CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

:: U3 ::

Introducción a la programación robótica

ROS
Robot Operating System

2. Preparación y gestión del entorno

Curso 2024-25

Preparación del entorno

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

1. Instalación
2. Estructura de un “workspace”
3. Navegación por el sistema de ficheros
4. Creación de paquetes
5. Entornos de desarrollo

ROS
Robot Operating System

1. Instalación

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- En cualquier de las versiones de ROS implica:
  a. instalar los paquetes base
  b. inicializar **rosdep**
     - necesario para instalación de dependencias previa compilación
     - necesario para ejecutar algunos componentes principales en ROS
  c. añadir el setup.bash al entorno de shell del usuario

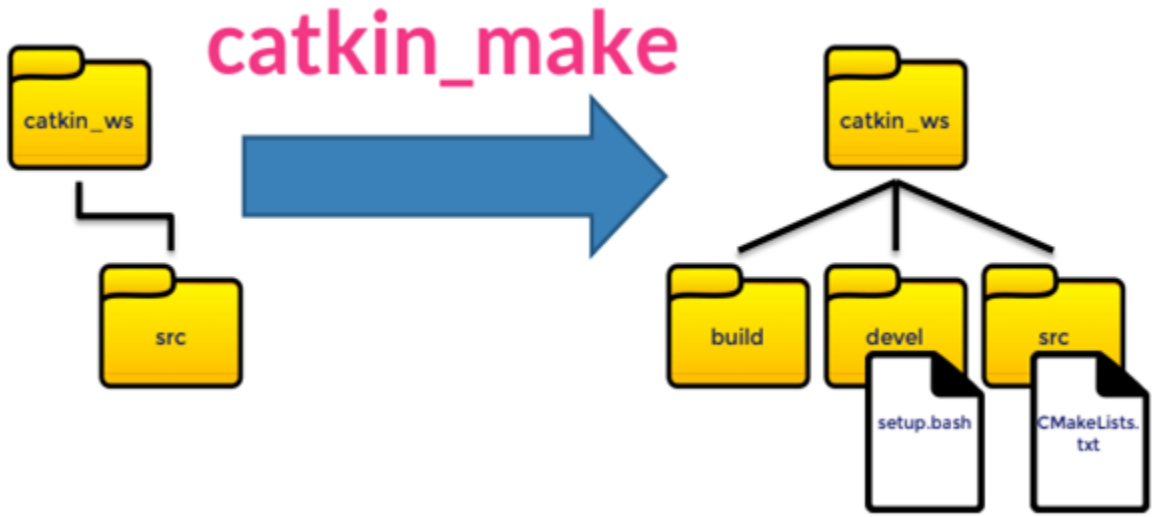
- Ejemplo de pasos en **Indigo** / **Noetic**

ROS
Robot Operating System

2. Estructura de un *workspace*

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- **src**: contendrá los paquetes que desarrollemos y vayamos a compilar.
- **build**: contendrá los ficheros intermedios generados al compilar.
- **devel**: contendrá el **setup.bash**, los binarios y librerías generados de los paquetes.



ROS
Robot Operating System

2. Estructura de un *workspace*

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Creación del **workspace** (área de trabajo):

```
# mkdir -p ~/catkin_ws/src
# cd ~/catkin_ws/
# catkin_make          // genera la estructura del workspace
# echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
# source ~/.bashrc
```

ROS
Robot Operating System

2. Estructura de un *workspace*

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Comprobación de la operatividad del **workspace**

 # roscd

(Lleva al directorio "catkin_ws")

 # catkin_make

(herramienta de compilcación por línea de comandos que facilita el flujo de trabajo estándar de catkin)

Robot Operating System

3. Navegación sistema ficheros

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Comandos útiles

roscd

- Permite cambiar desde el directorio actual hacia un paquete.

  Análogo al comando 'cd' de Linux.

Uso:

roscd [nombre_del_paquete[/subdir]]

ROS
Robot Operating System

3. Navegación sistema ficheros

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Comandos útiles

rosls

- Permite listar los contenidos de un paquete por nombre en lugar de por ruta. Análogo al comando 'ls' de Linux.

Uso:

rosls [nombre_del_paquete[/subdir]]

ROS
Robot Operating System

4. Creación de paquetes

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

¿Qué es un paquete en ROS?

"En ROS, un paquete es la unidad de organización software básica de código de programación ROS. Cada paquete puede contener librerías, ejecutables y scripts, entre otros, siguiendo un esquema de organización determinado"

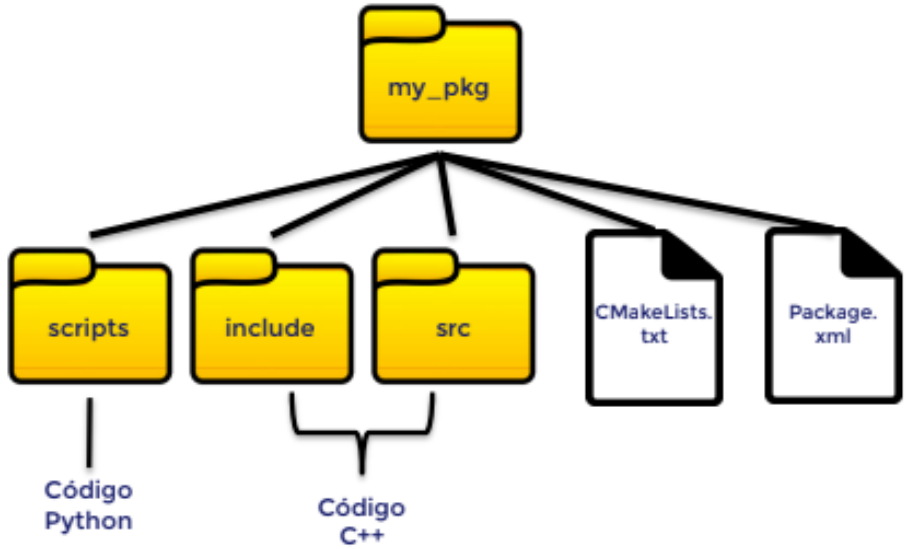
ROS
Robot Operating System

4. Creación de paquetes

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- **src**: contiene los paquetes Carpeta src: contendrá el código C++ de nuestras aplicaciones.
- **include**: contiene las cabeceras de nuestro código C++.
- **scripts**: contendrá el código Python de nuestras aplicaciones.
- **package.xml**: define las dependencias de nuestro paquete.
- **CMakeLists.txt**: define cómo compilar el código, generar mensajes, servicios, etc.

Estructura de un paquete



ROS
Robot Operating System

3. Navegación sistema ficheros

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Comandos útiles

rospack

- Con el argumento **find**, imprime la ruta de un paquete.
- Con el argumento **profile**, imprime el tiempo que lleva navegar a lo largo de la estructura de directorios.

Uso:

rospack find [nombre_del_paquete]

rospack profile

ROS
Robot Operating System

4. Creación de paquetes

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Comando → **catkin_create_package**

Uso

catkin_create_pkg <package_name> [depend1] [depend2] ...

Argumentos

nombre del paquete y lista de dependencias

Indicaciones

Lanzarlo dentro de la carpeta src del workspace!!

ROS
Robot Operating System

4. Creación de paquetes

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- **Ejemplo**: creación de un paquete **my_pkg** con dependencias de **std_msgs, rospy** y **roscpp**

```
# You should have created this in the Creating a
Workspace Tutorial

$ cd ~/catkin_ws/src

$ catkin_create_pkg my_pkg std_msgs rospy roscpp
```

ROS
Robot Operating System

4. Creación de paquetes

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- Las dependencias son añadidas en "**package.xml**" como dependencias de compilación **(build_depend)** o de ejecución **(exec_depend).**

*package.xml ×

```
<?xml version="1.0"?>
<package format="2">
  <name>my_pkg</name>
  <version>0.0.0</version>
  <description>The my_pkg package</description>

  <maintainer email="robot@todo.todo">robot</maintainer>
  <license>TODO</license>

  <buildtool_depend>catkin</buildtool_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>std_msgs</build_depend>
  <build_export_depend>roscpp</build_export_depend>
  <build_export_depend>rospy</build_export_depend>
  <build_export_depend>std_msgs</build_export_depend>
  <exec_depend>roscpp</exec_depend>
  <exec_depend>rospy</exec_depend>
  <exec_depend>std_msgs</exec_depend>

  <!-- The export tag contains other, unspecified, tags -->
  <export>
    <!-- Other tools can request additional information be placed here
-->
  </export>
</package>
```

ROS
Robot Operating System

4. Creación de paquetes

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- El fichero **CMakeLists.txt** es generado a modo de plantilla.
- Debe **modificarse** para que nuestro código pueda ser compilado (cuando ejecutamos **catkin_make**).

CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8.3)
project(my_pkg)

## Compile as C++11, supported in ROS Kinetic and newer
# add_compile_options(-std=c++11)

## Find catkin macros and libraries
## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
## is used, also find other catkin packages
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
)

## System dependencies are found with CMake's conventions
# find_package(Boost REQUIRED COMPONENTS system)


## Uncomment this if the package has a setup.py. This macro ensures
## modules and global scripts declared therein get installed
## See http://ros.org/doc/api/catkin/html/user_guide/setup_dot_py.html
# catkin_python_setup()

################################################
## Declare ROS messages, services and actions ##
################################################
```

ROS
Robot Operating System

5. Entornos de desarrollo

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

ROS
Robot Operating System

Tiempo para ejercitarse

CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- **Práctica 1**
  - Realiza la práctica "**ROBOT - P1 :: Explorar y usar el paquete Turtlesim"**, cuyo enunciado encontrarás en el moodle

ROS
Robot Operating System