

10. Trabajo final

El trabajo final se divide en 2 partes:

1. Creación de nodos con publicadores, suscriptores y mensajes
2. Creación de servicios

¿Qué tengo que entregar?

- Código fuente desarrollado y comentado (comprimir en un archivo .zip todo el Workspace).
- Una memoria con lo que se ha realizado: máximo 15 páginas en PDF
- Un vídeo de máx 5 minutos explicando el funcionamiento de la práctica en mp4. Se puede comprimir el vídeo para que ocupe menos con HandBrake.

El objetivo es crear un nuevo paquete llamado `interaccion`.

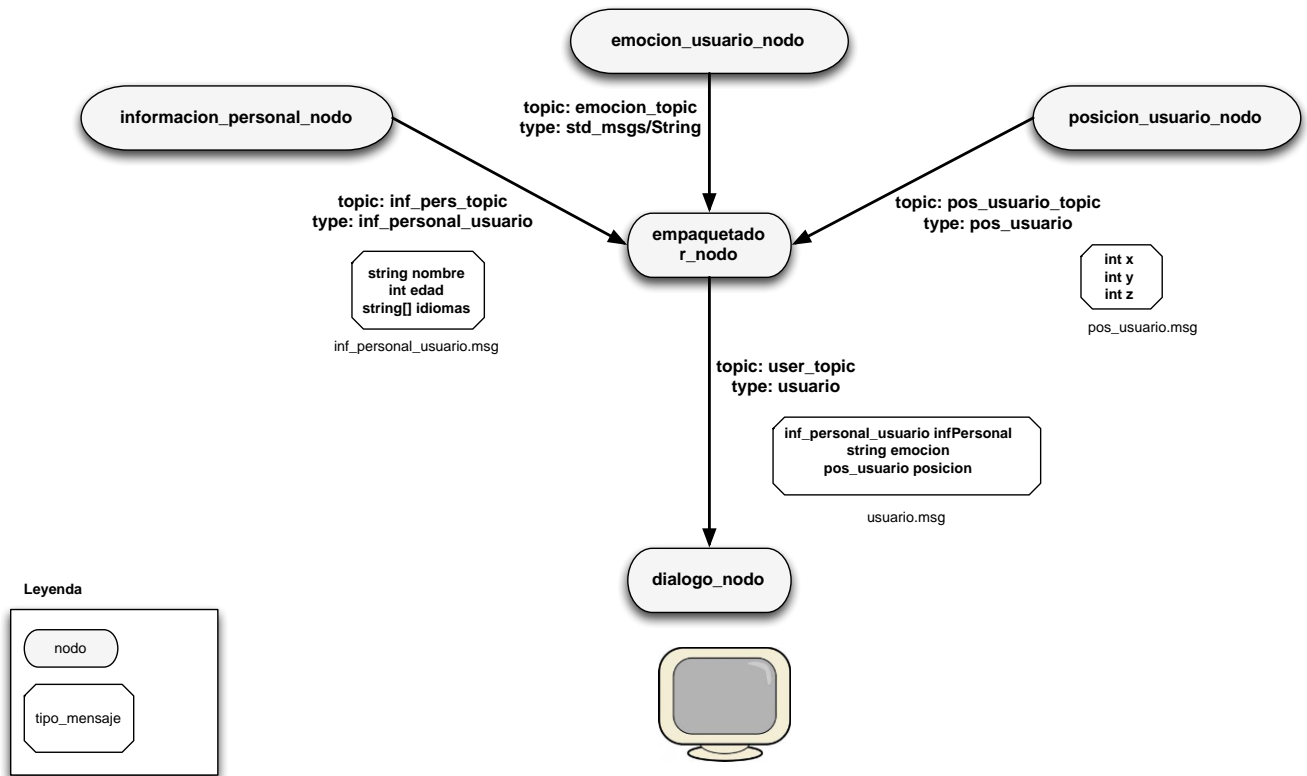
Creación de nodos con publicadores, suscriptores y mensajes

Será necesario definir tres tipos de mensajes y cinco nodos. La estructura es sencilla y se puede ver a continuación:

1. `informacion_personal_nodo`: pedirá por teclado el nombre, la edad y los posibles idiomas que habla un usuario inventado. Este nodo estará en un bucle pidiendo información de tal manera que se pueden introducir todos los usuarios que se desee.
2. `emocion_usuario_nodo`: de manera similar al anterior se pedirá indefinidamente, por teclado, y en bucle la emoción expresada por el usuario.
3. `posicion_usuario_nodo`: se pedirá por teclado las coordenadas del usuario en el espacio. Se podrán introducir tantas veces como se quiera.
4. `empaquetador_nodo`: se encarga de juntar la información recibida por cada uno de los tres nodos en un único mensaje de ros que se lo enviará al `dialogo_nodo`. Hasta que no se reciban todos los campos del mensaje a enviar, este no podrá ser enviado. Si se reciben varios mensajes de un tipo sin haberse recibido el resto de los campos, se deberá quedarse con el último mensaje recibido. Una vez se envié el mensaje al `dialogo_nodo`, se empezará a construir un mensaje nuevo.
5. `dialogo_nodo`: se encarga de mostrar por terminal cada mensaje que recibe del nodo empaquetador

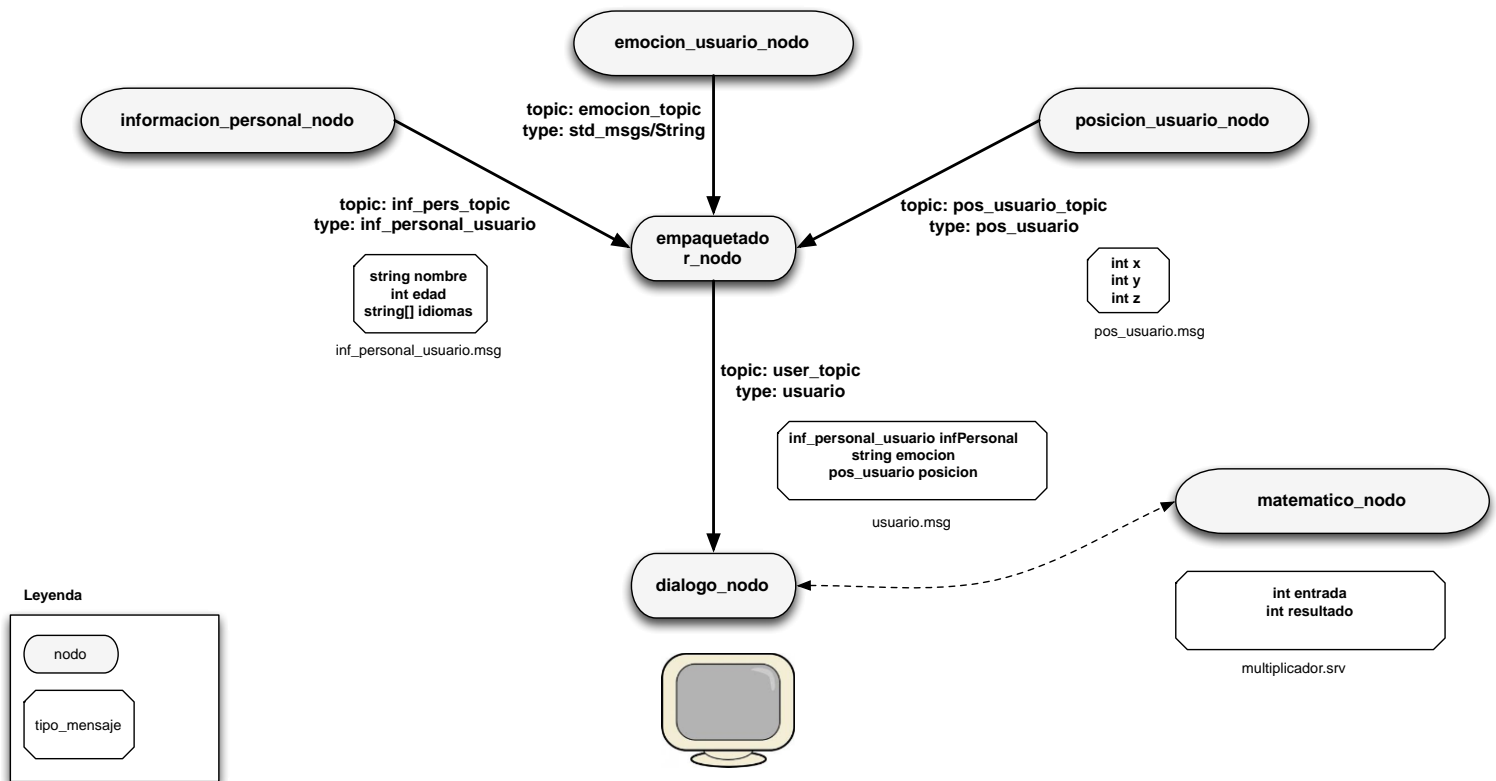
Los **objetivos** son:

- Crear la estructura correcta y que compile.
- Comprobar que el paso de mensajes es correcto.
- Generar el grafo de nodos y paso de mensajes.
- Grabar mediante *rosvbag* unos ejemplos de uso.
- Crear un launcher que lance todos los nodos.
- Reproducir la interacción grabada con *rosvbag* para que el profesor lo vea.



Añadiendo Servicios al ejercicio

Al ejercicio anteriormente propuesto vamos a añadirle un nuevo nodo que se encargue de ofrecer servicios que realicen operaciones matemáticas. En este caso simplemente queremos `dialogo_nodo` multiplique la edad recibida del usuario por dos. Para ello deberá invocar el servicio `servicio_multiplicador` registrado por el nodo `matematico_nodo`. El resultado debe ser mostrado por terminal por `dialogo_nodo`.



Extras ejercicio trabajo final

Ejecutando los nodos remotamente

ROS está concebido para ser usado de manera distribuida, con lo cual, si hemos seguidos los pasos correctamente, vamos a probar a lanzar los nodos en diferentes ordenadores. Para ello los ordenadores deben estar conectados en el mismo segmento de red (en el caso de la universidad podrán estar conectados por VPN para conseguir estar bajo el mismo segmento de red), cada uno de estos ordenadores lanzará uno de los nodos del ejercicio anterior. El ordenador que ejecute `dialogo_nodo` debe recibir y mostrar en pantalla correctamente la información suministrada.

Es necesario en cada PC establecer el ordenador donde corre el *roscore*. Sólo uno de los pc's ejecutará *roscore* (al que llamaremos MASTER), en cambio todos deben establecer la variable de entorno que indica en que IP corre el master:

```
export ROS_MASTER_URI= http://ip_donde_corre_ros_core:11311
```

Podemos encontrarnos con algún problema de red, permisos, etc. Se recomienda usar ping, netcat, cortagueos:

```
/etc/hosts.deny  
/etc/hosts.allow  
/etc/hosts
```

Para probar el funcionamiento distribuido de ROS se recomienda al usuario que implemente un chat sencillo. En el cual se pueda realizar una conversación multiusuario.