA:

- cuando compilamos, los mensajes definidos en ficheros **.msg** son traducidos a código **Python** y **C++**.
- los tipos de datos que aparecen en los ficheros **.msg** representan tipos primitivos que posteriormente son compilados a tipos específicos de cada lenguaje de forma automática.
- esto permite definir un mensaje sin tener que crear el código Python y C++.

## 2. Desarrollo de un publicador-suscriptor



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

### Ejemplo básico en la [página oficial](#)

ROS.org

[About](#) | [Support](#) | [Discussion Forum](#) | [Service Status](#) | [Q&A answers.ros.org](#)

Search:

Documentation

Browse Software

News

Download

[ROS/ Tutorials/ WritingPublisherSubscriber\(python\)](#)

**Note:** This tutorial assumes that you have completed the previous tutorials: creating a ROS msg and srv.

Please ask about problems and questions regarding this tutorial on [answers.ros.org](#). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

### Writing a Simple Publisher and Subscriber (Python)

**Description:** This tutorial covers how to write a publisher and subscriber node in python.

**Tutorial Level:** BEGINNER

**Next Tutorial:** [Examining the simple publisher and subscriber](#)

catkin

roscpp

#### Tabla de Contenidos

1. Writing the Publisher Node
  1. The Code
  2. The Code Explained
2. Writing the Subscriber Node
  1. The Code
  2. The Code Explained
3. Building your nodes
  1. Additional Resources
4. Video Demonstration

#### ROS 2 Documentation

The ROS Wiki is for ROS 1. Are you using ROS 2 (Dashing/Foxy/Rolling)? Check out the [ROS 2 Documentation](#)

#### Wiki

[Distributions](#)  
[ROS/Installation](#)  
[ROS/Tutorials](#)  
[RecentChanges](#)  
[WritingPublisherSubscriber\(python\)](#)

#### Página

[Página inmutable](#)  
[Información](#)  
[Adjuntos](#)

#### Más Acciones

Texto sin formato

Hacer

#### Usuario

[Ingresar](#)

ROS  
Robot Operating System

## 2. Desarrollo de un publicador-suscriptor



CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

### Práctica 2

- Realiza la práctica **“ROBOT - P2 :: Creación de un publicador-suscriptor”**, cuyo enunciado encontrarás en el moodle

### 3. Turtlebot 3D

#### Gazebo

Simulador 3D que permite probar algoritmos rápidamente, diseñar robots, realizar pruebas y entrenar el sistema de IA utilizando escenarios realistas



## 3. Turtlebot 3D

### Gazebo

#### a. Instalación

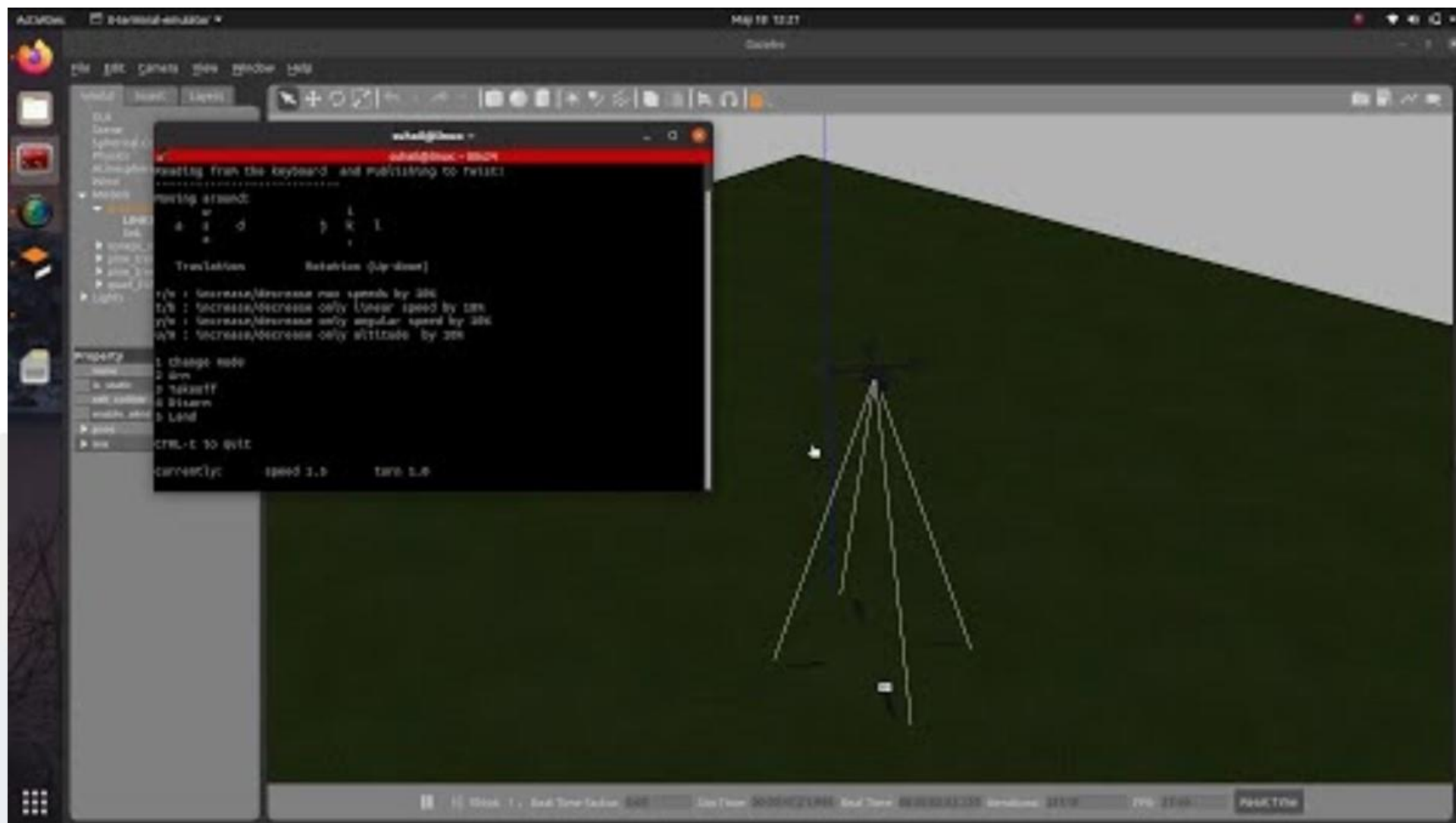
```
# apt-get install ros-noetic-gazebo-ros-pkgs ros-noetic-  
gazebo-ros-control  
# source ~/catkin_ws/devel/setup.bash
```

#### b. Ejecución

```
# roscore & rosrun gazebo_ros gazebo  
  
... o alternativamente  
  
# roslaunch gazebo_ros empty_world.launch
```



# 3. Turtlebot 3D





## 3. Turtlebot 3D

### Gazebo

#### c. Movimiento del robot

```
# roslaunch turtlebot_teleop keyboard_teleop.launch
```



## 3. Turtlebot 3D

### Gazebo

#### d. Publicación de tópicos

```
ros@ros-VirtualBox:~/catkin_ws$ rostopic list
/clock
/cmd_vel
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/performance_metrics
/gazebo/set_link_state
/gazebo/set_model_state
/imu
/joint_states
/odom
/rosout
/rosout_agg
/scan
/tf
```



## 3. Turtlebot 3D

### Práctica 3

- o Escribir un script bajo el modelo publicador-suscriptor para la monitorización de la posición de un robot en relación a otros objetos en un escenario dado.

