

```

classdef AppTFG_JR_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        Cuaternios              matlab.ui.control.Label
        AnguloGiro              matlab.ui.control.Label
        MargenAux4              matlab.ui.control.EditField
        MargenAux3              matlab.ui.control.EditField
        CUATERNIOSLabel         matlab.ui.control.Label
        ANGULOGIROLabel         matlab.ui.control.Label
        Coordenadas             matlab.ui.control.Label
        Objeto                  matlab.ui.control.Label
        COORDENADASLabel        matlab.ui.control.Label
        OBJETOLabel             matlab.ui.control.Label
        MostrarResultados      matlab.ui.control.Button
        IniciarRastreo          matlab.ui.control.Button
        Tick_Loading            matlab.ui.control.Image
        Tick_Rojo               matlab.ui.control.Image
        Tick_Verde              matlab.ui.control.Image
        OBSERVACIONESLabel     matlab.ui.control.Label
        Observaciones           matlab.ui.control.Label
        MargenDerecho           matlab.ui.control.TextArea
        MargenIzquierdo         matlab.ui.control.TextArea
        MargenSuperior          matlab.ui.control.TextArea
        MargenInferior          matlab.ui.control.TextArea
        MargenAux2              matlab.ui.control.TextArea
        MargenAux1              matlab.ui.control.TextArea
        Lamp_Conexion           matlab.ui.control.Lamp
        CONECTARSwitch          matlab.ui.control.Switch
        CONECTARSwitchLabel     matlab.ui.control.Label
        Lamp_Mesa               matlab.ui.control.Lamp
        Lamp_Generador          matlab.ui.control.Lamp
        GENERARPIEZAALATORIASwitch matlab.ui.control.Switch
        GENERARPIEZAALATORIASwitchLabel matlab.ui.control.Label
        GIRARMESASwitch         matlab.ui.control.Switch
        GIRARMESASwitchLabel    matlab.ui.control.Label
    end

    properties (Access = private)

        % Objetos de la clase empleados
        DA;
        C1 = 'DESKTOP-4BSJL2A_';
        C2 = 'Controlador2_';

        grupo_C;    %Grupo Conectar
        grupo_G;    %Grupo Generador
        grupo_M;    %Grupo Mesa
        grupo_R;    %Grupo Rastreo
        grupo_MR;   %Grupo Mostrar Resultados
        grupo_Fig;  %Grupo Figuras

        Switch_Generador_Piezas_On;    %di25
        Switch_Generador_Piezas_Off;    %di26
    end
end

```

```
Switch_Mesa_Giratoria_1_On;      %di27
Switch_Mesa_Giratoria_1_Off;     %di28
Button_Rastreo;                  %di29
Mostrar_Resultados;              %NO ES INTERRUPCION
```

```
%Variables empleadas para el C1
```

```
Ejecucion_Rastreo;
Rastreo_OK;
Contador_Piezas = 0;
Contador_Reset_Piezas = 0;
Valor_Pieza = 0;
Contador_Giro_Mesa = 0;
Contador_Paro_Mesa = 0;
Valor_Mesa = 0;
```

```
Contador_CalcArea = 0;
Contador_Rastreo = 0;
Contador_MR = 0;
Contador_Giro = 0;
```

```
%Variables empleadas para el C2
```

```
Inicio_CogerPieza;
Fin_CogerPieza;
Tiempo;
Q_1 = 0;
Q_2 = 0;
Q_3 = 0;
Q_4 = 0;
Figura = '';
f1 = [];
f2 = [];
f3 = [];
f4 = [];
```

```
%Variables adicionales empleadas
```

```
Color_Rojo = [1,0,0];
Color_Verde = [0,1,0];
Color_Negro = [0,0,0];
Stop = 0;
```

```
end
```

```
methods (Access = private)
```

```
function Conectar(app)
```

```
    %Creamos la comunicación
```

```
    app.DA = opcda('localhost', 'ABB.IRC5.OPC.Server.DA');
    connect(app.DA);
    app.grupo_C = addgroup(app.DA, 'Grupo_Conectar');
    set(app.grupo_C, 'UpdateRate', 0.001);
```

```
    app.Lamp_Conexion.Color = app.Color_Verde;
    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' CONECTADO AL SERVIDOR CON ÉXITO.';
```

```
app.TickRojo(0);
app.TickLoading(0);
app.TickVerde(1);

app.Stop = 0;

app.Reset_Controlador1;
pause(1);
app.Reset_Controlador2;

app.Ejecucion_Rastreo = additem(app.grupo_C, [app.C1, 'Controlador1.RAPID.↵
T_ROB1.Module1.ejecucion_rastreo']);
app.Rastreo_OK = additem(app.grupo_C, [app.C1, 'Controlador1.RAPID.T_ROB1.↵
Module1.rastreo_ok']);

write(app.Ejecucion_Rastreo, 0);
pause(0.1);
write(app.Rastreo_OK, 0);
pause(0.1);

while app.Stop == 0

    serveritems(app.DA, [app.C1, 'Controlador1.RAPID.T_ROB1.Module1.↵
ejecucion_rastreo']);
    serveritems(app.DA, [app.C1, 'Controlador1.RAPID.T_ROB1.Module1.↵
rastreo_ok']);

    mostrar1 = read(app.Ejecucion_Rastreo);
    pause(1);
    disp(mostrar1);

    mostrar2 = read(app.Rastreo_OK);
    pause(1);
    disp(mostrar2);

end
end

function Desconectar(app)

    %Hacemos reset de los contadores e interrupciones de ambos controladores
    app.Lamp_Conexion.Color = app.Color_Rojo;
    app.Reset_Controlador1;
    pause(1);
    app.Reset_Controlador2;
    pause(1);

    %Dejamos de leer las variables del servidor.
    app.Stop = 1;
    pause(1);

    %Nos desconectamos del servidor y eliminamos el objeto creado para la↵
conexion.
    disconnect(app.DA);
```

```
    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' DESCONECTADO DEL SERVIDOR CON ÉXITO.';
    app.TickRojo(0);
    app.TickLoading(0);
    app.TickVerde(1);
    delete(app.DA)
end

function Generar_Pieza_Aleatoria(app)

    app.Lamp_Generador.Color = app.Color_Verde;

    if app.Contador_Piezas > 0
        write(app.Switch_Generador_Piezas_Off,0)
        pause(0.1);
    end

    write(app.Switch_Generador_Piezas_On, 1);
    pause(0.1);

    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' PIEZA ALEATORIA GENERADA.';
    app.TickRojo(0);
    app.TickLoading(0);
    app.TickVerde(1);
    app.Contador_Piezas = app.Contador_Piezas + 1;
    app.Valor_Pieza = 1;

end

function Reset_Pieza_Aleatoria(app)

    app.Lamp_Generador.Color = app.Color_Rojo;

    write(app.Switch_Generador_Piezas_On, 0);
    pause(0.1);
    write(app.Switch_Generador_Piezas_Off,1)
    pause(0.1);

    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' PIEZA ALEATORIA ELIMINADA.';
    app.TickRojo(0);
    app.TickLoading(0);
    app.TickVerde(1);
    app.Contador_Reset_Piezas = app.Contador_Reset_Piezas + 1;
    app.Valor_Pieza = 0;

end

function Girar_Mesa_1(app)

    app.Lamp_Mesa.Color = app.Color_Verde;

    if app.Contador_Giro_Mesa > 0
        write(app.Switch_Mesa_Giratoria_1_Off, 0)
```

```
        pause(0.1);
    end

    write(app.Switch_Mesa_Giratoria_1_On, 1);
    pause(0.1);

    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' MESA 1 GIRANDO...';
    app.TickVerde(0)
    app.TickRojo(0);
    app.TickLoading(1);
    app.Contador_Giro_Mesa = app.Contador_Giro_Mesa + 1;
    app.Valor_Mesa = 1;
    app.Contador_Giro = app.Contador_Giro + 1;

end

function Parar_Mesa_1(app)

    app.Lamp_Mesa.Color = app.Color_Rojo;

    write(app.Switch_Mesa_Giratoria_1_On, 0);
    pause(0.1);

    write(app.Switch_Mesa_Giratoria_1_Off, 1)
    pause(0.1);

    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' MESA 1 PARADA.';
    app.TickRojo(0);
    app.TickLoading(0);
    app.TickVerde(1);

    app.Contador_Paro_Mesa = app.Contador_Paro_Mesa + 1;
    app.Valor_Mesa = 0;

end

function Reset_Controlador1(app)

    %Hacemos Reset a las señales de interrupcion
    if app.Switch_Generador_Piezas_On ~= 0
        write(app.Switch_Generador_Piezas_On, 0);
        pause(0.1);

    elseif app.Switch_Generador_Piezas_Off ~=0
        write(app.Switch_Generador_Piezas_Off, 0);
        pause(0.1);

    elseif app.Switch_Mesa_Giratoria_1_On ~= 0
        write(app.Switch_Mesa_Giratoria_1_On, 0);
```

```
        pause(0.1);

elseif app.Switch_Mesa_Giratoria_1_Off ~= 0
    write(app.Switch_Mesa_Giratoria_1_Off, 0);
    pause(0.1);

elseif app.Button_Rastreo ~= 0
    write(app.Button_Rastreo, 0);
    pause(0.1);
end

%Reset de otras variables empleadas
if app.Mostrar_Resultados ~= 0
    write(app.Mostrar_Resultados, 0);
    pause(0.1);

elseif app.Ejecucion_Rastreo ~= 0
    write(app.Ejecucion_Rastreo, 0);
    pause(0.1);

elseif app.Rastreo_OK ~= 0
    write(app.Rastreo_OK, 0);
    pause(0.1);
end

%Hacemos Reset a todos los contadores
app.Contador_Piezas = 0;
app.Contador_Reset_Piezas = 0;
app.Valor_Pieza = 0;
app.Contador_Giro_Mesa = 0;
app.Contador_Paro_Mesa = 0;
app.Valor_Mesa = 0;

app.Contador_CalcArea = 0;
app.Contador_Rastreo = 0;
app.Contador_MR = 0;
pause(0.1)

end

function Reset_Controlador2(app)

%Hacemos Reset a todas las variables
if app.Inicio_CogerPieza ~= 0
    write(app.Inicio_CogerPieza, 0);
    pause(0.1);

elseif app.Fin_CogerPieza ~= 0
    write(app.Fin_CogerPieza, 0);
    pause(0.1);

elseif app.C_X ~= 0
    write(app.C_X, 0);
    pause(0.1);
```

```
elseif app.C_Y ~= 0
    write(app.C_Y,0);
    pause(0.1);

elseif app.C_Z ~= 0
    write(app.C_Z,0);
    pause(0.1);

elseif app.Q_1 ~= 0
    write(app.Q_1,0);
    pause(0.1);

elseif app.Q_2 ~= 0
    write(app.Q_2,0);
    pause(0.1);

elseif app.Q_3 ~= 0
    write(app.Q_3,0);
    pause(0.1);

elseif app.Q_4 ~= 0
    write(app.Q_4,0);
    pause(0.1);

elseif app.Figura ~= ' '
    write(app.Figura,'');
    pause(0.1);
end

end

function TickVerde(app, opcion)
    if opcion == 1
        app.Tick_Verde.Enable = 'off';
        app.Tick_Verde.Visible = 'off';
        pause(0.2)
        app.Tick_Verde.Enable = 'on';
        app.Tick_Verde.Visible = 'on';
    else
        app.Tick_Verde.Enable = 'off';
        app.Tick_Verde.Visible = 'off';
    end
end

function TickRojo(app, opcion)
    if opcion == 1
        app.Tick_Rojo.Enable = 'off';
        app.Tick_Rojo.Visible = 'off';
        pause(0.2)
        app.Tick_Rojo.Enable = 'on';
        app.Tick_Rojo.Visible = 'on';
    else
        app.Tick_Rojo.Enable = 'off';
        app.Tick_Rojo.Visible = 'off';
    end
end
```

```

    end
end

function TickLoading(app, opcion)
    if opcion == 1
        app.Tick_Loading.Enable = 'on';
        app.Tick_Loading.Visible = 'on';
    else
        app.Tick_Loading.Enable = 'off';
        app.Tick_Loading.Visible = 'off';
    end
end

function Leer_Ficheros(app)

    cd
    ruta = cd('C:\Users\JORGE\Documents\RobotStudio\Projects\TFG_JR\Virtual
Controllers\Controlador1\HOME');
    ficheros = ls(ruta)

    fichJR = fopen('Fichero_Sensores.txt');

    NF = 4951;          %Filas del FICHERO
    contador = 1;      %Contador para recorrer todas las filas del fichero
    fila = 1;          %Los múltiplos de 25 contienen la fila del rastreo donde
se encuentra

    filas = 198;        %Filas AVANCE DETECTOR
    columnas = 24;      %Numero de sensores
    Matriz = zeros(filas,columnas);
    Mascara = ones(filas, columnas);

    while contador<NF %Recorrer el fichero fila por fila

        aux = fgetl(fichJR);
        datos = str2num(aux)

        if contador/fila == 1

            j=25;        %j hace referencia al contador de las columnas
            i = datos    %i hace referencia a la fila donde se encuentra
            fila = fila + 25; %Necesitamos 1+24 digitos para realizar 1
avance

        else

            j = j - 1
            Matriz(i,j) = datos; %guardamos los datos en la matriz

        end

        contador = contador + 1;
    end
end

```



```
end

Matriz

%Recorro la matriz elemento a elemento para detectar los 1
for x=1:filas

    v = ['Fila: ', num2str(x)]

    for y=1:columnas

        if Mascara(x,y) .* Matriz(x,y) == 1
            disp('Yes');
            app.f3 = [app.f3,x];
            app.f4 = [app.f4,y];

        else
            disp('No')
            app.f1 = [app.f1,x];
            app.f2 = [app.f2,y];
        end

    end

    disp('-----')
end

length(app.f3)

end

function Calcular_Area(app)

    t = 'Controlador1.RAPID.T_ROB1.Module1.Tiempo';
    X = 'Controlador2.RAPID.T_ROB1.Module2.C_X';
    Y = 'Controlador2.RAPID.T_ROB1.Module2.C_Y';
    Z = 'Controlador2.RAPID.T_ROB1.Module2.C_Z';
    q1 = 'Controlador2.RAPID.T_ROB1.Module2.Q_1';
    q2 = 'Controlador2.RAPID.T_ROB1.Module2.Q_2';
    q3 = 'Controlador2.RAPID.T_ROB1.Module2.Q_3';
    q4 = 'Controlador2.RAPID.T_ROB1.Module2.Q_4';
    I = 'Controlador2.RAPID.T_ROB1.Module2.Inicio_CogerPieza';
    F = 'Controlador2.RAPID.T_ROB1.Module2.Fin_CogerPieza';
    Fig = 'Controlador2.RAPID.T_ROB1.Module2.Opcion_Figura';

    if app.Contador_CalcArea == 0

        app.grupo_Fig = addgroup(app.DA, 'Grupo_Figuras');
        set(app.grupo_Fig, 'UpdateRate', 0.001);

        app.Tiempo = additem(app.grupo_Fig, [app.C1, t]);
        app.C_X = additem(app.grupo_Fig, [app.C2, X]);
        app.C_Y = additem(app.grupo_Fig, [app.C2, Y]);
```

```
app.C_Z = additem(app.grupo_Fig, [app.C2, Z]);
app.Q_1 = additem(app.grupo_Fig,[app.C2, q1]);
app.Q_2 = additem(app.grupo_Fig,[app.C2, q2]);
app.Q_3 = additem(app.grupo_Fig,[app.C2, q3]);
app.Q_4 = additem(app.grupo_Fig,[app.C2, q4]);

app.Inicio_CogerPieza = additem(app.grupo_Fig, [app.C2, I]);
app.Fin_CogerPieza = additem(app.grupo_Fig, [app.C2, F]);
app.Figura = additem(app.grupo_Fig, [app.C2, Fig]);

end

serveritems(app.DA, [app.C1, 'Controlador1.RAPID.T_ROB1.Module1.Tiempo']);

tiempo_simulac = read(app.Tiempo);
pause(1);
disp(tiempo_simulac);
app.Tiempo.Value

%-----

fprintf('Maximo F3: ');
disp(max(app.f3));
fprintf(' ');
fprintf('Minimo F3: ');
disp(min(app.f3));

fprintf('Maximo F4: ');
disp(max(app.f4));
fprintf(' ');
fprintf('Minimo F4: ');
disp(min(app.f4));

%Distinguimos los 3 posibles casos ("AREA FIGURA")

if length(app.f3)>240 && length(app.f3)<320

    Opcion = 'Rectangulo';

elseif length(app.f3)>440 && length(app.f3)<510

    Opcion = 'Triangulo';

elseif length(app.f3)>580 && length(app.f3)<660

    Opcion = 'Pentagono';

end

plot(app.f2,app.f1,'o',12.5,100,'*');

%-----
```

```
%Calculamos el Ángulo de Giro
```

```
if app.Contador_Giro == 0
```

```
    if strcmp(Opcion, 'Triangulo') == 1
```

```
        ANG = -30;
```

```
    elseif strcmp(Opcion, 'Rectangulo') == 1
```

```
        ANG = 0;
```

```
    elseif strcmp(Opcion, 'Pentagono') == 1
```

```
        ANG = -36;
```

```
    end
```

```
elseif app.Contador_Giro > 0
```

```
    if strcmp(Opcion, 'Triangulo') == 1
```

```
        ANG = -30 - (app.Tiempo.Value*90)/3.1455 - ((0.022*90)/3.1455)/4
```

```
    elseif strcmp(Opcion, 'Rectangulo') == 1
```

```
        ANG = - (app.Tiempo.Value*90)/3.1455 - ((0.022*90)/3.1455)/4
```

```
    elseif strcmp(Opcion, 'Pentagono') == 1
```

```
        ANG = -36 - (app.Tiempo.Value*90)/3.1455 - ((0.022*90)/3.1455)/4
```

```
    end
```

```
end
```

```
%-----
```

```
%Calculamos los Cuaternios (Q1,Q2,Q3,Q4)
```

```
ANG_Rad = (ANG*pi)/180;
```

```
eul = [-ANG_Rad 0 pi];
```

```
Q = eul2quat(eul);    %ZYX
```

```
%-----
```

```
%Mostramos Resultados
```

```
switch Opcion
```

```
    case 'Triangulo'
```

```
        if app.Figura ~= ' '
```

```
            %Mostramos al usuario la informacion por pantalla
```

```
            app.Objeto.FontColor = app.Color_Negro;
```

```

        app.Objeto.Text = ' PRISMA TRIANGULAR.';
        app.Coordenadas.FontColor = app.Color_Negro;
        app.Coordenadas.Text = [' X : 0 mm', ' Y : 0 mm', ' Z : ↵
260 mm'];

        app.AnguloGiro.FontColor = app.Color_Negro;
        app.AnguloGiro.Text = [' ', num2str(round(-ANG-30,2,"↵
decimals")), ' °'];

        app.Cuaternios.FontColor = app.Color_Negro;
        app.Cuaternios.Text = [' Q1 : ', num2str(round(Q(1),2,"↵
decimals")), ' Q2 : ', num2str(round(Q(2),2,"decimals")), ' Q3 : ', num2str(round(Q↵
(3),2,"decimals")), ' Q4 : ', num2str(round(Q(4),2,"decimals"))];

        %Pasamos la informacion al Robot 2
        write(app.Figura, 'Triangulo');
        pause(0.1);

        write(app.Q_1, Q(1));
        write(app.Q_2, Q(2));
        write(app.Q_3, Q(3));
        write(app.Q_4, Q(4));
        pause(1);

        %Colocamos finalmente la pieza en el lugar correspondiente
        write(app.Inicio_CogerPieza, 1);
        pause(0.1);

    end

    case 'Rectangulo'

        if app.Figura ~= ' '

            %Mostramos al usuario la informacion por pantalla
            app.Objeto.FontColor = app.Color_Negro;
            app.Objeto.Text = ' PRISMA RECTANGULAR.';
            app.Coordenadas.FontColor = app.Color_Negro;
            app.Coordenadas.Text = [' X : 0 mm', ' Y : 0 mm', ' Z : ↵
260 mm'];

            app.AnguloGiro.FontColor = app.Color_Negro;
            app.AnguloGiro.Text = [' ', num2str(round(-ANG,2,"decimals")), ↵
' °'];

            app.Cuaternios.FontColor = app.Color_Negro;
            app.Cuaternios.Text = [' Q1 : ', num2str(round(Q(1),2,"↵
decimals")), ' Q2 : ', num2str(round(Q(2),2,"decimals")), ' Q3 : ', num2str(round(Q↵
(3),2,"decimals")), ' Q4 : ', num2str(round(Q(4),2,"decimals"))];

            %Pasamos la informacion al Robot 2
            write(app.Figura, 'Rectangulo');
            pause(0.1);

            write(app.Q_1, Q(1));
            write(app.Q_2, Q(2));
            write(app.Q_3, Q(3));
            write(app.Q_4, Q(4));
            pause(1);

```

```

        %Colocamos finalmente la pieza en el lugar correspondiente
        write(app.Inicio_CogerPieza, 1);
        pause(0.1);

    end

    case 'Pentagono'

        if app.Figura ~= ' '

            %Mostramos al usuario la informacion por pantalla
            app.Objeto.FontColor = app.Color_Negro;
            app.Objeto.Text = ' PRISMA PENTAGONAL.';
            app.Coordenadas.FontColor = app.Color_Negro;
            app.Coordenadas.Text = [ ' X : 0 mm', ' Y : 0 mm', ' Z : ↙
260 mm'];

            app.AnguloGiro.FontColor = app.Color_Negro;
            app.AnguloGiro.Text = [ ' ', num2str(round(-ANG-36,2,"↙
decimals")), ' °'];

            app.Cuaternios.FontColor = app.Color_Negro;
            app.Cuaternios.Text = [ ' Q1 : ', num2str(round(Q(1),2,"↙
decimals")), ' Q2 : ', num2str(round(Q(2),2,"decimals")), ' Q3 : ', num2str(round(Q(
3),2,"decimals")), ' Q4 : ', num2str(round(Q(4),2,"decimals"))];

            %Pasamos la informacion al Robot 2
            write(app.Figura, 'Pentagono');
            pause(0.1);

            write(app.Q_1, Q(1));
            write(app.Q_2, Q(2));
            write(app.Q_3, Q(3));
            write(app.Q_4, Q(4));
            pause(1);

            %Colocamos finalmente la pieza en el lugar correspondiente
            write(app.Inicio_CogerPieza, 1);
            pause(0.1);

        end

    end

    app.Contador_CalcArea = app.Contador_CalcArea + 1;
    app.f3 = 0;
    app.f4 = 0;
    app.f1 = 0;
    app.f2 = 0;

end

end

% Callbacks that handle component events

```

```

methods (Access = private)

% Value changed function: GENERARPIEZAALERTORIASwitch
function GENERARPIEZAALERTORIASwitchValueChanged(app, event)
    value_G = app.GENERARPIEZAALERTORIASwitch.Value;

    di25 = 'Controlador1.IOSYSTEM.IOSIGNALS.di25';
    di26 = 'Controlador1.IOSYSTEM.IOSIGNALS.di26';

    if strcmp(value_G, 'On')
        if app.Contador_Piezas == 0
            app.grupo_G = addgroup(app.DA, 'Grupo_GeneradorPieza');
            set(app.grupo_G, 'UpdateRate', 0.001);
            app.Switch_Generador_Piezas_On = additem(app.grupo_G, [app.C1, ↵
di25]);

            Generar_Pieza_Aleatoria(app);
        else
            Generar_Pieza_Aleatoria(app);
        end

    else %(value_G, 'Off')
        if app.Contador_Reset_Piezas == 0
            app.Switch_Generador_Piezas_Off = additem(app.grupo_G, [app.C1, ↵
di26]);

            Reset_Pieza_Aleatoria(app);
        else
            Reset_Pieza_Aleatoria(app);
        end
    end

end

% Value changed function: GIRARMESASwitch
function GIRARMESASwitchValueChanged(app, event)
    value_M = app.GIRARMESASwitch.Value;

    di27 = 'Controlador1.IOSYSTEM.IOSIGNALS.di27';
    di28 = 'Controlador1.IOSYSTEM.IOSIGNALS.di28';

    if strcmp(value_M, 'On')
        if app.Contador_Giro_Mesa == 0
            app.grupo_M = addgroup(app.DA, 'Grupo_GirarMesa');
            set(app.grupo_M, 'UpdateRate', 0.001);
            app.Switch_Mesa_Giratoria_1_On = additem(app.grupo_M, [app.C1, ↵
di27]);

            Girar_Mesa_1(app);
        else
            Girar_Mesa_1(app);
        end

    else %(value_M, 'Off')
        if app.Contador_Paro_Mesa == 0
            app.Switch_Mesa_Giratoria_1_Off = additem(app.grupo_M, [app.C1, ↵
di28]);

            Parar_Mesa_1(app);

```

```
        else
            Parar_Mesa_1(app);
        end
    end

end

% Value changed function: CONECTARSwitch
function CONECTARSwitchValueChanged(app, event)
    value_C = app.CONECTARSwitch.Value;

    if strcmp(value_C, 'On')
        Conectar(app);
    else %(value_C, 'Off')
        Desconectar(app);
    end

end

% Button pushed function: IniciarRastreo
function IniciarRastreoButtonPushed(app, event)

    di29 = 'Controlador1.IOSYSTEM.IOSIGNALS.di29';

    app.Objeto.Text = ' ';
    app.AnguloGiro.Text = ' ';
    app.Cuaternios.Text = ' ';

    if app.Valor_Pieza == 0

        write(app.Ejecucion_Rastreo, 0);
        app.Observaciones.FontColor = app.Color_Rojo;
        app.Observaciones.Text = ' DEBE HABERSE GENERADO UNA PIEZA ↙
PREVIAMENTE.';
        app.TickVerde(0);
        app.TickLoading(0);
        app.TickRojo(1);

    elseif app.Valor_Mesa == 1

        write(app.Ejecucion_Rastreo, 0);
        app.Observaciones.FontColor = app.Color_Rojo;
        app.Observaciones.Text = ' LA MESA DEBE ESTAR PARADA PREVIAMENTE.';
        app.TickVerde(0);
        app.TickLoading(0);
        app.TickRojo(1);

    else

        if app.Contador_Rastreo == 0
            app.grupo_R = addgroup(app.DA, 'Grupo_Rastreo');
            set(app.grupo_R, 'UpdateRate', 0.001);
            app.Button_Rastreo = additem(app.grupo_R, [app.C1, di29]);
        end
    end
end
```

```
write(app.Button_Rastreo, 1);
pause(1);    %MUY IMPORTANTE

while app.Ejecucion_Rastreo.value == 1
    app.Observaciones.FontColor = app.Color_Negro;
    app.Observaciones.Text = ' REALIZANDO RASTREO...';
    app.TickVerde(0);
    app.TickRojo(0);
    app.TickLoading(1);

    disp('Realizando Rastreo...');
    pause(0.5);
end

write(app.Button_Rastreo, 0);
pause(0.1);

app.Observaciones.FontColor = app.Color_Negro;
app.Observaciones.Text = ' RASTREO REALIZADO CORRECTAMENTE.';
app.TickRojo(0);
app.TickLoading(0);
app.TickVerde(1);

app.Contador_Rastreo = app.Contador_Rastreo + 1;

end
end

% Button pushed function: MostrarResultados
function MostrarResultadosButtonPushed(app, event)

    if app.Rastreo_OK.value == 0 && app.Ejecucion_Rastreo.value == 1

        app.Observaciones.FontColor = app.Color_Rojo;
        app.Observaciones.Text = ' DEBE ESPERAR A QUE FINALICE EL PROCESO DE RASTREO.';

        app.TickVerde(0);
        app.TickLoading(0);
        app.TickRojo(1);
        pause(2);

    elseif app.Rastreo_OK.value == 0 && app.Ejecucion_Rastreo.value == 0

        Leer_Ficheros(app);

        app.Observaciones.FontColor = app.Color_Negro;
        app.Observaciones.Text = ' FICHEROS LEÍDOS CORRECTAMENTE.';
        app.TickRojo(0);
        app.TickLoading(0);
        app.TickVerde(1);

        Calcular_Area(app);
```



```

        app.Contador_MR = app.Contador_MR + 1;
        app.Contador_Giro = 0;

    end

end

end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Color = [0.651 0.651 0.651];
        app.UIFigure.Position = [100 100 640 480];
        app.UIFigure.Name = 'MATLAB App';

        % Create GIRARMESASwitchLabel
        app.GIRARMESASwitchLabel = uilabel(app.UIFigure);
        app.GIRARMESASwitchLabel.BackgroundColor = [0.651 0.651 0.651];
        app.GIRARMESASwitchLabel.HorizontalAlignment = 'center';
        app.GIRARMESASwitchLabel.FontWeight = 'bold';
        app.GIRARMESASwitchLabel.Position = [487 360 82 22];
        app.GIRARMESASwitchLabel.Text = 'GIRAR MESA';

        % Create GIRARMESASwitch
        app.GIRARMESASwitch = uiswitch(app.UIFigure, 'slider');
        app.GIRARMESASwitch.ValueChangedFcn = createCallbackFcn(app, ↵
@GIRARMESASwitchValueChanged, true);
        app.GIRARMESASwitch.Position = [504 397 45 20];

        % Create GENERARPIEZAALAEATORIASwitchLabel
        app.GENERARPIEZAALAEATORIASwitchLabel = uilabel(app.UIFigure);
        app.GENERARPIEZAALAEATORIASwitchLabel.BackgroundColor = [0.651 0.651 ↵
0.651];
        app.GENERARPIEZAALAEATORIASwitchLabel.HorizontalAlignment = 'center';
        app.GENERARPIEZAALAEATORIASwitchLabel.FontWeight = 'bold';
        app.GENERARPIEZAALAEATORIASwitchLabel.Position = [251 360 176 22];
        app.GENERARPIEZAALAEATORIASwitchLabel.Text = 'GENERAR PIEZA ALEATORIA';

        % Create GENERARPIEZAALAEATORIASwitch
        app.GENERARPIEZAALAEATORIASwitch = uiswitch(app.UIFigure, 'slider');
        app.GENERARPIEZAALAEATORIASwitch.ValueChangedFcn = createCallbackFcn(app, ↵
@GENERARPIEZAALAEATORIASwitchValueChanged, true);
        app.GENERARPIEZAALAEATORIASwitch.Position = [311 397 45 20];

        % Create Lamp_Generador
        app.Lamp_Generador = uilamp(app.UIFigure);
        app.Lamp_Generador.Position = [324 426 20 20];
        app.Lamp_Generador.Color = [1 0 0];

        % Create Lamp_Mesa

```

```
app.Lamp_Mesa = uilamp(app.UIFigure);
app.Lamp_Mesa.Position = [517 426 20 20];
app.Lamp_Mesa.Color = [1 0 0];

% Create CONECTARSwitchLabel
app.CONECTARSwitchLabel = uilabel(app.UIFigure);
app.CONECTARSwitchLabel.BackgroundColor = [0.651 0.651 0.651];
app.CONECTARSwitchLabel.HorizontalAlignment = 'center';
app.CONECTARSwitchLabel.FontWeight = 'bold';
app.CONECTARSwitchLabel.Position = [68 360 73 22];
app.CONECTARSwitchLabel.Text = 'CONECTAR';

% Create CONECTARSwitch
app.CONECTARSwitch = uiswitch(app.UIFigure, 'slider');
app.CONECTARSwitch.ValueChangedFcn = createCallbackFcn(app, ↵
@CONECTARSwitchValueChanged, true);
app.CONECTARSwitch.Position = [80 397 45 20];

% Create Lamp_Conexion
app.Lamp_Conexion = uilamp(app.UIFigure);
app.Lamp_Conexion.Position = [92 426 20 20];
app.Lamp_Conexion.Color = [1 0 0];

% Create MargenAux1
app.MargenAux1 = uitextarea(app.UIFigure);
app.MargenAux1.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenAux1.Position = [1 322 640 20];

% Create MargenAux2
app.MargenAux2 = uitextarea(app.UIFigure);
app.MargenAux2.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenAux2.Position = [190 341 20 121];

% Create MargenInferior
app.MargenInferior = uitextarea(app.UIFigure);
app.MargenInferior.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenInferior.Position = [1 1 640 20];

% Create MargenSuperior
app.MargenSuperior = uitextarea(app.UIFigure);
app.MargenSuperior.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenSuperior.Position = [1 461 640 20];

% Create MargenIzquierdo
app.MargenIzquierdo = uitextarea(app.UIFigure);
app.MargenIzquierdo.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenIzquierdo.Position = [1 1 20 480];

% Create MargenDerecho
app.MargenDerecho = uitextarea(app.UIFigure);
app.MargenDerecho.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenDerecho.Position = [621 1 20 480];

% Create Observaciones
app.Observaciones = uilabel(app.UIFigure);
```

```
app.Observaciones.BackgroundColor = [1 1 1];
app.Observaciones.FontColor = [1 1 1];
app.Observaciones.Position = [175 279 374 22];
app.Observaciones.Text = '';

% Create OBSERVACIONESLabel
app.OBSERVACIONESLabel = uilabel(app.UIFigure);
app.OBSERVACIONESLabel.BackgroundColor = [1 1 0];
app.OBSERVACIONESLabel.FontWeight = 'bold';
app.OBSERVACIONESLabel.Position = [42 279 120 22];
app.OBSERVACIONESLabel.Text = ' OBSERVACIONES :';

% Create Tick_Verde
app.Tick_Verde = uiimage(app.UIFigure);
app.Tick_Verde.Enable = 'off';
app.Tick_Verde.Visible = 'off';
app.Tick_Verde.Position = [568 277 22 26];
app.Tick_Verde.ImageSource = 'tick_verde.png';

% Create Tick_Rojo
app.Tick_Rojo = uiimage(app.UIFigure);
app.Tick_Rojo.Enable = 'off';
app.Tick_Rojo.Visible = 'off';
app.Tick_Rojo.Position = [566 279 25 23];
app.Tick_Rojo.ImageSource = 'tick_rojo.png';

% Create Tick_Loading
app.Tick_Loading = uiimage(app.UIFigure);
app.Tick_Loading.Enable = 'off';
app.Tick_Loading.Visible = 'off';
app.Tick_Loading.Position = [563 277 31 27];
app.Tick_Loading.ImageSource = 'tick_loading.png';

% Create IniciarRastreo
app.IniciarRastreo = uibutton(app.UIFigure, 'push');
app.IniciarRastreo.ButtonPushedFcn = createCallbackFcn(app, ↵
@IniciarRastreoButtonPushed, true);
app.IniciarRastreo.BackgroundColor = [0.9608 0.9608 0.9608];
app.IniciarRastreo.Position = [64 175 74 48];
app.IniciarRastreo.Text = {' INICIAR '; 'RASTREO'};

% Create MostrarResultados
app.MostrarResultados = uibutton(app.UIFigure, 'push');
app.MostrarResultados.ButtonPushedFcn = createCallbackFcn(app, ↵
@MostrarResultadosButtonPushed, true);
app.MostrarResultados.Position = [54 105 100 48];
app.MostrarResultados.Text = {'MOSTRAR'; 'RESULTADOS'};

% Create OBJETOLabel
app.OBJETOLabel = uilabel(app.UIFigure);
app.OBJETOLabel.FontWeight = 'bold';
app.OBJETOLabel.Position = [241 188 64 23];
app.OBJETOLabel.Text = ' OBJETO :';

% Create COORDENADASLabel
```

```
app.COORDENADASLabel = uilabel(app.UIFigure);
app.COORDENADASLabel.FontWeight = 'bold';
app.COORDENADASLabel.Position = [242 141 108 23];
app.COORDENADASLabel.Text = 'COORDENADAS :';

% Create Objeto
app.Objeto = uilabel(app.UIFigure);
app.Objeto.BackgroundColor = [1 1 1];
app.Objeto.Position = [361 188 225 22];
app.Objeto.Text = '';

% Create Coordenadas
app.Coordenadas = uilabel(app.UIFigure);
app.Coordenadas.BackgroundColor = [1 1 1];
app.Coordenadas.Position = [361 141 225 22];
app.Coordenadas.Text = '';

% Create ANGULOGIROLabel
app.ANGULOGIROLabel = uilabel(app.UIFigure);
app.ANGULOGIROLabel.FontWeight = 'bold';
app.ANGULOGIROLabel.Position = [243 97 99 22];
app.ANGULOGIROLabel.Text = 'ANGULO GIRO :';

% Create CUATERNIOSLabel
app.CUATERNIOSLabel = uilabel(app.UIFigure);
app.CUATERNIOSLabel.FontWeight = 'bold';
app.CUATERNIOSLabel.Position = [244 52 91 22];
app.CUATERNIOSLabel.Text = 'CUATERNIOS : ';

% Create MargenAux3
app.MargenAux3 = uieditfield(app.UIFigure, 'text');
app.MargenAux3.FontColor = [0.9294 0.6941 0.1255];
app.MargenAux3.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenAux3.Position = [21 238 602 22];

% Create MargenAux4
app.MargenAux4 = uieditfield(app.UIFigure, 'text');
app.MargenAux4.FontColor = [0.9294 0.6941 0.1255];
app.MargenAux4.BackgroundColor = [0.9294 0.6941 0.1255];
app.MargenAux4.Position = [190 20 21 219];

% Create AnguloGiro
app.AnguloGiro = uilabel(app.UIFigure);
app.AnguloGiro.BackgroundColor = [1 1 1];
app.AnguloGiro.Position = [361 97 225 22];
app.AnguloGiro.Text = '';

% Create Cuaternios
app.Cuaternios = uilabel(app.UIFigure);
app.Cuaternios.BackgroundColor = [1 1 1];
app.Cuaternios.Position = [361 52 225 22];
app.Cuaternios.Text = '';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
```

```
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = AppTFG_JR_exported

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargin == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```