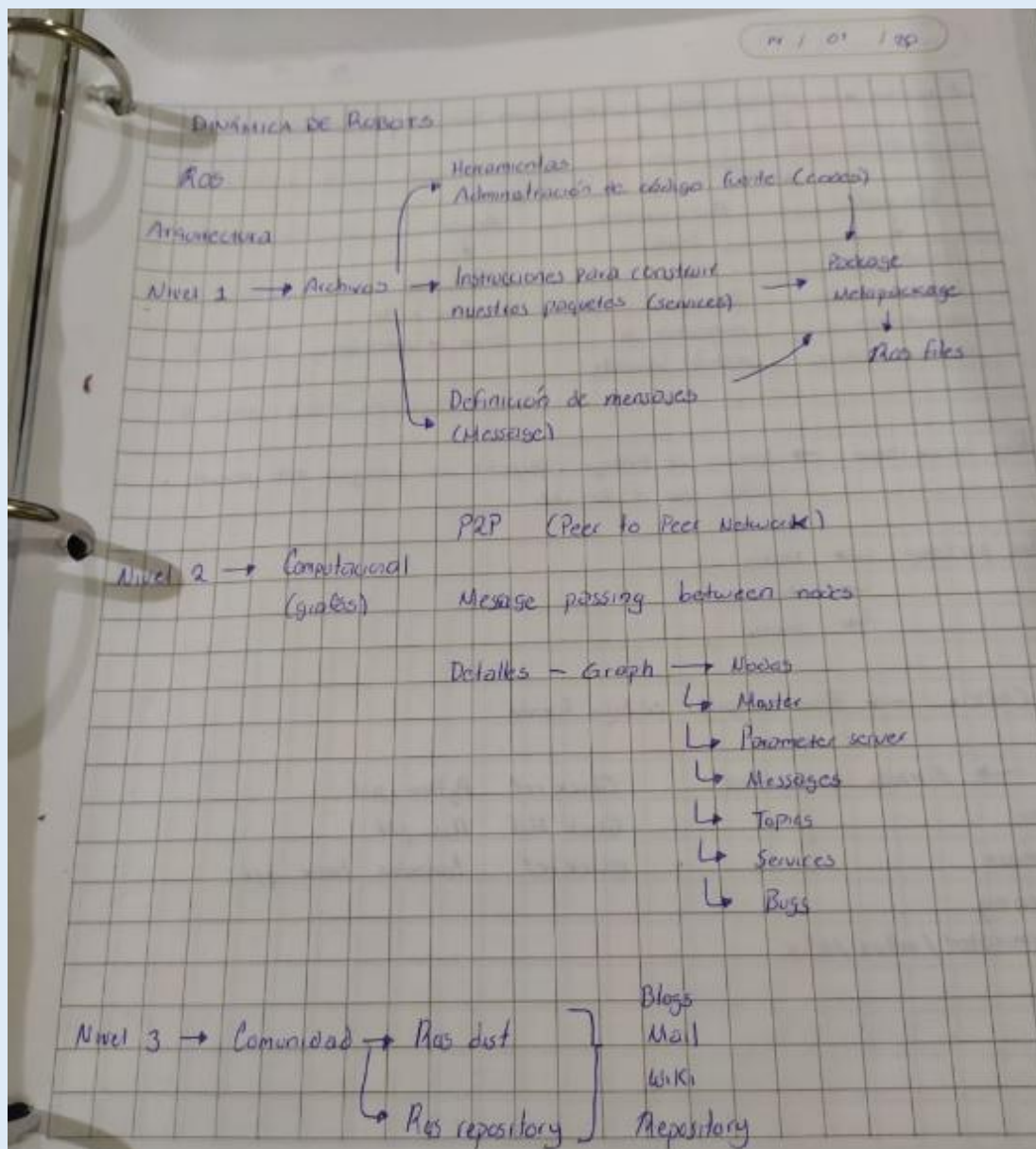
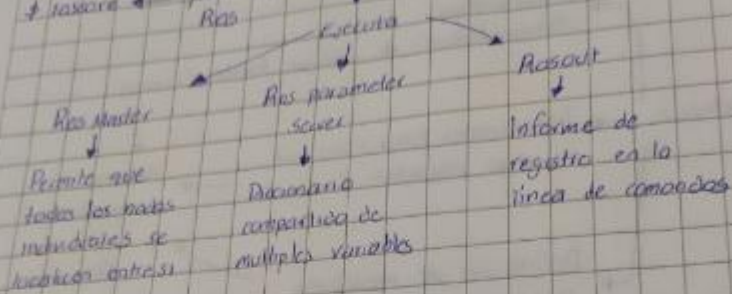


## Tapia Casillas Víctor Gabriel



\$ taskcat → proceso principal que gestiona todo el proceso de Ros



Ros workspace → Espacio de trabajo (carpeta con código) solo puede haber uno a la vez

Ros package →  
↳ Nodes  
↳ Messages  
↳ Service

\$ Catkin → Para compilar código fuente

\$ → símbolo de sistema	Quickref	Python.pdf
\$ mc	QuickRef	Ros.pdf
\$ emac	Quickref	Comandos /mux.pdf
\$ geany		
\$ terminator/mxux/tmux		

14 / 01 / 20

## CREACIÓN DEL ESPACIO DE TRABAJO EN ROS

```
$ mkdir -p ~/catkin_ws_apps/src
```

← único directorio que modificaremos

```
$ cd ~/catkin_ws_apps
```

/Build ← donde es invocado  
/devel ← despliegue de ejecutables y librerías

```
$ catkin_make
```

```
$ source devel/setup.bash
```

```
$ echo $ROS_PACKAGE_PATH
```

14 / 01 / 20

## CREACIÓN DE UN PAQUETE EN ROS

```
$ catkin_create_pkg package-name [dependencia]
```

↑ programa      ↑ Nombre del paquete a crear      ↑ dependencias que necesita

```
$ catkin_create_pkg package-robot roscpp rospy std_msgs
```

↑ Nombre de nuestro paquete

```
message-generation actionlib actionlib-msgs
```

```
$ cd ~/catkin_ws_apps/
```

```
$ catkin_make
```

```
$ source devel/setup.bash
```

## MODIFICAR EL BASH

```
$ echo "source ~/catkin_ws/apps/clevel/setup.bash" >> ~/.bashrc
```

```
$ source ~/.bashrc
```

```
$ cat ~/.bashrc
```

Si se equivoca

```
$ nano ~/.bashrc      ctrl x
```

## ROS NETWORK

Creación de nodos (craspy)  
(biblioteca de Python)



## Estructura

```
#!/usr/bin/env Python
# encoding: utf-8
def nodo():
    hello_str = "hola mundo"
    print(hello_str)
if __name__ == '__main__':
    tag:
        nodo()
    except:
        pass
```



descompar  
emaco

que

15-1

- Resistencias
- LEDs
- LCD 16x2
- Potenciómetros
- Transistores 2N2222A
- 4N24 / optoacoplador
- Relay
- L435
- Driver P2P
- Motor P2P

- cd src
- ls
- cd package-robot
- ls
- cd ..
- cd ..

Nodo - ros.py - P9

# } igual que el anterior

```
import rospy
def nodo():
    rospy.init_node('nodo1')
    message = "hola mundo"
    # print(message)
    rospy.loginfo(message)
```

```
if __name__ == '__main__':
    try:
```

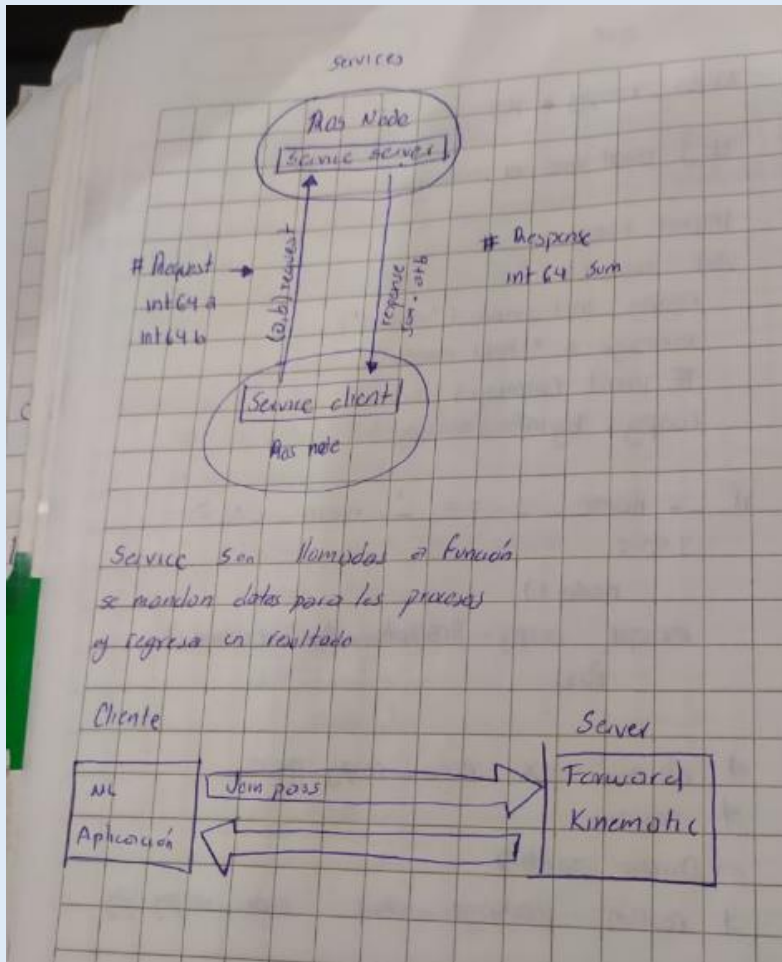
```
        nodo()
    except rospy.RosInterruptException:
        pass
```

\$ chmod +x nodo-ros.py

\$ roscore

.. Diridir pantalla

\$ roslaunch package-robot nodo-ros.py



package.xml  
línea 54  
permiso tener las mismas habilidades

