

Trabajo

Ya hemos visto como crear nuestro propio paquete, con la estructura de carpetas deseada, crear mensajes personalizados, enviar y recibir mensajes, etc. El objetivo es crear un nuevo paquete llamado *interaccion*. Será necesario definir tres tipos de mensajes y cinco nodos. La estructura es sencilla y se puede ver a continuación:

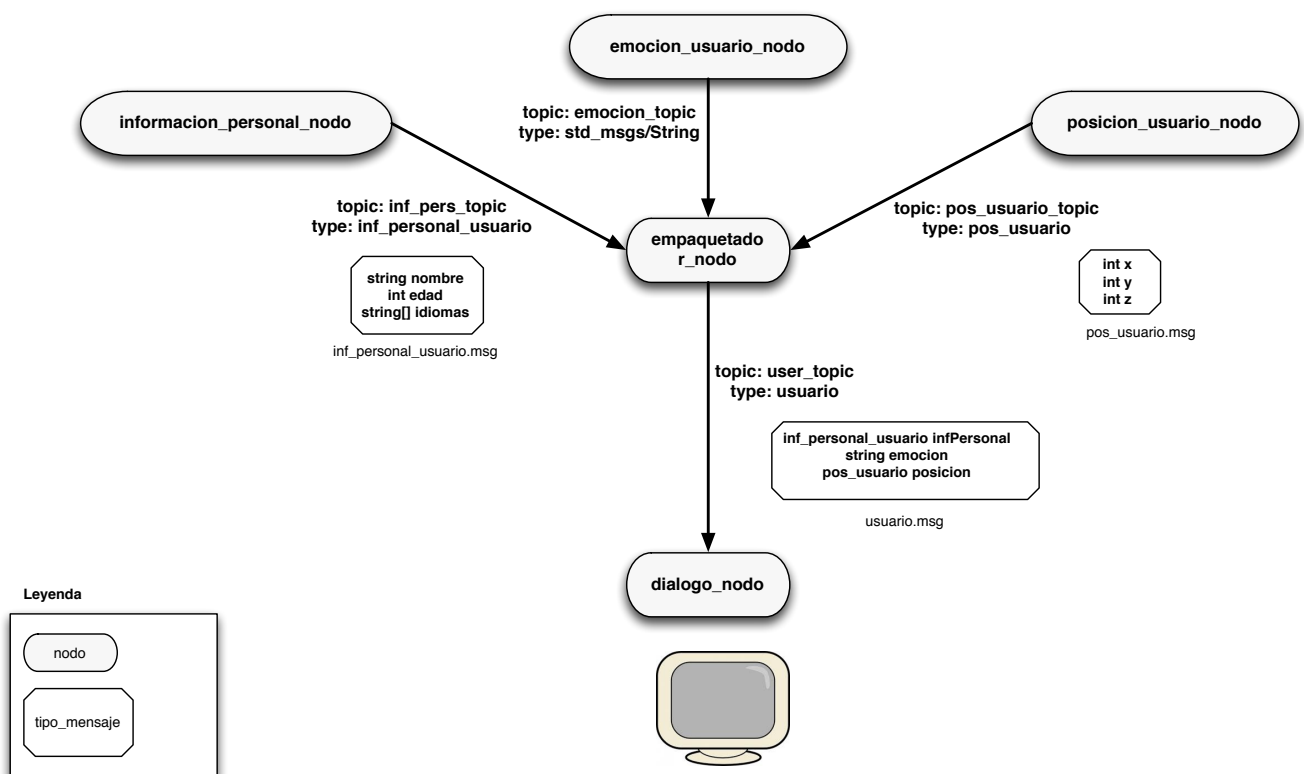
informacion_personal_nodo: pedirá por teclado el nombre, la edad y los posibles idiomas que habla un usuario inventado. Este nodo estará en un bucle pidiendo información de tal manera que se pueden introducir todos los usuarios que se desee.

emocion_usuario_nodo: de manera similar al anterior se pedirá indefinidamente, por teclado, y en bucle la emoción expresada por el usuario.

posicion_usuario_nodo: se pedirá por teclado las coordenadas del usuario en el espacio. Se podrán introducir tantas veces como se quiera.

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 envíe el mensaje al dialogo_nodo, se empezará a construir un mensaje nuevo.

dialogo_nodo: se encarga de mostrar por terminal cada mensaje que recibe del nodo empaquetador.

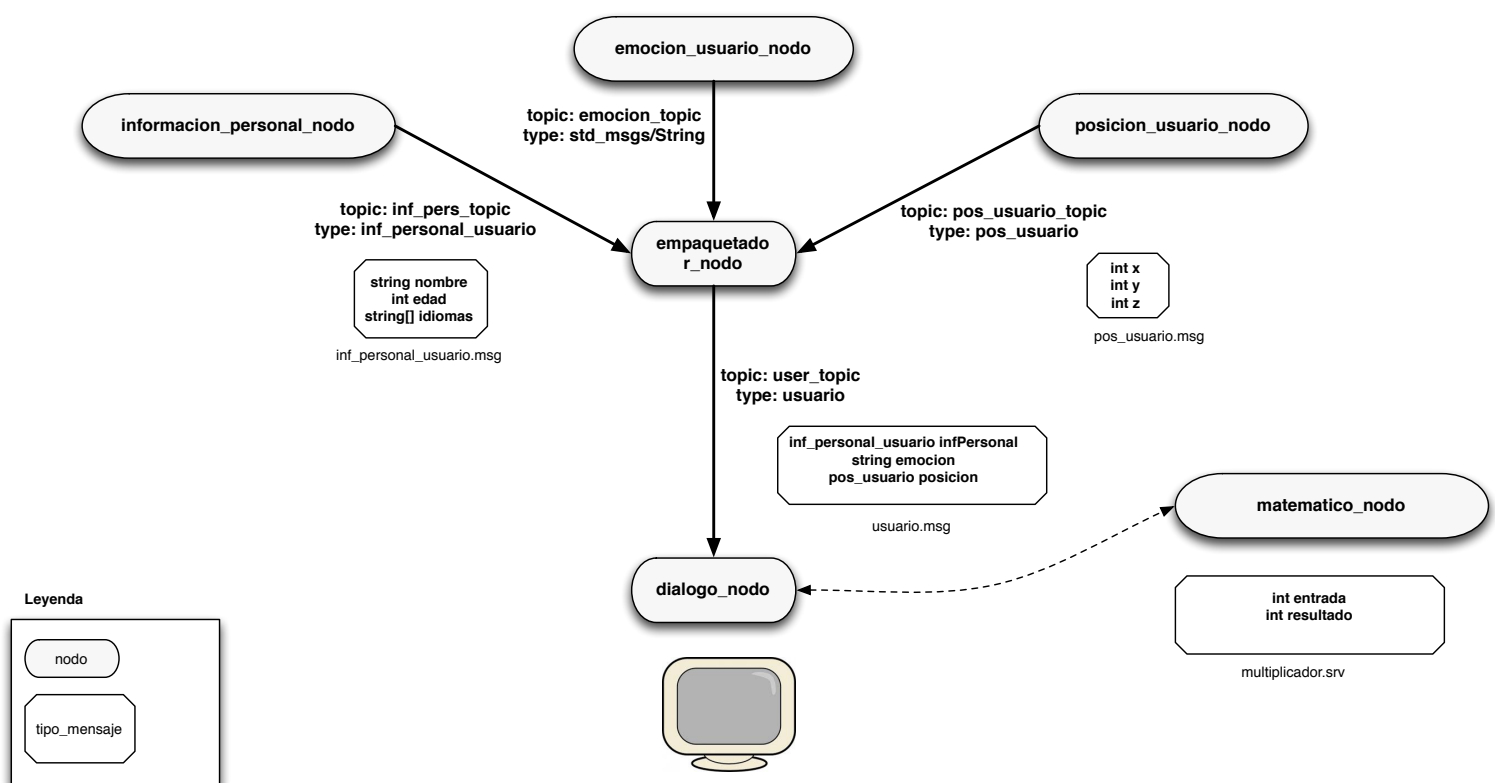


Los **objetivos** son:

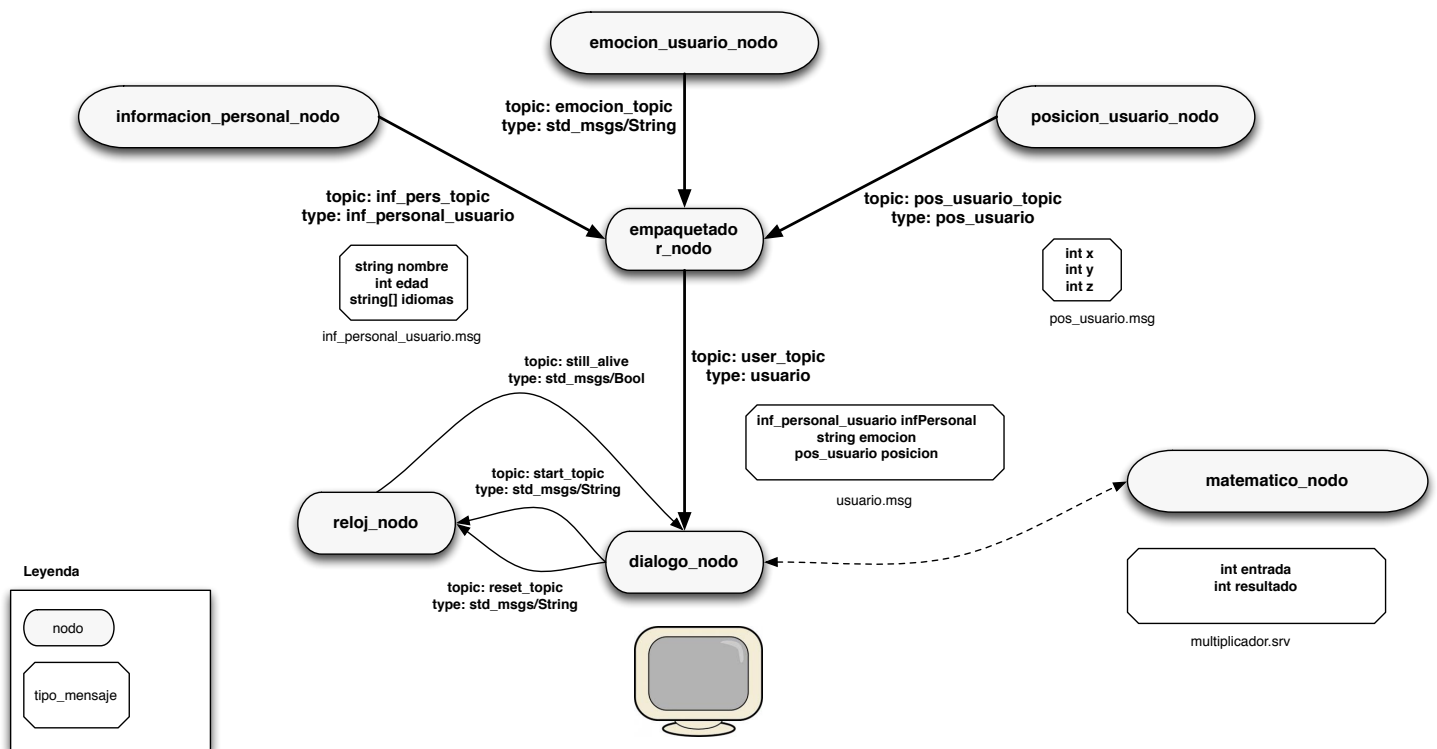
- Crear la estructura correcta y que compile.
- Comprobar que el paso de mensajes es correcto.
- Generar el grafo de nodos y paso de mensajes.
- Grabar mediante *rosbag* unos ejemplos de uso.
- Crear un launcher que lance todos los nodos.
- Reproducir la interacción grabada con *rosbag* para que el profesor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



Añadiendo Temporizadores y nodos que trabajan con el tiempo



Al ejercicio anteriormente propuesto vamos a añadirle un nuevo nodo *reloj_nodo*. Este nodo se activará mediante un mensaje que le envíe *dialogo_nodo* llamado *start_topic*. Este mensaje es enviado cuando *dialogo_nodo* recibe la información suministrada por el empaquetador y el nodo matemático. La primera vez enviará *start_topic*, las sucesivas veces enviará *reset_topic*.

El nodo *reloj_nodo* cuando reciba el mensaje de comenzar empezará a mostrar en pantalla la hora actual (tanto en formato UTC como en local) a una frecuencia de 3 veces por segundo. Además de mostrar los segundos transcurridos desde el instante actual hasta que se recibió el último mensaje de start o de reset.

Además el nodo_reloj deberá tener un Timer que se venza cada minuto. Una vez que se venza dicho Timer enviará un mensaje de *still_alive* al nodo *dialogo_nodo*. Este último deberá mostrar en pantalla que el reloj sigue funcionando correctamente.

Se deberá añadir al launcher los nodos necesarios.

Añadiendo Síntesis de voz (TTS)

Adicionalmente a la salida por pantalla del nodo *dialogo_nodo* se pide que se emita por los altavoces del sistema mediante un software de Text To Speech. Para ello instalar el paquete *espeak*.

```
$ sudo apt-get install espeak
```

Desde el *nodo_dialogo* hacer una llamada al sistema con el texto a sintetizar, de una manera similar a la siguiente:

```
string text= "Este es el texto a sintetizar";  
string command = "espeak -v es \"" + text + "\"";  
system (command.c_str());
```

Ejecutando los nodos remotamente

ROS esta 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 univerisad 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, cortaguegos:

```
/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 conversiación multiusuario.