



Universidad
Zaragoza

ANEXO

Generación de escenarios y programación de
brazos robóticos con RobotStudio y RAPID (ABB)

Autor

Daniel Vicente Moya

Directores

Rosario Aragüés

Raúl Igual

Escuela Universitaria Politécnica de Teruel

2016

1. Especificaciones Robots

En este anexo se recogen las especificaciones de los robots del Laboratorio de Tecnologías Industriales y los robots de la réplica virtual del laboratorio.

1.1. Especificaciones robot IRB6400F 2.4-150

Technical specification

**IRB 6400 /2.4-120, /2.4-150, /2.4-200, /2.8-120, /3.0-75
IRB 6400FHD**

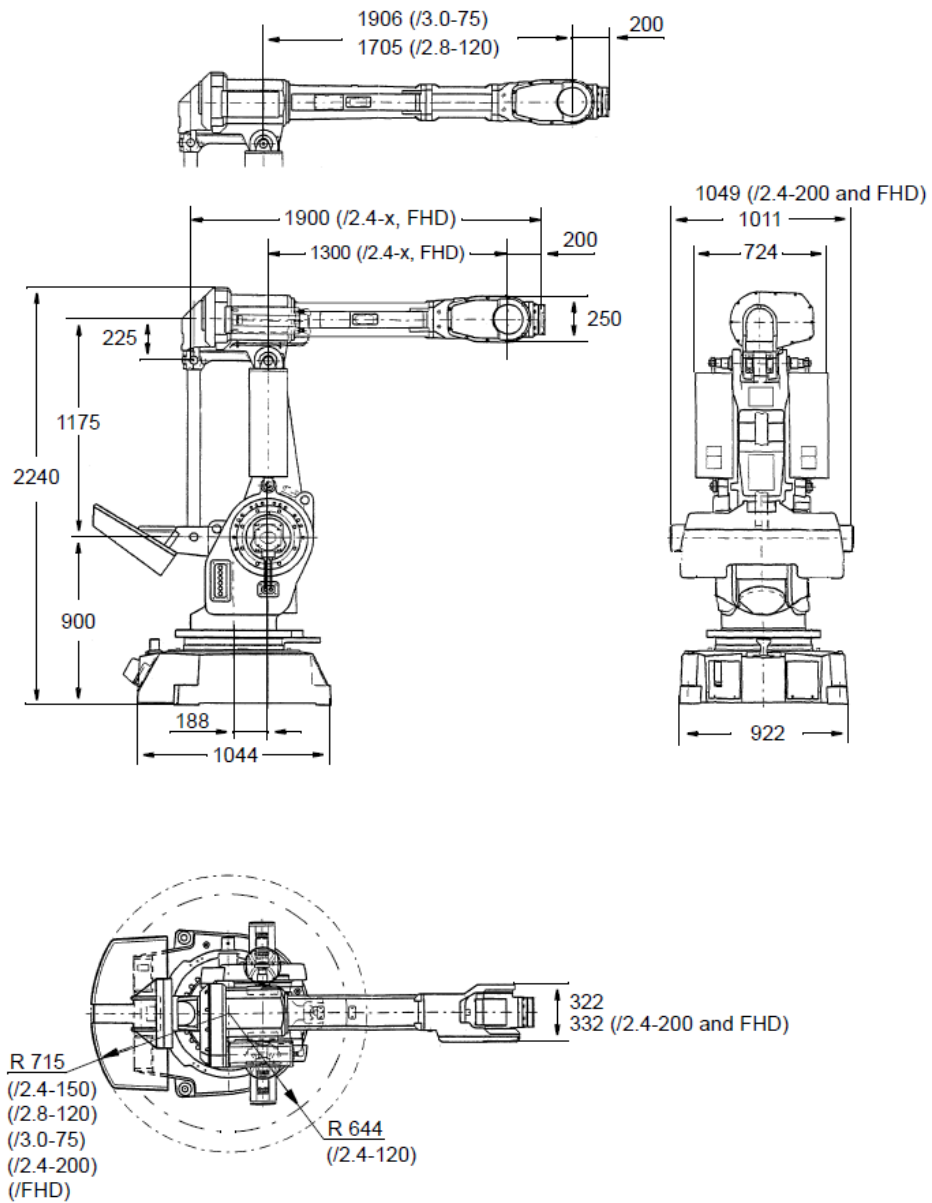


Ilustración 1 Especificaciones robot IRB6400F (medidas en mm)

1.2. Especificaciones robot IRB6400R 2.5-150

Technical specification

IRB 6400R /2.5-120, /2.5-150, /2.5-200, /2.8-150, /2.8-200 and /3.0-100

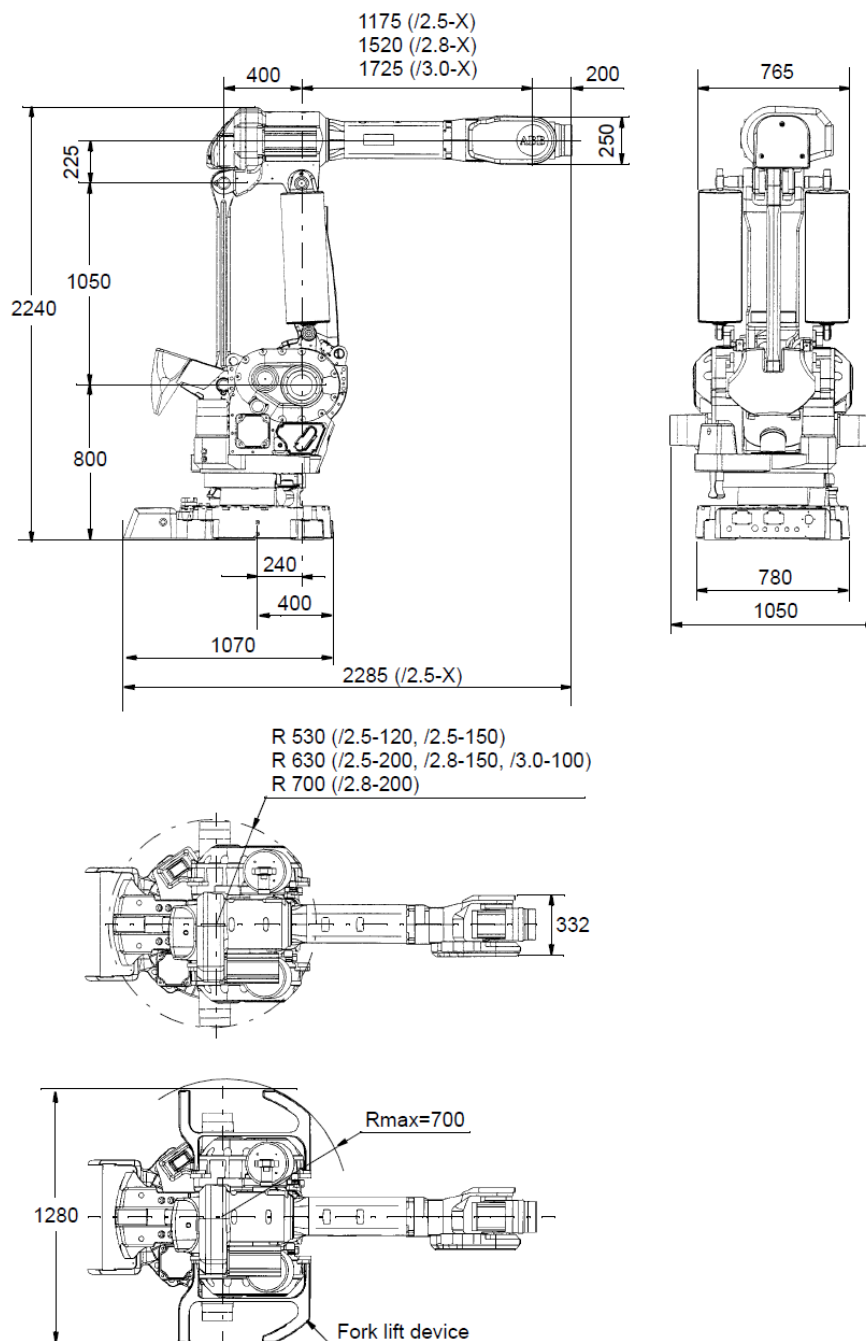


Ilustración 2 Especificaciones robot IRB6400R (medidas en mm)

2. Especificaciones de las mesas del área de trabajo.

Este anexo recoge los detalles dimensionales de las mesas del área de trabajo.

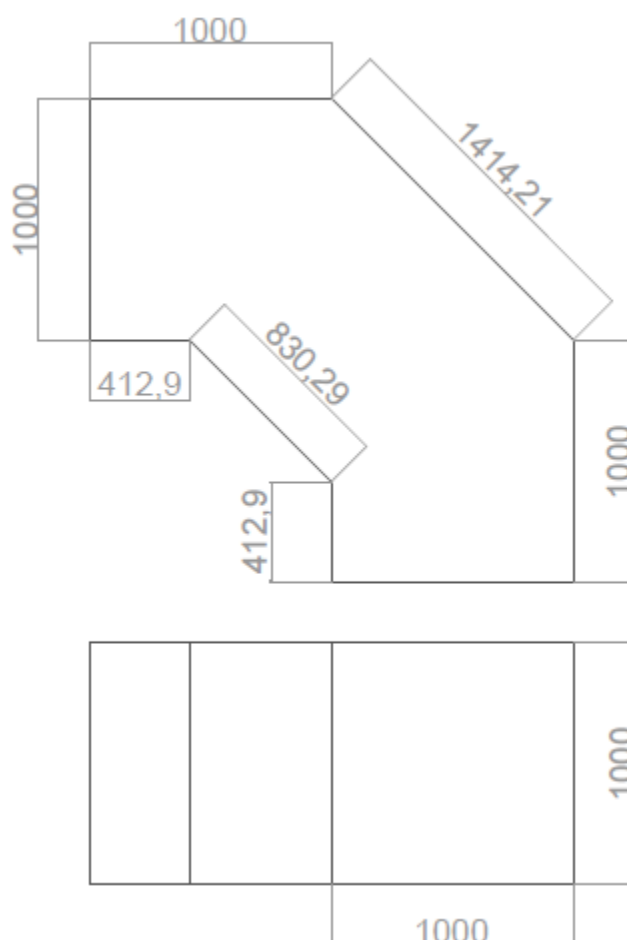


Ilustración 3 Especificaciones mesas laboratorio (medidas en mm)

3. Cálculos corte cilindro

Para dar forma a la parte cilíndrica del soporte de los aros del apartado 2.1, se ha tenido que hacer uso del **teorema de Tales**. Con él se calculó la altura del cono al que se elimina un tramo para que tome la forma requerida. La Ilustración 76 hace referencia a la aplicación del teorema.

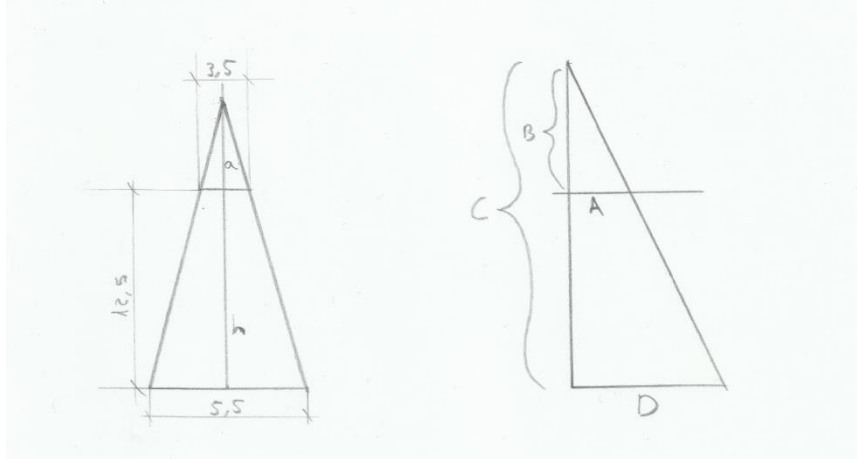


Ilustración 4 Aplicación del teorema de Tales al soporte de los aros

$$\frac{A}{B} = \frac{D}{C}$$

$$h = 12,5 + a$$

$$\frac{12,5 + a}{\frac{5,5}{2}} = \frac{a}{\frac{3,5}{2}}$$

$$a = 21,875cm \approx 22cm$$

La altura final del cono es de $h=34,5cm$. Posteriormente con la herramienta de **Resta** se ha eliminado la sección sobrante de $22cm$.

4. Detalles del Escenario 2

En este anexo continúa el desarrollo del escenario 2 desde donde se dejó, en el apartado 3.2.2 del trabajo.

4.1. Asignación de entradas y salidas

Tal y como se hizo en el escenario 1 apartado 3.1.2, en este escenario también es necesario crear un bus y una unidad en el controlador. Para los sensores se necesitará para su activación tres señales de salida “*sensor1act*”, “*sensor2act*” y “*sensor3act*”; y tres señales de entrada para la toma de datos “*sensor1*”, “*sensor2*”, “*sensor3*”. Para la herramienta hará falta una señal de salida “*herramienta*”. Se reinicia el controlador para que se apliquen los cambios.

También como en el escenario 1 se crearan los eventos de conectar y desconectar objetos a la herramienta y se les asignará la señal “*herramienta*”.

4.2. Lógica de estación

Se conectan las señales de activación de cada sensor con las creadas en el controlador para activar los sensores y las salidas de datos con las señales de datos del sensor. El diseño de las conexiones queda según la Ilustración 77.

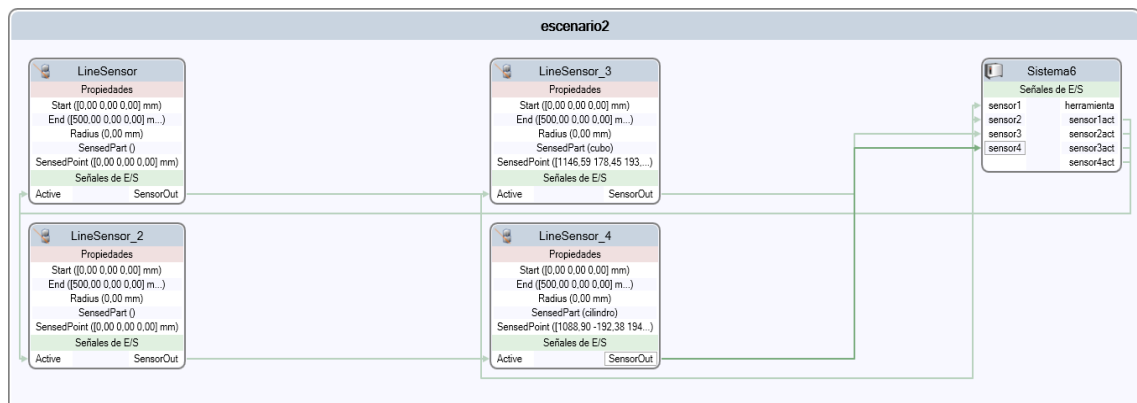


Ilustración 5 Lógica de estación del Escenario2

4.3. Creación de conjuntos de colisiones

Se crea para cada robot un conjunto de colisiones. Para el robot de transporte se tendrá: en el **grupo A** la herramienta y el robot, y en el **grupo B** los otros robots y sus herramientas junto con las mesas y piezas. Los conjuntos para los robots de tratado tendrán: en el **grupo A** los respectivos robots y herramienta y en el **grupo B** solamente la mesa y piezas de su área de trabajo. Como en el robot central se comprueba la colisión con los otros robots, no es necesario volver a introducir el robot en los conjuntos de los otros dos.

Trayectoria llevar cilindro a mesa 2 (Path_10):

Se crean dos puntos de aproximación, *target_110* $[0,0,256]$ para coger la pieza de entrada, y otro para dejarla en la mesa, *target_120* $[-1186.146,-987.434,656]$. La ventosa cogerá el cilindro en el punto central de la base superior, *target_30* $[-1186.146,-987.434,356]$. Y lo soltará en la mesa sobre el punto *target_20* $[0,0,0]$. La orientación de los puntos será perpendicular a la superficie de la pieza con la misma orientación que la herramienta y variará noventa grados de mesa a mesa para mantener la orientación de la pieza con el movimiento del robot. Ver la trayectoria en amarillo en la Ilustración 78.

La secuencia queda: reposo → punto aproximación → punto agarre → señal activar ventosa → punto aproximación → punto aproximación → punto soltar pieza → señal desactivar ventosa → punto aproximación → reposo

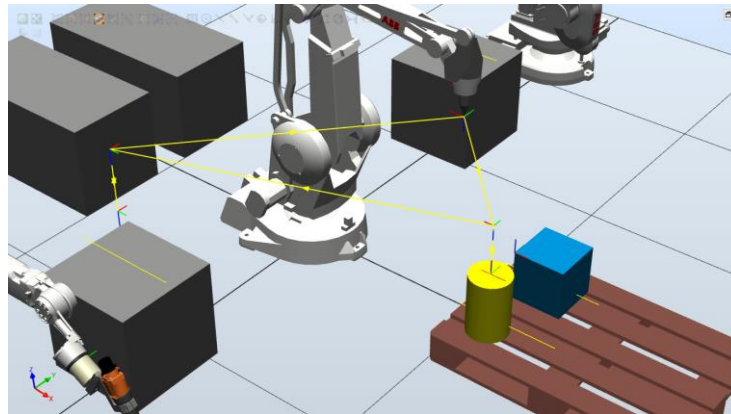


Ilustración 6 Path10 T_ROB1

Trayectoria llevar cilindro a mesa 4 (Path_20):

Los puntos de la mesa 2 serán los mismos que el 'Path' anterior, solo se requiere un punto auxiliar, *target_40* $[-2386.146,-137.434,356]$ y otro punto para depositar la pieza en la mesa 4, *target_130* $[-2386.146,-137.434,556]$. Como en el caso anterior la orientación de los puntos es perpendicular a la pieza y varia noventa grados de mesa a mesa. Ver la trayectoria en amarillo en la Ilustración 79.

La secuencia queda: reposo → punto aproximación → punto agarre → señal activar ventosa → punto aproximación → punto aproximación → punto soltar pieza → señal desactivar ventosa → punto aproximación → reposo

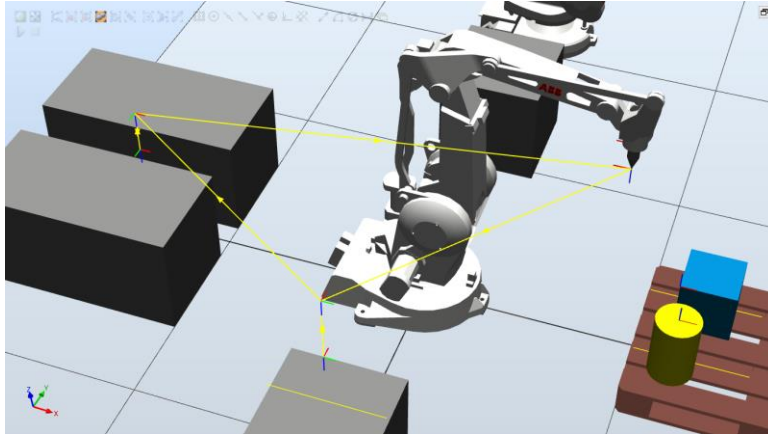


Ilustración 7 Path20 T_ROB1

Trayectoria llevar cubo a mesa 1 (Path_30):

Similar y casi simétrico al *Path_10* se crean los puntos auxiliares, *target_80* [125,125,305.991] y *target_90* [-1146.588,1163.501,705.991] y los puntos para coger y dejar la pieza, *target_50* [125,125,0] y *target_60* [-1146.588,1163.501,355.991]. Ver trayectoria en amarillo en la Ilustración 80.

La secuencia queda: reposo → punto aproximación → punto agarre → señal activar ventosa → punto aproximación → punto aproximación → punto soltar pieza → señal desactivar ventosa → punto aproximación → reposo

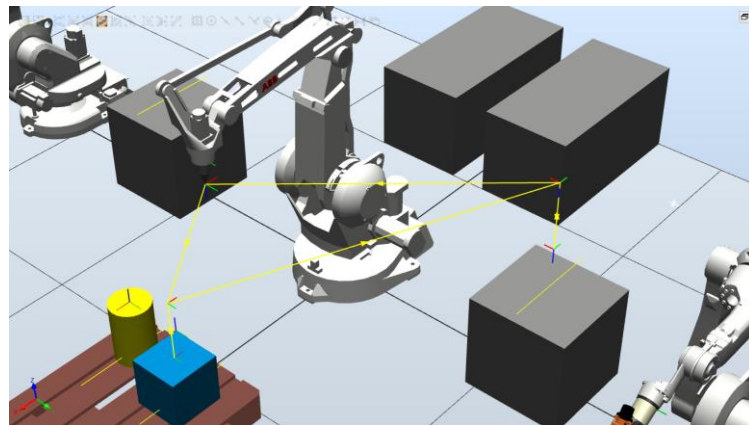


Ilustración 8 Path30 T_ROB1

Trayectoria llevar cubo a mesa 3 (Path_40):

Similar al *Path_20*, los puntos *target_90* [-1146.588,1163.501,705.991] y *target_100* [-2396.588,313.501,605.991] se utilizan como puntos de aproximación y los puntos *target_60* [-1146.588,1163.501,355.991] y *target_70* [-2396.588,313.501,355.991] para coger y dejar la pieza. Ver la trayectoria en amarillo en la Ilustración 81.

La secuencia queda: reposo → punto aproximación → punto agarre → señal activar ventosa → punto aproximación → punto aproximación → punto soltar pieza → señal desactivar ventosa → punto aproximación → reposo

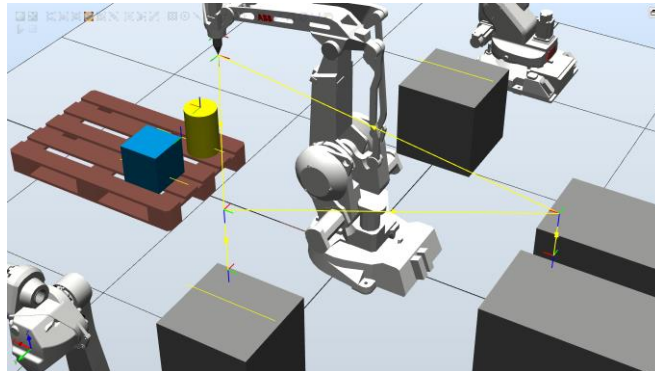


Ilustración 9 Path40 T_ROB1

5. Programas RAPID

En este anexo se recogen los programas RAPID de los diferentes escenarios.

5.1. Programa RAPID del escenario 1

```
MODULE Module1
  CONST robtarget Target_10:=[[506.291643024,0,679.499985759],
[0.499999989,0,0.866025467,0],[0,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_60:=[[0,-50,125], [0,0.707106781,0.707106781,0], [-
1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_30:=[[0,-50,25], [0,0.707106781,0.707106781,0], [-
1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_90:=[[-250,300,550], [0,0.707106781,0.707106781,0],
[0,0,1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_80:=[[-250,300,191.667],
[0,0.707106781,0.707106781,0], [0,0,1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_70:=[[-250,75,191.667],
[0,0.707106781,0.707106781,0], [0,0,1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_130:=[[0,-75,308.333], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_100:=[[0,-125,333.333], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_140:=[[0,-75,413.333], [0,-
0.707106781,0.707106781,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_120:=[[-250,300,500], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_110:=[[-250,0,500], [0,-0.707106781,0.707106781,0],
[0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_180:=[[550,-325,393.333], [0,-
0.707106781,0.707106781,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_170:=[[550,-325,333.333], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_160:=[[550,-575,413.333], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_150:=[[550,-575,308.333], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

  PROC Path_10()
    MoveJ Target_10,v1000,z100,wrist0\WObj:=wobj0;
    MoveJ Target_60,v1000,z100,wrist0\WObj:=Workobject_3;
    MoveL Target_30,v500,fine,wrist0\WObj:=Workobject_3;
    SetDO actpinza,0;
    MoveL Target_60,v1000,z100,wrist0\WObj:=Workobject_3;
    MoveJ Target_10,v1000,z100,wrist0\WObj:=wobj0;
  ENDPROC

  PROC Path_20()
    MoveJ Target_10,v1000,z100,wrist0\WObj:=wobj0;
    MoveJ Target_90,v1000,z100,wrist0\WObj:=Workobject_1;
    MoveL Target_80,v500,fine,wrist0\WObj:=Workobject_1;
    MoveL Target_70,v500,fine,wrist0\WObj:=Workobject_1;
```

```

        SetDO actpinza,1;
        MoveL Target_90,v500,fine,wrist0\WObj:=Workobject_1;
        MoveJ Target_10,v1000,z100,wrist0\WObj:=wobj0;
    ENDPROC

    PROC Path_30()
        MoveJ Target_10,v1000,z100,wrist0\WObj:=wobj0;
        MoveJ Target_140,v1000,z100,wrist0\WObj:=Workobject_2;
        MoveL Target_130,v500,fine,wrist0\WObj:=Workobject_2;
        SetDO actpinza,0;
        MoveJ Target_180,v1000,z100,wrist0\WObj:=Workobject_2;
        MoveL Target_170,v500,fine,wrist0\WObj:=Workobject_2;
        MoveL Target_150,v500,fine,wrist0\WObj:=Workobject_2;
        SetDO actpinza,1;
        MoveL Target_160,v500,fine,wrist0\WObj:=Workobject_2;
        MoveJ Target_10,v1000,z100,wrist0\WObj:=wobj0;
    ENDPROC

    PROC main()
        Path_10; !Agarre de pieza central
        Path_20; !Colocación pieza central
        Path_30; !Colocación pieza mitad izquierda
    ENDPROC
ENDMODULE

```

5.2. Programación RAPID escenario 2

5.2.1. Tarea T_ROB1

```

MODULE Module1
    CONST robtarget Target_10:=[[870,0,1009], [0,0,1,0], [-1,0,0,0],
    [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_110:=[[0,0,256], [0,0,1,0], [-1,0,-1,0],
    [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_20:=[[0,0,0], [0,0,1,0], [-1,0,-1,0],
    [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_120:=[[-1186.146,-987.434,656],
    [0,0.707106781,0.707106781,0], [-2,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_30:=[[-1186.146,-987.434,356],
    [0,0.707106781,0.707106781,0], [-2,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_130:=[[-2386.146,-137.434,556], [0,1,0,0], [-2,0,0,0],
    [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_40:=[[-2386.146,-137.434,356],[0,1,0,0],[-
    2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_80:=[[125,125,305.991], [0,0,1,0], [0,0,0,0],
    [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_50:=[[125,125,0], [0,0,1,0], [0,0,0,0],
    [9E9,9E9,9E9,9E9,9E9,9E9]];

```

```

CONST robtarget Target_90:= [[-1146.588,1163.501,705.991], [0,-
0.707106781,0.707106781,0], [1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_60:= [[-1146.588,1163.501,355.991], [0,-
0.707106781,0.707106781,0], [1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_100:= [[-2396.588,313.501,605.991], [0,1,0,0], [1,0,-
1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_70:= [[-2396.588,313.501,355.991], [0,1,0,0], [1,0,-
1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
PERS bool tarea1 := FALSE;
PERS bool tarea2 := FALSE;
PERS bool mesa1libre := TRUE;
PERS bool mesa2libre := TRUE;

```

```
PROC Path_10()
```

```

MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;
MoveJ Target_110,v1000,z100,ventosa\WObj:=Workobject_1;
MoveL Target_20,v300,fine,ventosa\WObj:=Workobject_1;
SetDO herramienta,1;
MoveL Target_110,v300,fine,ventosa\WObj:=Workobject_1;
MoveJ Target_120,v1000,z100,ventosa\WObj:=Workobject_1;
MoveL Target_30,v300,fine,ventosa\WObj:=Workobject_1;
SetDO herramienta,0;
MoveL Target_120,v300,fine,ventosa\WObj:=Workobject_1;
MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;

```

```
ENDPROC
```

```
PROC Path_20()
```

```

MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;
MoveJ Target_120,v1000,z100,ventosa\WObj:=Workobject_1;
MoveL Target_30,v300,fine,ventosa\WObj:=Workobject_1;
SetDO herramienta,1;
MoveL Target_120,v300,fine,ventosa\WObj:=Workobject_1;
MoveJ Target_130,v1000,z100,ventosa\WObj:=Workobject_1;
MoveL Target_40,v300,fine,ventosa\WObj:=Workobject_1;
SetDO herramienta,0;
MoveL Target_130,v300,fine,ventosa\WObj:=Workobject_1;
MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;

```

```
ENDPROC
```

```
PROC Path_30()
```

```

MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;
MoveJ Target_80,v1000,z100,ventosa\WObj:=Workobject_2;
MoveL Target_50,v300,fine,ventosa\WObj:=Workobject_2;
SetDO herramienta,1;
MoveL Target_80,v300,fine,ventosa\WObj:=Workobject_2;
MoveJ Target_90,v1000,z100,ventosa\WObj:=Workobject_2;
MoveL Target_60,v300,fine,ventosa\WObj:=Workobject_2;
SetDO herramienta,0;
MoveL Target_90,v300,fine,ventosa\WObj:=Workobject_2;
MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;

```

ENDPROC

PROC Path_40()

```
MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;  
MoveJ Target_90,v1000,z100,ventosa\WObj:=Workobject_2;  
MoveL Target_60,v300,fine,ventosa\WObj:=Workobject_2;  
SetDO herramienta,1;  
MoveL Target_90,v300,fine,ventosa\WObj:=Workobject_2;  
MoveJ Target_100,v1000,z100,ventosa\WObj:=Workobject_2;  
MoveL Target_70,v300,fine,ventosa\WObj:=Workobject_2;  
SetDO herramienta,0;  
MoveL Target_100,v300,fine,ventosa\WObj:=Workobject_2;  
MoveJ Target_10,v1000,z100,ventosa\WObj:=wobj0;
```

ENDPROC

PROC inicsensor()

```
SetDO sensor1act,1;  
SetDO sensor2act,1;  
SetDO sensor3act,1;  
SetDO sensor4act,1;
```

ENDPROC

PROC main()

inicsensor; !Obtención datos sensores

!Pone cilindro en mesa1

IF sensor1 < 1 AND sensor4 > 0 THEN

mesa1libre := FALSE; !Marca trayectrias de la mesa1 ocupadas

Path_10;

mesa1libre := TRUE; !Marca trayectorias de la mesa1 libres de

colisiones

ENDIF

!Pone cilindro en mesa2

IF sensor2 < 1 AND sensor3 > 0 THEN

mesa2libre := FALSE; !Marca trayectorias de la mesa2 ocupadas

Path_30;

mesa2libre := TRUE; !Marca trayectorias de la mesa2 libres de

colisiones

ENDIF

!Pone cubo en mesa4

IF sensor1 > 0 AND tarea1 = TRUE THEN

Path_20;

tarea1 := FALSE; !Reinicializa la tarea1 para otra ejecución de

esta

ENDIF

!pone cubo en mesa3

IF sensor2 > 0 AND tarea2 = TRUE THEN

```

Path_40;
tarea2 := FALSE; !Reinicializa la tarea2 para otra ejecución de la
misma
ENDIF
ENDPROC
ENDMODULE

```

5.2.2. Tarea T_ROB2

```

MODULE Module1
  CONST robtarget Target_10:=[[1237.30815974,-2200,891.685578915],
[0.069881481,0,0.997555301,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_70:=[[-1086.146,-987.434,456], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_20:=[[-1086.146,-987.434,356], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_30:=[[-1186.146,-887.434,356], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_40:=[[-1286.146,-987.434,356], [0,-
0.707106781,0.707106781,0], [1,-1,0,0], [9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_50:=[[-1186.146,-1087.434,356], [0,-
0.707106781,0.707106781,0], [1,-1,0,0], [9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_60:=[[-1086.146,-987.434,356], [0,-
0.707106781,0.707106781,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9]];
  PERS bool tarea1;
  PERS bool mesa1libre;

  PROC Path_10()
    MoveJ Target_10,v1000,z100,PKI_500\WObj:=wobj0;
    MoveJ Target_70,v1000,z100,PKI_500\WObj:=Workobject_1;
    MoveL Target_20,v1000,fine,PKI_500\WObj:=Workobject_1;
    MoveC Target_30,Target_40,v300,z100,
PKI_500\WObj:=Workobject_1;
    MoveC Target_50,Target_60,v300,z100,
PKI_500\WObj:=Workobject_1;
    MoveL Target_70,v1000,fine,PKI_500\WObj:=Workobject_1;
    MoveJ Target_10,v1000,z100,PKI_500\WObj:=wobj0;
  ENDPROC

  !Tarea soldado cubo
  PROC main()
    SetDO sensor1act,1; !Obtención de datos sensor

    !La tarea se activa si hay un pieza en la mesa, la tarea no ha sido
realizada y la trayectoria del robot esta despejada de otro robot
    IF ((sensor1 > 0) AND (tarea1 = FALSE))
      AND (mesa1libre = TRUE) THEN
        Path_10;
        tarea1 := TRUE; !Señal fin tarea
      ENDIF
    ENDPROC

```

ENDMODULE

5.2.3. Tarea T_ROB3

MODULE Module1

CONST robtarget Target_10:=[[1237.30815974,2200,891.685578915],
[0.069881481,0,0.997555301,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_60:=[[-1021.588,-1161.499,505.991],
[0,0.707106781,0.707106781,0], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_20:=[[-1021.588,-1161.499,355.991],
[0,0.707106781,0.707106781,0], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_30:=[[-1021.588,-911.499,355.991],
[0,0.707106781,0.707106781,0], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_40:=[[-1271.588,-911.499,355.991],
[0,0.707106781,0.707106781,0], [-2,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_50:=[[-1271.588,-1161.499,355.991],
[0,0.707106781,0.707106781,0], [-2,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

PERS bool tarea2;

PERS bool mesa2libre;

PROC Path_10()

MoveJ Target_10,v1000,z100,PKI_500\WObj:=wobj0;

MoveJ Target_60,v1000,z100,PKI_500\WObj:=Workobject_2;

MoveL Target_20,v300,fine,PKI_500\WObj:=Workobject_2;

MoveL Target_30,v300,fine,PKI_500\WObj:=Workobject_2;

MoveL Target_40,v300,fine,PKI_500\WObj:=Workobject_2;

MoveL Target_50,v300,fine,PKI_500\WObj:=Workobject_2;

MoveL Target_20,v300,fine,PKI_500\WObj:=Workobject_2;

MoveL Target_60,v1000,fine,PKI_500\WObj:=Workobject_2;

MoveJ Target_10,v1000,z100,PKI_500\WObj:=wobj0;

ENDPROC

!Tarea soldado cilindro

PROC main()

SetDO sensor2act,1; !Obtención datos sensor

!La tarea se activa si hay un pieza en la mesa, la tarea no ha sido
realizada y la trayectoria del robot esta despejada de otro robot

IF ((sensor2 > 0) AND (tarea2 = FALSE))

AND (mesa2libre = TRUE) THEN

Path_10;

tarea2 := TRUE; !Señal tarea finalizada

ENDIF

ENDPROC

ENDMODULE

5.3. Programación RAPID escenario 3

5.3.1. Programa Sistema7

MODULE Module1

```
CONST robtarget Target_10:=[[1073.515940163,0,1077.499948513],  
[0.499999989,0,0.866025467,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_30:=[[100,1800,200],  
[0,0.707106781,0.707106781,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_20:=[[100,1800,0], [0,0.707106781,0.707106781,0], [-  
1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_80:=[[977.417557498,2600,460.117093341],  
[0.015784004,0,0.999875425,0], [0,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_50:=[[100,3400,200], [0,-  
0.707106781,0.707106781,0], [1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_40:=[[100,3400,0], [0,-0.707106781,0.707106781,0],  
[1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_70:=[[1324.028,2597.668,500], [0,0,1,0], [-1,0,-1,0],  
[9E9,9E9,9E9,9E9,9E9,9E9]];  
CONST robtarget Target_60:=[[1424.028,2597.668,274.083], [0,0,1,0], [-1,0,-  
1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];  
VAR intnum parada;  
VAR intnum iniciar;
```

PROC Path_10()

```
MoveJ Target_10,v1000,z100,Ventosa1\WObj:=wobj0;  
MoveJ Target_30,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveL Target_20,v1000,fine,Ventosa1\WObj:=Workobject_1;  
SetDO herramienta1,1;  
MoveL Target_30,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveJ Target_80,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveJ Target_50,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveL Target_40,v1000,fine,Ventosa1\WObj:=Workobject_1;  
SetDO herramienta1,0;  
MoveL Target_50,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveJ Target_10,v1000,z100,Ventosa1\WObj:=wobj0;
```

ENDPROC

PROC Path_20()

```
MoveJ Target_10,v1000,z100,Ventosa1\WObj:=wobj0;  
MoveJ Target_50,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveL Target_40,v1000,fine,Ventosa1\WObj:=Workobject_1;  
SetDO herramienta1,1;  
MoveL Target_50,v1000,fine,Ventosa1\WObj:=Workobject_1;  
MoveJ Target_80,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveJ Target_70,v1000,z100,Ventosa1\WObj:=Workobject_1;  
MoveL Target_60,v1000,fine,Ventosa1\WObj:=Workobject_1;  
SetDO herramienta1,0;  
MoveL Target_70,v1000,fine,Ventosa1\WObj:=Workobject_1;  
MoveJ Target_10,v1000,z100,Ventosa1\WObj:=wobj0;
```

ENDPROC

```

PROC ActSensores()
    SetDO ActivarsensorMesa1,1;
ENDPROC

TRAP Paro_estacion
    StopMove; !Detiene al robot
    StorePath; !Guarda en memoria el recorrido que se estaba
    haciendo
    ClearPath; !Limpia la cola de tareas y de esta forma deja de
    ejecutarse la última instrucción
ENDTRAP

TRAP Marcha_estacion
    RestoPath; !Carga el path guardado en la cola
    StartMove; !Permite movimiento del robot
ENDTRAP

PROC main()

    IEnable; !Activa interrupciones
    CONNECT parada WITH Paro_estacion; !Conecta la señal con la
    interrupción

    CONNECT iniciar WITH Marcha_estacion;
    ISignalDI interrupcionparo,1,parada;
    ISignalDI continuar,1,iniciar;

    ActSensores; !Toma datos de los sensores
    SetDO PiezaSoltada,0; !Reinicia la variable

    inicio: !Etiqueta ejecución cíclica

    !Coloca pieza en la mesa2 para realizar tarea
    IF (DatosSensorMesa1 = 1) AND (DatosSensorMesa2 = 0)
    THEN

        SetDO OcuparMesa,1;
        Path_10;
        SetDO OcuparMesa,0;
        ENDIF

        !Quita la pieza de la mesa2 con la tarea finalizada
        IF ((DatosSensorMesa2 = 1) AND (EstadoTarea = 1))
        AND (DatosSensorCinta = 0) THEN
            SetDO OcuparMesa,1;
            Path_20;
            SetDO OcuparMesa,0;

            !Para el funcionamiento de la simulación
            SetDO PiezaSoltada,1;

```

ENDIF

GOTO inicio;

ENDPROC
ENDMODULE

5.3.2. Programa Sistema8

MODULE Module1

CONST robtarget Target_100:= [[-285.459217565,1008.970109218,
1016.016708033], [0.005676789,0.305737381,-0.952097214,0.00182293], [0,-1,-1,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_10:= [[1073.515940163,0,1077.499948513],
[0.499999891,0,0.866025467,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_120:= [[-992.86,1703.256,1056], [0,-
0.707106781,0.707106781,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_110:= [[-992.86,1703.256,829.553], [0,-
0.707106781,0.707106781,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_30:= [[200,100,356], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_20:= [[200,100,200], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_50:= [[200,350,356], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_40:= [[200,300,200], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_70:= [[200,550,356], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_60:= [[200,500,200], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_90:= [[200,750,356], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_80:= [[200,700,200], [0,0,1,0], [0,0,0,0],
[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST intnum DISTANCIA_CAJAS := 200;

VAR robtarget aprox := Target_30;

VAR robtarget ptapilado := Target_20;

VAR intnum cuenta := 0;

VAR intnum parada;

VAR intnum iniciar;

PROC Path_10()

MoveJ Target_10,v1000,z100,Ventosa2\WObj:=wobj0;

MoveJ Target_120,v1000,z100,Ventosa2\WObj:=Workobject_1;

MoveL Target_110,v1000,fine,Ventosa2\WObj:=Workobject_1;

SetDO herramienta2,1;

MoveL Target_120,v1000,z100,Ventosa2\WObj:=Workobject_1;

MoveJ Target_100,v1000,z100,Ventosa2\WObj:=Workobject_1;

ENDPROC

```

PROC Path_20()
    MoveJ Target_100,v1000,z100,Ventosa2\WObj:=Workobject_1;
    MoveJ aprox,v1000,z100,Ventosa2\WObj:=Workobject_1;
    MoveL ptapilado,v1000,fine,Ventosa2\WObj:=Workobject_1;
    SetDO herramienta2,0;
    MoveL aprox,v1000,z100,Ventosa2\WObj:=Workobject_1;
    MoveJ Target_10,v1000,z100,Ventosa2\WObj:=wobj0;
ENDPROC
PROC ActSensores()
    SetDO Actsensor1,1;
    SetDO Actsensor2,1;
    SetDO Actsensor3,1;
ENDPROC

TRAP Paro_estacion
    StopMove; !Detiene al robot
    StorePath; !Guarda en memoria la trayectoria que se estaba
    haciendo
    ClearPath; !Limpia la cola de trayectorias y de esta forma deja de
    ejecutarse la última
ENDTRAP

TRAP Marcha_estacion
    RestoPath; !Carga el path guardado en la cola
    StartMove; !Permite movimiento del robot
ENDTRAP

PROC main()

    VAR intnum i:=0; !variable auxiliar

    IEnable; !Activa interrupciones
    CONNECT parada WITH Paro_estacion; !Conecta la señal con la
    interrupción

    CONNECT iniciar WITH Marcha_estacion;
    ISignalDI interrupcionparo,1,parada;
    ISignalDI continuar,1,iniciar;

    ActSensores; !Toma datos del sensor

    inicio: !Etiqueta ejecución cíclica

    !Al llegar pieza la colocará en su posición correspondiente
    IF Datossensor2 = 1 THEN
    IF cuenta < 4 THEN
        Path_10; !Recoge la pieza de la cinta
        Path_20; !La deja en la posición vacía
        cuenta := cuenta + 1;
        !Incremento del punto de aproximación
        i := aprox.trans.y;

```

```

        aprox.trans.y := i + DISTANCIA_CAJAS;
        !Incremento del punto de apilado
        i := ptapilado.trans.y;
        ptapilado.trans.y := i + DISTANCIA_CAJAS;
    ELSE
        SetDO PaleLleno,1;
        WaitDI cambiarpale,1;
        cuenta := 0; !Con el cambio de pale se reinician las
variables
        aprox := Target_30;
        ptapilado := Target_20;
        SetDO PaleLleno,0;
    ENDIF
    ENDIF

    !Reinicio de cuenta por si surge la necesidad de cambiar el pale
    IF palenuevo = 1 THEN
        cuenta := 0; !Con el cambio de pale se reinician las variables
        aprox := Target_30;
        ptapilado := Target_20;
    ENDIF

    GOTO inicio;

ENDPROC

ENDMODULE

```

5.3.3. Programa Sistema9

```

MODULE Module1
    CONST robtarget Target_10:=[[1330.30815974,0,838.185578915],
[0.069881481,0,0.997555301,0], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_60:=[[300,0,300], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_20:=[[200,0,200], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_30:=[[200,0,100], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_40:=[[200,200,100], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_70:=[[200,300,200], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_50:=[[200,400,100], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_80:=[[300,400,100], [0.27059805,-0.653281482,-
0.653281482,0.27059805], [-1,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
    PERS bool Tarealista := FALSE;
    VAR intnum parada;
    VAR intnum iniciar;

    PROC ActSensores()

```

```

        SetDO ActSensorMesa2,1;
ENDPROC

PROC InformarTarea()
    IF (Tarealista = TRUE) THEN
        SetDO EstadoTarea,1;
    ENDIF
    IF (Tarealista = FALSE) THEN
        SetDO EstadoTarea,0;
    ENDIF
ENDPROC

TRAP Paro_estacion
    StopMove; !Detiene al robot
    StorePath; !Guarda en memoria el recorrido que se estaba
    haciendo
    ClearPath; !Limpia la cola de tareas y de esta forma deja de
    ejecutarse la última instrucción
ENDTRAP

TRAP Marcha_estacion
    RestoPath; !Carga el path guardado en la cola
    StartMove; !Permite movimiento del robot
ENDTRAP

PROC main()

    IEnable; !Activa interrupciones
    CONNECT parada WITH Paro_estacion; !Conecta la señal con la
    interrupción
    CONNECT iniciar WITH Marcha_estacion;
    ISignalDI interrupcionparo,1,parada;
    ISignalDI continuar,1,iniciar;

    ActSensores; !Toma datos del sensor

    inicio: !Etiqueta ejecución cíclica

    !Con este if evitamos repeticiones
    IF (DatosSensorMesa2 = 0) THEN
        Tarealista := FALSE; !Re-inicialización de la variable al quitar
        pieza de la mesa
    ENDIF

    !Realización tarea con mesa libre
    IF ((MesaOcupada = 0) AND (DatosSensorMesa2 = 1))
    AND (Tarealista = FALSE) THEN
        Path_10;
        Tarealista := TRUE; !Marca fin tarea

```

ENDIF

InformarTarea; !Comunica al otro robot si la tarea se realizó

GOTO inicio;

ENDPROC
ENDMODULE

5.4. Ejemplos estación laboratorio

5.4.1. Programa 1:

```
MODULE Module1
  CONST robtarget Target_10:=[[1221.0115689, 1221.008711157, 1894.999921126],
  [0.461939777, -0.33141313, 0.800103398, 0.191341404], [0, -1, 0, 0], [9E9, 9E9, 9E9,
  9E9, 9E9, 9E9]];
  CONST robtarget Target_20:=[[32.5, 32.5, 45], [0,-0.707106781, 0.707106781,
  0], [0, 0, -1, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_30:=[[33.416, -351.87, 45], [0, -0.707106781,
  0.707106781, 0], [0, 0, -1, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_40:=[[928.204, -1027.415, 45], [0.000000004, 0, 1, 0],
  [0, -1, 0, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_50:=[[1336.73, -970.855, 45], [0.000000004, 0, 1, 0],
  [0, -1, 0, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_60:=[[653.888999899, -515.748999912, 45], [-
  0.000000003, -0.382683432, 0.923879533, 0.000000002], [0, 0, 0, 0], [9E9, 9E9, 9E9,
  9E9, 9E9, 9E9]];
  CONST robtarget Target_70:=[[32.5, 32.5, 165], [0, -0.707106781,
  0.707106781, 0], [0, 0, -1, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_80:=[[33.416, -351.87, 165], [0, -0.707106781,
  0.707106781, 0], [0, 0, -1, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_90:=[[1336.73, -970.855, 165], [0.000000004, 0, 1, 0],
  [0, -1, 0, 0], [9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
  CONST robtarget Target_100:=[[928.204, -1027.415,165], [0.000000004,0,1,0],
  [0,-1,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_110:=[[653.889,-515.749,165], [-0.000000003,-
  0.382683432, 0.923879533, 0.000000002], [0, 0, 0, 0], [9E9, 9E9, 9E9, 9E9, 9E9,
  9E9]];
  CONST robtarget Target_120:=[[653.889, -515.749, 465], [0.353553388, -
  0.353553389, 0.853553392, 0.146446611], [0, -1, 0, 0], [9E9, 9E9, 9E9, 9E9, 9E9,
  9E9]];
  CONST intnum DISTANCIA_CUBOS := 65;
  VAR robtarget aprox_apilado := Target_60;
  VAR robtarget pt_apilado := Target_110;
  VAR robtarget aprox := Target_70;
  VAR robtarget pt_coger_cubo := Target_20;
  VAR intnum cuenta := 0;

  VAR wzstationary work_limit;
```

```

VAR wzstationary max_workarea;

PROC power_on()
    VAR shapedata joint_space;
    VAR shapedata volumen;

    !Definición de los límites para los tres primeros ejes
    CONST jointtarget low_pos := [ [ 18, 9E9, 9E9, 9E9, 9E9, 9E9],[
9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
    CONST jointtarget high_pos := [ [ 79, 9E9, 9E9, 9E9,9E9, 9E9],[
9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
    !Definición cajas límite
    CONST pos corner1 := [-3700,0,0];
    CONST pos corner2 := [-6150,2400,2500];

    !Supervisión de los límites de los ejes;
    WZLimJointDef \Outside, joint_space, low_pos, high_pos;
    WZLimSup \Stat, work_limit, joint_space;
    !Supervisión de los límites del area de trabajo
    WZBoxDef \Inside, volumen, corner1, corner2;
    WZLimSup \Stat, max_workarea, volumen;

ENDPROC

PROC Path_10()
    MoveJ
    aprox,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
    SetDO herramienta,1;
    MoveL
    pt_coger_cubo,v1000,fine,Mi_mecanismo_1\WObj:=Workobject_1;
    SetDO herramienta,0;
    WaitTime 2;
    MoveL
    aprox,v1000,fine,Mi_mecanismo_1\WObj:=Workobject_1;
    MoveJ
    Target_120,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
    MoveJ
    aprox_apilado,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
    MoveL
    pt_apilado,v1000,fine,Mi_mecanismo_1\WObj:=Workobject_1;
    SetDO herramienta,1;
    WaitTime 2;
    MoveL
    aprox_apilado,v1000,fine,Mi_mecanismo_1\WObj:=Workobject_1;
    MoveJ
    Target_120,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
ENDPROC

PROC rutina1()
    VAR intnum i:=0; !variable auxiliar

```

```

    aprox := Target_70;
    pt_coger_cubo := Target_20;
    aprox_apilado := Target_110;
    pt_apilado := Target_60;
ENDPROC

PROC rutina2()
    VAR intnum i:=0; !variable auxiliar

    aprox := Target_80;
    pt_coger_cubo := Target_30;
    !Incremento del punto de aproximación
    i := aprox_apilado.trans.z;
    aprox_apilado.trans.z := i + DISTANCIA_CUBOS;
    !Incremento del punto de apilado
    i := pt_apilado.trans.z;
    pt_apilado.trans.z := i + DISTANCIA_CUBOS;
ENDPROC

PROC rutina3()
    VAR intnum i:=0; !variable auxiliar

    aprox := Target_100;
    pt_coger_cubo := Target_40;
    !Incremento del punto de aproximación
    i := aprox_apilado.trans.z;
    aprox_apilado.trans.z := i + DISTANCIA_CUBOS;
    !Incremento del punto de apilado
    i := pt_apilado.trans.z;
    pt_apilado.trans.z := i + DISTANCIA_CUBOS;
ENDPROC

PROC rutina4()
    VAR intnum i:=0; !variable auxiliar

    aprox := Target_90;
    pt_coger_cubo := Target_50;
    !Incremento del punto de aproximación
    i := aprox_apilado.trans.z;
    aprox_apilado.trans.z := i + DISTANCIA_CUBOS;
    !Incremento del punto de apilado
    i := pt_apilado.trans.z;
    pt_apilado.trans.z := i + DISTANCIA_CUBOS;
ENDPROC

PROC main()
    power_on; !Rutina de limites articular

    cuenta := 0; !inicia la cuenta
    SetDO herramienta,1; !Inicia la herramienta abierta

```

```

inicio: !Etiqueta ejecución ciclica

!Cambio de los puntos según las piezas apiladas
TEST cuenta
CASE 0: rutina1;
CASE 1: rutina2;
CASE 2: rutina3;
CASE 3: rutina4;
DEFAULT: GOTO fin;
ENDTEST

Path_10;
cuenta := cuenta + 1;

GOTO inicio;

fin:
MoveL Target_10,v1000,z100,Mi_mecanismo_1\WObj:=wobj0;
!Posición de reposo

ENDPROC
ENDMODULE

```

5.4.2. Programa 2:

```

MODULE Module1
  CONST robtarget Target_10:=[[1221.0115689,-
1221.008711157,1894.999921126], [0.461939777,0.33141313,0.800103398,-
0.191341404], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_120:=[[103.436,397.836,-38.5],
[0,0.382683433,0.923879532,0], [-1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_30:=[[103.436,397.836,-138.5],
[0,0.382683433,0.923879532,0], [-1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_170:=[[0,0,87.5], [0,0.382683433,0.923879532,0], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_110:=[[0,0,-112.5], [0,0.382683433,0.923879532,0],
[-1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_160:=[[ -588.627,-327.953,-38.5],
[0,0.382683433,0.923879532,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_40:=[[ -588.627,-327.953,-138.5],
[0,0.382683433,0.923879532,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_100:=[[0,0,-87.5], [0,0.382683433,0.923879532,0], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_150:=[[ -429.489,-548.171,-38.5],
[0,0.382683433,0.923879532,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
  CONST robtarget Target_50:=[[ -429.489,-548.171,-138.5],
[0,0.382683433,0.923879532,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

```

```

CONST robtarget Target_90:=[[0,0,-62.5], [0,0.382683433,0.923879532,0], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_130:=[[708.833,97.746,-38.5],
[0,0.382683433,0.923879532,0], [-1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_20:=[[708.833,97.746,-138.5],
[0,0.382683433,0.923879532,0], [-1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_80:=[[0,0,-37.5], [0,0.382683433,0.923879532,0], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_140:=[[-47.06,-553.06,-38.5],
[0,0.382683433,0.923879532,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_60:=[[-41.069,-535.798,-138.5],
[0,0.382683433,0.923879532,0], [-1,0,-1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget Target_70:=[[0,0,-12.5], [0,0.382683433,0.923879532,0], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
VAR wzstationary work_limit;
VAR wzstationary max_workarea;

PROC power_on()
    VAR shapedata joint_space;
    VAR shapedata volumen;

    !Definición de los límites para los tres primeros ejes
    CONST jointtarget low_pos := [ [-79, 9E9, 9E9, 9E9, 9E9, 9E9],[
9E9, 9E9, 9E9, 9E9, 9E9, 9E9]];
    CONST jointtarget high_pos := [ [-18, 9E9, 9E9, 9E9,9E9, 9E9],[
9E9, 9E9, 9E9, 9E9, 9E9, 9E9] ];
    !Definición cajas límite
    CONST pos corner1 := [3700,0,0];
    CONST pos corner2 := [6150,2400,2500];

    !Supervisión de los límites de los ejes;
    WZLimJointDef \Outside, joint_space, low_pos, high_pos;
    WZLimSup \Stat, work_limit, joint_space;
    !Supervisión de los límites del area de trabajo
    WZBoxDef \Inside, volumen, corner1, corner2;
    WZLimSup \Stat, max_workarea, volumen;

ENDPROC
PROC Path_10()
    MoveL Target_10,v1000,z100,Mi_mecanismo_1\WObj:=wobj0;
    MoveJ Target_120,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
    SetDO herramienta,1;
    MoveL Target_30,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
    WaitTime 2;
    SetDO herramienta,0;
    WaitTime 2;
    MoveL Target_120,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
    MoveJ Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;

```

```

        MoveL Target_110,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,1;
        WaitTime 2;
        MoveL Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_160,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        SetDO herramienta,1;
        MoveL Target_40,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,0;
        WaitTime 2;
        MoveL Target_160,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveL Target_100,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,1;
        WaitTime 2;
        MoveL Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_150,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        SetDO herramienta,1;
        MoveL
Target_50,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,0;
        WaitTime 2;
        MoveL Target_150,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveL Target_90,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,1;
        WaitTime 2;
        MoveL Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_130,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        SetDO herramienta,1;
        MoveL
Target_20,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,0;

```

```

        WaitTime 2;
        MoveL Target_130,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveL Target_80,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,1;
        WaitTime 2;
        MoveL Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_140,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        SetDO herramienta,1;
        MoveL
Target_60,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,0;
        WaitTime 2;
        MoveL Target_140,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveL Target_70,v1000,z100,Mi_mecanismo_1\WObj:=Workobject_1;
        WaitTime 2;
        SetDO herramienta,1;
        WaitTime 2;
        MoveL Target_170,v1000,z100,
Mi_mecanismo_1\WObj:=Workobject_1;
        MoveJ Target_10,v1000,z100,Mi_mecanismo_1\WObj:=wobj0;
ENDPROC

PROC main()
    power_on; !Rutina de limites articulares
    SetDO herramienta,1;
    Path_10;
ENDPROC

ENDMODULE

```

6. Índice de ilustraciones

Ilustración 73 Especificaciones IRB6400F (medidas en mm)	2
Ilustración 74 Especificaciones IRB6400R (medidas en mm)	3
Ilustración 75 Especificaciones mesas laboratorio (medidas en mm)	4
Ilustración 76 Aplicación del teorema de Tales al soporte de los aros	5
Ilustración 77 Lógica de estación del Escenario2	6
Ilustración 78 Path10 T_ROB1	7
Ilustración 79 Path20 T_ROB1	8
Ilustración 80 Path30 T_ROB1	8
Ilustración 81 Path40 T_ROB1	9