



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Electrónica Industrial y Automática

MÁSTER EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA DE INGENIERÍAS INDUSTRIALES
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

Diseño, simulación y fabricación de una
estación robotizada universitaria

ANEXOS

Autor: D. Víctor Lobo Granado

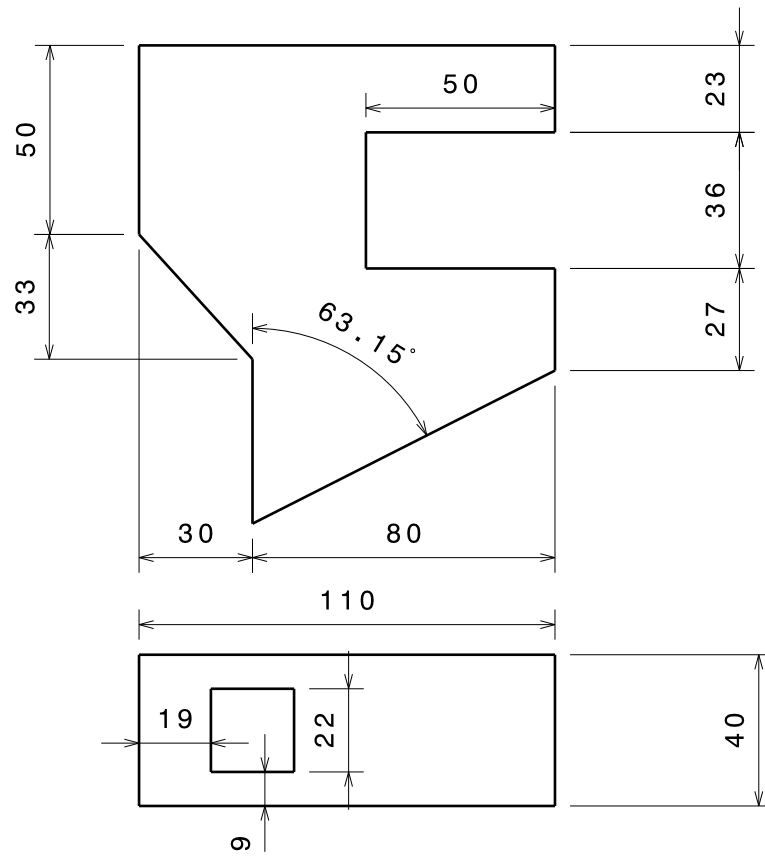
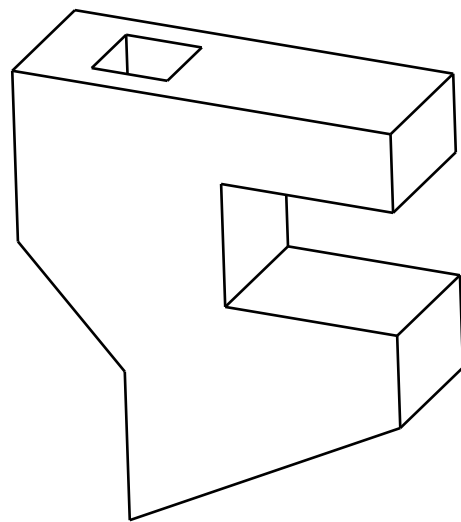
Tutor: D. Alberto Herreros López

Valladolid, Julio, 2020.

Indice

ANEXO I. PLANOS DE PIEZAS	83
ANEXO II. CODIGOS MATLAB	105
Clase Robot.m	105
Clase Robot_Simulacion.m.....	117
Clase Herramienta.m	130
Clase tcp_ABB.m	132
Clase Kin.m.....	138
ANEXO III. CODIGO RAPID	143
ANEXO IV. ESTUDIO ECONOMICO	151
Costes directos del proyecto	151
Costes asociados al equipo.....	151
Coste asociado a la mano de obra.....	152
Costes indirectos del proyecto	153
Coste total del proyecto	153

ANEXO I. PLANOS DE PIEZAS

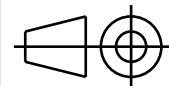


DESIGNED BY:
Victor Lobo
 DATE:
17/05/2020

CHECKED BY:
Alberto Herreros
 DATE:
17/05/2020

Anclaje Mesa

SIZE
A4



DASSAULT SYSTEMES

SCALE
1:2

WEIGHT (kg)
XXX

DRAWING NUMBER
XXX

SHEET
1/1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D C B A

4

3

2

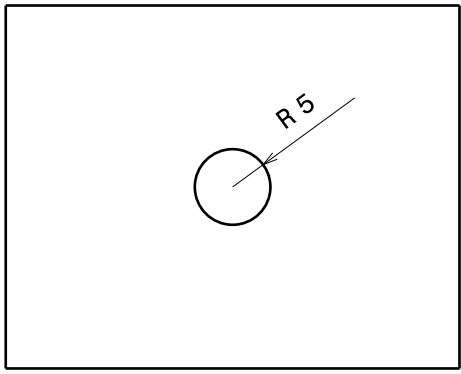
1

4

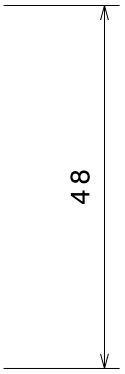
3

2

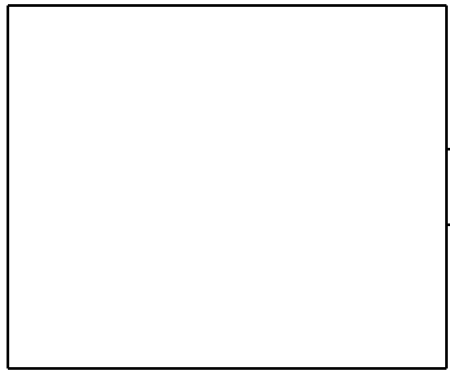
1



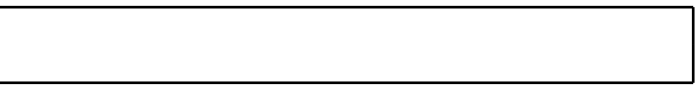
60



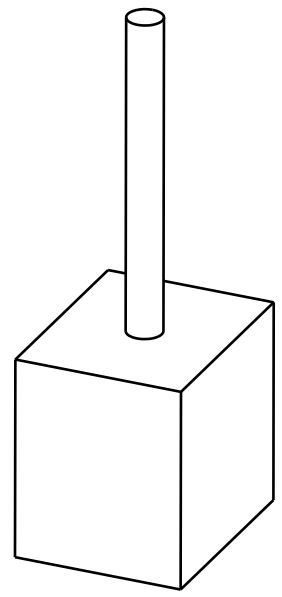
48



58



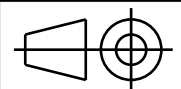
150



DESIGNED BY:
Victor Lobo
DATE:
17/05/2020
CHECKED BY:
Alberto Herreros
DATE:
17/05/2020

Barra 15cm

SIZE
A4



DASSAULT SYSTEMES

SCALE
1:2

WEIGHT (kg)
XXX

DRAWING NUMBER
XXX

SHEET
1/1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D A

D

C

B

A

4

4

3

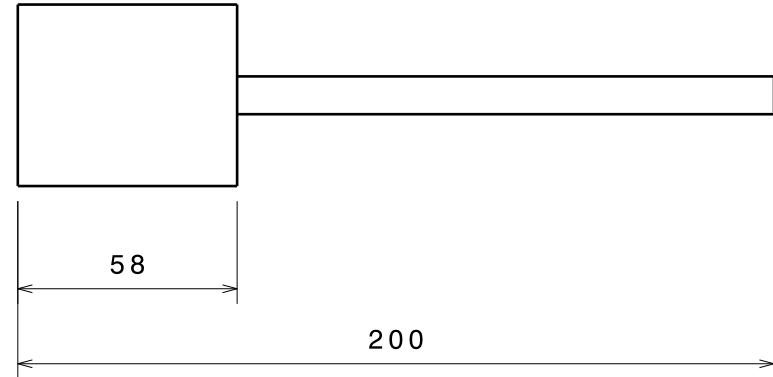
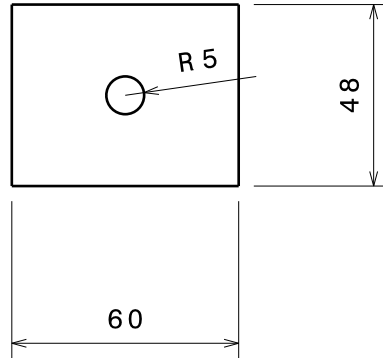
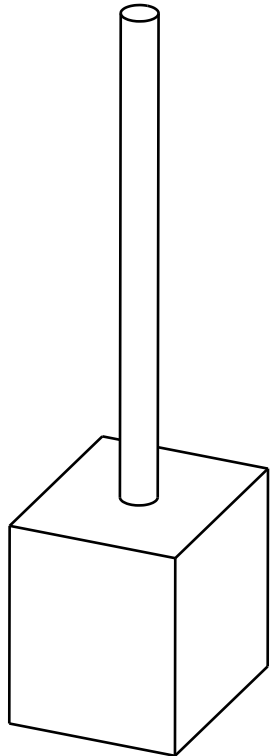
3

2

2

1

1



DESIGNED BY: Victor Lobo		<h1>Barra 20cm</h1>
DATE: 17/05/2020		
CHECKED BY: Alberto Herreros		<h2>DASSAULT SYSTEMES</h2>
DATE: 17/05/2020		
SIZE A4		
SCALE 1:4	WEIGHT (kg) XXX	DRAWING NUMBER XXX
		SHEET 1/1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

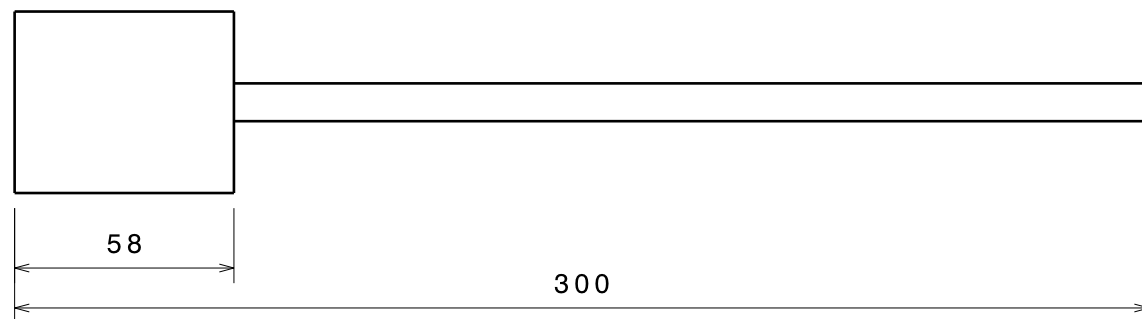
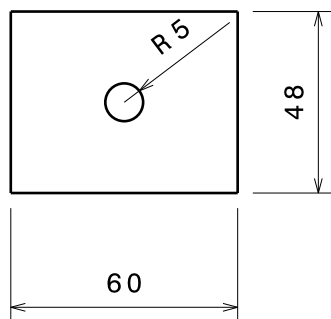
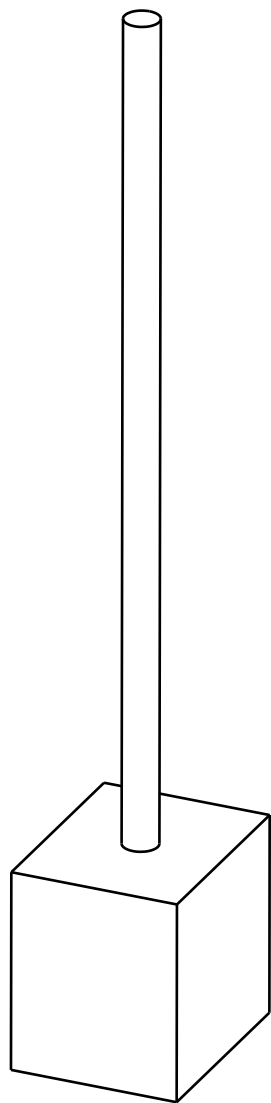
A

4

3

2

1

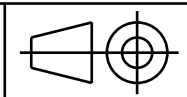


DESIGNED BY:
Victor Lobo
DATE:
17/05/2020

CHECKED BY:
Alberto Herreros
DATE:
17/05/2020

Barra 30cm

SIZE
A4



DASSAULT SYSTEMES

SCALE
1:2

WEIGHT (kg)
XXX

DRAWING NUMBER
XXX

SHEET
1 / 1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

A

4

4

3

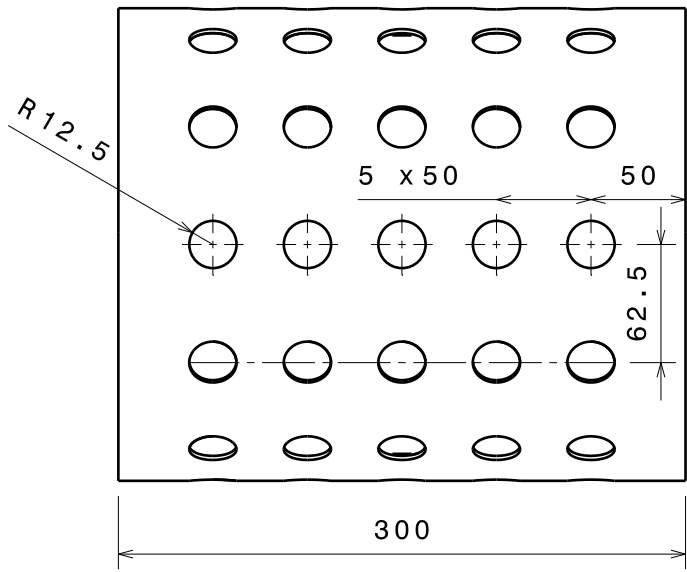
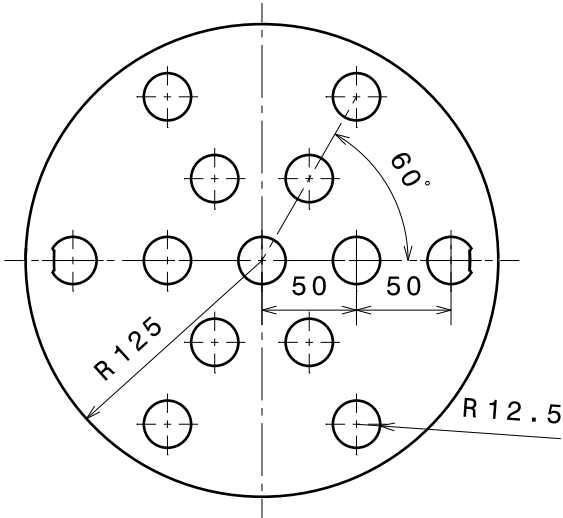
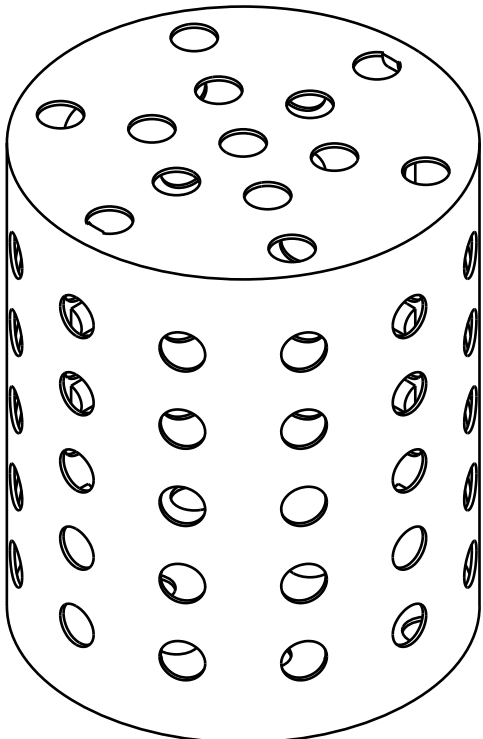
3

2

2

1

1



DESIGNED BY: Victor Lobo		CILINDRO
DATE: 16/05/2020		
CHECKED BY: Alberto Herreros		DASSAULT SYSTEMES
DATE: 16/05/2020		
SIZE A4		
SCALE 1:4	WEIGHT (kg) 1	DRAWING NUMBER 1
		SHEET 1/1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

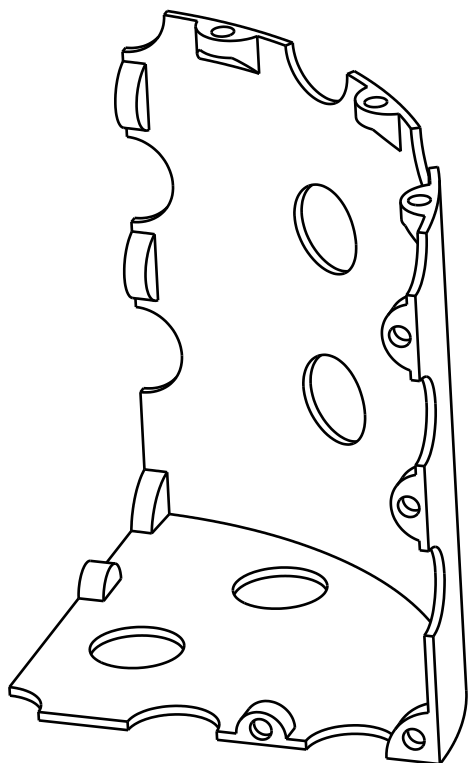
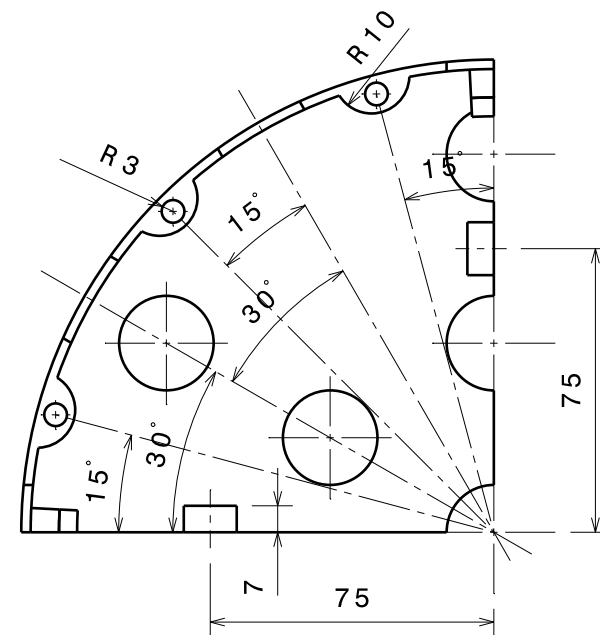
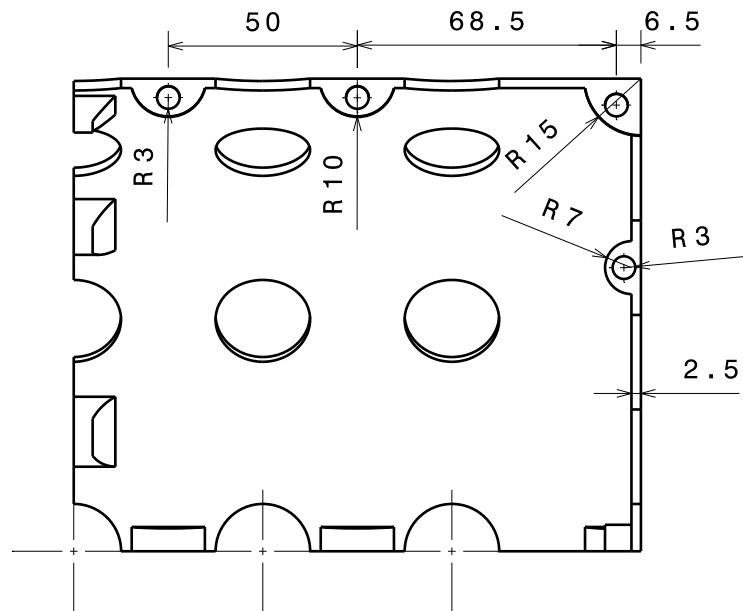
A

4

3

2

1



DESIGNED BY: Victor Lobo	
DATE: 17/05/2020	
CHECKED BY: Alberto Herreros	
DATE: 17/05/2020	
SIZE A4	
SCALE 1:4	WEIGHT (kg) 0.1

<h1>Cilindro Cortado</h1>	
DRAWING NUMBER XXX	
SHEET 1 / 1	

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

A

4

4

3

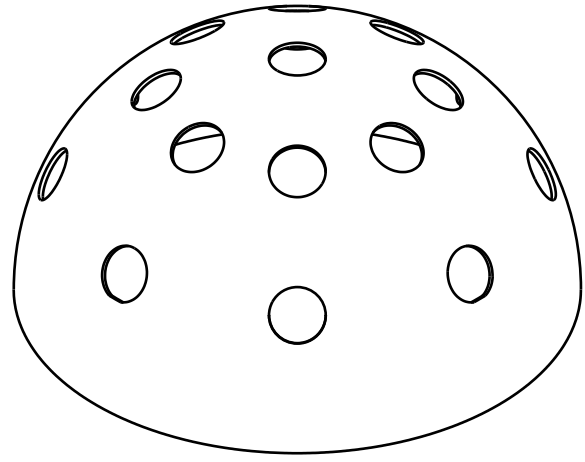
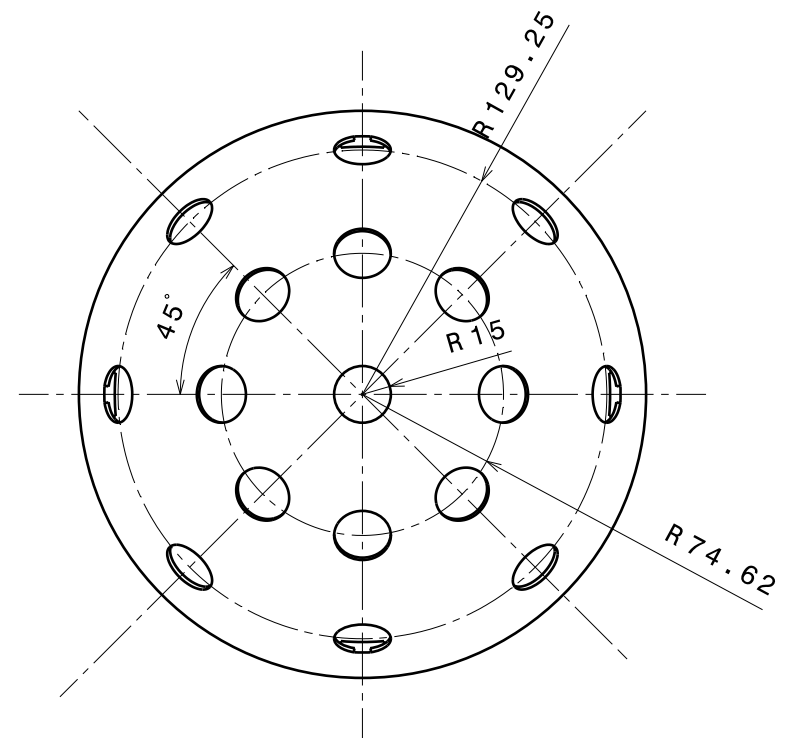
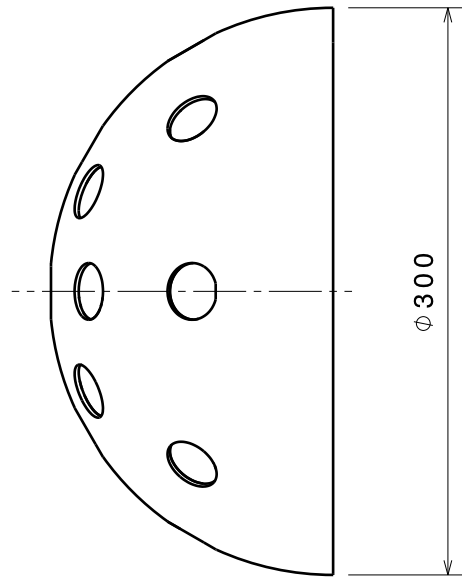
3

2

2

1

1

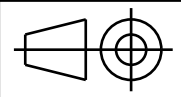


DESIGNED BY:
Victor Lobo
 DATE:
17/05/2020

CHECKED BY:
Alberto Herreros
 DATE:
17/05/2020

Esfera

SIZE
A4



DASSAULT SYSTEMES

SCALE
1:4

WEIGHT (kg)
0.1

DRAWING NUMBER
XXX

SHEET
1/1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

A

4

4

3

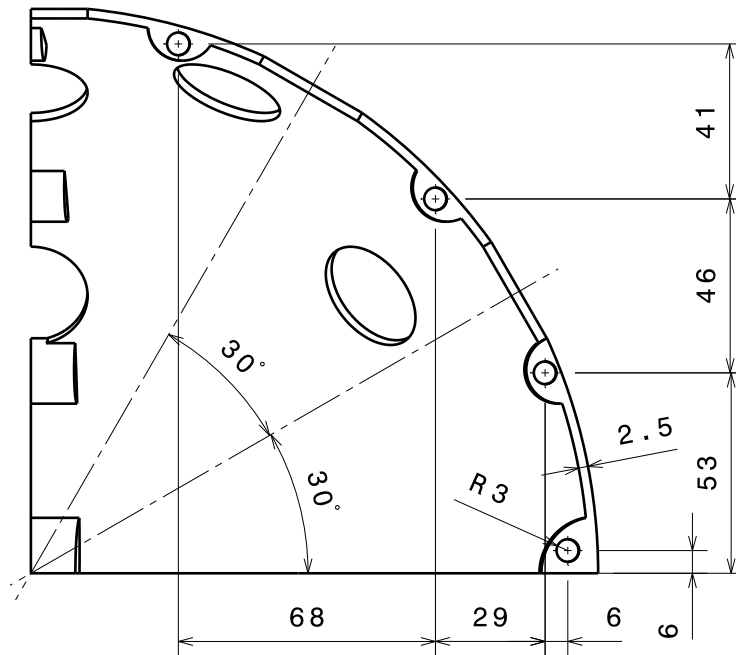
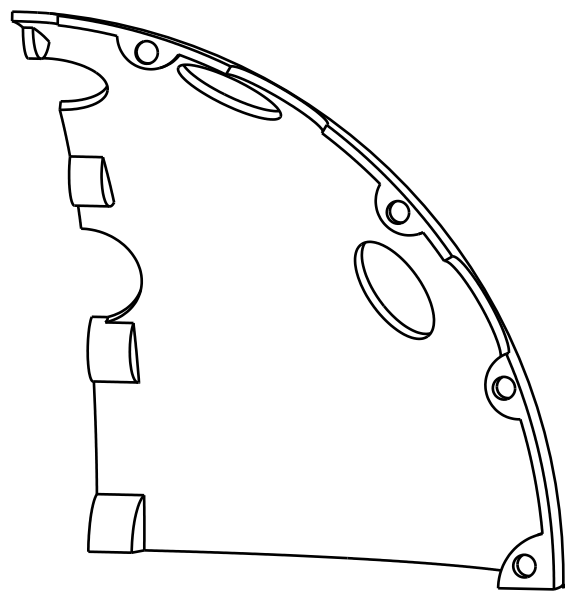
3

2

2

1

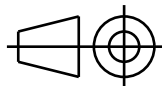
1



DESIGNED BY:
Victor Lobo
 DATE:
17/05/2020
 CHECKED BY:
Alberto Herreros
 DATE:
17/05/2020

Esfera Cortada

SIZE
A4



DASSAULT SYSTEMES

SCALE
1:2

WEIGHT (kg)
0.1

DRAWING NUMBER
XXX

SHEET
1 / 1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

A

4

4

3

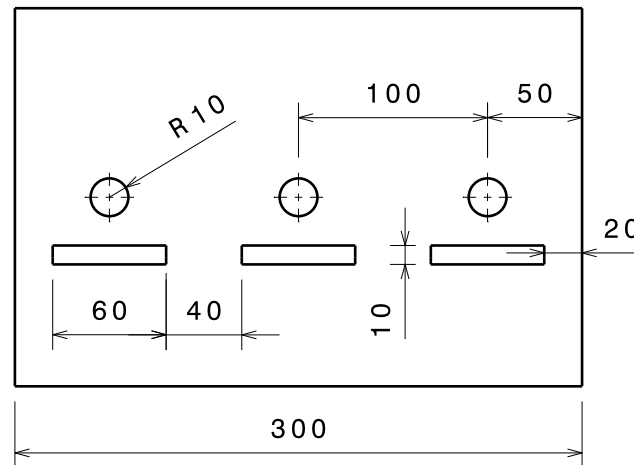
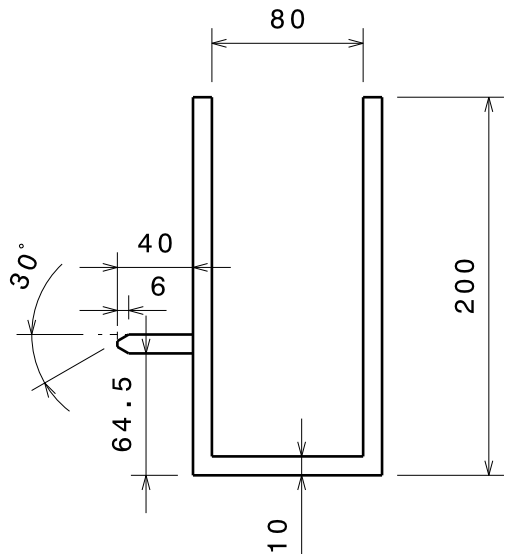
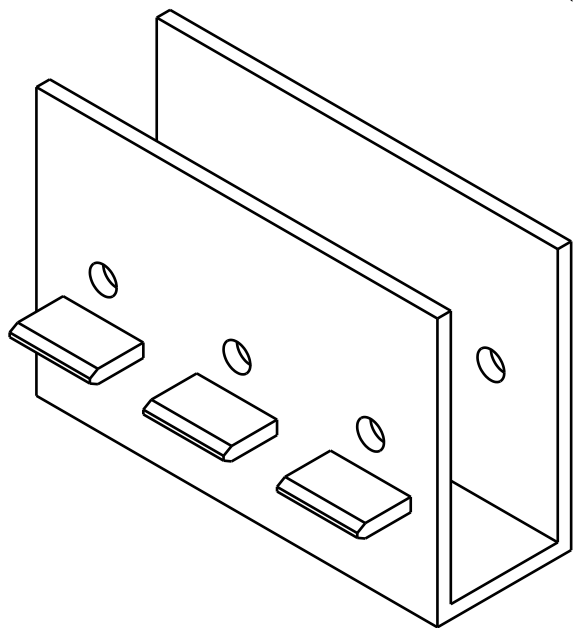
3

2

2

1

1



DESIGNED BY:
Victor Lobo
 DATE:
17/05/2020
 CHECKED BY:
Alberto Herreros
 DATE:
17/05/2020

Soporte Barras

SIZE
A4

DASSAULT SYSTEMES

SCALE
1 : 4

WEIGHT (kg)
XXX

DRAWING NUMBER
XXX

SHEET
1 / 1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

D

C

B

A

4

4

3

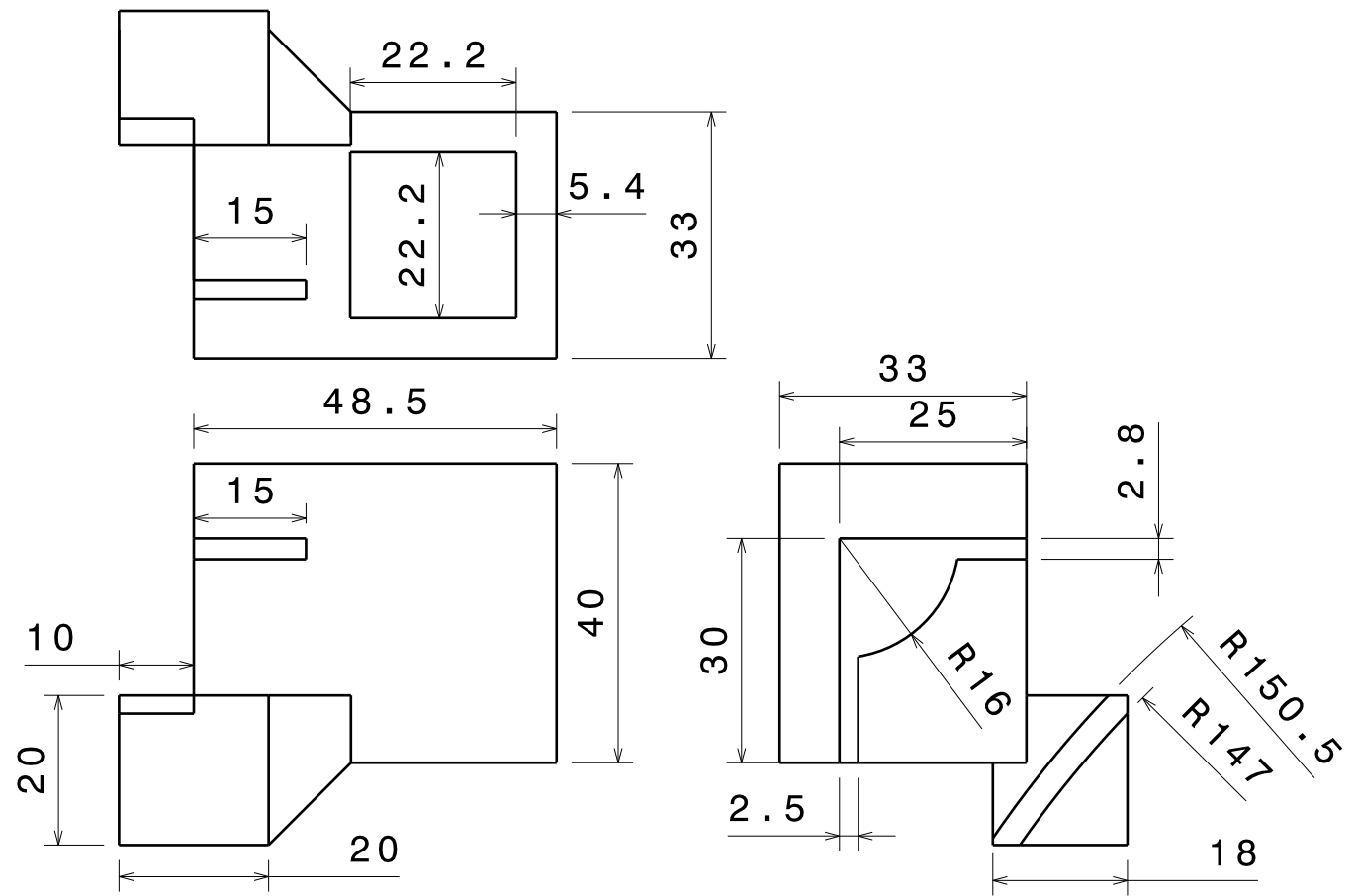
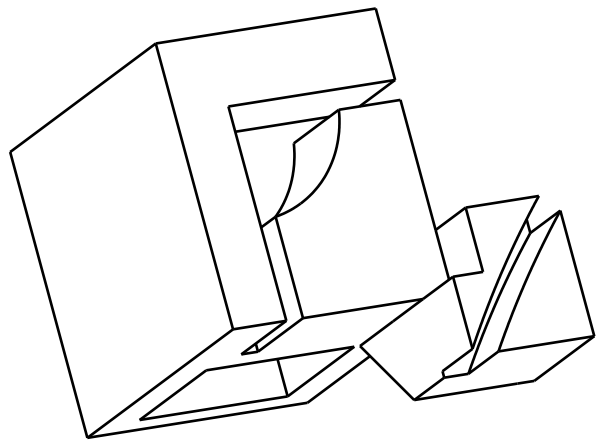
3

2

2

1

1



DESIGNED BY:
Victor Lobo
 DATE:
 17/05/2020
 CHECKED BY:
Alberto Herreros
 DATE:
 17/05/2020

Soporte objetos

SIZE
A4

DASSAULT SYSTEMES

SCALE
1:1

WEIGHT (kg)
XXX

DRAWING NUMBER
XXX

SHEET
1 / 1

I	-
H	-
G	-
F	-
E	-
D	-
C	-
B	-
A	-

This drawing is our property; it can't be reproduced or communicated without our written agreement.

D

A

ANEXO II. CODIGOS MATLAB

Clase Robot.m

```
classdef Robot < handle
    %UNTITLED2 Summary of this class goes here
    % Detailed explanation goes here

    properties
        R;
        robot_simulado;

        h = Hmat;
        a = [0;0;0];
        b = [0,0,0,1];
        radio_esfera = 160;
        radio_cilindro = 140;
        altura_cilindro = 307;

        herramienta;
        tPinza;
        tRoja;
        tVerde;
        tNaranja;

        wBase;
        wBase_Herramienta;
        wBarras;
        wEsfera;
        wEsfera_Pared_Derecha;
        wEsfera_Techo;
        wCilindro;
        wCilindro_Pared_Derecha;
        wCilindro_Techo;

    end

    methods

        function this = Robot()
            this.tPinza = this.h.t([0,0,83]);
            this.tRoja = this.h.t([0,0,260]);
            this.tVerde = this.h.t([0,0,210]);
            this.tNaranja = this.h.t([0,0,360]);
            this.wBase = this.h.tQ([0,0,0],[1,0,0,0]);
            this.wBase_Herramienta = this.h.tQ([[0,0,0],[0,0,1,0]]);
            this.wBarras =
            this.h.tQ([250.541304602,250,0],[0.707106781,0.707106781,0,0]);
            this.wEsfera = this.h.tQ([0.541304602,-350,0],[1,0,0,0]);
            this.wEsfera_Pared_Derecha = this.h.tQ([0,-
            750,350],[0,0,0.707106781,0.707106781]);
            this.wEsfera_Techo =
            this.h.tQ([400.541304602,0,950],[0,0.707106781,0.707106781,0]);
            this.wCilindro = this.h.tQ([200,-300,0],[1,0,0,0]);
            this.wCilindro_Pared_Derecha = this.h.tQ([-150,-
            750,350],[0,0,0.707106781,0.707106781]);
            %this.wCilindro_Techo =
            this.h.tQ([200,150,975],[0,0.707106781,0.707106781,0]);
            this.wCilindro_Techo =
            this.h.tQ([200,0,950],[0,0.707106781,0.707106781,0]);
            this.R = tcpABB('127.0.0.1',30000);
        end

        function Desconectar(this)
            %UNTITLED2 Construct an instance of this class
            % Detailed explanation goes here
        end
    end
end
```

```

        disp(this.R);
        this.R.Disconnect;
        this.R = [];
        disp(this.R);
    end
    function Stop(this)
        this.R.Stop;
    end
    function SetPinza(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Tool(this.tPinza);
    end
    function AbrirPinza(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.OpenTool();
    end
    function CerrarPinza(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.CloseTool();
    end
    function SetBarraRoja(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Tool(this.tRoja);
    end
    function SetBarraVerde(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Tool(this.tVerde);
    end
    function SetBarraNaranja(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Tool(this.tNaranja);
    end
    function SetBase(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Wobj(this.wBase);
    end
    function SetBarras(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Wobj(this.wBarras);
    end
    function SetEsfera(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Wobj(this.wEsfera);
    end
    function SetEsferaParedDerecha(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Wobj(this.wEsfera_Pared_Derecha);
    end
    function SetEsferaTecho(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Wobj(this.wEsfera_Techo);
    end
    function SetCilindro(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Wobj(this.wCilindro);
    end
end

```

```

function SetCilindroParedDerecha(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    this.R.Wobj(this.wCilindro_Pared_Derecha);
end
function SetCilindroTecho(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    this.R.Wobj(this.wCilindro_Techo);
end
function MovimientoAbsoluto(this, ang_articulacion)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    this.R.MoveAbsJ(ang_articulacion, 'v500', 'z0');
end
function MovimientoJ(this, punto, rotacion)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    switch nargin
        case 2
            this.R.MoveJ(this.h.t(punto(1), punto(2), punto(3)) * this.wBase_Herramienta, 'v300', 'z0');
        case 3
            this.R.MoveJ(this.h.t(punto(1), punto(2), punto(3)) * (this.wBase_Herramienta * rotacion), 'v300', 'z0');
        end
    end
function MovimientoL(this, punto, rotacion)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    switch nargin
        case 2
            this.R.MoveL(this.h.t(punto(1), punto(2), punto(3)) * this.wBase_Herramienta);
        case 3
            this.R.MoveL(this.h.t(punto(1), punto(2), punto(3)) * (this.wBase_Herramienta * rotacion));
        end
    end
function GuardarPunto(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    this.R.Path(1);
end
function RepetirSecuencia(this, wobj)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    switch wobj
        case 'Base'
            this.MovimientoAbsoluto([0,0,0,0,90,0]);
        case 'Mesa'
            this.MovimientoAbsoluto([-90,0,0,0,90,0]);
        case 'Pared derecha'
            this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
            this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
        case 'Techo'
            this.MovimientoAbsoluto([0,-40,50,0,-100,0]);
    end
    this.R.PathJ(1);
    pause(1);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function BorrarSecuencia(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here

```

```

        this.R.DelPath;
    end
    function ResetVariables(this)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        this.R.Reset;
    end
    function obj = RotacionX(this,angulo)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        RotX = [1,0,0;0,cos(angulo),-
sin(angulo);0,sin(angulo),cos(angulo)];
        RotX = [RotX this.a];
        RotX = [RotX;this.b];
        obj = RotX;
    end
    function obj = RotacionY(this,angulo)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        RotY = [cos(angulo),0,sin(angulo);0,1,0;-
sin(angulo),0,cos(angulo)];
        RotY = [RotY this.a];
        RotY = [RotY;this.b];
        obj = RotY;
    end
    function obj = RotacionZ(this,angulo)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        RotZ = [cos(angulo),-
sin(angulo),0;sin(angulo),cos(angulo),0;0,0,1];
        RotZ = [RotZ this.a];
        RotZ = [RotZ;this.b];
        obj = RotZ;
    end
    function obj = Translacion(this,direccion)
        %UNTITLED2 Construct an instance of this class
        % Detailed explanation goes here
        Trans = [1,0,0;0,1,0;0,0,1];
        Trans = [Trans direccion];
        Trans = [Trans;this.b];
        obj = Trans;
    end
    function MovimientoManual(this,punto,rotacion,wobj,objeto)
        rx = this.RotacionX(rotacion(1) * pi / 180);
        ry = this.RotacionY(rotacion(2) * pi / 180);
        rz = this.RotacionZ(rotacion(3) * pi / 180);
        r = rx * ry * rz;
        switch wobj
            case 'Base'
                this.SetBase;
            case 'Mesa'
                switch objeto
                    case 'Esfera'
                        this.SetEsfera;
                    case 'Cilindro'
                        this.SetCilindro
                end
            case 'Pared derecha'
                switch objeto
                    case 'Esfera'
                        this.SetEsferaParedDerecha;
                    case 'Cilindro'
                        this.SetCilindroParedDerecha;
                end
            case 'Techo'
                switch objeto
                    case 'Esfera'
                        this.SetEsferaTecho;

```

```

        case 'Cilindro'
            this.SetCilindroTecho;
        end
    end
    this.MovimientoJ(punto,r);
end

function TomarBarra(this,barra)
    switch barra
        case 'Roja'
            this.herramienta =
Herramienta([50,100,120],[50,100,40],70,40,15,120,1);
        case 'Verde'
            this.herramienta =
Herramienta([150,100,120],[150,100,40],50,70,40,80,1);
        case 'Naranja'
            this.herramienta =
Herramienta([250,100,120],[250,100,40],170,30,10,160,1);
        end
        rot = this.RotacionZ(pi/2);
        this.SetBarras;
        this.SetPinza;
        this.AbrirPinza;
        this.MovimientoAbsoluto([0,0,0,0,90,0]);
        pause(1);
        this.MovimientoJ(this.herramienta.GetPunto_2,rot);
        pause(1);
        this.MovimientoL(this.herramienta.GetPunto_3,rot);
        pause(1);
        this.CerrarPinza;
        pause(1);
        this.MovimientoL(this.herramienta.GetPunto_4,rot);
        pause(1);
        this.MovimientoL(this.herramienta.GetPunto_5,rot);
        pause(1);
        this.MovimientoAbsoluto([0,0,0,0,90,0]);
        switch barra
            case 'Roja'
                this.SetBarraRoja;
            case 'Verde'
                this.SetBarraVerde;
            case 'Naranja'
                this.SetBarraNaranja;
            end
        end
        this.SetBase;
    end
function DejarBarra(this)
    rot = this.RotacionZ(pi/2);
    this.SetBarras;
    this.SetPinza;
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
    pause(1);
    this.MovimientoJ(this.herramienta.GetPunto_5,rot);
    pause(1);
    this.MovimientoL(this.herramienta.GetPunto_4,rot);
    pause(1);
    this.MovimientoL(this.herramienta.GetPunto_3,rot);
    pause(2);
    this.AbrirPinza;
    pause(1);
    this.MovimientoL(this.herramienta.GetPunto_2,rot);
    pause(1);
    this.MovimientoJ(this.herramienta.GetPunto_6,rot);
    pause(1);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
    this.SetBase;
end
function EsferaMesaSencilla(this)

```

```

this.SetEsfera;
this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
switch this.herramienta.GetEntradaEsfera
    case 30
        for theta = linspace(pi/6,-pi/6,3)

            x = this.radio_esfera*sin(theta);
            x2 = this.herramienta.entrada_esfera*sin(theta);
            y = 0;
            z = this.radio_esfera*cos(theta);
            z2 = this.herramienta.entrada_esfera*cos(theta);

            RotY = this.RotacionY(theta);

            this.MovimientoL([x,y,z],RotY);
            this.MovimientoL([x2,y,z2],RotY);
            this.MovimientoL([x,y,z],RotY);
            this.MovimientoL([x-15,y+30,z+50],RotY);
        end
    otherwise
        for theta = linspace(pi/3,-pi/3,5)

            x = this.radio_esfera*sin(theta);
            x2 = this.herramienta.entrada_esfera*sin(theta);
            y = 0;
            z = this.radio_esfera*cos(theta);
            z2 = this.herramienta.entrada_esfera*cos(theta);

            RotY = this.RotacionY(theta);

            this.MovimientoL([x,y,z],RotY);
            %this.GuardarPunto(guardar);
            this.MovimientoL([x2,y,z2],RotY);
            %this.GuardarPunto(guardar);
            this.MovimientoL([x,y,z],RotY);
            %this.GuardarPunto(guardar);
            this.MovimientoL([x-15,y+30,z+50],RotY);
        end
    end

this.MovimientoAbsoluto([-90,0,0,0,90,0]);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaMesaRotando(this)
    this.SetEsfera;
    this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
    for theta = linspace(pi/6,-pi/6,3)
        switch this.herramienta.GetEntradaEsfera
            case 30
                for alpha = linspace(pi/4,0,2)

                    x = this.radio_esfera*sin(theta)*cos(alpha);
                    x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha); %70
                    z = this.radio_esfera*cos(theta);
                    z2 = this.herramienta.entrada_esfera*cos(theta);

                    RotY = this.RotacionY(theta);

                    if (theta == -pi/6)
                        y = this.radio_esfera*sin(theta)*sin(-alpha);
                        y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(-alpha);
                        RotZ = this.RotacionZ(alpha);
                    else
                        y = this.radio_esfera*sin(theta)*sin(alpha);
                        y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);

```



```

        RotZ = this.RotacionZ(-alpha);
    end

    Rotacion = RotZ*RotY;

    this.MovimientoJ([x,y,z],Rotacion);
    this.MovimientoL([x2,y2,z2],Rotacion);
    this.MovimientoL([x,y,z],Rotacion);
    this.MovimientoL([x,y,z+9],RotY);
    if(theta == 0)
        break
    end
end
otherwise
    for alpha = linspace(-pi/4,pi/4,3)

        x = this.radio_esfera*sin(theta)*cos(alpha);
        x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
        y = this.radio_esfera*sin(theta)*sin(alpha);
        y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
        z = this.radio_esfera*cos(theta);
        z2 = this.herramienta.entrada_esfera*cos(theta);

        RotY = this.RotacionY(theta);
        RotZ = this.RotacionZ(-alpha);

        Rotacion = RotZ*RotY;

        this.MovimientoJ([x,y,z],Rotacion);
        this.MovimientoL([x2,y2,z2],Rotacion);
        this.MovimientoL([x,y,z],Rotacion);
        this.MovimientoL([x,y,z+9],RotY);
        if(theta == 0)
            break
        end
    end
end
end
this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaParedDerechaSencilla(this)
    this.SetEsferaParedDerecha;
    this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
    this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
    this.MovimientoJ([210,0,150]);
    for theta = linspace(pi/6,-pi/6,3)

        x = this.radio_esfera*sin(theta);
        x2 = this.herramienta.entrada_esfera*sin(theta);
        y = 0;
        z = this.radio_esfera*cos(theta);
        z2 = this.herramienta.entrada_esfera*cos(theta);

        RotY = this.RotacionY(theta);

        this.MovimientoL([x,y,z],RotY);
        this.MovimientoL([x2,y,z2],RotY);
        this.MovimientoL([x,y,z],RotY);
        this.MovimientoL([x-110,y+60,z+75],RotY);
    end
    this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
    this.MovimientoAbsoluto([-90,-10,0,0,30,0]);
    for theta = linspace(pi/6,-pi/6,3)

        y = this.radio_esfera*sin(theta);

```

```

        y2 = this.herramienta.entrada_esfera*sin(theta);
        x = 0;
        z = this.radio_esfera*cos(theta);
        z2 = this.herramienta.entrada_esfera*cos(theta);

        RotX = this.RotacionX(theta);

        this.MovimientoL([x,y,z],RotX);
        this.MovimientoL([x,y2,z2],RotX);
        this.MovimientoL([x,y,z],RotX);
        if(theta == pi/6)
            this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
        elseif(theta == 0)
            this.MovimientoL([x,y,z+35],RotX);
        else
            this.MovimientoL([x,y,z+35]);
        end
    end
    this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaParedDerechaRotando(this)
    this.SetEsferaParedDerecha;
    this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
    this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
    this.MovimientoJ([210,0,175]);
    for theta = linspace(pi/6,-pi/6,3)
        for alpha = linspace(pi/4,-pi/4,3)

            x = this.radio_esfera*sin(theta)*cos(alpha);
            x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
            y = this.radio_esfera*sin(theta)*sin(alpha);
            y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
            z = this.radio_esfera*cos(theta);
            z2 = this.herramienta.entrada_esfera*cos(theta);

            RotY = this.RotacionY(theta);
            RotZ = this.RotacionZ(-alpha);
            Rotacion = RotZ*RotY;

            this.MovimientoL([x,y,z],Rotacion);
            this.MovimientoL([x2,y2,z2],Rotacion);
            this.MovimientoL([x,y,z],Rotacion);
            this.MovimientoL([x-40,y+30,z+75],Rotacion);
            if(theta == 0)
                break
            end
        end
    end
    this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
    this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaTechoSencilla(this)
    this.SetEsferaTecho;
    this.MovimientoAbsoluto([-30,-40,50,0,-100,0]);
    this.MovimientoJ([-250,-100,100]);
    for theta = linspace(-pi/6,pi/6,3)

        x = this.radio_esfera*sin(theta);
        x2 = this.herramienta.entrada_esfera*sin(theta);
        y = 0;
        z = this.radio_esfera*cos(theta);
        z2 = this.herramienta.entrada_esfera*cos(theta);
        RotY = this.RotacionY(theta);

```

```

        this.MovimientoL([x,y,z],RotY);
        this.MovimientoL([x2,y,z2],RotY);
        this.MovimientoL([x,y,z],RotY);
        this.MovimientoL([x+25,y+15,z+5.95],RotY);
    end
    this.MovimientoJ([250,-100,100]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaTechoRotando(this)
    this.SetEsferaTecho;
    this.MovimientoAbsoluto([-30,-40,50,0,-100,0]);
    this.MovimientoJ([-200,-100,100]);
    for theta = linspace(-pi/6,pi/6,3)
        for alpha = linspace(-pi/4,0,2)

            x = this.radio_esfera*sin(theta)*cos(alpha);
            x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
            z = this.radio_esfera*cos(theta);
            z2 = this.herramienta.entrada_esfera*cos(theta);

            RotY = this.RotacionY(theta);

            if (theta == -pi/6)
                y = this.radio_esfera*sin(theta)*sin(-alpha);
                y2 = this.herramienta.entrada_esfera*sin(theta)*sin(-
alpha);
                RotZ = this.RotacionZ(alpha);
            else
                y = this.radio_esfera*sin(theta)*sin(alpha);
                y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
                RotZ = this.RotacionZ(-alpha);
            end

            Rotacion = RotZ*RotY;

            this.MovimientoL([x,y,z],Rotacion);
            %this.GuardarPunto;
            this.MovimientoL([x2,y2,z2],Rotacion);
            %this.GuardarPunto;
            this.MovimientoL([x,y,z],Rotacion);
            %this.GuardarPunto;
            %this.MovimientoL([x+12,y+12,z+3],Rotacion);
            if(theta == 0)
                break
            end
        end
    end
    this.MovimientoJ([200,-100,100]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroMesaSencilla(this)
    this.SetCilindro;
    this.MovimientoAbsoluto([-60,-10,0,0,90,0]);
    this.MovimientoJ([0,0,300]);
    for radio = linspace(100,0,3)
        for angulo = linspace(0,2*pi,7)
            if(angulo == 2*pi)
                break;
            end

            x = radio*cos(angulo);
            y = radio*sin(angulo);
            z = this.altura_cilindro;
            z2 = this.altura_cilindro -
this.herramienta.entrada_cilindro_base;

```

```

        this.MovimientoJ([x,y,z]);
        this.MovimientoL([x,y,z2]);
        this.MovimientoL([x,y,z]);
        if (radio == 0)
            break;
        end
    end
    if (radio == 0)
        break;
    end
end
this.MovimientoJ([0,0,350]);
this.MovimientoAbsoluto([-60,-10,0,0,90,0]);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroMesaRotando(this)
    this.SetCilindro;
    this.MovimientoAbsoluto([-120,-10,0,0,90,0]);
    angulo_1 = -pi/2;
    RotY = this.RotacionY(angulo_1);
    this.MovimientoJ([-150,0,100],RotY);
    for altura = linspace(50,250,5)
        Trans = this.Translacion([-altura;0;0]);

        RotTran = RotY*Trans;

        x = -this.radio_cilindro;
        x2 = -this.herramienta.entrada_cilindro;
        y = 0;
        z = 0;

        this.MovimientoJ([x,y,z],RotTran);
        this.MovimientoL([x2,y,z],RotTran);
        this.MovimientoL([x,y,z],RotTran);
    end
    this.MovimientoJ([x,y,z+50],RotTran);
    this.MovimientoAbsoluto([-120,-10,0,0,90,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroParedDerechaSencilla(this)
    this.SetCilindroParedDerecha;
    this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
    this.MovimientoAbsoluto([-90,-10,50,0,-90,0]);
    this.MovimientoJ([-150,0,150]);
    for altura = linspace(50,250,5)
        Trans = this.Translacion([altura;0;0]);

        x = 0;
        y = 0;
        z = this.radio_cilindro;
        z2= this.herramienta.entrada_cilindro;

        this.MovimientoJ([x,y,z],Trans);
        this.MovimientoL([x,y,z2],Trans);
        this.MovimientoL([x,y,z],Trans);
    end
    this.MovimientoJ([-150,0,150]);
    this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroParedDerechaRotando(this)
    this.SetCilindroParedDerecha;
    this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
    this.MovimientoAbsoluto([-90,-10,0,0,40,0]);
    for angulo = linspace(pi/6,-pi/6,3)
        for altura = linspace(50,250,5)

            Trans = this.Translacion([altura;0;0]);

```

```

        RotX = this.RotacionX(angulo);
        RotTrans = RotX*Trans;

        x = 0;
        y = this.radio_cilindro*sin(angulo);
        y2 = this.herramienta.entrada_cilindro*sin(angulo);
        z = this.radio_cilindro*cos(angulo);
        z2= this.herramienta.entrada_cilindro*cos(angulo);

        this.MovimientoL([x,y,z],RotTrans);
        this.MovimientoL([x,y2,z2],RotTrans);
        this.MovimientoL([x,y,z],RotTrans);
    end
    if(angulo == pi/6)
        this.MovimientoAbsoluto([-90,-10,50,0,-50,0]);
    elseif(angulo == 0)
        this.MovimientoL([x+50,y,z+50],RotTrans);
    else
        break;
    end
    end
    this.MovimientoJ([0,0,200],RotTrans);
    this.MovimientoAbsoluto([-60,-10,0,0,90,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroTechoSencilla(this)
    this.SetCilindroTecho;
    this.MovimientoAbsoluto([40,-30,40,0,-100,0]);
    this.MovimientoJ([0,-50,150]);
    for altura = linspace(50,250,5)
        Trans = this.Translacion([0;altura;0]);

        x = 0;
        y = 0;
        z = this.radio_cilindro;
        z2= this.herramienta.entrada_cilindro;

        this.MovimientoJ([x,y,z],Trans);
        this.MovimientoL([x,y,z2],Trans);
        this.MovimientoL([x,y,z],Trans);
        this.MovimientoL([x,y,z+7],Trans);
    end
    this.MovimientoJ([0,0,150],Trans);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroTechoRotando(this)
    switch this.herramienta.GetEntradaCilindro
        case 15
            t = linspace(150,200,2);
        otherwise
            t = linspace(50,200,4);
    end
    this.SetCilindroTecho;
    this.MovimientoAbsoluto([-40,-30,40,0,-100,0]);
    this.MovimientoL([-125,0,100]);
    for angulo = linspace(-pi/6,pi/6,3)
        for altura = t
            Trans = this.Translacion([0;altura;0]);

            RotY = this.RotacionY(angulo);

            y = 0;
            x = this.radio_cilindro*sin(angulo);
            x2 = this.herramienta.entrada_cilindro*sin(angulo);
            z = this.radio_cilindro*cos(angulo);
            z2= this.herramienta.entrada_cilindro*cos(angulo);

            RotTran = RotY*Trans;

```

```
        this.MovimientoL([x,y,z],RotTran);
        this.MovimientoL([x2,y,z2],RotTran);
        this.MovimientoL([x,y,z],RotTran);
    end
    this.MovimientoL([x+10,y,z+7.5],RotTran)
end
this.MovimientoJ([0,0,150],RotTran);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
end
end
```

Clase Robot_Simulacion.m

```
classdef Robot_Simulacion < handle
    %UNTITLED3 Summary of this class goes here
    % Detailed explanation goes here

    properties
        robot_simulado;
        h = Hmat;
        trayectoria;
        i = 1;

        trayectoria_guardada;
        cont = 1;
        articulacion_guardada;
        arts = 1;

        rotacion = [0,0,pi];
        deg = pi/180;
        radio_esfera = 160 * 1e-3;
        radio_cilindro = 140 * 1e-3;
        altura_cilindro = 307 * 1e-3;
        herramienta;

        wBase;
        wBarras;
        wEsfera;
        wEsfera_Pared_Derecha;
        wEsfera_Techo;
        wCilindro;
        wCilindro_Pared_Derecha;
        wCilindro_Techo;
    end

    methods
        function this = Robot_Simulacion mdl
            %UNTITLED3 Construct an instance of this class
            % Detailed explanation goes here
            load('./irb120_urdf/Robot_2.mat');
            this.robot_simulado = Kin(robot, mdl);
            set_param('Robot_q/tool1/Visual','ExtGeomFileName','');

            this.wBase = this.h.tQ([[0,0,0],[1,0,0,0]]);
            this.wBarras =
            this.h.tQ([[0.250541304602,0.250,0],[0.707106781,0.707106781,0,0]]);
            this.wEsfera = this.h.tQ([[0.00541304602,-0.350,0],[1,0,0,0]]);
            this.wEsfera_Pared_Derecha = this.h.tQ([[0,-
            0.750,0.350],[0,0,0.707106781,0.707106781]]);
            this.wEsfera_Techo =
            this.h.tQ([[0.400541304602,0,0.950],[0,0.707106781,0.707106781,0]]);
            this.wCilindro = this.h.tQ([[0.200,-0.300,0],[1,0,0,0]]);
            this.wCilindro_Pared_Derecha = this.h.tQ([[0,-
            0.750,0.350],[0,0,0.707106781,0.707106781]]);
            this.wCilindro_Techo =
            this.h.tQ([[0.350,0,0.950],[0,0.707106781,0.707106781,0]]);
        end

        function MovimientoAbsoluto(this, ang_articulacion)
            %UNTITLED2 Construct an instance of this class
            % Detailed explanation goes here
            this.robot_simulado.MoveAbsJ(ang_articulacion*this.deg,30);
        end
        function MovimientoManual(this,punto,rotacion,wobj,objeto)
            p = punto / 1000;
            rotacion(3) = 180 + rotacion(3);
            r = rotacion * this.deg;
            switch wobj
                case 'Base'
```

```

        w = this.wBase;
    case 'Mesa'
        switch objeto
            case 'Esfera'
                w = this.wEsfera;
                this.MostrarEsfera;
                this.MostrarEsferaMesa;
            case 'Cilindro'
                w = this.wCilindro;
                this.MostrarCilindro;
                this.MostrarCilindroMesa;
            end
        case 'Pared derecha'
            switch objeto
                case 'Esfera'
                    w = this.wEsfera_Pared_Derecha;
                    this.MostrarEsfera;
                    this.MostrarEsferaParedDerecha;
                case 'Cilindro'
                    w = this.wCilindro_Pared_Derecha;
                    this.MostrarCilindro;
                    this.MostrarCilindroParedDerecha;
                end
            case 'Techo'
                switch objeto
                    case 'Esfera'
                        w = this.wEsfera_Techo;
                        this.MostrarEsfera;
                        this.MostrarEsferaTecho;
                    case 'Cilindro'
                        w = this.wCilindro_Techo;
                        this.MostrarCilindro;
                        this.MostrarCilindroTecho;
                    end
                end
            end
        this.robot_simulado.MoveJ([p,r],w);
    end
    function GuardarPunto(this,punto,rotacion)
        punto = punto * 1e-3;
        rotacion(3) = 180 + rotacion(3);
        rotacion = rotacion * this.deg;

        this.trayectoria_guardada(this.cont,:) = [punto,rotacion];
        this.cont = this.cont + 1;
    end
    function BorrarSecuencia(this)
        this.cont = 1;
        this.trayectoria_guardada = [];
    end
    function RepetirSecuencia(this,wobj,objeto)
        switch wobj
            case 'Base'
                w = this.wBase;
            case 'Mesa'
                switch objeto
                    case 'Esfera'
                        w = this.wEsfera;
                        this.MostrarEsfera;
                        this.MostrarEsferaMesa;
                    case 'Cilindro'
                        w = this.wCilindro;
                        this.MostrarCilindro;
                        this.MostrarCilindroMesa;
                    end
                case 'Pared derecha'
                    switch objeto
                        case 'Esfera'
                            w = this.wEsfera_Pared_Derecha;

```



```

        this.MostrarEsfera;
        this.MostrarEsferaParedDerecha;
    case 'Cilindro'
        w = this.wCilindro_Pared_Derecha;
        this.MostrarCilindro;
        this.MostrarCilindroParedDerecha;
    end
case 'Techo'
    switch objeto
        case 'Esfera'
            w = this.wEsfera_Techo;
            this.MostrarEsfera;
            this.MostrarEsferaTecho;
        case 'Cilindro'
            w = this.wCilindro_Techo;
            this.MostrarCilindro;
            this.MostrarCilindroTecho;
        end
    end
end
this.robot_simulado.MoveJ(this.trayectoria_guardada,w);

end
function GuardarArticulaciones(this,articulaciones)
    articulaciones = articulaciones * this.deg;
    this.articulacion_guardada(this.arts,:) = articulaciones;
    this.arts = this.arts + 1;
end
function BorrarArticulaciones(this)
    this.articulacion_guardada = [];
    this.arts = 1;
end
function RepetirArticulaciones(this)
    this.robot_simulado.MoveAbsJ(this.articulacion_guardada);
end

function TomarBarra(this,barra)
    this.rotacion = [0,0,pi];
    switch barra
        case 'Roja'
            this.herramienta =
Herramienta([0.050,0.100,0.120],[0.050,0.100,0.040],0.070,0.040,0.015,0.120,10
00);
        case 'Verde'
            this.herramienta =
Herramienta([0.150,0.100,0.120],[0.150,0.100,0.040],0.050,0.070,0.040,0.080,10
00);
        case 'Naranja'
            this.herramienta =
Herramienta([0.250,0.100,0.120],[0.250,0.100,0.040],0.170,0.030,0.010,0.160,10
00);
    end
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
    this.trayectoria = [this.herramienta.GetPunto_2,this.rotacion;...
        this.herramienta.GetPunto_3,this.rotacion;...
        this.herramienta.GetPunto_4,this.rotacion;...
        this.herramienta.GetPunto_5,this.rotacion];
    this.robot_simulado.MoveJ(this.trayectoria,this.wBarras);
    pause(4);
    switch barra
        case 'Roja'

set_param('Robot_q/tool1/Visual','ExtGeomFileName','C:\Victor\Universidad\Mast
er Electronica Industrial Automatica\TFM\Piezas\STLs\Barra_20.stl');
        this.robot_simulado.Tool('tool2','tool1',
this.h.t([0,0,0.198]), ...
        path, this.h.t([0,0,-0.198]));
        case 'Verde'

```

```

set_param('Robot_q/tool1/Visual','ExtGeomFileName','C:\Victor\Universidad\Master
Electronica Industrial Automatica\TFM\Piezas\STLs\Barra_15.stl');
    this.robot_simulado.Tool('tool2','tool1',
this.h.t([0,0,0.148]), ...
    path, this.h.t([0,0,-0.148]));
    case 'Naranja'

set_param('Robot_q/tool1/Visual','ExtGeomFileName','C:\Victor\Universidad\Master
Electronica Industrial Automatica\TFM\Piezas\STLs\Barra_30.stl');
    this.robot_simulado.Tool('tool2','tool1',
this.h.t([0,0,0.298]), ...
    path, this.h.t([0,0,-0.298]));
    end
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function DejarBarra(this)
this.robot_simulado.DelBody('tool2');
this.rotacion = [0,0,pi];
this.MovimientoAbsoluto([0,0,0,0,90,0]);
this.trayectoria = [this.herramienta.GetPunto_5,this.rotacion;...
    this.herramienta.GetPunto_4,this.rotacion;...
    this.herramienta.GetPunto_3,this.rotacion;...
    this.herramienta.GetPunto_2,this.rotacion;...
    this.herramienta.GetPunto_6,this.rotacion];
this.robot_simulado.MoveJ(this.trayectoria,this.wBarras);
pause(4);
set_param('Robot_q/tool1/Visual','ExtGeomFileName','');
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaMesaSencilla(this)
this.MostrarEsfera;
this.MostrarEsferaMesa;
this.i = 1;
switch this.herramienta.GetEntradaEsfera
    case 0.030
        this.trayectoria = zeros(12,6);
        for theta = linspace(pi/6,-pi/6,3)

            x = this.radio_esfera*sin(theta);
            x2 = this.herramienta.entrada_esfera*sin(theta);
            y = 0;
            z = this.radio_esfera*cos(theta);
            z2 = this.herramienta.entrada_esfera*cos(theta);

            this.rotacion(2) = theta;

            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
            this.i = this.i+1;
            this.trayectoria(this.i,:) = [x2,y,z2,this.rotacion];
            this.i = this.i+1;
            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
            this.i = this.i+1;
            this.trayectoria(this.i,:) = [x-
0.015,y+0.03,z+0.050,this.rotacion];
            this.i = this.i+1;
        end
    otherwise
        this.trayectoria = zeros(20,6);
        for theta = linspace(pi/3,-pi/3,5)

            x = this.radio_esfera*sin(theta);
            x2 = this.herramienta.entrada_esfera*sin(theta);
            y = 0;
            z = this.radio_esfera*cos(theta);
            z2 = this.herramienta.entrada_esfera*cos(theta);

            this.rotacion(2) =theta;

```

```

        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i+1;
        this.trayectoria(this.i,:) = [x2,y,z2,this.rotacion];
        this.i = this.i+1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i+1;
        this.trayectoria(this.i,:) =
[x,y,(z+0.008),this.rotacion];
        this.i = this.i+1;
    end
end

this.MovimientoAbsoluto([-90,0,0,0,90,0]);
this.robot_simulado.MoveJ(this.trayectoria,this.wEsfera);
pause(7);
this.MovimientoAbsoluto([-90,0,0,0,90,0]);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaMesaRotando(this)
this.MostrarEsfera;
this.MostrarEsferaMesa;
this.i = 1;
this.rotacion = [0,0,pi];
    switch this.herramienta.GetEntradaEsfera
    case 0.03
        this.trayectoria = zeros(20,6);
        for theta = linspace(pi/6,-pi/6,3)
            for alpha = linspace(pi/4,0,2)
                x = this.radio_esfera*sin(theta)*cos(alpha);
                x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
                z = this.radio_esfera*cos(theta);
                z2 =
this.herramienta.entrada_esfera*cos(theta);

                RotY = theta;

                if (theta == -pi/6)
                    y = this.radio_esfera*sin(theta)*sin(-
alpha);
                    y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(-alpha);
                    RotZ = -alpha;
                else
                    y =
this.radio_esfera*sin(theta)*sin(alpha);
                    y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
                    RotZ = alpha;
                end

                this.rotacion(1) = RotZ;
                this.rotacion(2) = RotY;

                this.trayectoria(this.i,:) =
[x,y,z,this.rotacion];
                this.i = this.i+1;
                this.trayectoria(this.i,:) =
[x2,y2,z2,this.rotacion];
                this.i = this.i+1;
                this.trayectoria(this.i,:) =
[x,y,z,this.rotacion];
                this.i = this.i+1;
                this.trayectoria(this.i,:) =
[x,y,(z+0.08),this.rotacion];
                this.i = this.i+1;
            end
        end
    end
end

```

```

        if(theta == 0)
            break
        end
    end
end
end
otherwise
    this.trayectoria = zeros(28,6);
    for theta = linspace(pi/6,-pi/6,3)
        for alpha = linspace(-pi/4,pi/4,3)

            x = this.radio_esfera*sin(theta)*cos(alpha);
            x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
            y = this.radio_esfera*sin(theta)*sin(alpha);
            y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
            z = this.radio_esfera*cos(theta);
            z2 =
this.herramienta.entrada_esfera*cos(theta);

            RotY = theta;
            RotZ = alpha;

            this.rotacion = [RotZ, RotY, pi];
            this.trayectoria(this.i, :) =
[x, y, z, this.rotacion];

            this.i = this.i+1;
            this.trayectoria(this.i, :) =
[x2, y2, z2, this.rotacion];

            this.i = this.i+1;
            this.trayectoria(this.i, :) =
[x, y, z, this.rotacion];

            this.i = this.i+1;
            this.trayectoria(this.i, :) =
[x, y, z+0.09, this.rotacion];

            this.i = this.i+1;
            if(theta == 0)
                break
            end
        end
    end
end
end
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
    this.MovimientoAbsoluto([-90,0,0,0,90,0]);
    this.robot_simulado.MoveJ(this.trayectoria,this.wEsfera);
    pause(12);
    this.MovimientoAbsoluto([-90,0,0,0,90,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaParedDerechaSencilla(this)
    this.MostrarEsfera;
    this.MostrarEsferaParedDerecha;
    t = zeros(8,6);
    e = 1;
    t2 = zeros(4,6);
    this.rotacion = [0,0,pi - pi/6];
    t2(1, :) =
[0,this.radio_esfera*sin(pi/6),this.radio_esfera*cos(pi/6),this.rotacion];
    t2(2, :) =
[0,this.herramienta.entrada_esfera*sin(pi/6),this.herramienta.entrada_esfera*c
os(pi/6),this.rotacion];
    t2(3, :) =
[0,this.radio_esfera*sin(pi/6),this.radio_esfera*cos(pi/6),this.rotacion];
    t2(4, :) =
[0,this.radio_esfera*sin(pi/6),this.radio_esfera*cos(pi/6)+0.035,this.rotacion
];

    this.trayectoria = zeros(12,6);

```

```

this.i = 1;
this.rotacion = [0,0,pi];
for theta = linspace(pi/6,-pi/6,3)

    x = this.radio_esfera*sin(theta);
    x2 = this.herramienta.entrada_esfera*sin(theta);
    y = 0;
    z = this.radio_esfera*cos(theta);
    z2 = this.herramienta.entrada_esfera*cos(theta);

    this.rotacion(2) = theta;

    this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
    this.i = this.i + 1;
    this.trayectoria(this.i,:) = [x2,y,z2,this.rotacion];
    this.i = this.i + 1;
    this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
    this.i = this.i + 1;
    this.trayectoria(this.i,:) = [x-
0.11,y+0.06,z+0.075,this.rotacion];
    this.i = this.i + 1;
end
for theta = linspace(0,-pi/6,2)

    this.rotacion = [0,0,pi - theta];

    y = this.radio_esfera*sin(theta);
    y2 = this.herramienta.entrada_esfera*sin(theta);
    x = 0;
    z = this.radio_esfera*cos(theta);
    z2 = this.herramienta.entrada_esfera*cos(theta);

    t(e,:) = [x,y,z,this.rotacion];
    e = e + 1;
    t(e,:) = [x,y2,z2,this.rotacion];
    e = e + 1;
    t(e,:) = [x,y,z,this.rotacion];
    e = e + 1;
    t(e,:) = [x,y,z+0.035,this.rotacion];
    e = e + 1;
end
this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
this.robot_simulado.MoveJ([0.210,0,0.150]);

this.robot_simulado.MoveJ(this.trayectoria,this.wEsfera_Pared_Derecha);
pause(8);
this.robot_simulado.MoveJ(t2,this.wEsfera_Pared_Derecha);
pause(6);
this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
this.robot_simulado.MoveJ(t,this.wEsfera_Pared_Derecha);
pause(8)
this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
this.MovimientoAbsoluto([-90,-10,0,0,30,0]);

this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaParedDerechaRotando(this)
this.MostrarEsfera;
this.MostrarEsferaParedDerecha;
this.trayectoria = zeros(28,6);
this.i = 1;
this.rotacion = [0,0,pi];
for theta = linspace(pi/6,-pi/6,3)
    for alpha = linspace(pi/4,-pi/4,3)

        x = this.radio_esfera*sin(theta)*cos(alpha);

```

```

        x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
        y = this.radio_esfera*sin(theta)*sin(alpha);
        y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
        z = this.radio_esfera*cos(theta);
        z2 = this.herramienta.entrada_esfera*cos(theta);

        this.rotacion(2) = theta;
        this.rotacion(1) = alpha;

        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x2,y2,z2,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x-
0.04,y+0.03,z+0.075,this.rotacion];
        this.i = this.i + 1;
        if(theta == 0)
            break
        end
    end
end
    this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
    this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);

this.robot_simulado.MoveJ([0.210,0,0.175],this.wEsfera_Pared_Derecha);

this.robot_simulado.MoveJ(this.trayectoria,this.wEsfera_Pared_Derecha);
    pause(11);
    this.MovimientoAbsoluto([-90,-30,50,0,-20,0]);
    this.MovimientoAbsoluto([0,-10,50,0,-40,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function EsferaTechoSencilla(this)
    this.MostrarEsfera;
    this.MostrarEsferaTecho;
    this.trayectoria = zeros(12,6);
    this.i = 1;
    for theta = linspace(-pi/6,pi/6,3)

        x = this.radio_esfera*sin(theta);
        x2 = this.herramienta.entrada_esfera*sin(theta);
        y = 0;
        z = this.radio_esfera*cos(theta);
        z2 = this.herramienta.entrada_esfera*cos(theta);

        this.rotacion = [0,theta,pi];

        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x2,y,z2,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) =
[x+0.025,y+0.015,z+0.00595,this.rotacion];
        this.i = this.i + 1;
    end
    this.MovimientoAbsoluto([-30,-40,50,0,-100,0]);
    this.robot_simulado.MoveJ([-0.250,-
0.100,0.100],this.wEsfera_Techo);
    this.robot_simulado.MoveJ(this.trayectoria,this.wEsfera_Techo);
    pause(5);
    this.robot_simulado.MoveJ([0.250,-
0.100,0.100],this.wEsfera_Techo);

```

```

        this.MovimientoAbsoluto([0,0,0,0,90,0]);
    end
    function EsferaTechoRotando(this)
        this.MostrarEsfera;
        this.MostrarEsferaTecho;
        this.trayectoria = zeros(20,6);
        this.i = 1;
        for theta = linspace(-pi/6,pi/6,3)
            for alpha = linspace(-pi/4,0,2)

                x = this.radio_esfera*sin(theta)*cos(alpha);
                x2 =
this.herramienta.entrada_esfera*sin(theta)*cos(alpha);
                z = this.radio_esfera*cos(theta);
                z2 = this.herramienta.entrada_esfera*cos(theta);

                this.rotacion = [0,theta,pi];

                if (theta == -pi/6)
                    y = this.radio_esfera*sin(theta)*sin(-alpha);
                    y2 = this.herramienta.entrada_esfera*sin(theta)*sin(-
alpha);
                    this.rotacion(1) = -alpha;
                else
                    y = this.radio_esfera*sin(theta)*sin(alpha);
                    y2 =
this.herramienta.entrada_esfera*sin(theta)*sin(alpha);
                    this.rotacion(1) = alpha;
                end

                this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
                this.i = this.i + 1;
                this.trayectoria(this.i,:) = [x2,y2,z2,this.rotacion];
                this.i = this.i + 1;
                this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
                this.i = this.i + 1;
                this.trayectoria(this.i,:) =
[x+0.015,y+0.015,z+0.005,this.rotacion];
                this.i = this.i + 1;
                if(theta == 0)
                    break
                end
            end
        end
        this.MovimientoAbsoluto([-30,-40,50,0,-100,0]);
        this.robot_simulado.MoveJ([-0.250,-
0.100,0.100],this.wEsfera_Techo);
        this.robot_simulado.MoveJ(this.trayectoria,this.wEsfera_Techo);
        pause(5);
        this.robot_simulado.MoveJ([0.250,-
0.100,0.100],this.wEsfera_Techo);
        this.MovimientoAbsoluto([0,0,0,0,90,0]);
    end
    function CilindroMesaSencilla(this)
        this.MostrarCilindro;
        this.MostrarCilindroMesa;
        this.trayectoria = zeros(39,6);
        this.i = 1;
        for radio = linspace(0.100,0,3)
            for angulo = linspace(0,2*pi,7)
                if(angulo == 2*pi)
                    break;
                end

                x = radio*cos(angulo);
                y = radio*sin(angulo);
                z = this.altura_cilindro;
            end
        end
    end
end

```

```

        z2 = this.altura_cilindro -
this.herramienta.entrada_cilindro_base;

        this.rotacion = [0,0,pi];

        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z2,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        if (radio == 0)
            break;
        end
    end
end
this.MovimientoAbsoluto([-60,-10,0,0,90,0]);
this.robot_simulado.MoveJ(this.trayectoria,this.wCilindro);
pause(13);
this.robot_simulado.MoveJ([0,0,0.350],this.wCilindro);
this.MovimientoAbsoluto([-60,-10,0,0,90,0]);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroMesaRotando(this)
    this.MostrarCilindro;
    this.MostrarCilindroMesa;
    this.trayectoria = zeros(15,6);
    this.i = 1;
    this.rotacion = [0,-pi/2,pi];
    for altura = linspace(0.05,0.250,5)

        x = -this.radio_cilindro;
        x2 = -this.herramienta.entrada_cilindro;
        y = 0;
        z = altura;

        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x2,y,z,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
    end
    this.MovimientoAbsoluto([-120,-10,0,0,90,0]);
    this.robot_simulado.MoveJ([-
0.150,0,0.100,this.rotacion],this.wCilindro);
    this.robot_simulado.MoveJ(this.trayectoria,this.wCilindro);
    pause(6);

    this.robot_simulado.MoveJ([x,y,z+0.050,this.rotacion],this.wCilindro);
    this.MovimientoAbsoluto([-120,-10,0,0,90,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroParedDerechaSencilla(this)
    this.MostrarCilindro;
    this.MostrarCilindroParedDerecha;
    this.trayectoria = zeros(15,6);
    this.i = 1;
    this.rotacion = [0,0,pi];
    for altura = linspace(-0.10,0.10,5)

        x = -altura;
        y = 0;
        z = this.radio_cilindro;
        z2 = this.herramienta.entrada_cilindro;

        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
    end
end

```



```

        this.trayectoria(this.i,:) = [x,y,z2,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
    end
    this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
    this.MovimientoAbsoluto([-90,-10,50,0,-90,0]);
    this.robot_simulado.MoveJ([-
0.150,0,0.150],this.wCilindro_Pared_Derecha);

this.robot_simulado.MoveJ(this.trayectoria,this.wCilindro_Pared_Derecha);
    pause(6);
    this.robot_simulado.MoveJ([-
0.150,0,0.150],this.wCilindro_Pared_Derecha);
    this.MovimientoAbsoluto([-90,-10,0,0,90,0]);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroParedDerechaRotando(this)
    this.MostrarCilindro;
    this.MostrarCilindroParedDerecha;
    t = zeros(15,6);
    e = 1;
    this.trayectoria = zeros(30,6);
    this.i = 1;
    for altura = linspace(-0.10,0.10,5)

        this.rotacion = [0,0,pi - pi/6];

        x = -altura;
        y = this.radio_cilindro*sin(pi/6);
        y2 = this.herramienta.entrada_cilindro*sin(pi/6);
        z = this.radio_cilindro*cos(pi/6);
        z2= this.herramienta.entrada_cilindro*cos(pi/6);

        t(e,:) = [x,y,z,this.rotacion];
        e = e + 1;
        t(e,:) = [x,y2,z2,this.rotacion];
        e = e + 1;
        t(e,:) = [x,y,z,this.rotacion];
        e = e + 1;
    end
    for angulo = linspace(0,-pi/6,2)
        for altura = linspace(-0.10,0.10,5)

            this.rotacion = [0,0,pi - angulo];

            x = -altura;
            y = this.radio_cilindro*sin(angulo);
            y2 = this.herramienta.entrada_cilindro*sin(angulo);
            z = this.radio_cilindro*cos(angulo);
            z2= this.herramienta.entrada_cilindro*cos(angulo);

            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
            this.i = this.i + 1;
            this.trayectoria(this.i,:) = [x,y2,z2,this.rotacion];
            this.i = this.i + 1;
            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
            this.i = this.i + 1;
        end
        if(angulo == 0)
            this.trayectoria(this.i,:) =
[x+0.05,y,z+0.05,this.rotacion];
            this.i = this.i + 1;
        else
            break;
        end
    end
end
this.MovimientoAbsoluto([-90,-10,0,0,90,0]);

```

```

        this.MovimientoAbsoluto([-90,-10,0,0,40,0]);
        this.robot_simulado.MoveJ(t,this.wCilindro_Pared_Derecha);
        pause(6);
        this.MovimientoAbsoluto([-90,-10,50,0,-50,0]);

this.robot_simulado.MoveJ(this.trayectoria,this.wCilindro_Pared_Derecha);
    pause(11);

this.robot_simulado.MoveJ([0,0,0.200,this.rotacion],this.wCilindro_Pared_Derecha);

        this.MovimientoAbsoluto([-60,-10,0,0,90,0]);
        this.MovimientoAbsoluto([0,0,0,0,90,0]);
    end
    function CilindroTechoSencilla(this)
        this.MostrarCilindro;
        this.MostrarCilindroTecho;
        this.trayectoria = zeros(20,6);
        this.i = 1;
        this.rotacion = [0,0,pi];
        for altura = linspace(-0.1,0.10,5)

            x = 0;
            y = altura;
            z = this.radio_cilindro;
            z2= this.herramienta.entrada_cilindro;

            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
            this.i = this.i + 1;
            this.trayectoria(this.i,:) = [x,y,z2,this.rotacion];
            this.i = this.i + 1;
            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
            this.i = this.i + 1;
            this.trayectoria(this.i,:) = [x,y,z+0.007,this.rotacion];
            this.i = this.i + 1;
        end
        this.MovimientoAbsoluto([40,-30,40,0,-100,0]);
        this.robot_simulado.MoveJ([0,-0.050,0.150],this.wCilindro_Techo);
        this.robot_simulado.MoveJ(this.trayectoria, this.wCilindro_Techo);
        pause(6);

this.robot_simulado.MoveJ([0,0,0.150,this.rotacion],this.wCilindro_Techo);
    this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
function CilindroTechoRotando(this)
    this.MostrarCilindro;
    this.MostrarCilindroTecho;
    this.trayectoria = zeros(20,6);
    this.i = 1;
    switch this.herramienta.GetEntradaCilindro
        case 0.015
            t = linspace(0.0,0.05,2);
            this.trayectoria = zeros(18,6);
        otherwise
            t = linspace(-0.100,0.050,4);
            this.trayectoria = zeros(36,6);
    end
    for angulo = linspace(-pi/6,pi/6,3)
        for altura = t

            this.rotacion = [0,angulo,pi];

            y = altura;
            x = this.radio_cilindro*sin(angulo);
            x2 = this.herramienta.entrada_cilindro*sin(angulo);
            z = this.radio_cilindro*cos(angulo);
            z2= this.herramienta.entrada_cilindro*cos(angulo);

            this.trayectoria(this.i,:) = [x,y,z,this.rotacion];

```

```

        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x2,y,z2,this.rotacion];
        this.i = this.i + 1;
        this.trayectoria(this.i,:) = [x,y,z,this.rotacion];
        this.i = this.i + 1;
    end
    this.trayectoria(this.i,:) =
[x+0.0015,y,z+0.0015,this.rotacion];
    this.i = this.i + 1;
end
this.MovimientoAbsoluto([-40,-30,40,0,-100,0]);
this.robot_simulado.MoveJ([-0.1,0,0.050],this.wCilindro_Techo);
this.robot_simulado.MoveJ(this.trayectoria,this.wCilindro_Techo);
pause(13);
this.robot_simulado.MoveJ([0,0,0.150],this.wCilindro_Techo);
this.MovimientoAbsoluto([0,0,0,0,90,0]);
end
end
methods (Static)
    function MostrarEsfera()

set_param('Robot_q/base/Objeto','ExtGeomFileName','C:\Victor\Universidad\Maste
r Electronica Industrial Automatica\TFM\Piezas\STLs\Esfera.stl');
    end
    function MostrarCilindro()

set_param('Robot_q/base/Objeto','ExtGeomFileName','C:\Victor\Universidad\Maste
r Electronica Industrial Automatica\TFM\Piezas\STLs\Cilindro.stl');
    end
    function MostrarEsferaMesa()

set_param('Robot_q/base/ObjetoOriginTransform1','TranslationCartesianOffset',
[0,-350,0]);

set_param('Robot_q/base/ObjetoOriginTransform1','RotationSequenceAngles','[0,0
,0]');
    end
    function MostrarEsferaParedDerecha()

set_param('Robot_q/base/ObjetoOriginTransform1','TranslationCartesianOffset',
[0,-750,350]);

set_param('Robot_q/base/ObjetoOriginTransform1','RotationSequenceAngles','[-
90,0,0]');
    end
    function MostrarEsferaTecho()

set_param('Robot_q/base/ObjetoOriginTransform1','TranslationCartesianOffset',
[400,0,950]);

set_param('Robot_q/base/ObjetoOriginTransform1','RotationSequenceAngles','[180
,0,180]');
    end
    function MostrarCilindroMesa()

set_param('Robot_q/base/ObjetoOriginTransform1','TranslationCartesianOffset',
[200,-300,150]);

set_param('Robot_q/base/ObjetoOriginTransform1','RotationSequenceAngles','[0,0
,0]');
    end
    function MostrarCilindroParedDerecha()

set_param('Robot_q/base/ObjetoOriginTransform1','TranslationCartesianOffset',
[0,-750,350]);

set_param('Robot_q/base/ObjetoOriginTransform1','RotationSequenceAngles','[-
90,0,-90]');

```

```

        end
        function MostrarCilindroTecho()

set_param('Robot_q/base/ObjetoOriginTransform1','TranslationCartesianOffset',[350,0,950]);

set_param('Robot_q/base/ObjetoOriginTransform1','RotationSequenceAngles',[90,0,90]);

        end
    end
end

```

Clase Herramienta.m

```

classdef Herramienta
    %UNTITLED Summary of this class goes here
    % Detailed explanation goes here

    properties
        punto_2;
        punto_3;
        punto_4;
        punto_5;
        punto_6;
        entrada_esfera;
        entrada_cilindro;
        entrada_cilindro_base;
    end

    methods
        function this =
Herramienta(p2,p3,extra,in_esfera,in_cilindro,in_cilindro_base,unidades)
            %UNTITLED Construct an instance of this class
            % Detailed explanation goes here
            this.punto_2 = p2;
            this.punto_3 = p3;
            this.punto_4 = this.punto_2;
            this.punto_4(3) = this.punto_4(3) + extra;
            this.punto_5 = this.punto_4;
            this.punto_5(2) = this.punto_5(2) + 150 / unidades;
            this.punto_6 = this.punto_2;
            this.punto_6(2) = this.punto_6(2) + 150 / unidades;
            this.entrada_esfera = in_esfera;
            this.entrada_cilindro = in_cilindro;
            this.entrada_cilindro_base = in_cilindro_base;
        end
        function obj = GetPunto_2(this)
            %UNTITLED2 Construct an instance of this class
            % Detailed explanation goes here
            obj = this.punto_2;
        end
        function obj = GetPunto_3(this)
            %UNTITLED2 Construct an instance of this class
            % Detailed explanation goes here
            obj = this.punto_3;
        end
        function obj = GetPunto_4(this)
            %UNTITLED2 Construct an instance of this class
            % Detailed explanation goes here
            obj = this.punto_4;
        end
        function obj = GetPunto_5(this)
            %UNTITLED2 Construct an instance of this class
            % Detailed explanation goes here
            obj = this.punto_5;
        end
    end
end

```

```

function obj = GetPunto_6(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    obj = this.punto_6;
end
function obj = GetEntradaEsfera(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    obj = this.entrada_esfera;
end
function obj = GetEntradaCilindro(this)
    %UNTITLED2 Construct an instance of this class
    % Detailed explanation goes here
    obj = this.entrada_cilindro;
end
function MostrarHerramienta(this)
    disp(this.punto_2);
    disp(this.punto_3);
    disp(this.punto_4);
    disp(this.punto_5);
    disp(this.punto_6);
    disp(this.entrada_esfera);
    disp(this.entrada_cilindro);
    disp(this.entrada_cilindro_base);
end
end
end

```

Clase tcp_ABB.m

```
classdef tcpABB < handle

    properties
        t
    end % properties

    % Ordenes de RAPID
    % ["MoveAbsJ","MoveJ","MoveL",
    % "CJointT","CRobT","CTool","CWobj",,"CSpeed","CZone",
    % "CalcJointT","CalcRobT","TestJoint",
    % "OpenTool","CloseTool","Reset","Stop",""];

    % Ordenes de asignación en RAPID
    % ["join","rob.pos","rob.orient","tool.pos","tool.orient",
    % "wobj.pos","wobj.orient","speed","zone"];

    methods

        function this= tcpABB(ip, port)
            this.t= tcpclient(ip, port);
            msg= this.Read();
            %fprintf([msg,'\n']);
        end

        function msg= Read(this)
            while (this.t.BytesAvailable ==0)
                pause(0.1)
            end
            msg= char(this.t.read(this.t.BytesAvailable));
        end

        function Write(this, msg)
            this.t.write(uint8(msg));
        end

        %
        %
        %
        function Close(this)
            this.Write();
        end

        function val= Joint(this, val)
            if nargin==1
                this.Write('CJointT');
                msg= this.Read();
                val= sscanf(msg, '[%f,%f,%f,%f,%f,%f]');
            else
                if length(val) ~= 6
                    error('joint(1x6)');
                end
                msg= sprintf('[%.1f,%.1f,%.1f,%.1f,%.1f,%.1f]', ...
                    val(1), val(2), val(3), val(4), val(5), val(6));
                this.Write(['join=', msg]);
                this.Error;
            end
        end

        function val= Rob(this, val)
            if nargin==1
                this.Write('CRobT');
                msg= this.Read();
                val= sscanf(msg, '[%f,%f,%f][%f,%f,%f,%f][%f,%f,%f,%f]');
                T= Hmat;
                val= T.tQ(val);
            else
                T= Hmat(val);
                val= T.tQ();
                msg= sprintf('[%.1f,%.1f,%.1f]', ...
```

```

        val(1), val(2), val(3));
    this.Write(['rob.pos=', msg]);
    this.Error;
    msg= sprintf(['%.4f,%.4f,%.4f,%.4f'], ...
        val(4), val(5), val(6), val(7));
    this.Write(['rob.orient=', msg]);
    this.Error;
end
end

function val= Tool(this, val)
    if nargin==1
        this.Write('CTool');
        msg= this.Read();
        val= sscanf(msg, '%f,%f,%f [%f,%f,%f,%f]');
        T= Hmat;
        val= T.tQ(val);
    else
        T= Hmat(val);
        val= T.tQ();
        msg= sprintf(['%.1f,%.1f,%.1f'], ...
            val(1), val(2), val(3));
        this.Write(['tool.pos=', msg]);
        this.Error;
        msg= sprintf(['%.4f,%.4f,%.4f,%.4f'], ...
            val(4), val(5), val(6), val(7));
        this.Write(['tool.orient=', msg]);
        this.Error;
    end
end

function val= Wobj(this, val)
    if nargin==1
        this.Write('CWobj');
        msg= this.Read();
        val= sscanf(msg, '%f,%f,%f [%f,%f,%f,%f]');
        T= Hmat;
        val= T.tQ(val);
    else
        T= Hmat(val);
        val= T.tQ();
        msg= sprintf(['%.1f,%.1f,%.1f'], ...
            val(1), val(2), val(3));
        this.Write(['wobj.pos=', msg]);
        this.Error;
        msg= sprintf(['%.4f,%.4f,%.4f,%.4f'], ...
            val(4), val(5), val(6), val(7));
        this.Write(['wobj.orient=', msg]);
        this.Error;
    end
end

function val= Fkin(this, val)
    if nargin==1
        val= [];
    end
    if ~isempty(val)
        this.Joint(val);
    end
    this.Write('CalcRobT');
    msg= this.Read();
    val= sscanf(msg, '%f,%f,%f [%f,%f,%f,%f]');
    T= Hmat;
    val= T.tQ(val);
end

function val= Ikin(this, val)
    if nargin==1

```

```

        val=[];
    end
    if ~isempty(val)
        this.Rob(val);
    end
    this.Write('CalcJointT');
    msg= this.Read();
    val= sscanf(msg, '%f,%f,%f,%f,%f,%f');
end

function val= TestJoint(this, val)
    if nargin==1
        val=[];
    end
    if ~isempty(val)
        this.Joint(val);
    end
    this.Write('TestJoint');
    msg= this.Read();
    val= sscanf(msg, '%f,%f,%f,%f,%f,%f');
end

function val= Speed(this, val)
    if nargin==1
        this.Write('CSpeed');
        msg= this.Read();
        val= sscanf(msg, '%f,%f,%f,%f');
    else
        if ischar(val)
            val=[str2double(val(2:end)),500,5000,1000];
        end
        msg= sprintf('%0.1f,%0.1f,%0.1f,%0.1f', val);
        this.Write(['speed=', msg]);
        this.Error;
    end
end

function val= Zone(this, val)
    if nargin==1
        this.Write('CZone');
        msg= this.Read();
        val= sscanf(msg, '%d,%f,%f,%f,%f,%f,%f');
    else
        if strcmp(val,'fine')
            val=[str2double(val),500,5000,1000];
            msg= sprintf('[true,0,0,0,0,0,0]');
            this.Write(['zone=', msg]);
            this.Error;
        else
            switch val
            case 'z0'
                val= [0.3,0.3,0.3,0.03,0.3,0.03];
            case 'z5'
                val= [5,8,8,0.8,8,0.8];
            otherwise
                val= str2double(val(2:end));
                val=
[val,val*1.5,val*1.5,val*0.15,val*1.5,val*0.15];
            end
            msg= sprintf('[false,%0.2f,%0.2f,%0.2f,%0.2f,%0.2f]', val);
            this.Write(['zone=', msg]);
            this.Error;
        end
    end
end

function Error(this)
    msg= this.Read();

```



```

    if msg=='0'
        errordlg('Datos mal enviados','Error en envio');
    end
end

function MoveAbsJ(this, join, speed, zone, tool)
    if nargin==1
        join=[]; speed=[]; zone=[]; tool=[];
    elseif nargin==2
        speed=[]; zone=[]; tool=[];
    elseif nargin==3
        zone=[]; tool=[];
    elseif nargin==4
        tool=[];
    end

    if ~isempty(join)
        this.Joint(join);
    end
    if ~isempty(speed)
        this.Speed(speed);
    end
    if ~isempty(zone)
        this.Zone(zone);
    end
    if ~isempty(tool)
        this.Tool(tool);
    end
    val= this.TestJoint(join);
    if val(1)>1e8
        errordlg('Posición no accesible','Error de Posicion');
    else
        this.Write('MoveAbsJ');
        this.Error;
    end
end

function MoveJ(this, rob, speed, zone, tool, wobj)
    if nargin==1
        rob=[]; speed=[]; zone=[]; tool=[]; wobj= [];
    elseif nargin==2
        speed=[]; zone=[]; tool=[]; wobj= [];
    elseif nargin==3
        zone=[]; tool=[]; wobj= [];
    elseif nargin==4
        tool=[]; wobj= [];
    elseif nargin==5
        wobj= [];
    end

    if ~isempty(rob)
        this.Rob(rob);
    end
    if ~isempty(speed)
        this.Speed(speed);
    end
    if ~isempty(zone)
        this.Zone(zone);
    end
    if ~isempty(tool)
        this.Tool(tool);
    end
    if ~isempty(wobj)
        this.Wobj(wobj);
    end
    join= this.Ikin(rob);
    if join(1)>1e8
        errordlg('Posición no accesible','Error de Posicion');
    end
end

```

```

        else
            this.Write('MoveJ');
            this.Error;
        end
    end

function MoveL(this, rob, speed, zone, tool, wobj)
    if nargin==1
        rob=[]; speed=[]; zone=[]; tool=[]; wobj= [];
    elseif nargin==2
        speed=[]; zone=[]; tool=[]; wobj= [];
    elseif nargin==3
        zone=[]; tool=[]; wobj= [];
    elseif nargin==4
        tool=[]; wobj= [];
    elseif nargin==5
        wobj= [];
    end

    if ~isempty(rob)
        this.Rob(rob);
    end
    if ~isempty(speed)
        this.Speed(speed);
    end
    if ~isempty(zone)
        this.Zone(zone);
    end
    if ~isempty(tool)
        this.Tool(tool);
    end
    if ~isempty(wobj)
        this.Wobj(wobj);
    end
    join= this.Ikin(rob);
    if join(1)>1e8
        error('Posición no accesible','Error de Posicion');
    else
        this.Write('MoveL');
        this.Error;
    end
end

function OpenTool(this)
    this.Write('OpenTool');
    this.Error;
end

function CloseTool(this)
    this.Write('CloseTool');
    this.Error;
end

function Stop(this)
    this.Write('Stop');
    this.Error;
end

function Reset(this)
    this.Write('Reset');
    this.Error;
end

function TPwrite(this)
    this.Write('TPwrite');
    this.Error;
end
end

```

```

function Path(this, n)
    if nargin==1
        n= 1;
    end
    this.Write(sprintf('Path=%d',n));
    this.Error;
end

function DelPath(this, n)
    if nargin==1
        n= 0;
    end
    this.Write(sprintf('DelPath=%d',n));
    this.Error;
end

function PathJ(this, n)
    if nargin==1
        n= 1;
    end
    this.Write(sprintf('PathJ=%d',n));
    this.Error;
end

function PathL(this, n)
    if nargin==1
        n= 1;
    end
    this.Write(sprintf('PathL=%d',n));
    this.Error;
end

function Disconnect(this)
    this.Write('Disconnect');
    %this.Error;
    %disp('He pasado error');
end

end % methods
end

```

Clase Kin.m

```
classdef Kin < handle

    properties
        robot
        q
        tool
        h
        ejes
        psim
        save
        value
    end %properties

    methods
        function this= Kin(robot, mdl)
            if nargin==0
                error('this= Kin(robot, [mdl])');
            elseif nargin==1
                mdl= [];
            end
            robot.DataFormat= 'column';
            this.robot= robot;
            this.h= Hmat(); this.h.Rad();
            % Parámetros por defecto
            this.tool= robot.BodyNames(end);
            fprintf('Herramienta: %s\n', this.tool);
            this.q= [0,0,0,0,0,0];
            this.ejes.axis= [-700,700,-700,700,-100,1000]*1e-3;
            this.ejes.view= [120,30];
            if isempty(mdl)
                this.psim.mdl= [];
            else
                this.Sim(mdl, 0.3);
                fprintf('Simulación n. Vel= %.3f m/s\n', this.psim.value);
            end
            this.save.q= [];
            this.save.t= [];
            this.save.pose= [];
            this.save.value= 0;
            this.Show();
        end

        function Body(this, name, body0, Htool, stl, Hstl, type)
            if nargin<6
                error('Tool(this, name, body0, Htool, stlname, Hstl, [type])')
            elseif nargin==6
                type= 'fixed';
            end
            body1= robotics.RigidBody(name);
            % Se crea un eje fijo
            jtool1 = robotics.Joint(name,type);
            % setFixedTransform permite matrices homogéneas
            setFixedTransform(jtool1, Htool);
            body1.Joint= jtool1;
            % Se une al cuerpo final del robot 'tool0'
            addBody(this.robot, body1, body0);
            % Definir la posición de lapiz con respecto al eje del tool1
            addVisual(this.robot.Bodies(end), "Mesh", stl, Hstl);
            Show(this);
        end

        function DelBody(this,name)
            removeBody(this.robot, name);
            if strcmp(this.tool,name)==1
                this.tool= this.robot.BodyNames(end);
            end
        end
    end
end
```

```

        Show(this);
    end

function Tool(this, name, body0, Htool, stl, Hstl)
    if nargin==2
        this.tool= name;
        return
    elseif nargin<6
        error('Tool(this, name, body0, Htool, stl, Hstl)')
    end
    this.Body(name, body0, Htool, stl, Hstl);
    this.tool= name;
end

function Wobj(this, name, body0, Hwobj, stl, Hstl)
    if nargin<6
        error('Wobj(this, name, body0, Hwobj, stl, Hstl)')
    end
    this.Body(name, body0, Hwobj, stl, Hstl);
    this.tool;
end

function robot= Robot(this)
    robot= this.robot;
end

function pose= Pose(this, q)
    if nargin==1
        q= this.q;
    end
    nq= size(q, 1);
    pose= zeros(nq,6);
    for i= 1:nq
        this.h.H(getTransform(this.robot, q(i,:)', this.tool));
        pose(i,:)= this.h.tRzyx;
    end
end

function q= Joint(this, pose, w)
    if nargin==1
        q= this.q;
        return
    elseif nargin==2
        w= ones(1,6);
    end
    ik = robotics.InverseKinematics('RigidBodyTree', this.robot);
    npose= size(pose, 1);
    q= zeros(npose, length(this.q));
    q0= this.q';
    %pose(:,4:6)= rotm2eul(eul2rotm(pose(:,4:6)));
    for i= 1:npose
        H= this.h.tRzyx(pose(i,:));
        q0= ik(this.tool, H, w, q0);
        q(i,:)= q0';
    end
end

function q= MoveAbsJ(this, qN, nPts)
    if nargin<2
        error('q= MoveAbsJ(this, q1, [nPts])')
    elseif nargin==2
        nPts= 2;
    end
    if nPts>2
        nq= length(this.q);
        q= zeros(nPts, nq);
        % Interpolaci3n de q
        for i= 1:nq

```

```

        q(:,i)= linspace(this.q(1,i), qN(1,i), nPts)';
    end
else
    q= [this.q; qN];
end
this.Show(q);
this.q= q(end,:);
end

function q= MoveJ(this, poseN, Hwobj, nPts)
    if nargin<2
        error('q= MoveJ(this, poseN, [nPts])');
    elseif nargin==2
        Hwobj=[]; nPts= 2;
    elseif nargin==3
        nPts= 2;
    end

    % Se desean no cambiar el giro actual
    [fil, col]= size(poseN);
    if col==3
        pose0= this.Pose(this.q);
        poseN(:, 4:6)= repmat(pose0(1,4:6), [fil,1]);
        w= [0,0,0,1,1,1];
    else
        w= ones(1,6);
    end

    if ~isempty(Hwobj)
        this.h.H(Hwobj);
        poseN= this.h.Tpose(poseN);
    end

    % Solver con la estructura actual
    nq= length(this.q);
    if nPts>2
        qN= this.Joint(poseN(1,:),w);
        q= zeros(nPts, nq);
        % Interpolaci3n de q
        for i= 1:nq
            q(:,i)= linspace(this.q(1,i), qN(1,i), nPts)';
        end
    else
        q= [this.q; this.Joint(poseN,w)];
    end
    this.Show(q, poseN);
    this.q= q(end,:);
end

function q= MoveL(this, pose1, Hwobj, nPts)
    if nargin<2
        error('q= MoveL(this, pose1, [nPts])');
    elseif nargin==2
        Hwobj= []; nPts= 10;
    elseif nargin==3
        nPts= 10;
    end

    pose0= this.Pose(this.q);
    if length(pose1)==3
        pose1(4:6)= pose0(4:6);
        w= [0,0,0,1,1,1];
    else
        w= ones(1,6);
    end

    if ~isempty(Hwobj)
        this.h.H(Hwobj);
    end

```

```

        pose1= this.h.Tpose(pose1);
    end

    poseN= zeros(nPts,6);
    % Interpolaci3n en todos los elementos
    for i= 1:6
        poseN(:,i)= linspace(pose0(i), pose1(i), nPts)';
    end
    q= this.Joint(poseN,w);
    this.Show(q, poseN);
    this.q= q(end, :);
end

function Show(this, q, pose)
    if nargin==1
        q= this.q; pose= [];
    elseif nargin==2
        pose=[];
    end
    if isempty(this.psim.mdl) % Gr3fica matlab
        for i= 1:size(q,1)
            hold off
            show(this.robot, q(i,:));
            axis(this.ejes.axis);
            if ~isempty(pose)
                hold on
                this.h.H(eye(4));
                this.h.Tpose(pose,0.05);
            end
            view(this.ejes.view(1),this.ejes.view(2));
            drawnow
        end
    else % Simulaci3n Simulink multi-body
        if this.psim.type=='T'
            t= linspace(0,this.psim.value,size(q,1))';
        else
            nPts= size(q,1);
            t= zeros(nPts, 1);
            qpose= this.Pose(q);
            qpose= qpose(:,1:3);
            for i= 2:nPts
                d= sqrt(sum((qpose(i-1,:)- qpose(i,:)).^2));
                t(i)= t(i-1)+ d/this.psim.value+ 0.01;
            end
        end
        opt= simset('SrcWorkspace','Current');
        q0= q(1,:); % Posici3n inicial del robot
        if size(pose,1)<3 % Para dibujar trayectoria en pantalla
            pose= repmat(qpose(end,:), [3,1]).*(1+ randn(3,3)*1e-3);
        else
            pose= pose(:,1:3);
        end
        sim(this.psim.mdl,t,opt,[t,q]);
        % Guarda valores para reproducci3n final
        if this.save.value==1
            this.save.q= [this.save.q; q];
            if isempty(this.save.t)
                this.save.t= [this.save.t; t];
            else
                this.save.t= [this.save.t; t+ this.save.t(end)+1];
            end
            this.save.pose= [this.save.pose; pose];
        end
        end
        this.q= q(end,:);
    end

function Sim(this, mdl, value, type)

```

```

        if nargin==1
            this.psim.mdl=[];
            return
        elseif nargin<3
            error('Sim(this, mdl, par, [type])');
        elseif nargin==3
            this.psim.type= 'V';
        else % 'T'
            this.psim.type= type;
        end
        this.psim.value= value;
        this.psim.mdl= mdl;
    end

    function Rec(this, value)
    % Control de grabaci3n de simulaci3n
        if (nargin==1 | value == 1)
            this.save.value= 1;
        elseif value==0
            this.value= 0;
        elseif value==2
            this.save.q= [];
            this.save.t= [];
            this.save.pose= [];
            this.save.value= 1;
        else
            error('Valor incorrecto');
        end
    end

    function Rep(this)
        if size(this.save.q,1)>3
            opt= simset('SrcWorkspace','Current');
            q0= this.save.q(1,:); % Posici3n inicial del memoria
            % Para evitar dos valores iguales.
            pose= this.save.pose+ randn(size(this.save.pose,1),3)*1e-5;
            sim(this.psim.mdl,this.save.t,opt,[this.save.t,this.save.q]);
        else
            error('No hay record')
        end
    end

end % methods

end % class

```


ANEXO III. CODIGO RAPID

```
MODULE M_tcpABB

    LOCAL PERS tooldata
    tool:=[TRUE,[[0,0,260],[1,0,0,0]],[0.001,[0,0,0.001],[1,0,0,0],0,0,0]];
    LOCAL PERS wobjdata
    wobj:=[FALSE,TRUE,"",[[200,0,950],[0,0.7071,0.7071,0]],[[0,0,0],[1,0,0,0]]];
    ! local PERS loaddata load:=[0.001,[0,0,0.001],[1,0,0,0],0,0,0];
    LOCAL pers robtarget rob:=[[0,102.54,651.969],[0.7071,-6.88427E-06,-
6.88427E-06,-0.7071],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    ! LOCAL pers robtarget
    robC:=[[0,0,0],[1,0,0,0],[0,0,0,0],[1E+09,1E+09,1E+09,1E+09,1E+09,1E+09]];
    LOCAL pers jointtarget join:=[[0,-0.000216056,-
0.00335314,0,90.0025,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

    LOCAL PERS speeddata speed:=[500,500,5000,1000];
    LOCAL pers zonedata zone:=[FALSE,0.3,0.3,0.3,0.03,0.3,0.03];

    ! Salvar puntos robtarget para después moverse a través de ellos de forma
rápida
    LOCAL const num NrobPath:= 100; ! Máximo número de puntos
    LOCAL VAR num idxPath{NrobPath};
    LOCAL var robtarget robPath{NrobPath}; ! Vector de Nrobm puntos

    ! Ordenes que se pueden recibir
    LOCAL CONST num nOrden:=18;
    LOCAL const string orden{nOrden}:=["MoveAbsJ","MoveJ","MoveL",
"CJointT","CRobT","CTool","CWobj","CSpeed","CZone",
"CalcJointT","CalcRobT","TestJoint",
"OpenTool","CloseTool","Reset","Stop","TPwrite","Disconnect"];

    ! Modificación del valor de los datos básicos
    LOCAL CONST num nVarTipo:=13;
    LOCAL CONST string
VarTipo{nVarTipo}:=["join","rob.pos","rob.orient","tool.pos","tool.orient",
"wobj.pos","wobj.orient","speed","zone","Path","DelPath","PathJ","PathL"];

    ! Conexión vía TCP/IP con un cliente
    ! Activa las conexiones "192.168.125.1" IP local robot;
    LOCAL CONST string IPdir:="127.0.0.1";
    LOCAL CONST num IPport:=30000;
    ! define el socket del cliente (robot) servidor (Matlab)
    LOCAL VAR socketdev server_socket;
    LOCAL VAR socketdev client_socket;
    ! Ver mensajes
    LOCAL VAR bool ver:=true;

    PROC mainTcpABB()

        VAR num opcion;

        !CONNECT perslint WITH iroutinel;
        !IPers parar,perslint;

        ConfL\off;
        ConfJ\off;

        WHILE TRUE DO
            TPerase;
            TPwrite "Menú Comunicación TCP_IP:";
            TPwrite " 1.- Reseteo Variables";
            TPwrite " 2.- Ver/no Ver Mensajes";
```

```

        TPWrite "    3.- TCP: Iniciar Comunicaciones";
        TPReadFK opcion,"Opción:", "Reset", "", "Ver", "", "tcpABB";

        TEST opcion
        CASE 1:
            ResetVar;
        CASE 3:
            ver:=NOT ver;
            IF ver THEN
                TPWrite "Ver Mensajes";
            ELSE
                TPWrite "No Ver Mensajes";
            ENDIF
            waittime 2;
        CASE 5:
            AbrirComunicación;
        ENDTEST
    ENDWHILE
ENDPROC

LOCAL PROC Init()
    WHILE TRUE DO
        MsgClient;
    ENDWHILE
ENDPROC

LOCAL PROC ResetVar()
    tool:=[TRUE, [[0,0,0], [1,0,0,0]], [0.001, [0,0,0.001], [1,0,0,0], 0,0,0]];
    wobj:=[FALSE, TRUE, "", [[0,0,0], [1,0,0,0]], [[0,0,0], [1,0,0,0]]];
    join:=[[0,0,0,0,90,0], [1e9,1e9,1e9,1e9,1e9,1e9]];
    rob:=CalcRobT(join,tool\WObj:=wobj);
    speed:=[100,500,5000,1000];
    zone:=[true,0,0,0,0,0,0];
    ver:=False;
    ! Posición inicial
    MoveAbsJ join,speed,zone,tool;
ENDPROC

! Redondea los valores para que entren en una cadena de 80 caracteres
local FUNC robjoint RoundJoint(robjoint joint)
    joint.rax_1:=Round(joint.rax_1\Dec:=1);
    joint.rax_2:=Round(joint.rax_2\Dec:=1);
    joint.rax_3:=Round(joint.rax_3\Dec:=1);
    joint.rax_4:=Round(joint.rax_4\Dec:=1);
    joint.rax_5:=Round(joint.rax_5\Dec:=1);
    joint.rax_6:=Round(joint.rax_6\Dec:=1);
    RETURN joint;
ENDFUNC

local FUNC pos RoundPos(pos trans)
    trans.x:=Round(trans.x\Dec:=1);
    trans.y:=Round(trans.y\Dec:=1);
    trans.z:=Round(trans.z\Dec:=1);
    RETURN trans;
ENDFUNC

local FUNC orient RoundOrient(orient rot)
    rot.q1:=Round(rot.q1\Dec:=4);
    rot.q2:=Round(rot.q2\Dec:=4);
    rot.q3:=Round(rot.q3\Dec:=4);
    rot.q4:=Round(rot.q4\Dec:=4);
    RETURN rot;
ENDFUNC

LOCAL PROC MsgOrden(string cad)
    VAR num caso;
    VAR num pos;
    VAR bool err:=FALSE;

```

```

FOR i FROM 1 TO nOrden DO
  pos:=StrMatch(cad,1,orden{i});
  IF pos=1 THEN
    caso:=i;
    IF ver THEN
      TPWrite "Orden: "+orden{i};
    ENDIF
  ENDIF
ENDFOR

TEST caso
CASE 1:
  ! "MoveAbsJ"
  rob:=CalcRobT(join,tool\WObj:=wobj);
  MoveAbsJ join,speed,zone,tool;
  MsgSend "1";
CASE 2:
  ! "MoveJ"
  join:=CalcJointT(rob,tool\WObj:=wobj);
  MoveJ rob,speed,zone,tool\WObj:=wobj;
  MsgSend "1";
CASE 3:
  ! "MoveL"
  join:=CalcJointT(rob,tool\WObj:=wobj);
  MoveL rob,speed,zone,tool\WObj:=wobj;
  MsgSend "1";
CASE 4:
  ! "CJointT"
  join:=CJointT();
  MsgSend ValToStr(join.robax);
CASE 5:
  ! "CRobT"
  rob:=CRobT(\Tool:=tool\WObj:=wobj);
  MsgSend ValToStr(RoundPos(rob.trans));
  MsgSend ValToStr(RoundOrient(rob.rot));
  ! MsgSend ValToStr(rob.robconf);
CASE 6:
  ! "CTool"
  tool:=CTool();
  MsgSend ValToStr(RoundPos(tool.tframe.trans));
  MsgSend ValToStr(RoundOrient(tool.tframe.rot));
CASE 7:
  ! "CWobj"
  wobj:=CWobj();
  MsgSend ValToStr(RoundPos(wobj.uframe.trans));
  MsgSend ValToStr(RoundOrient(wobj.uframe.rot));
CASE 8:
  ! CSpeed
  MsgSend ValToStr(speed);
CASE 9:
  ! CZone
  MsgSend ValToStr(zone);
CASE 10:
  ! "CalcJointT"
  join:=CalcJointT(rob,tool\WObj:=wobj);
  IF err THEN
    MsgSend "[1e9,1e9,1e9,1e9,1e9,1e9]";
  ELSE
    MsgSend ValToStr(RoundJoint(join.robax));
  ENDIF
CASE 11:
  ! "CalcRobT"
  rob:=CalcRobT(join,tool\WObj:=wobj);
  MsgSend ValToStr(RoundPos(rob.trans));
  MsgSend ValToStr(RoundOrient(rob.rot));
  !MsgSend ValToStr(rob.robconf);
CASE 12:

```

```

        ! "TestJoint"
        join:=CalcJointT(CalcRobT(join,tool\WObj:=wobj),tool\WObj:=wobj);
        IF err THEN
            MsgSend "[1e9,1e9,1e9,1e9,1e9,1e9]";
        ELSE
            MsgSend ValToStr(RoundJoint(join.robax));
        ENDIF
    CASE 13:
        ! "OpenTool"
        AbrirPinza;
        MsgSend "1";
    CASE 14:
        ! "CloseTool"
        CerrarPinza;
        MsgSend "1";
    CASE 15:
        ! "Reset"
        ResetVar;
        MsgSend "1";
    CASE 16:
        ! "Stop"
        Stop;
        MsgSend "1";
    CASE 17:
        ! TPWrite
        Ver:=NOT Ver;
        MsgSend "1";
    CASE 18:
        ! Disconnect
        !SocketClose server_socket;
        TPWrite "Comunicación Finalizada";
        WaitTime 2;
        AbrirComunicación;
        !MsgSend "1";
    DEFAULT:
        MsgSend "0";
    ENDTEST

    ! Errores en las ordenes
    ERROR
    IF ERRNO=Err_Roblimit THEN
        IF ver THEN
            TPWrite "Error en límites Joint";
        ENDIF
        err:=TRUE;
        TRYNEXT;
    ENDIF

    ENDPROC

    LOCAL PROC MsgVar(string cad)
        VAR string strdato;
        VAR string strvar;
        VAR num len;
        VAR num pos;
        VAR num caso;
        VAR bool ok;
        VAR bool ok1;
        VAR bool err:=False;
        VAR num n;
        VAR num idx;

        !          tpwrite cad;
        len:=StrLen(cad);
        pos:=StrFind(cad,1,"=");
        strvar:=StrPart(cad,1,pos-1);
        strdato:=StrPart(cad,pos+1,len-pos);
        !          TPWrite strvar;

```

```

!           TPWrite strdato;

FOR i FROM 1 TO nVarTipo DO
  pos:=StrMatch(cad,1,VarTipo{i});
  IF pos=1 THEN
    caso:=i;
    IF ver THEN
      TPWrite "Orden:= "+cad;
    ENDIF
  ENDIF
ENDFOR

TEST caso
CASE 1:
  ! "join"
  ok:=StrToVal(strdato,join.robax);
CASE 2:
  ! "rob.pos"
  ok:=StrToVal(strdato,rob.trans);
CASE 3:
  ! "rob.orient"
  ok:=StrToVal(strdato,rob.rot);
CASE 4:
  ! "tool.pos"
  ok:=StrToVal(strdato,tool.tframe.trans);
CASE 5:
  ! "tool.orient"
  ok:=StrToVal(strdato,tool.tframe.rot);
CASE 6:
  ! "wobj.pos"
  ok:=StrToVal(strdato,wobj.uframe.trans);
CASE 7:
  ! "wobj.orient"
  ok:=StrToVal(strdato,wobj.uframe.rot);
CASE 8:
  ! "speed"
  ok:=StrToVal(strdato,speed);
CASE 9:
  ! "zone"
  ok:=StrToVal(strdato,zone);
CASE 10:
  ! Path
  ok1:= TRUE;
  FOR i FROM 1 TO NrobPath DO
    IF ok1 and idxPath{i}=0 THEN
      ok:= StrToVal(strdato, idxPath{i});
      IF ok THEN
        robPath{i}:= rob;
        ok1:= FALSE;
      ENDIF
    ENDIF
  ENDFOR
CASE 11:
  ! DelPath
  ok:= StrToVal(strdato,idx);
  IF ok THEN
    IF idx=0 THEN
      FOR i FROM 1 TO NrobPath DO
        idxPath{i}:= 0;
      ENDFOR
    ELSE
      FOR i FROM 1 TO NrobPath DO
        IF idxPath{i}=idx THEN
          idxPath{i}:= 0;
        ENDIF
      ENDFOR
    ENDIF
  ENDIF
ENDIF

```

```

CASE 12:
    ! PathJ
    ok:= StrToVal(strdato,idx);
    IF ok THEN
        FOR i FROM 1 TO NrobPath DO
            IF idx= idxPath{i} THEN
                MoveJ robPath{i},speed,zone,tool\WObj:=wobj;
            ENDIF
        ENDFOR
    ENDIF
CASE 13:
    ! PathL
    ok:= StrToVal(strdato,idx);
    IF ok THEN
        FOR i FROM 1 TO NrobPath DO
            IF idx= idxPath{i} THEN
                MoveL robPath{i},speed,zone,tool\WObj:=wobj;
            ENDIF
        ENDFOR
    ENDIF
DEFAULT:
    err:=True;
ENDTEST

IF (not err) and ok THEN
    MsgSend "1";
ELSE
    MsgSend "0";
ENDIF
ENDPROC

local PROC MsgSend(string cad)
    IF ver THEN
        IF StrMatch(cad,1,"0")=1 THEN
            TPWrite "ERROR Mensaje";
        ELSE
            TPWrite "Server: "+cad;
        ENDIF
    ENDIF
    SocketSend client_socket\Str:=cad;
ENDPROC

local PROC AbrirComunicación()

    ! Cerrar posibles socket abiertos en anteriores ejecuciones
    SocketClose server_socket;
    SocketClose client_socket;
    TPErase;
    TPWrite "Conectando con cliente ";
    TPWrite "    IP robot (servidor): "+IPdir;
    TPWrite "    Port robot: "\Num:=IPport;
    ! Abrir sockets actual
    SocketCreate server_socket;
    ! Enlaza al zócalo
    SocketBind server_socket,IPdir,IPport;
    ! Escucha
    SocketListen server_socket;
    ! Espera para la conexión el máximo tiempo permitido
    SocketAccept server_socket,client_socket\Time:=WAIT_MAX;
    SocketSend client_socket\Str:="Comunicación Establecida";
    TPWrite "Comunicación Establecida";
    WaitTime 1;
    Init;
ENDPROC

local PROC MsgClient()
    VAR string msg;

```

```

VAR bool ok;
VAR num len;
VAR num pos;

!SocketSend client_socket\Str:="Orden/dato=valor";
SocketReceive client_socket\Str:=msg\Time:=WAIT_MAX;

IF ver THEN
    TPWrite "Client: "+msg;
ENDIF
pos:=StrFind(msg,1,"=");
len:=StrLen(msg);
IF pos<len THEN
    MsgVar(msg);
ELSE
    MsgOrden(msg);
ENDIF
ENDPROC

! Abre la pinza
local PROC AbrirPinza()
    Set dol5;
    ! Abrir y desattacher
    WaitTime 2;
    Reset dol5;
ENDPROC
! Cerrar Pinza
local PROC CerrarPinza()
    Set dol6;
    ! Activar los sensores de attacher
    WaitTime 2;
    Reset dol6;
ENDPROC
ENDMODULE

```


ANEXO IV. ESTUDIO ECONOMICO

En este capítulo se llevará a cabo una estimación del coste del proyecto al mismo tiempo que una enumeración de los materiales empleados. Se agruparán los costes según el origen y serán divididos en costes directos e indirectos. Finalmente, se detallará el presupuesto final.

Costes directos del proyecto

Entendemos como costes directos aquel tipo de gasto que tiene una relación directa a la realización y producción de un servicio o producto.

Costes asociados al equipo

En este apartado se resumirá el coste del material que forma parte de la instalación diseñada, tanto coste material como software.

Coste de material

El coste de material y equipo se recoge en la Tabla 1.

Tabla 1 Coste de material y equipo

Costes de material y equipo				
Concepto	Procedencia	Coste unitario (€)	Unidades	Coste total (€)
Robot ABB IRB-IRB-120 + licencia RobotStudio	ABB	15.000,00	1	15.000,00
Bobina impresión 3D (1kg)	Amazon	22,95	3	68,85
Imanes 6mm (105 unidades)	WukongMag (Amazon)	10,99	2	21,98
6 Listones 2.2x2,2x200cm	Leroy Merlin	9,49	1	9,49
			Total	15.100,32

Coste de software y herramientas

El coste del software empleado en este proyecto, puesto que es un proyecto educativo y la Universidad de Valladolid cuenta con licencias de estudiante, ha sido nulo y también se han empleado softwares gratuitos. Aun así, el coste se recogerá en la Tabla 2 para dejar constancia de los softwares empleados.

Tabla 2 Coste del software empleado

Coste del software empleado			
Concepto	Coste real licencia (€/año)	Coste alumno licencia Uva (€/año)	Coste total (€)
Matlab	500,00	0,00	0,00
Catia V5	1000,00	0,00	0,00
RobotStudio (incluido en compra robot ABB)	-	-	-
Ultimaker Cura	0,00	0,00	0,00
		Total	0,00

Tabla 3 Coste de herramientas empleadas

Coste herramientas empleadas			
Concepto	Coste unitario (€/hora)	Unidades (horas)	Coste total (€)
Impresora 3D Creality Ender 3	0,75	86	64,50
Banco de trabajo (herramientas varias)	3	20	60,00
Ordenador	0.3	900	270,00
		Total	394,50

Coste asociado a la mano de obra

En primer lugar, se calculan las horas de trabajo en un año estimando el sueldo por horas y para ajustar el tiempo necesario para realizar cada parte del trabajo, Tabla 4. Así se estiman los costes asociados a la mano de obra necesarios para la puesta en marcha del robot, programación y diseño CAD.

Tabla 4 Horas laborables por año

Horas laborables por año	
Concepto	Tiempo (días)
Año medio	365
Vacaciones	22
Días festivos	14
Fines de semana	104
Días perdidos	5
Total	220
x 8 horas/día	1760 h/año

Con las horas laborables, el sueldo medio de un ingeniero con un nivel medio de experiencia que puede aproximarse por 21€/hora y el tiempo dedicado al proyecto, de 6 meses trabajando 8 horas diarias, podemos obtener la cantidad total del coste de la mano de obra, Tabla 5.

Tabla 5 Costes de mano de obra

Coste de mano de obra			
Concepto	Coste por hora (€/hora)	Unidades (horas)	Coste total (€)
Diseño 3D y fabricación	21,00	150	3150,00
Programación Matlab	21,00	550	11550,00
Programación robot ABB	21,00	180	3780,00
	Total	880	18480,00

Sumando los costes de material y equipo y mano de obra obtenemos los costes totales a la parte de costes directos, Tabla 6.

Tabla 6 Costes directos totales

Costes directos totales	
Concepto	Importe (€)
Costes asociados a materiales y equipo	15.100,32 + 394,50
Costes de mano de obra	18.480,00
Total	33.974,82

Costes indirectos del proyecto

Entendemos como costes indirectos aquel tipo de gasto que tiene una relación directa con los recursos de luz consumidos durante la realización de un proyecto y que no pueden relacionarse directamente este.

Costes indirectos totales			
Concepto	Coste (€/kWh)	Unidades (horas)	Coste total (€)
Consumo eléctrico de herramientas y equipos (400W)	0,09	90	3,24
Otros consumos eléctricos: iluminación (200W)	0,09	300	5,4
		Total	8,64

Coste total del proyecto

Finalmente, el coste total de la realización del proyecto sería la suma de los costes director y los costes indirectos del proyecto. Esta cantidad se resume en la Tabla 7.

Tabla 7 Coste total del proyecto

Concepto	Importe (€)
Costes directos	33.974,82
Costes indirectos	8,64
Total	33.983,46



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Electrónica Industrial y Automática

MÁSTER EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA DE INGENIERÍAS INDUSTRIALES
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

Diseño, simulación y fabricación de una
estación robotizada universitaria

ANEXOS

Autor: D. Víctor Lobo Granado

Tutor: D. Alberto Herreros López

Valladolid, Julio, 2020.