

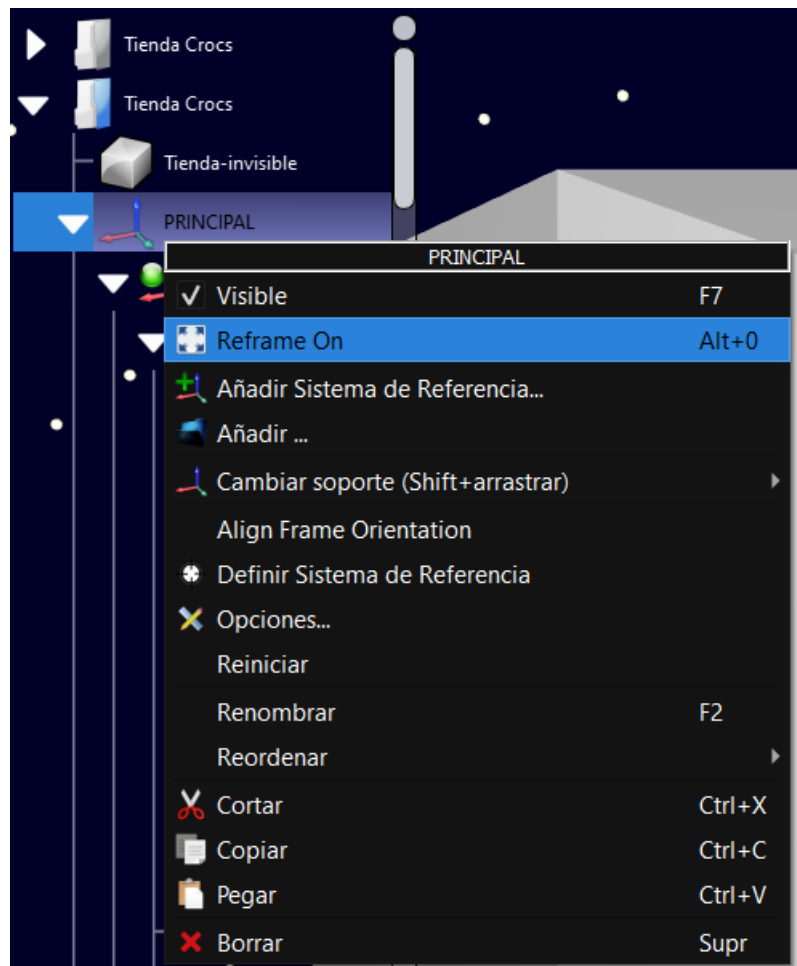
MANUAL DE ROBODK

Uso de usuario

Para el uso del archivo automatización_UNDEFINED.rdk, hay que tener pocas cosas en cuenta, de hecho, solo dos:

La primera es que al iniciar el programa no se verá nada, esto es porque hay que hacer un Reframe On, en el sistema de referencia PRINCIPAL.

1. Click derecho sobre PRINCIPAL (en el árbol de sistemas de referencias)
2. Seleccionar Reframe On
3. Hacer zoom out (rueda del ratón hacia atrás), hasta ver bien la estación.



La segunda es ejecutar el programa. Al final del árbol hay un archivo Python MAIN, simplemente hay que clicar en él, este reseteará todos los componentes y valores, y solo funcionará si recibe un mensaje MQTT.



Uso de programador

Si se desea modificar o utilizar el archivo Python. Aspectos para tener en cuenta:

1. En la primera parte del programa (líneas 1 – 218) se declaran todos los aspectos globales, y se reinician todos los parámetros para tener la vista inicial de la estación, no modificar nada.
2. hueco_random() y hueco_random_dev(): simplemente devuelven un hueco para coger, y en el caso de _dev, uno de los huecos libres que hay.

```
221 def hueco_random():
222     piso_aleatorio = random.randint(1, 3)
223     hueco_aleatorio = random.randint(1, 5)
224     es_random = [piso_aleatorio, hueco_aleatorio]
225
226     if disponible[piso_aleatorio][hueco_aleatorio] is True:
227         return es_random
228     else:
229         for i in range(1, 3):
230             for j in range(1, 5):
231                 if disponible[i][j] is True:
232                     return [i, j]
233
234 def hueco_random_dev():
235     piso_aleatorio = random.randint(1, 3)
236     hueco_aleatorio = random.randint(1, 5)
237     es_random = [piso_aleatorio, hueco_aleatorio]
238
239     if disponible[piso_aleatorio][hueco_aleatorio] is False:
240         return es_random
241     else:
242         for i in range(1, 3):
243             for j in range(1, 5):
244                 if disponible[i][j] is False:
245                     return [i, j]
```

3. Los hilos (líneas 248-636): mov_ur_pedido(), mov_cinta_pedido(), mov_estante_pedido (piso, caja), mov_ascensor_pedido (piso), hacer_devolucion (piso, caja), reponer(). NO MODIFICAR, es la coordinación entre robots, ya está todo optimizado.
4. hacer_pedido() y process_dev(): pieza central que gestiona las acciones a realizar, TAMPOCO MODIFICAR

5. `process_tasks(mqtt_client)`: gestiona como quiere que se realice el orden de pedidos, se puede cambiar la lógica si prefieres que se cambie dicho orden.

```
700 v def process_tasks(mqtt_client):
701     global pedidos hechos
702
703 v while True:
704     # 1) Si hay 2 ventas en curso y devoluciones esperando: procesar devolución primero
705 v     if pedidos_hechos >= 2 and not cola_devoluciones.empty():
706 v         try:
707             devol = cola_devoluciones.get(timeout=0.1)
708 v         except queue.Empty:
709             devol = None
710 v         if devol:
711             process_dev()
712             continue
713
714     # 2) Si hay menos de 2 ventas en curso y ventas esperando: procesar venta
715 v     if pedidos_hechos < 2 and not cola_ventas.empty():
716 v         try:
717             pedido = cola_ventas.get(timeout=0.1)
718 v         except queue.Empty:
719             pedido = None
720 v         if pedido:
721             hacer_pedido()
722             mqtt_client.publish("tienda/respuesta", json.dumps({"status": "en_proceso"}))
723             continue
724
725     # 3) Si no hay nada: pequeña espera
726     time.sleep(0.1)
```

6. `on_message(client, userdata, msg)`: se puede modificar el bróker y el puerto de mqtt si necesitas conectarte de otra forma. RECOMENDACIÓN: no cambiar los topics.

```
731 # CALLBACKS MQTT e INICIO
732 def on_message(client, userdata, msg):
733     pedido = json.loads(msg.payload.decode())
734     if pedido["tipo"] == "venta":
735         cola_ventas.put(pedido)
736     else:
737         cola_devoluciones.put(pedido)
738
739 if __name__ == "__main__":
740     mqtt_client = mqtt.Client()
741     mqtt_client.on_message = on_message
742     mqtt_client.connect("broker.hivemq.com", 1883)
743     mqtt_client.subscribe("tienda/pedidos")
744     mqtt_client.loop_start()
745
746     hilo = threading.Thread(target=process_tasks, args=(mqtt_client,), daemon=True)
747     hilo.start()
748
749     try:
750         while True:
751             time.sleep(1)
752     except KeyboardInterrupt:
753         mqtt_client.loop_stop()
```