

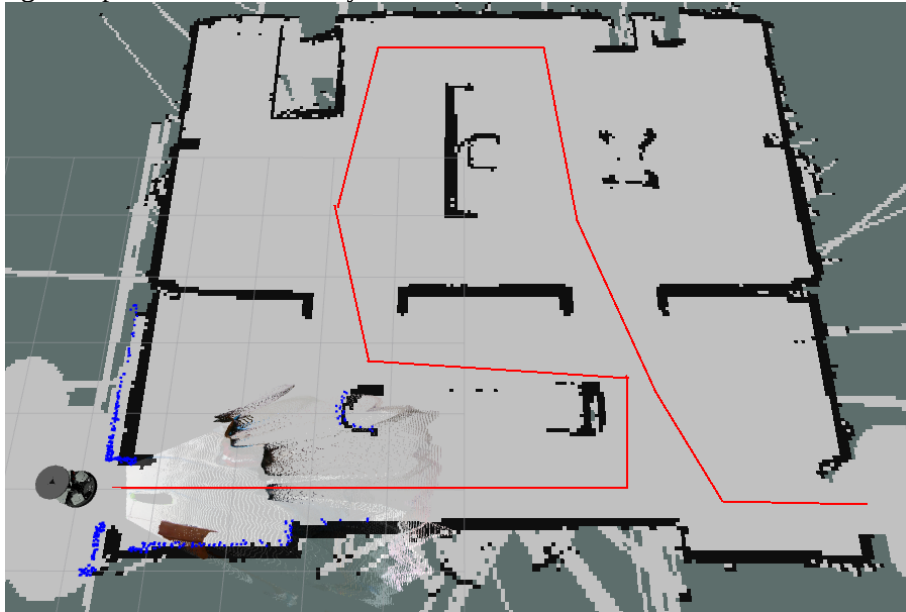
Uso del *navigation stack* para navegación en 2D

Primero abrimos la terminal de linux, en la que en la primera pestaña introducimos la instrucción `roslaunch bring_up_robotino_simul.launch`. Y en otra pestaña introducimos el siguiente código `roslaunch bring_up_navigation_move_base.launch`.

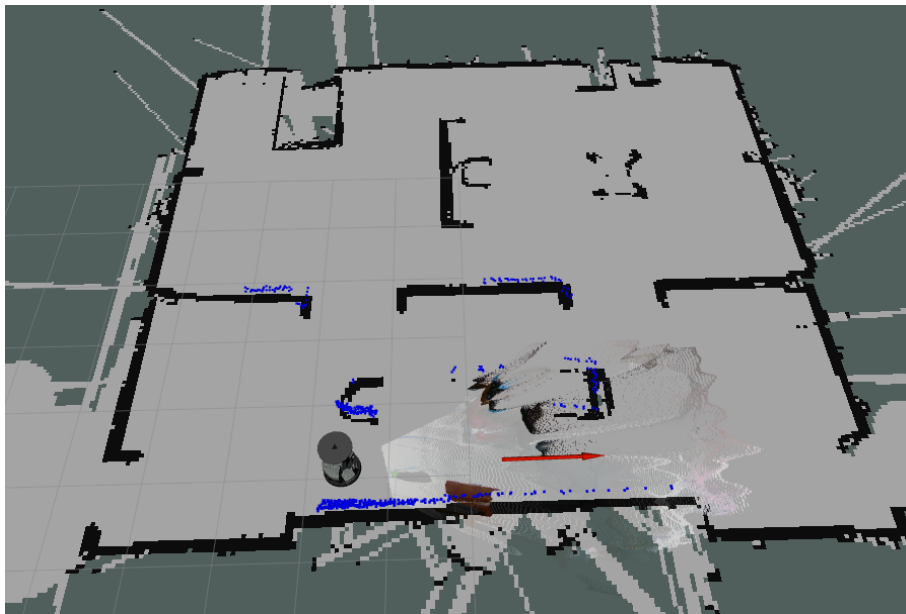
Posteriormente le configuramos en la pestaña de agregar displays en RVIZ, “ByTopic”->”goal”->”pose”.

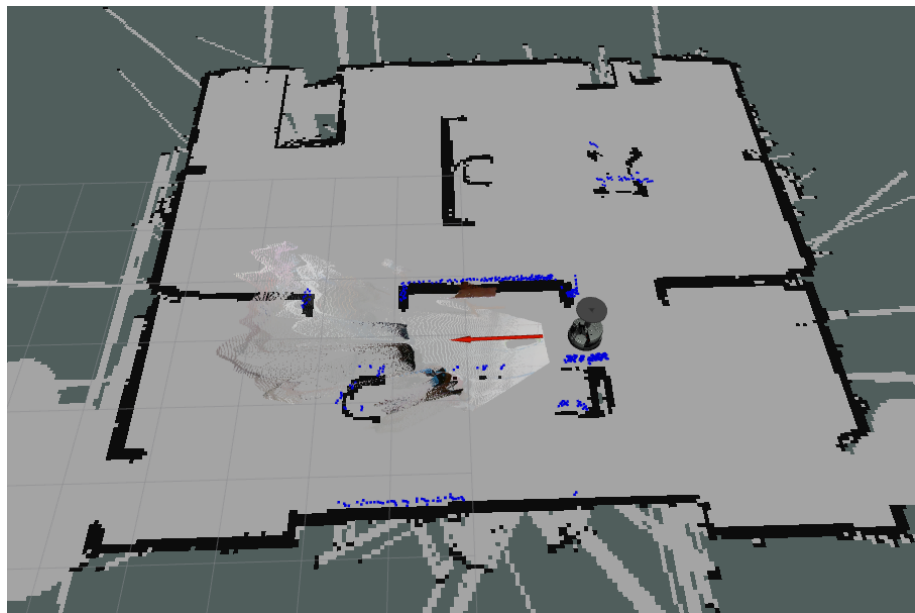
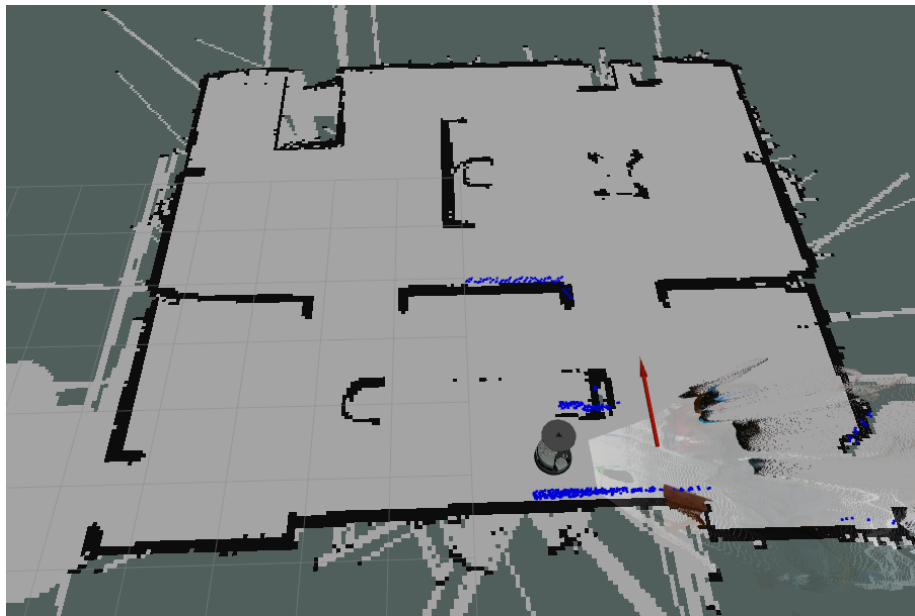
Posteriormente empezamos a trazar una meta con la flecha de la barra superior “2D Nav Goal”, ya que en donde dejaba a flecha era el objetivo que nuestro robot seguía, y la dirección de la flecha era la que el robot tendría una vez que llegará a su destino.

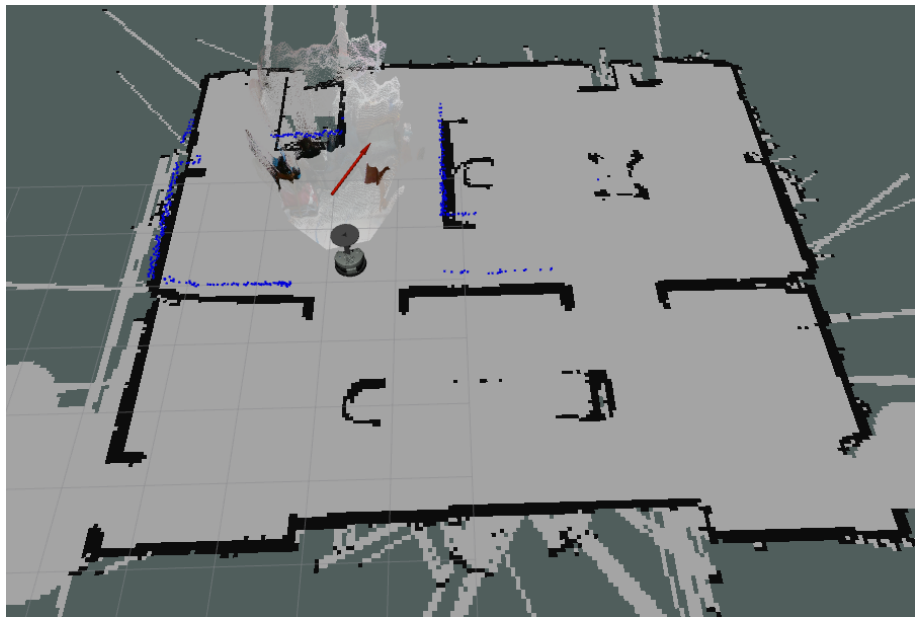
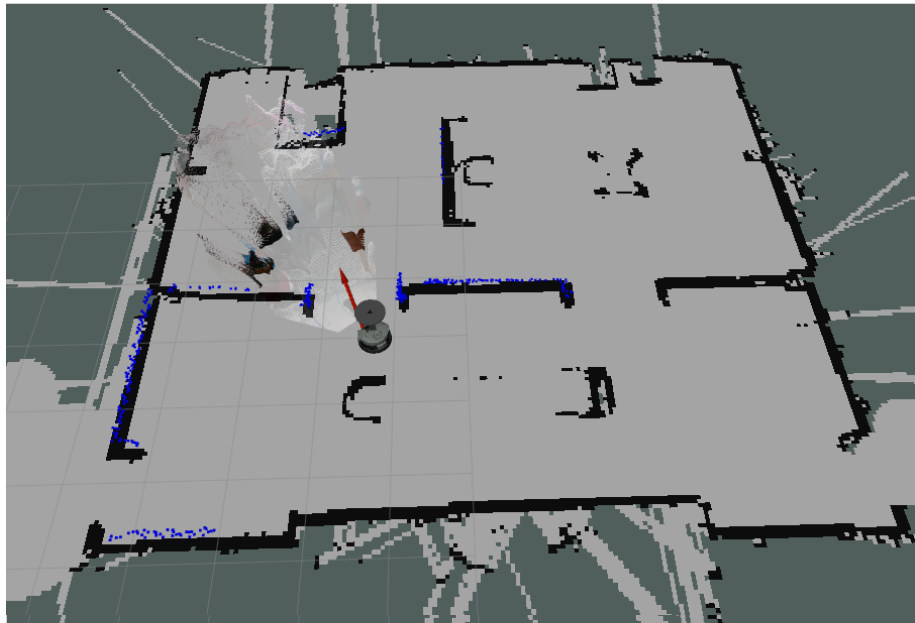
En la siguiente imagen se puede observar la trayectoria deseada:

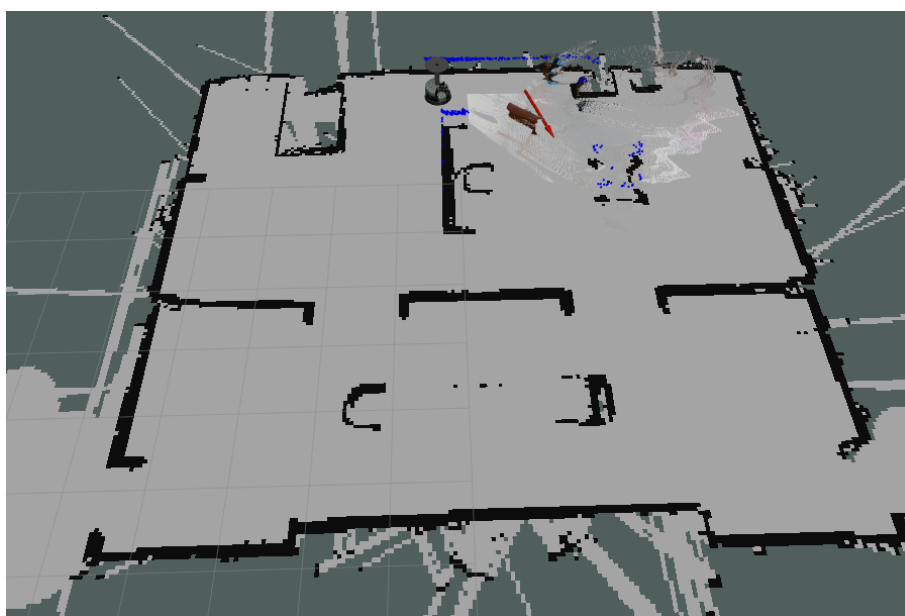
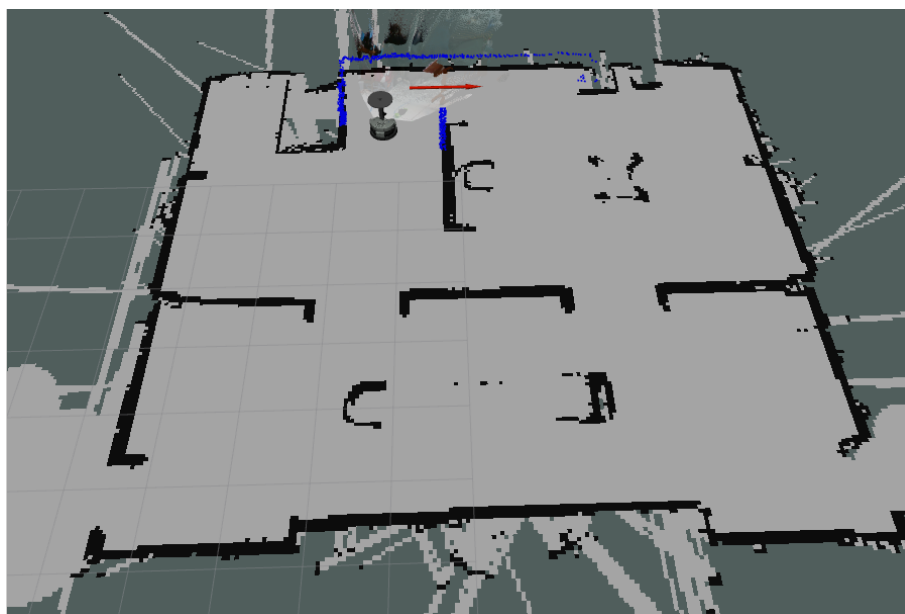


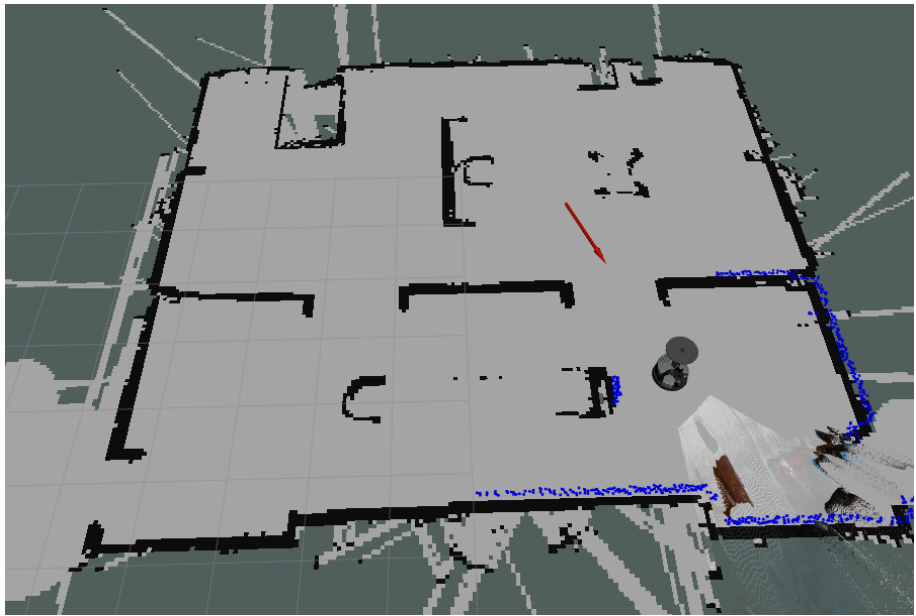
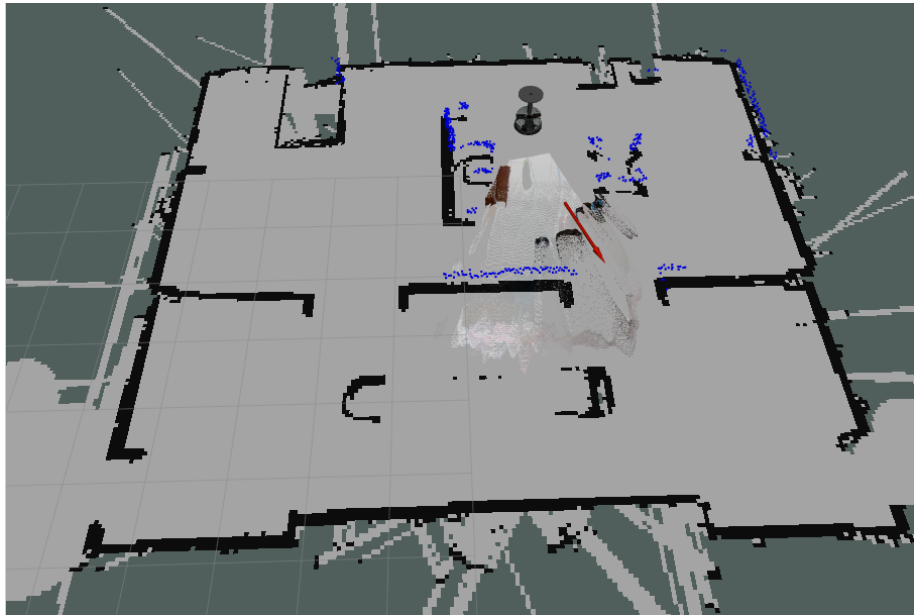
Aquí se observa como en base a la flecha de 2D Nav Goal el robot sigue la trayectoria deseada:

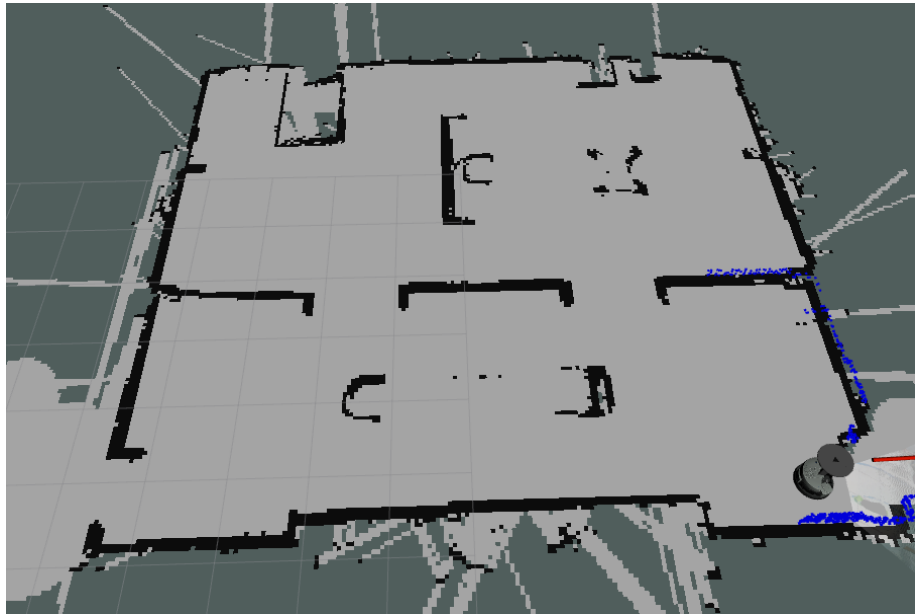
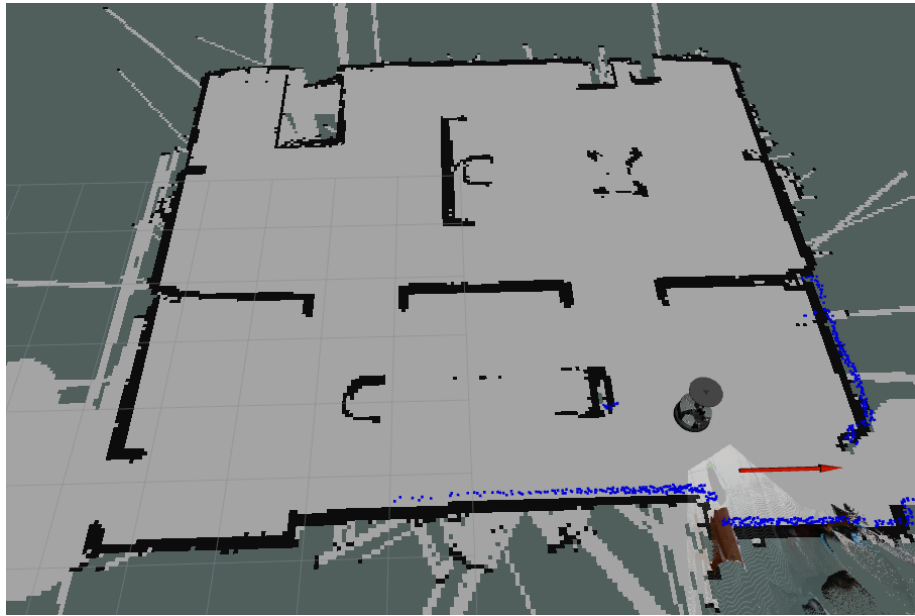












Posteriormente detuvimos la ejecución del `navigation_move_base.launch`, y después modificamos el archivo `sostmap_common_params.yaml`, ubicado en la carpeta `catkin_ws/src/config_files/move_base_params/`. Cambiamos los parámetros: `cost_scaling_factor` a 1.0 y `inflation_radius` a 2.5. Los cambios que se pudieron apreciar en la modificación, es que el robot se podía mover de una manera más fina, y al mismo tiempo no giraba muchas veces un poco a la izquierda o a la derecha cuando avanzaba, permitiendo así que no se desviara de la trayectoria deseada.

Por último detuvimos la ejecución del `navigation_move_base.launch` y entramos al archivo `dwa_local_planner_params.yaml`, ubicado en la carpeta `catkin_ws/src/config_files/move_base_params/`. Cambiamos los parámetros: `max_vel_x` a 2.0, `max_trans_vel` a 2.0 y `acc_lim_x` a 2.0. En esta modificación observamos que el robot se movía con mayor velocidad que en las pruebas pasadas, ya que anteriormente en ese parametro era de 0.5. Otra de las cosas que se pudo observar en esa modificación es que el robot no seguía una trayectoria recta, es decir que no se movía “tambalenándose de izquierda a derecha” durante la trayectoria, si no que su trayectoria era curvilínea sin presentar cualquier otra desviación, por lo que este tipo de modificación es recomendable unicamente para seguir objetivos largos.

Conclusiones

Pudimos apreciar como nuestro robot podía seguir trayectorias, en base a como se fijaba la flecha de 2D navegación ya que el robot de RVIZ adquiría posición en el mapa y su orientación para un próximo movimiento. Cabe destacar que tambien se le podía cambiar parámetros en cuestión del movimiento del robot, como la proximidad que su trayectoria fuera más recta para la posición deseada, así como su cambio de orientación (que viene relacionado con lo anterior), su velocidad, su rigidez o estabilidad de su movimiento, así como la adquisición de una trayectoria curvilínea (esta solo serviría para el seguimineto de trayectorías más distantes).