

Jesús Reyes C.I 24.152.665
Maribal Zambrano C.I 24.819.737
Loreana Zambrano C.I 24.356.109

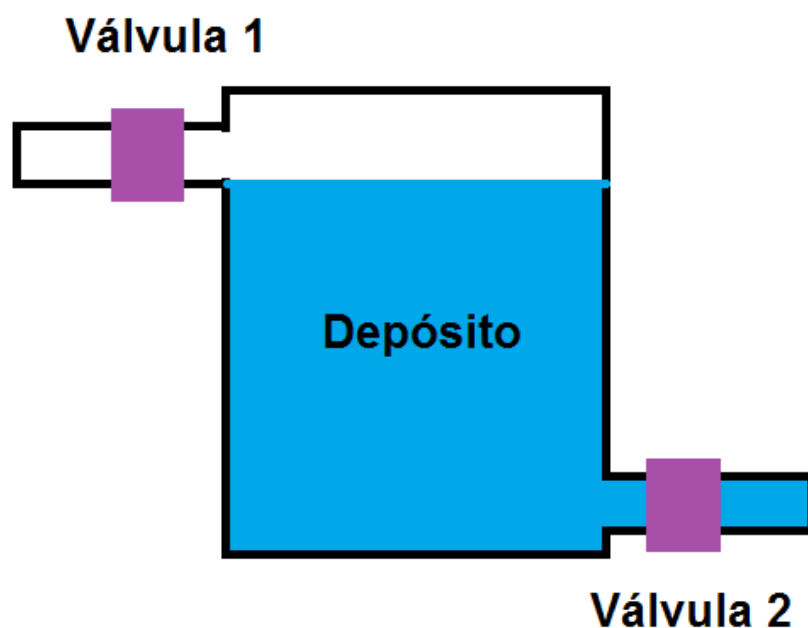
Informe Técnico del Proyecto

Descripción

Se pretende controlar el llenado y vaciado de un tanque de agua de capacidad finita utilizando para tal fin la teoría de autómatas. Se procedió en primera instancia a identificar las variables involucradas en el proceso y a asociarlas con un determinado símbolo que posee un rango de valores posible, con dichos símbolos construyó un lenguaje formal que representa el funcionamiento del sistema para luego implementarlo junto a un autómata finito determinista sobre la plataforma NodeMCU.

Identificación de variables

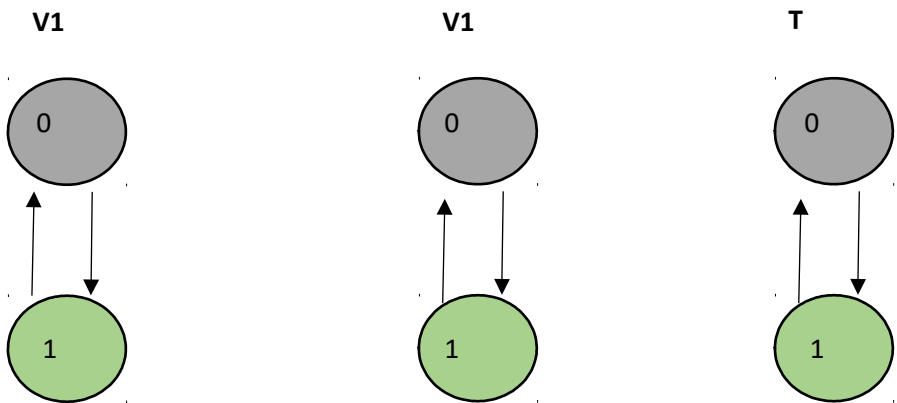
El sistema de llenado/vaciado del tanque de agua fue simplificado a la siguiente figura:



En este caso, la Válvula 1 permite el paso del agua hacia el depósito cuando se encuentra activa. Por otra parte, la Válvula 2 es la encargada de vaciar el tanque de agua. Teniendo en cuenta esta información, podemos extraer del sistema 3 símbolos desde 3 variables dicotómicas (2 estados posibles):

- V1 que hospeda el estado de la válvula 1 (abierta = 1, cerrada = 0).
- V2 que muestra el estado de la válvula 2 (abierta = 1, cerrada = 0).
- T que indica el estado del depósito de agua (vacío = 0, lleno = 1).

El siguiente diagrama describe el cambio de transición entre los estados de las variables antes mencionadas.



Lenguaje Formal

El lenguaje formal fue construido seleccionando las posibles combinaciones (palabras) válidas para nuestro sistema:

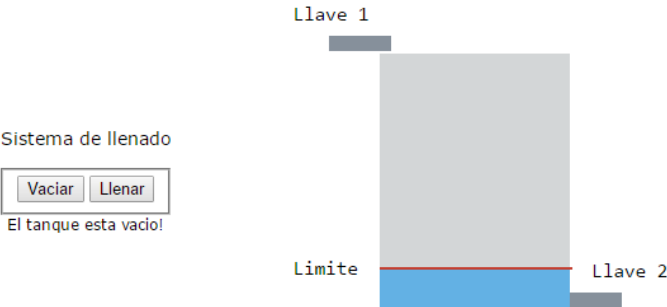
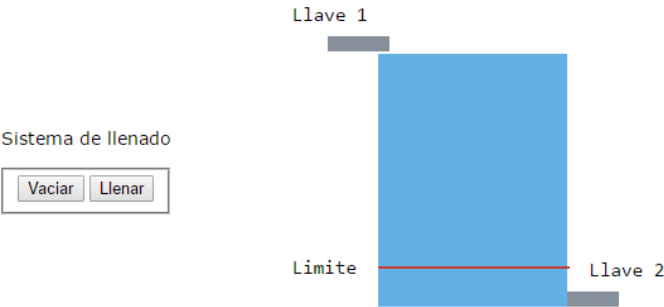
Combinaciones entre los posibles estados de los símbolos (Lenguaje generado)		
V1	V2	T
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

En la siguiente tabla se presenta el lenguaje marcado y el orden secuencial de las combinaciones (palabras) en el que se presentan los estados dentro del sistema

Combinaciones de estados válidas para el sistema (Lenguaje marcado)			
#	V1	V2	T
1	0	0	0
2	1	0	0
3	1	0	1
4	0	0	1
5	0	1	1
6	0	1	0

Implementación del autómata

Teniendo el orden secuencial de los posibles estados del autómata se procedió a implementarlo dentro de una aplicación web asentado sobre la plataforma NodeMCU. La aplicación web consta de 2 partes: la parte de control de las válvulas (llaves) compuesta por 2 botones para vaciar/llenar el depósito de agua y la parte de visualización del estado del tanque.



La página web está escrita en HTML + Javascript + AJAX y se despliega desde el NodeMCU creando un endpoint WiFi utilizando el siguiente código fuente escrito para ser compilado en Arduino IDE:

```
1  #include <ESP8266WiFi.h>
2  #include <WiFiClient.h>
3  #include <ESP8266WebServer.h>
4
5
6  const char *ssid = "ESPap";
7  const char *password = "picolinajlml";
8
9  ESP8266WebServer server(80);
10
11 /* Just a little test message. Go to http://192.168.4.1 in a web browser
12  * connected to this access point to see it.
13  */
14  String form;
15
16
17  int ledPin = 2;
18
19  void ini_vaciar(){
20    digitalWrite(D1, HIGH);
21    Serial.println("entrooooo");
22  }
23  void fin_vaciar(){
24    digitalWrite(ledPin, HIGH);
25    digitalWrite(D1, LOW);
26  }
27  void ini_llenar(){
28    digitalWrite(ledPin, LOW);
29    digitalWrite(D2, HIGH);
30  }
31  void fin_llenar(){
32    digitalWrite(D2, LOW);
33  }
34
35  void setup() {
36    delay(1000);
37    Serial.begin(9600);
38    Serial.println();
39    Serial.print("Configuring access point...");
40
41    /*-----*/
42    form="<!DOCTYPE html>";
43    form+="
```

En esta parte podemos apreciar que se crea un servidor web en el puerto 80 usando el objeto ESP8266WebServer. Luego se declaran las funciones encargadas de encender y apagar los LEDs físicos asociados a las válvulas que se conectaron al NodeMCU cuando se está llenando o vaciando el tanque. Por otra parte, se creó un String llamado **form** que es el encargado de alojar el código HTML de la página web y los scripts de AJAX que ejecuta cuando se presiona cada uno de los botones de la interfaz de usuario.

```

57
58 form+=" function functionProgress2(i,x) {";
59
60 form+=" barra1=document.getElementById('barra');";
61
62 form+=" if(x==1){";
63 form+="     msj=document.getElementById('mensaje');";
64 form+="     barra = document.getElementById('barra');";
65 form+="     l1 = document.getElementById('l11');";
66 form+="     l2 = document.getElementById('l12');";
67 form+="     max = 169;";
68 form+="     TIME = 80;";
69 form+="     if (barra1.style.height=='0px' || barra1.style.height=='') { ";
70 form+="         ban=0;";
71 form+="         vaciar_b=document.getElementById('v1');llenar_b=document.getElementById('l1');";
72 form+="         llenar_b.disabled = true;vaciar_b.disabled= true;";
73 form+="         msj.innerHTML='';";
74 form+="         Enviar('inivaciar');";
75 form+="     }else{";
76 form+="         ban=1;";
77 form+="         msj.innerHTML='El tanque esta vacio! ';;";
78
79 form+="     if (i > max){";
80 form+="         ban=1; llenar_b.disabled = false;vaciar_b.disabled=false;";
81 form+="         l2.setAttribute('style', 'height: 10px; background-color: #888896;');";
82 form+="         Enviar('finvaciar');";
83
84 form+="     if(ban==0){";
85 form+="         i=i+2;";
86 form+="         barra.setAttribute('style', 'height:'+i+'px; width: 150px; background: #080304;');";
87 form+="         l1.setAttribute('style', 'height: 10px; background-color: #888896;');";
88 form+="         l2.setAttribute('style', 'height: 10px; background-color: #CB4335;');";
89 form+="         setTimeout('functionProgress2(' + (i) + ',' + (0) + ')', TIME);}}";
90 form+=" function functionProgress1(i,x) {";
91
92 form+="     barra1=document.getElementById('barra');";
93 form+="     if(x==1){";
94 form+="         barra = document.getElementById('barra');";
95 form+="         msj=document.getElementById('mensaje');";
96 form+="         max = 0;";
97 form+="         TIME1 = 80;";
98 form+="         if (barra1.style.height=='170px') { ";
99 form+="             ban=0;";
100 form+="             vaciar_b=document.getElementById('v1');llenar_b=document.getElementById('l1');";
101 form+="             llenar_b.disabled = true;vaciar_b.disabled= true;";
102 form+="             msj.innerHTML='';";
103 form+="             Enviar('inllenar');";
104 form+="         }else{";
105 form+="             ban=1;";
106 form+="             msj.innerHTML='El tanque esta lleno! ';;";
107 form+="             l1 = document.getElementById('l11');l2 = document.getElementById('l12');";
108
109 form+="             if (i < max){ban=1; llenar_b.disabled = false;vaciar_b.disabled=false;";
110 form+="                 l1.setAttribute('style', 'height: 10px; background-color: #888896;');";
111 form+="                 Enviar('finllenar');";
112 form+="             if(ban==0){ i=i-2;";
113 form+="                 barra.setAttribute('style', 'height:'+i+'px; width: 150px; background: #080304;');";

```

Continúa la adición de líneas a la variable **form**.

```

116         form+= "                setTimeout('functionProgress1(' + (i) + ',' + (0) + ')', TIME1);});";
117         /*-----*/
118         form+= "    </script> </head>";
119 form+= "    <body style='overflow: hidden; height: 100%; display: -webkit-flex; display: -ms-flexbox; display: flex; -webkit-justify-content: center;
120
121 form+= "        <div id='llave1' style='position: absolute; top: 60px; left: 40%; font: 120% monospace;'>Llave 1 &nbsp;<div id='ll1' width=60% align=
122
123 form+= "        <div id='etiquetas' style='position: absolute; top: 260px; left: 40%; font: 120% monospace;'>Limite</div>";
124 form+= "    <div id='contenedor' style='width: 150px; height: 200px; background: #504DE2; position: absolute; top: 100px; left: 45%;>";
125 form+= "        <div id='barra' style='height: 0px; width: 150px; background: #000304;'>";
126 form+= "            <div width=100% align='center' color='#C0392B' style='position: absolute; top: 160px; left: 0%;> </div> </div>";
127
128 form+= "    <div id='llave2' style='position: absolute; top: 262px; left: 56%; font: 120% monospace;'> &nbsp;<div id='ll2' width=50% align=
129 form+= "        </div> <form name='form1' action='' method='post' style='position: relative; right: 300px; top: 150px; font-family: verdana, arial; font-s
130
131 form+= "        <legend id='eti' style='font-size: 14px; font-family: 'Tahoma', 'Geneva', sans-serif; font-weight: bold; color: #C0392B; text-align:
132 form+= "            <br> <input id='v1' type='button' name='boton1' value='Vaciar' onclick='functionProgress2(0,-1);' />";
133 form+= "            <input id='ll' type='button' name='boton' value='Llenar' onclick='functionProgress1(169,-1);' />";
134 form+= "        </fieldset> <label id='mensaje'> </label> </form> </body> </html> ";
135 /*-----*/
136
137 pinMode(LED_PIN, OUTPUT); // tanque
138 digitalWrite(LED_PIN, LOW);
139
140 pinMode(D2, OUTPUT); // llenar
141 digitalWrite(D2, LOW);
142
143 pinMode(D1, OUTPUT); // vaciar
144 digitalWrite(D1, LOW);
145
146 /* You can remove the password parameter if you want the AP to be open. */
147 WiFi.softAP(ssid, password);
148
149 IPAddress myIP = WiFi.softAPIP();
150 Serial.print("AP IP address: ");
151 Serial.println(myIP);
152
153 server.on("/", [](){
154     server.send(200, "text/html", form);
155 });
156 Serial.println("Pagina...");
157 server.on("/inivaciar.html", ini_vaciar);
158 server.on("/finvaciar.html", fin_vaciar);
159 server.on("/inilllenar.html", ini_llenar);
160 server.on("/finllenar.html", fin_llenar);
161 server.begin();
162 }
163
164 void loop() {
165     server.handleClient();
166 }

```

Luego de terminar la inicialización de la variable **form** se procede a definir los pines de salida de los LEDs como pines digitales de salida usando la función **pinMode()**; con el segundo parámetro en OUTPUT. Asimismo, se crea un IP Access Point usando la función **softAP()** del objeto WiFi; luego se llama al **server.on()** que se encarga de definir los manejadores de las peticiones hechas a la página con diferentes URI utilizando diferentes funciones (las que se codificaron al inicio del .ino). La página presenta diferentes URI para poder usar AJAX como método de envío y recepción de información entre el NodeMCU y el back-end de la página web.

Por último, se inicia la recepción en el servidor con **server.begin()** para comenzar a servir las páginas web a través del punto de acceso usando el puerto 80 (ahí fue declarado al principio del código), y se esperan los posibles clientes a conectarse usando **server.handleClient()**.

