

## Índice

Anexo I: Programación.....	1
1. Real Time Application.....	1
1.1. Funciones. ....	1
1.2. Preparación. ....	2
1.3. Arranque.....	2
1.4. Funcionamiento. ....	3
1.5. Parada.....	5
2. Node-Red.....	6
2.1. Enviar datos a Simulink. ....	6
2.2. Recibir datos de Simulink. ....	7
2.3. Guardar datos.....	8
2.4. Cálculo de la presión. ....	8
2.5. Envío de datos a Azure. ....	9
3. Interfaz Gráfica de Usuario. ....	9

# Anexo I: Programación.

Este anexo muestra el programa realizado en la RTA, Node-Red e Interfaz Gráfica de Usuario de Matlab.

## 1. Real Time Application.

### 1.1. Funciones.

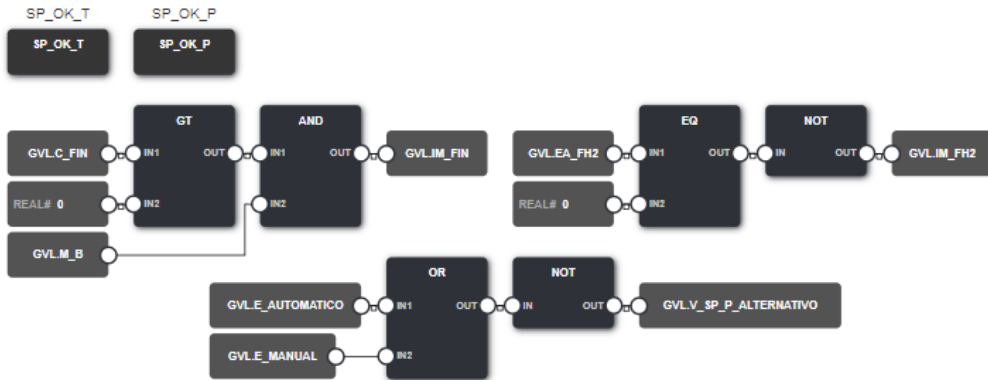


Ilustración 1: Programa de la función "Funciones".

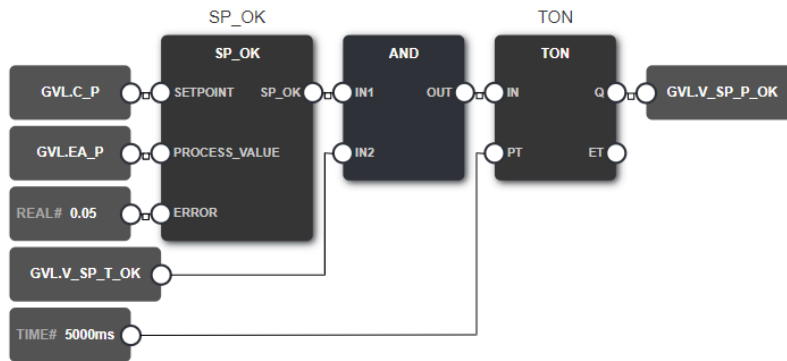


Ilustración 2: Programa de la función "SP\_OK\_T".

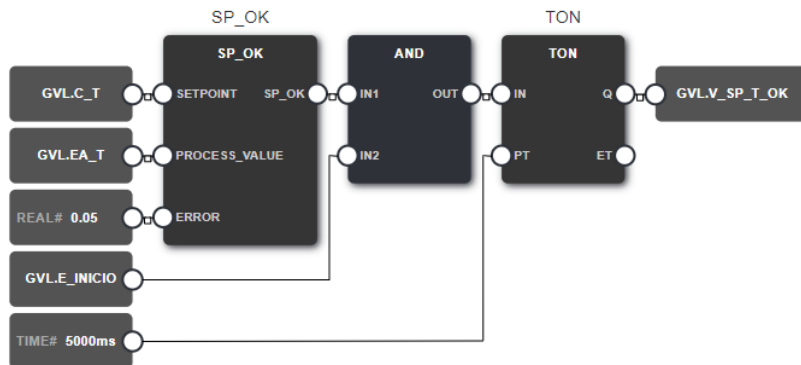


Ilustración 3: Programa de la función "SP\_OK\_P".

### 1.2. Preparación.

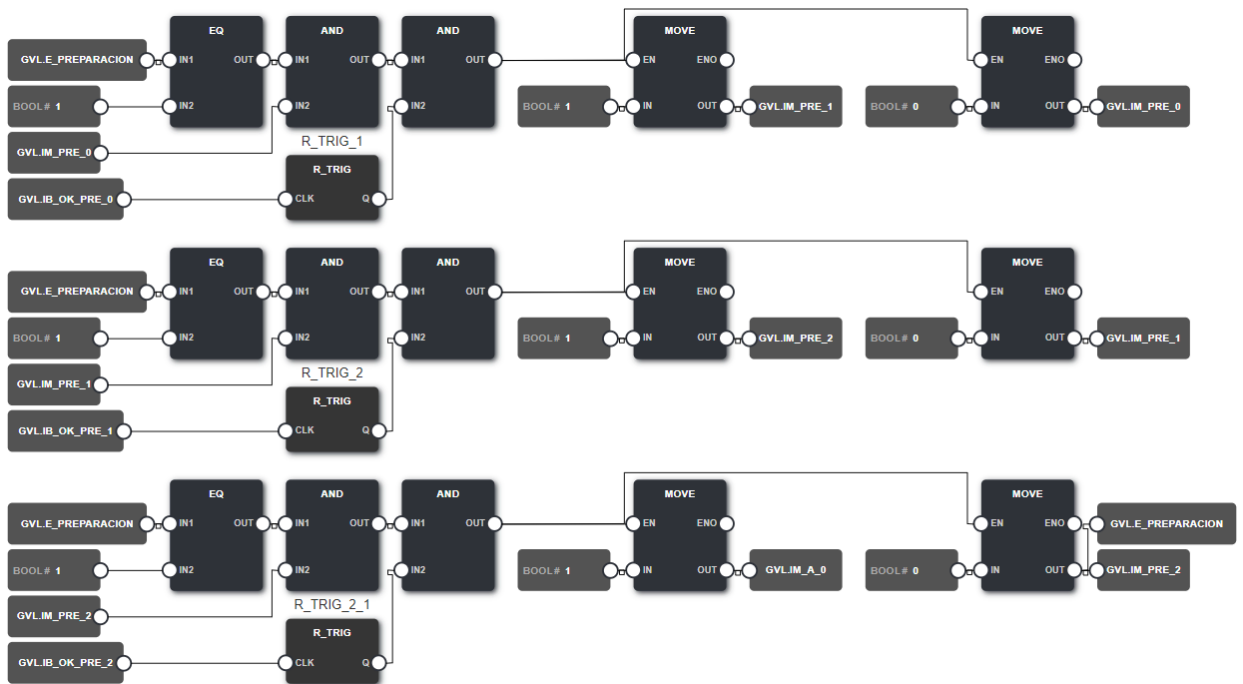


Ilustración 4: Programa de la función "PREPARACIÓN".

### 1.3. Arranque.



Ilustración 5: Programa de la función "ARRANQUE".

### 1.4. Funcionamiento.



Ilustración 6: Programa de función "FUNCIONAMIENTO" (parte 1).



Ilustración 7: Programa de función "FUNCIONAMIENTO" (parte 2).

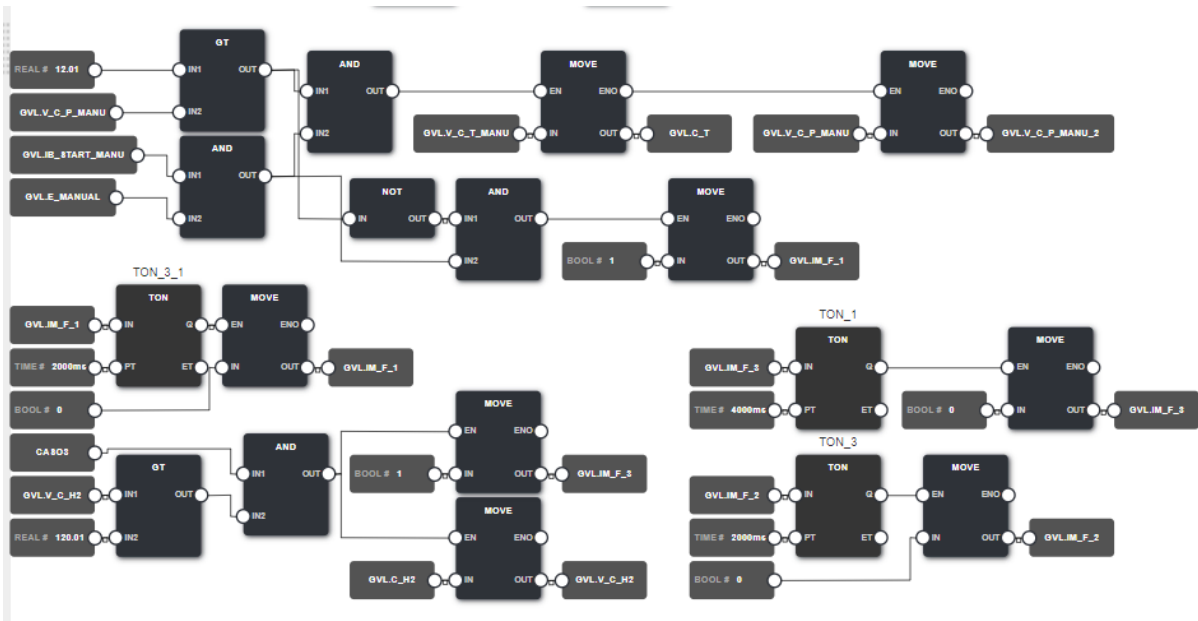


Ilustración 8: Programa de función "FUNCIONAMIENTO" (parte 3).

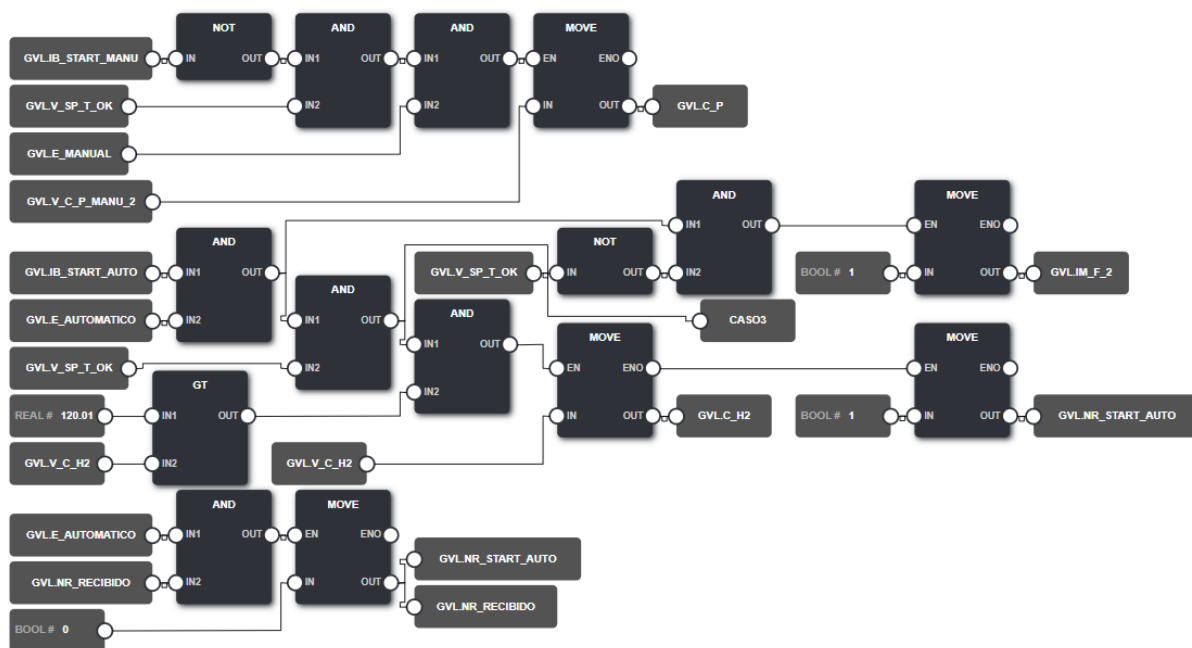


Ilustración 9: Programa de función "FUNCIONAMIENTO" (parte 4).

### 1.5. Parada.

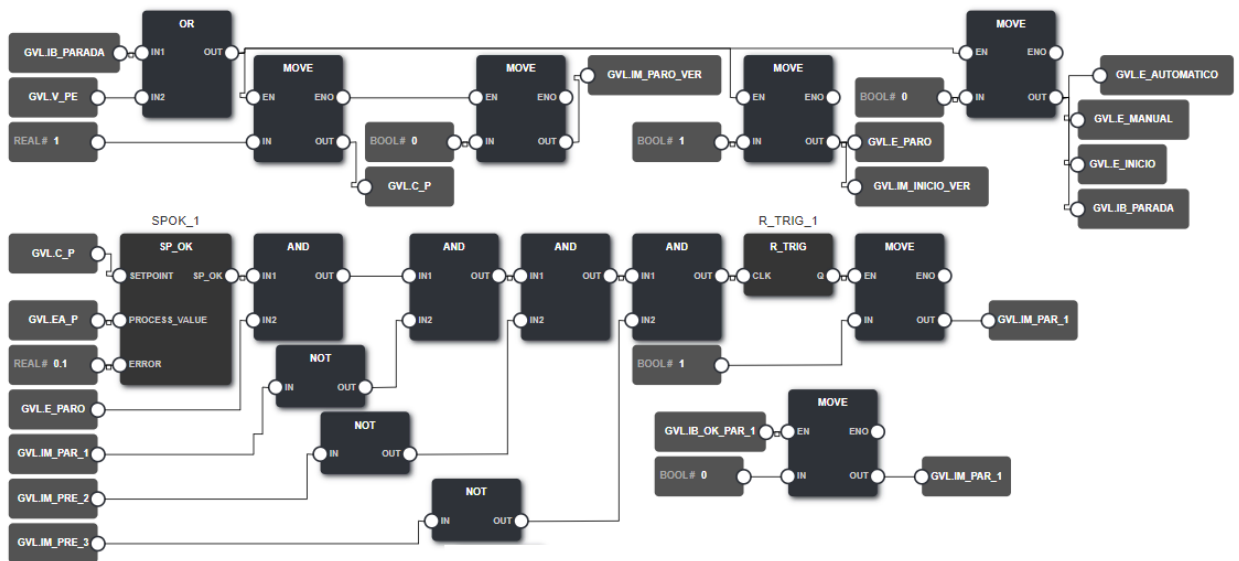


Ilustración 10: Programa de función "PARADA" (parte 1).

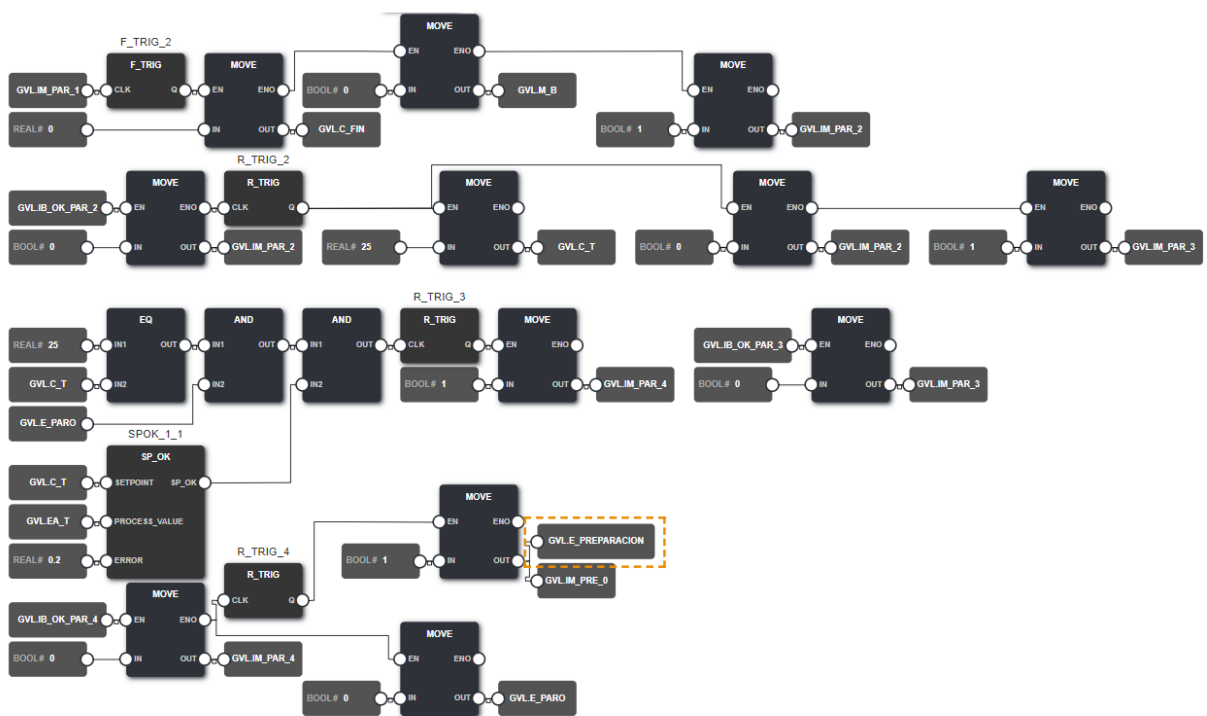
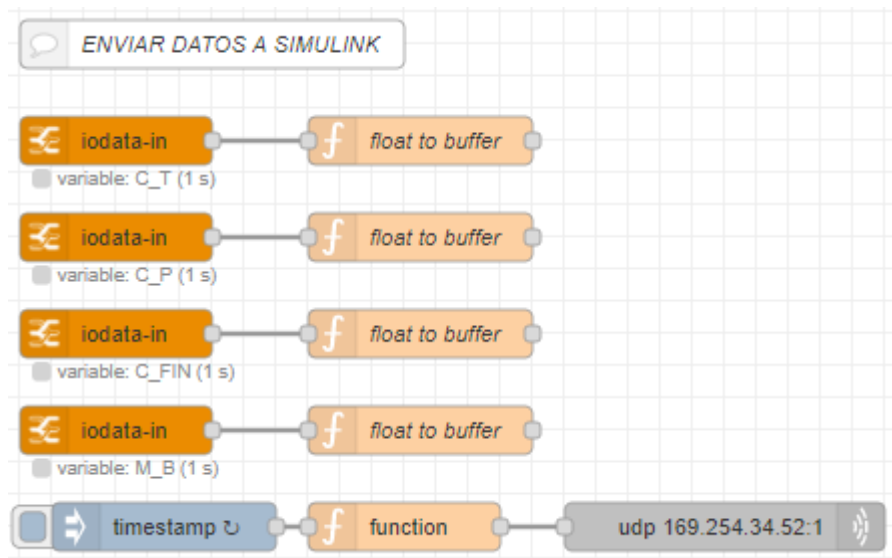


Ilustración 11: Programa de función "PARADA" (parte 2).

## 2. Node-Red.

### 2.1. Enviar datos a Simulink.



Il·lustració 12: Programa de envió de dades a Simulink.

## 2.2. Recibir datos de Simulink.

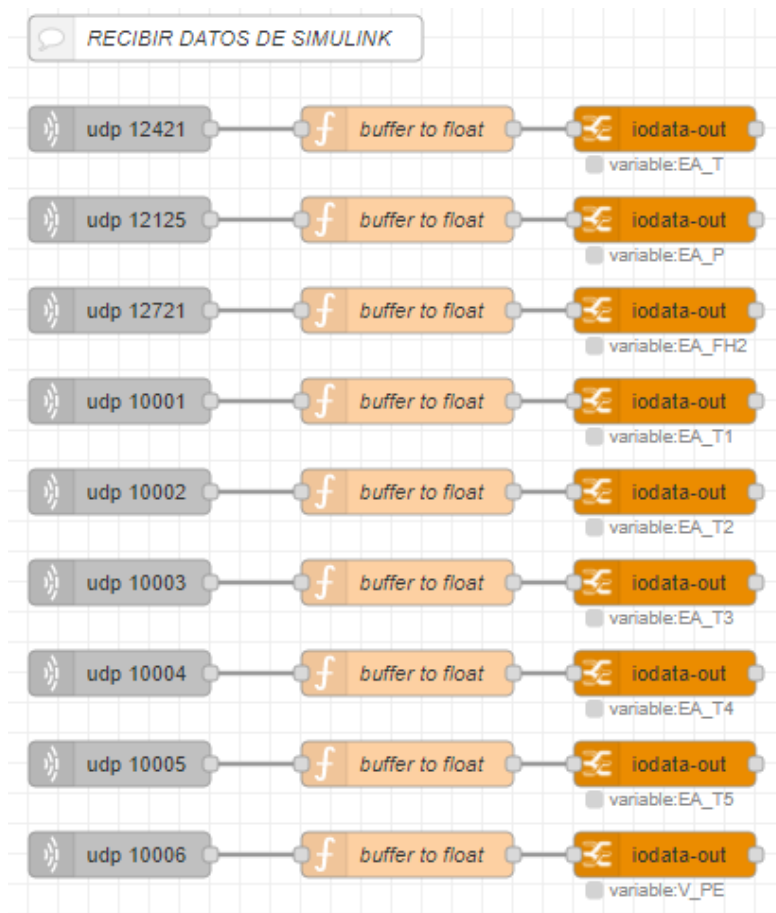


Ilustración 13: Programa de recepción de datos de Simulink.



## 2.3. Guardar datos.

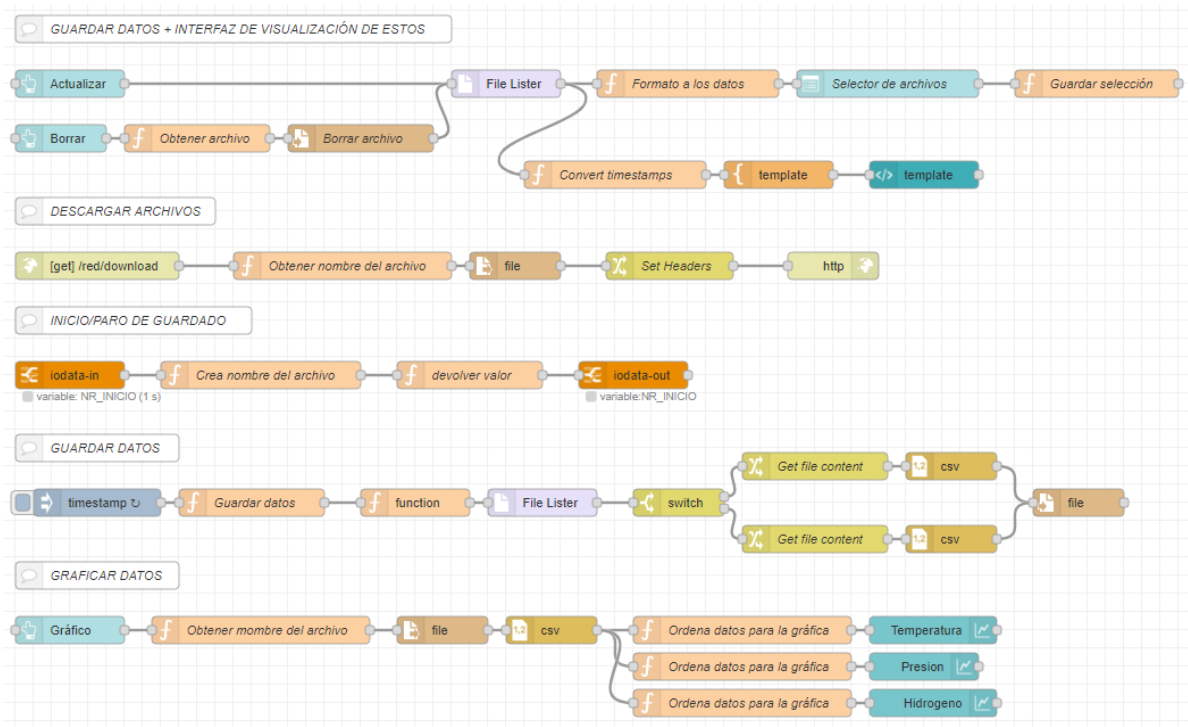


Ilustración 14: Programa de guardado de datos y creación de interfaz para visualizarlos.

## 2.4. Cálculo de la presión.

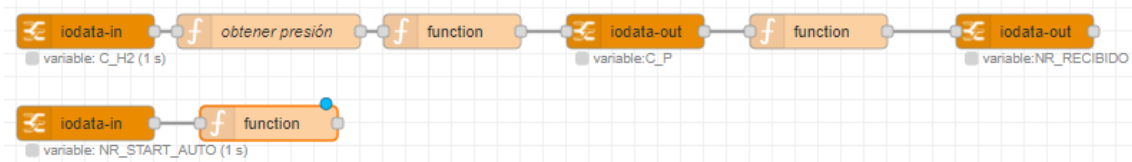


Ilustración 15: Programa de cálculo de presión en modo automático.

## 2.5. Envío de datos a Azure.

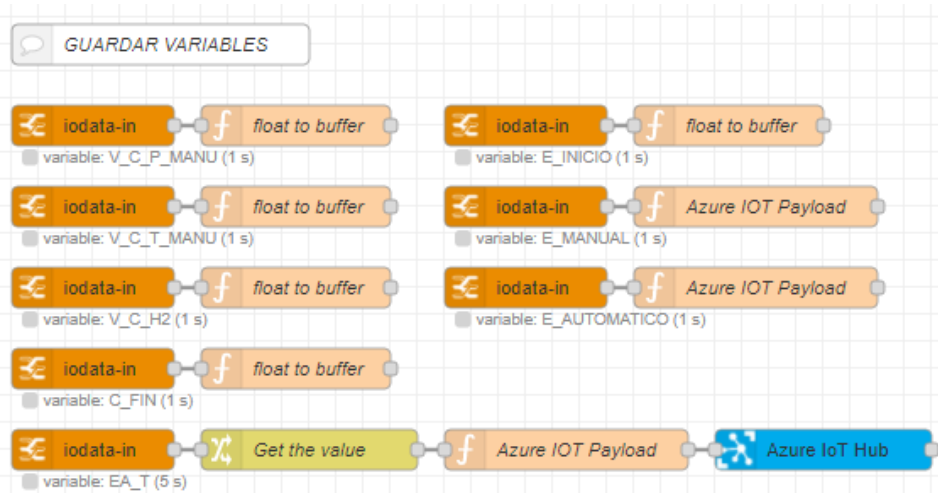


Ilustración 16: Programa de envío de datos al Centro de IoT.

## 3. Interfaz Gráfica de Usuario.

```

1  function varargout = guide_interfaz_tfg(varargin)
2  -   gui_Singleton = 1;
3  -   gui_State = struct('gui_Name',       mfilename, ...
4  -                   'gui_Singleton',   gui_Singleton, ...
5  -                   'gui_OpeningFcn', @guide_interfaz_tfg_OpeningFcn, ...
6  -                   'gui_OutputFcn',  @guide_interfaz_tfg_OutputFcn, ...
7  -                   'gui_LayoutFcn',  [], ...
8  -                   'gui_Callback',    []);
9  -   if nargin && ischar(varargin{1})
10 -     gui_State.gui_Callback = str2func(varargin{1});
11 -   end
12
13   if nargin
14     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
15   else
16     gui_mainfcn(gui_State, varargin{:});
17   end
18
19  function guide_interfaz_tfg_OpeningFcn(hObject, eventdata, handles, varargin)
20  -   %Se cargan las imagenes que muestran la planta:
21  -   axes(handles.axes2);
22  -   [x,map] = imread('Reformador.jpg');
23  -   image(x);
24  -   colormap(map);
25  -   axis off
26  -   hold on
27
28  -   axes(handles.axes3);
29  -   [x,map] = imread('Parte_1.jpg');
30  -   image(x);
31  -   colormap(map);
32  -   axis off
33  -   hold on
34
35  -   axes(handles.axes4);
36  -   [x,map] = imread('Parte_2.jpg');
37  -   image(x);

```

Ilustración 17: Programa de la GUI de Matlab (parte 1).

```

37 - colormap(map);
38 - axis off
39 - hold on
40
41 - axes(handles.axes5);
42 - [x,map] = imread('parte_3.jpg');
43 - image(x);
44 - colormap(map);
45 - axis off
46 - hold on
47
48 - axes(handles.axes6);
49 - [x,map] = imread('parte_4.jpg');
50 - image(x);
51 - colormap(map);
52 - axis off
53 - hold on
54
55 - axes(handles.axes7);
56 - [x,map] = imread('parte_5.jpg');
57 - image(x);
58 - colormap(map);
59 - axis off
60 - hold on
61 - guidata(hObject, handles);
62 - function varargout = guide_interfaz_tfg_OutputFcn(hObject, eventdata, handles)
63 - function boton_vh2_Callback(hObject, eventdata, handles)
64 - %función que se ejecutra al pulsar la valvula de H2.
65 - if get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant','Value') == '0'
66 -     set(handles.boton_vh2,'String','CERRAR VH2');
67 -     set_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant','Value','1');
68 -     axes(handles.axes11);
69 -     [x,map] = imread('verde.jpg');
70 -     image(x);
71 -     colormap(map);
72 -     axis off

```

*Ilustración 18: Programa de la GUI de Matlab (parte 2).*

```

73 -         hold on
74 -     elseif get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant','Value') == '1'
75 -         set(handles.boton_vh2,'String','ABRIR VH2');
76 -         set_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant','Value','0');
77 -         axes(handles.axes11);
78 -         [x,map] = imread('gris.jpg');
79 -         image(x);
80 -         colormap(map);
81 -         axis off
82 -         hold on
83 -     end
84 -     function boton_inicio_Callback(hObject, eventdata, handles)
85 -     %Función que se ejecuta al pulsar el botón Inicio
86 -     %cargar el archivo simulink
87 -     load_system('COMUNICACION_NODERED_SIMULINK_UDP_real');
88 -     find_system('Name','COMUNICACION_NODERED_SIMULINK_UDP_real');
89 -     %ejecutar el simulink
90 -     set_param('COMUNICACION_NODERED_SIMULINK_UDP_real','SimulationCommand','start');
91 -     %VALVULA V1 INICIO:
92 -     set(findobj('Tag','boton_v1'),'UserData','2');
93 -     %foto válvula v1 estado 2
94 -     axes(handles.axes9);
95 -     [x,map] = imread('Valvula2.jpg');
96 -     image(x);
97 -     colormap(map);
98 -     axis off
99 -     hold on
100 -     %VALVULA V2 INICIO:
101 -     set(findobj('Tag','boton_v2'),'UserData','2');
102 -     %foto válvula v2 estado 2
103 -     axes(handles.axes13);
104 -     [x,map] = imread('Valvula.jpg');
105 -     image(x);
106 -     colormap(map);
107 -     axis off
108 -     hold on

```

Ilustración 19: Programa de la GUI de Matlab (parte 3).

```

109 -     %VALVULA VH2 INICIO:
110 -     set_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant','Value','0');
111 -     set(handles.boton_vh2,'String','ABRIR VH2');
112 -     %foto válvula vh2 cerrada
113 -     axes(handles.axes11);
114 -     [x,map] = imread('gris.jpg');
115 -     image(x);
116 -     colormap(map);
117 -     axis off
118 -     hold on
119 -     %VALVULA Vd INICIO:
120 -     set(findobj('Tag','boton_vd'),'UserData','1');
121 -     %foto válvula vd_1 abierto
122 -     axes(handles.axes17);
123 -     [x,map] = imread('verde.jpg');
124 -     image(x);
125 -     colormap(map);
126 -     axis off
127 -     hold on
128 -     %foto válvula vd_2 abierto
129 -     axes(handles.axes18);
130 -     [x,map] = imread('verde.jpg');
131 -     image(x);
132 -     colormap(map);
133 -     axis off
134 -     hold on
135 -     while get(hObject,'Value')
136 -     %se cogen los valores del simulink cada segundo.
137 -     %temperatura
138 -     TEMPERATURA = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain','RuntimeObject').OutputPort(1).Data,2);
139 -     TE = findobj('Tag','Temperatura');
140 -     set(TE,'string',num2str(TEMPERATURA));
141 -     %presion
142 -     PRESION = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain1','RuntimeObject').OutputPort(1).Data,2);
143 -     PRE = findobj('Tag','Presion');
144 -     set(PRE,'string',num2str(PRESION));

```

Ilustración 20: Programa de la GUI de Matlab (parte 4).

```

145 %tc1
146 TC1 = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain2','RuntimeObject').OutputPort(1).Data,2);
147 TP1 = findobj('Tag','TC1');
148 set(TP1,'string',num2str(TC1));
149 %tc2
150 TC2 = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain4','RuntimeObject').OutputPort(1).Data,2);
151 TP2 = findobj('Tag','TC2');
152 set(TP2,'string',num2str(TC2));
153 %tc3
154 TC3 = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain5','RuntimeObject').OutputPort(1).Data,2);
155 TP3 = findobj('Tag','TC3');
156 set(TP3,'string',num2str(TC3));
157 %tc4
158 TC4 = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain6','RuntimeObject').OutputPort(1).Data,2);
159 TP4 = findobj('Tag','TC4');
160 set(TP4,'string',num2str(TC4));
161 %tc5
162 TC5 = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain7','RuntimeObject').OutputPort(1).Data,2);
163 TP5 = findobj('Tag','TC5');
164 set(TP5,'string',num2str(TC5));
165 HIDROGENO = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain3','RuntimeObject').OutputPort(1).Data,2);
166 H2 = findobj('Tag','H2');
167 set(H2,'string',num2str(HIDROGENO));
168 %caudal de entrada SP
169 CinSP = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain9','RuntimeObject').OutputPort(1).Data,2);
170 C_in_SP = findobj('Tag','CinSP');
171 set(C_in_SP,'string',num2str(CinSP));
172 pause(1);
173 %bomba on
174 bomba_on = round(get_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Gain10','RuntimeObject').OutputPort(1).Data,1);
175 B_ON = findobj('Tag','bomba_on');
176 if(bomba_on == 1)
177     set(B_ON,'string',"BOMBA ON");
178 else
179     set(B_ON,'string',"BOMBA OFF");
180 end

```

*Ilustración 21: Programa de la GUI de Matlab (parte 5).*

```

181 -     pause(1);
182 - end
183 - function boton_paro_Callback(hObject, eventdata, handles)
184 -     %funcion que se activa al pulsar el boton de paro
185 -     set_param('COMUNICACION_NODERED_SIMULINK_UDP_real','SimulationCommand','stop');
186 - function boton_vl_Callback(hObject, eventdata, handles)
187 -     %función que se activa al abrir la valvula vl.
188 -     if get(findobj('Tag','boton_vl'),'UserData') == '1'
189 -         set(findobj('Tag','boton_vl'),'UserData','2')
190 -
191 -         axes(handles.axes9);
192 -         [x,map] = imread('Valvula2.jpg');
193 -         image(x);
194 -         colormap(map);
195 -         axis off
196 -         hold on
197 -         set(handles.boton_vl,'String','Vl ESTADO 1');
198 -
199 -         axes(handles.axes8);
200 -         [x,map] = imread('nada.jpg');
201 -         image(x);
202 -         colormap(map);
203 -         axis off
204 -         hold on
205 -     elseif get(findobj('Tag','boton_vl'),'UserData') == '2'
206 -         set(findobj('Tag','boton_vl'),'UserData','1')
207 -         guidata(hObject,handles);
208 -         set(handles.boton_vl,'String','Vl ESTADO 2');
209 -
210 -         axes(handles.axes9);
211 -         [x,map] = imread('nada.jpg');
212 -         image(x);
213 -         colormap(map);
214 -         axis off
215 -         hold on

```

Il·lustració 22: Programa de la GUI de Matlab (parte 6).

```

217 - axes(handles.axes8);
218 - [x,map] = imread('Valvula.jpg');
219 - image(x);
220 - colormap(map);
221 - axis off
222 - hold on
223 - end
224
225 - function boton_v2_Callback(hObject, eventdata, handles)
226 - %función que se activa al abrir la válvula v2
227 - if get(findobj('Tag','boton_v2'),'UserData') == '1'
228 -     set(findobj('Tag','boton_v2'),'UserData','2')
229 -     axes(handles.axes13);
230 -     [x,map] = imread('Valvula.jpg');
231 -     image(x);
232 -     colormap(map);
233 -     axis off
234 -     hold on
235 -     set(handles.boton_v2,'String','V2 ESTADO 1');
236
237 -     axes(handles.axes12);
238 -     [x,map] = imread('nada.jpg');
239 -     image(x);
240 -     colormap(map);
241 -     axis off
242 -     hold on
243 - elseif get(findobj('Tag','boton_v2'),'UserData') == '2'
244 -     set(findobj('Tag','boton_v2'),'UserData','1')
245 -     guidata(hObject,handles);
246 -     set(handles.boton_v2,'String','V2 ESTADO 2');
247
248 -     axes(handles.axes13);
249 -     [x,map] = imread('nada.jpg');
250 -     image(x);
251 -     colormap(map);
252 -     axis off

```

Ilustración 23: Programa de la GUI de Matlab (parte 7).

```

253 -         hold on
254 -
255 -         axes(handles.axes12);
256 -         [x,map] = imread('Valvula2.jpg');
257 -         image(x);
258 -         colormap(map);
259 -         axis off
260 -         hold on
261 -     end
262 -
263 -
264 - function boton_vd_Callback(hObject, eventdata, handles)
265 - %función que se activa al abrir la valvula vd
266 - if get(findobj('Tag','boton_vd'),'UserData') == '1'
267 -     set(findobj('Tag','boton_vd'),'UserData','0')
268 -     set(handles.boton_vd,'String','ABRIR Vd');
269 -     axes(handles.axes17);
270 -     [x,map] = imread('gris.jpg');
271 -     image(x);
272 -     colormap(map);
273 -     axis off
274 -     hold on
275 -     axes(handles.axes18);
276 -     [x,map] = imread('gris.jpg');
277 -     image(x);
278 -     colormap(map);
279 -     axis off
280 -     hold on
281 - elseif get(findobj('Tag','boton_vd'),'UserData') == '0'
282 -     set(findobj('Tag','boton_vd'),'UserData','1')
283 -     set(handles.boton_vd,'String','CERRAR Vd');
284 -     %imagen tick.
285 -     axes(handles.axes17);
286 -     [x,map] = imread('verde.jpg');
287 -     image(x);
288 -     colormap(map);

```

Ilustración 24: Programa de la GUI de Matlab (parte 8).

```

289 -         axis off
290 -         hold on
291 -         axes(handles.axes18);
292 -         [x,map] = imread('verde.jpg');
293 -         image(x);
294 -         colormap(map);
295 -         axis off
296 -         hold on
297 -     end
298 -
299 -
300 - function boton_pe_Callback(hObject, eventdata, handles)
301 - %funcion que se activa al pulsar el boton paro de emergencia
302 - set_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant3','Value','1');
303 - pause(4);
304 - set_param('COMUNICACION_NODERED_SIMULINK_UDP_real/Constant3','Value','0');

```

Ilustración 25: Programa de la GUI de Matlab (parte 9).