

Diseño y construcción de un sistema automatizado de control de temperatura para el ahorro de agua en la ducha.

Design and construction of an automated temperature control system to save water in shower.

Fernando Alfonso May Arrijoa (1).
Tecnológico Nacional de México, Instituto Tecnológico de Tuxtla Gutiérrez.
fernando.ma@tuxtla.tecnm.mx.

Hernán Valencia Sánchez* (2). Instituto Tecnológico de Tuxtla Gutiérrez, hernan.vs@tuxtla.tecnm.mx.

Ignacio Arrijoa Cárdenas (3). Instituto Tecnológico de Tuxtla Gutiérrez, ignacio.ac@tuxtla.tecnm.mx.

Roberto Carlos García Gómez (4). Instituto Tecnológico de Tuxtla Gutiérrez, roberto.gg@tuxtla.tecnm.mx.

Juan Carlos Niños Torres (5). Instituto Tecnológico de Tuxtla Gutiérrez, juan.nt@tuxtla.tecnm.mx.

*corresponding author.

Artículo recibido en octubre 18, 2023; aceptado en noviembre 10, 2023.

Resumen.

El proyecto consiste en diseñar y construir un sistema automatizado que permita el ahorro de agua durante la ducha. Específicamente la cantidad de agua desperdiciada en las duchas, en espera de que el agua alcance la temperatura deseada de uso. Se usa el software **MATLAB** (*MATrix LABoratory*). Este software nos da la capacidad de crear una interfaz de usuario que nos proporciona la interactividad con el sistema. La automatización del sistema de control de temperatura, permite que el usuario establezca el volumen y la temperatura del agua que utilizará para la ducha, entonces determina el volumen de agua fría (tinaco) y agua caliente (boiler), en función de su temperatura, para obtener el volumen de mezcla a la temperatura deseada. Este proyecto se montará en una base experimental que incluirá sensores de temperatura, electroválvulas, recipientes para representar el tinaco, el boiler y el recipiente de equilibrio termodinámico, tubería de pvc y llaves universales.

Palabras claves: Ahorro de agua, ducha, equilibrio termodinámico.

Abstract.

The project consists of designing and building an automated system that allows water savings during showering. Specifically the amount of water wasted in showers, waiting for the water to reach the desired temperature for use. MATLAB software (*MATrix LABoratory*) is used. This software gives us the ability to create a user interface that provides us with interactivity with the system. The automation of the temperature control system allows the user to set the volume and temperature of the water to be used for the shower. Then determine the volume of cold water (water tank) and hot water (boiler), depending on their temperature, to obtain the volume of mixture at the desired temperature. This project will be set up on an experimental base that will include temperature sensors, solenoid valves, containers to represent the water tank, the boiler and the thermodynamic equilibrium container, PVC pipe and universal keys.

Keywords: Shower, thermodynamic balance, water saving.

1. Introducción.

Una gran cantidad de agua limpia es desperdiciada cada vez que la mayoría de las personas se dan un baño mientras esperan la salida de agua caliente. El problema no es menor, la ducha es el elemento por donde más agua se desperdicia en los hogares. Se estima que las duchas domésticas en México gastan en promedio 7 litros por minuto. El tiempo promedio para que el agua caliente fluya es de 1,8 minutos por lo que más de 12 litros se pierden al esperar a que salga (Twenergy, 2019). Para evitar lo anterior, se propone un sistema automatizado que a partir de la temperatura del agua caliente en el boiler y de la temperatura del agua fría en el tinaco, determine los volúmenes de ambas que deben mezclarse para obtener el volumen seleccionado por el usuario que toma la ducha, así como la temperatura deseada por el mismo (temperatura de equilibrio). Por ello, el software determina el tiempo de apertura de las electroválvulas de las líneas de agua caliente y agua fría, para obtener los volúmenes necesarios. Se utilizan 3 sensores de temperatura ubicados en la línea de agua fría, la línea de agua caliente y en el tanque de equilibrio. Para adquirir los datos del proceso (variables físicas) se usa Arduino mega. Los volúmenes de agua fría y caliente en el tanque de equilibrio se controlan midiendo su respectiva altura en el tanque de equilibrio, cuya sección transversal es constante, usando sensores ultrasónicos.

2. Métodos.

Configuración del sistema.

El sistema se integra, básicamente, de los siguientes elementos:

- 3 sensores de temperatura sumergibles.
- 2 electroválvulas de ½”.
- 2 módulos de relay.
- 1 arduino mega.
- 1 protoboard.
- Cables UTP y jumpers.
- 3 recipientes para agua.

El sistema considera un primer recipiente (Tanque 1) para el agua caliente (agua del boiler), un segundo recipiente (Tanque 2) para el agua fría (agua del tinaco) y, un tercer recipiente (Tanque 3), donde se mezclarán volúmenes calculados de agua caliente y agua fría, para obtener un volumen final de agua en el Tanque 3, a la temperatura seleccionada por el usuario. Se utilizan sensores de temperatura para medir la temperatura del agua fría en el tinaco, la temperatura del agua caliente en el boiler y la temperatura en el tanque de equilibrio termodinámico (Tanque 3).

Para controlar los volúmenes requeridos de agua fría y agua caliente, el programa calcula el tiempo de apertura de las dos electroválvulas, instaladas en la tubería de agua fría y agua caliente, que las conducen al Tanque 3. Se requiere que la sección transversal del Tanque 3, sea un área A constante a lo largo de toda su altura. La configuración general de los componentes se muestra en la figura 1.

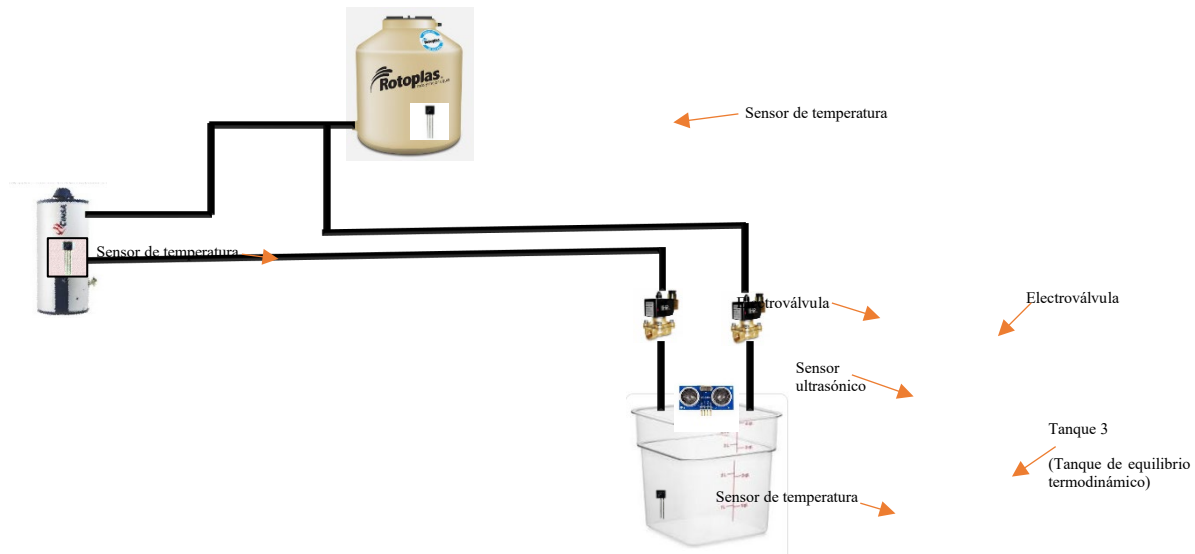


Figura 1. Configuración del sistema de equilibrio termodinámico (fuente propia).

Cálculo de alturas de agua fría y caliente en el tanque de equilibrio.

Se requiere determinar la masa m_1 (masa de agua caliente) y la masa m_2 (masa de agua fría) para obtener la masa m_3 (masa de agua a la temperatura deseada) a la temperatura de equilibrio.

La energía Q requerida para cambiar la temperatura de una masa m , de una temperatura inicial T_0 a una temperatura final T_f , manteniendo constante la presión (Cengel & Boles, 2011), está dada por:

$$Q = m c_p (T_f - T_0) \text{ --- (1)}$$

Donde:

c_p = calor específico a presión constante.

Entonces:

$$Q_1 + Q_2 = 0 \text{ --- (2)}$$

Donde:

Q_1 = calor que cede el agua caliente

Q_2 = calor que recibe el agua fría

Aplicando (1) en (2) con los correspondientes subíndices:

$$(m_1)(c_{p1})(T_f - T_{01}) + (m_2)(c_{p2})(T_f - T_{02}) = 0 \text{ --- (3)}$$

Donde:

T_f = temperatura final de ambos fluidos (temperatura de equilibrio)

T_{01} = temperatura inicial del agua caliente

T_{02} = temperatura inicial del agua fría

Para el agua, $c_{p1} = c_{p2}$, sustituyendo en (3):

$$(m_1)(T_f - T_{01}) + (m_2)(T_f - T_{02}) = 0 \text{ --- (4)}$$

Pero:

$$m_1 + m_2 = m_3 \text{ --- (5)}$$

Donde:

$m_3 = \text{masa de agua en equilibrio térmico}$

De (5), tenemos:

$$m_1 = m_3 - m_2 \text{ --- (6)}$$

Y sustituyendo (6) en (4):

$$\begin{aligned} (m_3 - m_2)(T_f - T_{01}) + (m_2)(T_f - T_{02}) &= 0 \\ m_2(T_f - T_{01}) - m_2(T_f - T_{02}) &= m_3(T_f - T_{01}) \\ m_2 &= \frac{m_3(T_f - T_{01})}{(T_f - T_{01}) - (T_f - T_{02})} \text{ --- (7)} \end{aligned}$$

Entonces:

$$m_1 = m_3 - \frac{m_3(T_f - T_{01})}{(T_f - T_{01}) - (T_f - T_{02})} \text{ --- (8)}$$

El control automático abrirá las electroválvulas el tiempo necesario para que fluyan al Tanque 3 (Tanque de equilibrio termodinámico), las masas m_1 y m_2 . Las temperaturas T_{01} y T_{02} serán medidas por los respectivos sensores de temperatura, instalados en el boiler y tinaco, respectivamente. El usuario, a través de la interfaz determinará la masa m_3 y la temperatura T_f .

Las masas m_1 y m_2 , se relacionan con los volúmenes V_1 y V_2 (figura 2) usando:

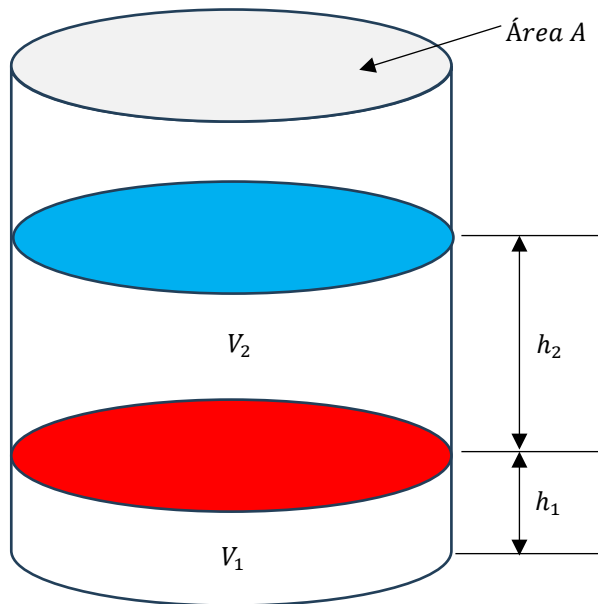


Figura 2. Masas de agua fría y caliente en Tanque 3 (fuente propia).

Se tiene:

$$V_1 = Ah_1 \text{ y } V_2 = Ah_2 \text{ --- (9)}$$

Donde:

- h_1 = altura de agua caliente
- V_1 = volumen de agua caliente
- h_2 = altura de agua fría
- V_2 = volumen de agua fría

Las alturas h_1 y h_2 serán medidas con el sensor ultrasónico instalado en el Tanque 3.

Sabiendo que $\rho = m/V$, donde ρ es la densidad, m es la masa y V es el volumen, entonces:

$$m = \rho V \text{ --- (10)}$$

Por consiguiente:

$$m_1 = \rho_1 V_1 \text{ y } m_2 = \rho_2 V_2$$

Donde ρ_1 y ρ_2 , son las densidades del agua caliente y del agua fría, respectivamente.

Entonces:

$$m_1 = \rho_1 Ah_1 \text{ y } m_2 = \rho_2 Ah_2$$

Despejando h_1 y h_2 :

$$h_1 = \frac{m_1}{\rho_1 A} \text{ y } h_2 = \frac{m_2}{\rho_2 A} \text{ --- (11)}$$

Sustituyendo (7) y (8) en (11), queda:

$$h_1 = \frac{m_3}{\rho_1 A} - \frac{m_3(T_f - T_{01})}{\rho_1 A[(T_f - T_{01}) - (T_f - T_{02})]} \text{ --- (12)}$$

$$h_2 = \frac{m_3(T_f - T_{01})}{\rho_2 A[(T_f - T_{01}) - (T_f - T_{02})]} \text{ --- (13)}$$

Para definir la masa m_3 el usuario define el volumen V_3 , dado por:

$$V_3 = A(h_1 + h_2) = \frac{m_3}{\rho_3} \text{ --- (14)}$$

Donde ρ_3 es la densidad del agua en el Tanque 3 (agua en equilibrio termodinámico) a la temperatura deseada T_f .

Entonces:

$$m_3 = \rho_3 V_3 \text{ --- (15)}$$

Sustituyendo (15) en (12) y (13), respectivamente, tenemos:

$$h_1 = \frac{\rho_3 V_3}{\rho_1 A} - \frac{\rho_3 V_3(T_f - T_{01})}{\rho_1 A[(T_f - T_{01}) - (T_f - T_{02})]} \text{ --- (16)}$$

$$h_2 = \frac{\rho_3 V_3(T_f - T_{01})}{\rho_2 A[(T_f - T_{01}) - (T_f - T_{02})]} \text{ --- (17)}$$

Como $V_s = \frac{1}{\rho}$, donde V_s es el volumen específico, tanto para el agua caliente, para el agua fría y el agua del Tanque 3, se tiene $\rho_1 = \frac{1}{V_{s1}}$, $\rho_2 = \frac{1}{V_{s2}}$ y $\rho_3 = \frac{1}{V_{s3}}$. Sustituyendo en (16) y (17), finalmente se tiene:

$$h_1 = \frac{V_{s1} V_3}{V_{s3} A} - \frac{V_{s1} V_3(T_f - T_{01})}{V_{s3} A[(T_f - T_{01}) - (T_f - T_{02})]} \text{ --- (18)}$$

$$h_2 = \frac{V_{s2} V_3(T_f - T_{01})}{V_{s3} A[(T_f - T_{01}) - (T_f - T_{02})]} \text{ --- (19)}$$

Programación del sistema.

La programación en *MATLAB* (The MathWorks, Inc., 2023) se inicia estableciendo la comunicación serial entre *MATLAB* y Arduino para el uso y librerías de los sensores (figura 3).

```

1 - a=arduino('COM11', 'Uno', 'Libraries',{ 'JRodrigoTech/HCSR04','PaulStoffregen/OneWire'});
2 - sensor=addon(a, 'JRodrigoTech/HCSR04', 'D12', 'D13'); %Pin D12 al Trigger, Pin D13 al Echo
3 - sensor1=addon(a, 'PaulStoffregen/OneWire', 'D9'); %Sensor de temperatura 1 al pin D9
4 - sensor2=addon(a, 'PaulStoffregen/OneWire', 'D9'); %Sensor de temperatura 2 al pin D9
5 - sensor3=addon(a, 'PaulStoffregen/OneWire', 'D9'); %Sensor de temperatura 3 al pin D9
6 - addr = sensor1.AvailableAddresses{1}; %Dirección IP a tomar para el sensor 1
7 - addr1 = sensor2.AvailableAddresses{2}; %Dirección IP a tomar para el sensor 2
8 - addr2 = sensor3.AvailableAddresses{3}; %Dirección IP a tomar para el sensor 3

```

Figura 3. Comunicación serial entre MATLAB y Arduino (fuente propia).

A continuación, se programa la adquisición de datos de los sensores de temperatura en los Tanques 1 y 2 (agua caliente y agua fría) (figura 4 y 5).

```

10 %Inicio de lectura para los sensores de temperatura
11 %Sensor 1 (tanque agua caliente)
12 reset(sensor1);
13 write(sensor1, addr, hex2dec('44'), true);
14 pause(1);
15 reset(sensor1);
16 write(sensor1, addr, hex2dec('BE')); % read command - 'BE'
17 data = read(sensor1, addr, 9);
18 crc = data(9);
19 sprintf('Data = %x %x %x %x %x %x %x %x CRC = %x\n', ...
20 data(1), data(2), data(3), data(4), data(5), data(6), data(7), data(8), crc);
21 if ~checkCRC(sensor1, data(1:8), crc, 'crc8')
22 error('Invalid data read.');
```

```

23 end
24 raw = bitshift(data(2),8)+data(1);
25 cfg = bitshift(bitand(data(5), hex2dec('60')), -5);
26 switch cfg
27 case bin2dec('00') % 9-bit resolution, 93.75 ms conversion time
28 raw = bitand(raw, hex2dec('fff8'));
29 case bin2dec('01') % 10-bit resolution, 187.5 ms conversion time
30 raw = bitand(raw, hex2dec('fffc'));
31 case bin2dec('10') % 11-bit resolution, 375 ms conversion time
32 raw = bitand(raw, hex2dec('ffff'));
33 case bin2dec('11') % 12-bit resolution, 750 ms conversion time
34 otherwise
35 error('Invalid resolution configuration');
```

```

36 end
37 raw = typecast(uint16(raw), 'int16');
38 celsius = double(raw) / 16.0;

```

Figura 4. Programación sensor agua caliente (fuente propia).

```

40 %Sensor 2 (tanque agua fría)
41 reset(sensor2);
42 write(sensor2, addr1, hex2dec('44'), true);
43 pause(1);
44 reset(sensor2);
45 write(sensor2, addr1, hex2dec('BE')); % read command - 'BE'
46 data = read(sensor2, addr1, 9);
47 crc = data(9);
48 sprintf('Data = %x %x %x %x %x %x %x %x CRC = %x\n', ...
49     data(1), data(2), data(3), data(4), data(5), data(6), data(7), data(8), crc);
50 if ~checkCRC(sensor2, data(1:8), crc, 'crc8')
51     error('Invalid data read.');
```

```

52 end
53 raw = bitshift(data(2),8)+data(1);
54 cfg = bitshift(bitand(data(5), hex2dec('60')), -5);
55 switch cfg
56     case bin2dec('00') % 9-bit resolution, 93.75 ms conversion time
57         raw = bitand(raw, hex2dec('fff8'));
58     case bin2dec('01') % 10-bit resolution, 187.5 ms conversion time
59         raw = bitand(raw, hex2dec('fffc'));
60     case bin2dec('10') % 11-bit resolution, 375 ms conversion time
61         raw = bitand(raw, hex2dec('ffff'));
62     case bin2dec('11') % 12-bit resolution, 750 ms conversion time
63         otherwise
64             error('Invalid resolution configuration');
```

```

65 end
66 raw = typecast(uint16(raw), 'int16');
67 celsius1 = double(raw) / 16.0;
```

Figura 5. Programación sensor agua fría (fuente propia).

Los valores medidos se reservan en las variables $t1$ y $t2$ (Figura 6).

```

70 t1=celsius; %Temperatura del sensor del agua caliente
71 t2=celsius1; %Temperatura del sensor del agua fría
72
73 fprintf('Temperatura del tanque de agua caliente es: %f\n',t1);
74 fprintf('Temperatura del tanque de agua fría es: %f\n',t2);
```

Figura 6. Temperaturas obtenidas por los sensores (fuente propia).

Enseguida se solicitan al usuario, los valores de volumen y temperatura del agua en equilibrio termodinámico (Tanque 3) (Figura 7).

```

76 fprintf('El volumen máximo a solicitar es 1.3lts \n');
77 t3=input('Temperatura del agua requerida:'); %Temperatura deseada por el usuario
78 v=input('Volumen de agua requerida:'); %Volumen deseado por el usuario
```

Figura 7. Parámetros que el usuario define (fuente propia).

A continuación, el programa calcula los volúmenes específicos del agua caliente, del agua fría y del agua en equilibrio, según la temperatura medida por los sensores o definida por el usuario (figuras 8-11).


```

80 %Memoria de cálculo
81 - b=0.0084948665;
82 - ih=Int(t1); %Interpolación para el agua caliente
83 - ic=Inte(t2); %Interpolación para el agua fría
84 - id=Inter(t3); %Interpolación para el agua pedida por el usuario

```

Figura 8. Subprogramas para interpolar volumen específico del agua (fuente propia).

```

1 %subprograma donde van los valores de temperatura y volumen específico para
2 %hacer la interpolación.
3 function ih=Int(t) %función a llamar en el programa principal
4 - T=[30,35,40,45,50,55,60,65,70,75,80,85,90,95]; %Temperatura del agua caliente
5 - Ps=[0.001004,0.001006,0.001008,0.001010,0.001012,0.001015,0.001017,0.001020,0.001023,0.001026,0.001029];
6 - ih=interp1(T,Ps,t);
7 - end

```

Figura 9. Interpolación volumen específico agua caliente (fuente propia).

```

1 %subprograma donde van los valores de temperatura y volumen específico para
2 %hacer la interpolación.
3 function ic=Inte(t) %función a llamar en el programa principal
4 - T=[5,10,15,20,25,30,35]; %Temperatura del agua fría
5 - Ps=[0.001000,0.001000,0.001001,0.001002,0.001003,0.001004,0.001006]; %volumen específico del agua
6 - ic=interp1(T,Ps,t);
7 - end

```

Figura 10. Interpolación volumen específico agua fría (fuente propia).

```

1 %subprograma donde van los valores de temperatura y volumen específico para
2 %hacer la interpolación.
3 function id=Inter(t) %función a llamar en el programa principal
4 - T=[5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95]; %Temperatura del agua requerida
5 - Ps=[0.001000,0.001000,0.001001,0.001002,0.001003,0.001004,0.001006,0.001008,0.001010,0.001012,0.001015];
6 - id=interp1(T,Ps,t);
7 - end

```

Figura 11. Interpolación volumen específico agua en equilibrio termodinámico (fuente propia).

Enseguida se resuelve la matriz con los volúmenes específicos para determinar los volúmenes V_1 y V_2 (figura 12).

```

86 %Matriz para obtener los valores del V1 y V2
87 - fa=2;
88 - f=[t1,t2];
89 - ca=2;
90 - c=[1,1];
91 - A=[f;c];
92 - Ainv=Minversa(A,fa);
93 - B=[(v*t3);v];
94 - P=ProductoMatrices(Ainv,B,fa,ca,1);
95 - F=P(1,1);
96 - C=P(2,1);

```

Figura 12. Cálculo de los volúmenes V_1 y V_2 (fuente propia).

Se calcula la matriz inversa de A (figura 13).

```

1  function Ainv=Minversa(A,n)
2      k = n+1;
3      for i=1:n
4          for j=n+1:2*n
5              if k ~= j
6                  A(i,j)= 0.0;
7              else
8                  A(i,j)= 1.0;
9              end
10         end
11         k=k+1;
12     end
13     % Se obtiene la matriz inversa de A(i,j).
14     for k=1:n
15         for j=2*n:-1:k
16             A(k,j)= A(k,j)/A(k,k);
17         end
18         for i=1:n
19             if i~=k
20                 for j=2*n:-1:k
21                     A(i,j)=A(i,j)- A(i,k)*A(k,j);
22                 end
23             end
24         end
25     end
26     % Se extrae la matriz inversa Ainv(i,j) de la matriz A(i,j).
27     for i=1:n
28         k = 1;
29         for j=n+1:2*n
30             Ainv(i,k)= A(i,j);
31             k=k+1;
32         end
33     end
34 end

```

Figura 13. Matriz inversa de A (fuente propia).

Se procede a obtener el producto de las matrices A y B (figura 14).

```

1  function P=ProductoMatrices(A,B,fa,ca,cb)
2      % Se desarrolla el producto de matrices
3      for i=1:fa
4          for j=1:cb
5              s=0;
6              for k=1:ca
7                  s=s+A(i,k)*B(k,j);
8              end
9              P(i,j)=s;
10         end
11     end
12 end

```

Figura 14. Producto de matrices (fuente propia).

Entonces se pueden calcular las alturas h_1 de agua caliente y h_2 de agua fría (figura 15).

```

98 %Cálculos para la apertura y cierre de las electroválvulas
99 - h1=(F*ih)/b; %Alura requerida para el agua caliente
100 - h2=(C*ic)/b; %Alura requerida para el agua fría
101 - h3=h1+h2; %Alura final en el recipiente intermedio
102 - h4=0.236-h1;
103 - h5=0.236-h3;

```

Figura 15. Cálculo de alturas h_1 y h_2 en Tanque 3 (fuente propia).

Ahora, se pueden establecer los tiempos de apertura de las electroválvulas en las líneas de agua caliente y agua fría, como una función de las alturas determinadas para cada tipo de fluido (figura 16).

```

107 - for i=15:-.01:1
108 -     i1=i;
109 -     i1=i1-.01;
110 -     if i1>1
111 -         d=readTravelTime(sensor); %Lee el tiempo de viaje de las ondas del sensor Ultrasónico
112 -         d1=340*d/2;
113 -         d2=d1+0.010;
114 -         d3=d1+0.008;
115 -         pause(.1);
116 -         end
117 -         configurePin(a,'D10','DigitalOutput'); %Configuración del pin D10 (Válvula 1) como salida
118 -         if d2<=h4
119 -             writeDigitalPin(a,'D10',1);
120 -             configurePin(a,'D11','DigitalOutput'); %Configuración del pin D11 (Válvula 2) como salida
121 -             writeDigitalPin(a,'D11',0);
122 -             if d3<=h5
123 -                 writeDigitalPin(a,'D11',1);
124 -                 break
125 -             else
126 -                 if d1>h4
127 -                     writeDigitalPin(a,'D10',0);
128 -                     end
129 -                 end
130 -             end
131 -         end

```

Figura 16. Tiempo de apertura de electroválvulas en función de las lecturas del sensor ultrasónico (fuente propia).

Finalmente se programa la lectura del sensor de temperatura en el Tanque 3, para medir que corresponda a la temperatura establecida por el usuario (figura 17).

```

135 - pause(12);
136 - reset(sensor3);
137 - write(sensor3, addr2, hex2dec('44'), true);
138 - pause(1);
139 - reset(sensor3);
140 - write(sensor3, addr2, hex2dec('BE')); % read command - 'BE'
141 - data = read(sensor3, addr2, 9);
142 - crc = data(9);
143 - sprintf('Data = %x %x %x %x %x %x %x %x CRC = %x\n', ...
144 -     data(1), data(2), data(3), data(4), data(5), data(6), data(7), data(8), crc);
145 - if ~checkCRC(sensor3, data(1:8), crc, 'crc8')
146 -     error('Invalid data read.');
```

```

147 - end
148 - raw = bitshift(data(2),8)+data(1);
149 - cfg = bitshift(bitand(data(5), hex2dec('60')), -5);
150 - switch cfg
151 -     case bin2dec('00') % 9-bit resolution, 93.75 ms conversion time
152 -         raw = bitand(raw, hex2dec('fff8'));
153 -     case bin2dec('01') % 10-bit resolution, 187.5 ms conversion time
154 -         raw = bitand(raw, hex2dec('fffc'));
155 -     case bin2dec('10') % 11-bit resolution, 375 ms conversion time
156 -         raw = bitand(raw, hex2dec('fffe'));
157 -     case bin2dec('11') % 12-bit resolution, 750 ms conversion time
158 -         otherwise
159 -             error('Invalid resolution configuration');
```

```

160 - end
161 - raw = typecast(uint16(raw), 'int16');
162 - celsius2 = double(raw) / 16.0;
163 - t5=celsius2; %Temperatura del sensor del tanque intermedio

```

Figura 17. Temperatura sensor Tanque 3 (fuente propia).

Pruebas experimentales.

Con el propósito de verificar de forma práctica las ecuaciones (18) y (19), así como validar la programación de los sensores de temperatura y el sensor ultrasónico, se construye un modelo a escala, utilizando recipientes plásticos para simular el tinaco, el boiler y el Tanque de equilibrio (Tanque 3) (figura 18).

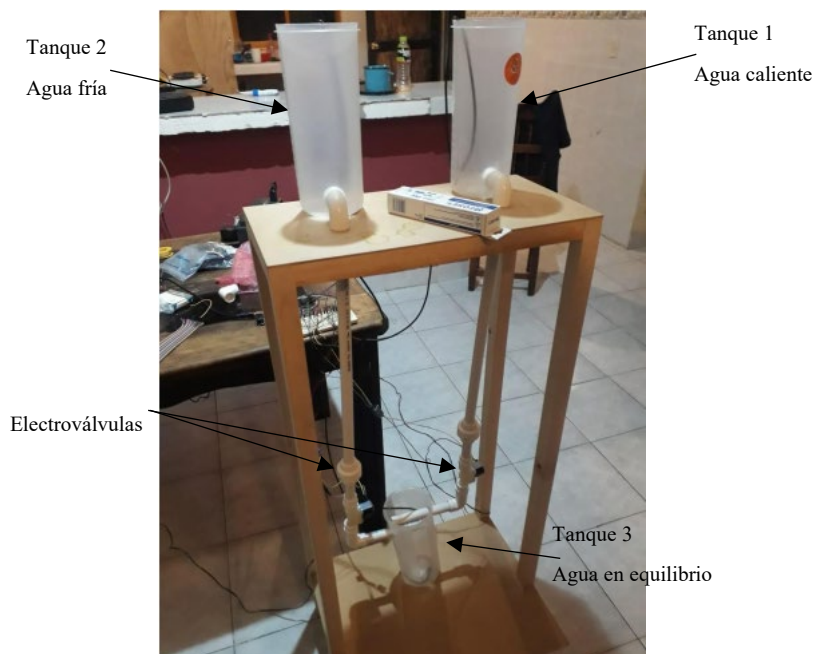


Figura 18. Modelo de experimentación (Fuente propia).

El agua en el Tanque 1 se calienta usando una simple resistencia eléctrica y la temperatura se eleva hasta los $58.687500\text{ }^{\circ}\text{C}$, mientras que al agua del Tanque 2 se enfría usando cubos de hielo, midiendo entonces una temperatura de $27.562400\text{ }^{\circ}\text{C}$. El volumen del Tanque 3 (experimental) es de 1.3 lt .

Por lo anterior el usuario puede, a través de la interface, seleccionar un volumen V , tal que $(0 \leq V \leq 1.3)\text{lt}$ y una temperatura tal que $27.562400\text{ }^{\circ}\text{C} \leq T \leq 58.687500\text{ }^{\circ}\text{C}$. Para la experimentación se selecciona un volumen $V = 0.7\text{ lt}$ y una temperatura $T = 35\text{ }^{\circ}\text{C}$.

En la figura 19 se puede ver la información obtenida en la interface.

```
Editor - C:\Users\GAMAU...MATLAB\Taller de investigacion\Subprogramas\Memoria1.m
Command Window
Temperatura del tanque de agua caliente es: 58.687500
Temperatura del tanque de agua fria es: 27.562500
Volumen máximo a solicitar 1.3lts
Temperatura del agua requerida:35
Volumen de agua requerida:.7
El volumen final es: 0.7004
La temperatura final con fórmula es: 35.0000
Temperatura del tanque intermedio medido del sensor : 34.312500
>>
```

Figura 19. Resultados obtenidos en la corrida experimental (fuente propia).

3. Resultados.

Se obtienen las ecuaciones (18) y (19) que permiten definir, dada la sección transversal constante (Área A) del Tanque 3, las alturas h_1 y h_2 que definen los volúmenes requeridos para obtener una mezcla con volumen y temperatura establecidos por el usuario.

Se usan tablas de propiedades del agua a diferentes temperaturas y presión atmosférica estándar, para interpolar los volúmenes específicos del agua fría, agua caliente y el agua en equilibrio.

Como se puede apreciar en la Figura 19, el usuario solicita un volumen de 0.7 lts a una temperatura de 35 °C y el sistema dispensa, sin desperdiciar ni una gota de agua, un volumen de 0.7004 lts a una temperatura de 34.3125 °C. Esto es, una eficiencia del 99.94% en el volumen dispensado y 98.04% en la temperatura equilibrada.

Por lo anterior, la implementación del sistema en una casa habitación, evitaría el desperdicio del agua que se deja correr durante la ducha, en espera de que se alcance la temperatura deseada por el usuario.

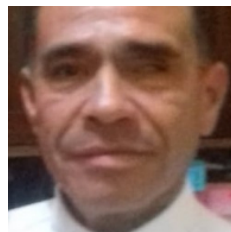
Referencias bibliográficas.

Cengel, Y. A., & Boles, M. A. (2011). *Thermodynamics: An Engineering Approach*. Boston, USA: McGraw-Hill. Recuperado el 2023 de Octubre de 2023.

The MathWorks, Inc. (2023). *Aprenda a utilizar MATLAB y Simulink mediante tutoriales gratuitos*. Recuperado el 5 de Octubre de 2023, de MathWorks: https://la.mathworks.com/support/learn-with-matlab-tutorials.html?s_tid=hp_ff_1_tutorials.

Twenergy. (27 de Marzo de 2019). ¿Cómo podemos evitar el desperdicio del agua al comenzar el baño? Recuperado el 24 de Octubre de 2023, de Twenergy: <https://twenergy.com/ecologia-y-reciclaje/como-ahorrar-agua/como-podemos-evitar-el-desperdicio-del-agua-al-comenzar-el-bano-1586/>.

Información de los autores.



Fernando Alfonso May Arrijoja, Licenciado en Ingeniería Mecánica por el Instituto Tecnológico de Tuxtla Gutiérrez, Maestro en Energías Renovables por la Universidad Politécnica de Chiapas y Doctor en Ingeniería Aplicada en el Colegio de Formación Educativa Tenam. Profesor de Carrera Enseñanza Superior en el Departamento de Metal-Mecánica del TecNM/Instituto Tecnológico de Tuxtla Gutiérrez. Perfil Deseable desde 2016.



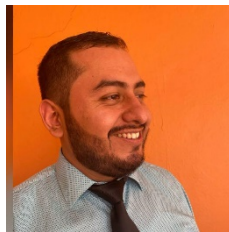
Hernán Valencia Sánchez, Licenciatura en Ingeniería Mecánica y con un Posgrado en Maestría en Ciencias en Ingeniería Mecatrónica Egresado del Instituto Tecnológico de Tuxtla Gutiérrez, Reconocimiento al perfil deseable Jefe de Proyectos de Investigación del Departamento de Metal-Mecánica, Docente del Departamento de Metal-Mecánica del TecNM/Instituto tecnológico de Tuxtla Gutiérrez.



Ignacio Arriola Cárdenas, es profesor de tiempo completo y jefe de proyectos de docencia en el departamento de Metal-Mecánica del Instituto Tecnológico de Tuxtla Gutiérrez/Tecnológico Nacional de México. Doctor en Ingeniería aplicada por el Colegio de Formación Educativa Tenam, Maestro en Ciencias en Ingeniería Mecánica por el Instituto Tecnológico de Veracruz, Maestro en Ciencias Físicas por Centro Mesoamericano de Física Teórica-Universidad Autónoma de Chiapas e Ingeniero Mecánico por el Instituto Tecnológico de Tuxtla Gutiérrez



Roberto Carlos García Gómez, Licenciatura en Ingeniería Mecánica, con tres posgrados: Especialidad en Mecatrónica, Maestría en Ciencias en Ingeniería Mecánica, Doctorado en Procesos de Manufactura. Egresado del Instituto Tecnológico de Tuxtla Gutiérrez, Jefe de Proyectos de Vinculación del Departamento de Metal-Mecánica, Docente del Departamento de Metal-Mecánica del TecNM/Instituto tecnológico de Tuxtla Gutiérrez.



Juan Carlos Niños Torres es Ingeniero Mecánico egresado del Instituto Tecnológico de Tuxtla Gutiérrez, cuenta con la Maestría en Ciencias en Ingeniería Mecánica del Instituto Tecnológico de Celaya, así también con el Doctorado en Ingeniería Aplicada de la Universidad TENAM. Actualmente es Profesor Investigador del Departamento de Metal Mecánica del Instituto Tecnológico de Tuxtla Gutiérrez y Jefe de la División de Estudios Profesionales del Instituto Tecnológico de Tuxtla Gutiérrez. Cuenta con el perfil PRODEP 2022-2025 como miembro del Cuerpo Académico Ingeniería Mecánica en la línea de investigación de Diseño Mecánico. Es miembro del comité de evaluadores del Consejo de Acreditación de la Enseñanza de la Ingeniería (CACEI A.C.) avalado por

CIEES desde 2016.