

Práctica 1

Sistemas de Tiempo Real

Grupo 17

Pablo Gómez Rivas

Jesús Fornieles Muñoz

Universidad de Almería

Almería, domingo 24 de marzo de 2024

Índice de Contenidos

Página

1. Descripción general del sistema	3
1.1 Descripción de los sensores.....	3
1.2 Aspectos generales del programa ADA	4
2. Ejercicio 1	6
2.1 Salida de resultados por pantalla	9
2.2 Representación gráfica	10
2.2.1 Representación gráfica temperaturas	10
2.2.2 Representación gráfica caudales	10
2.2.3 Representación gráfica radiación	11
2.2.4 Representación gráfica flujo de destilado	11
2.3 Pregunta 1: ¿Se está operando el campo solar y el módulo de forma óptima?	11
2.4 Pregunta 2: ¿Se incumple las restricciones de seguridad en algún momento?.....	11
3. Ejercicio 2	12
3.1 Salida de los datos.....	14
3.2 Representación gráfica	14
3.2.1 Representación gráfica temperaturas	14
3.2.2 Representación gráfica caudales	14
3.2.3 Representación gráfica radiación	15
3.2.4 Representación gráfica flujo de destilado	15
3.3 Pregunta: ¿Se cumplen en este caso las condiciones de operación óptimas?.....	15

1. Descripción general del sistema

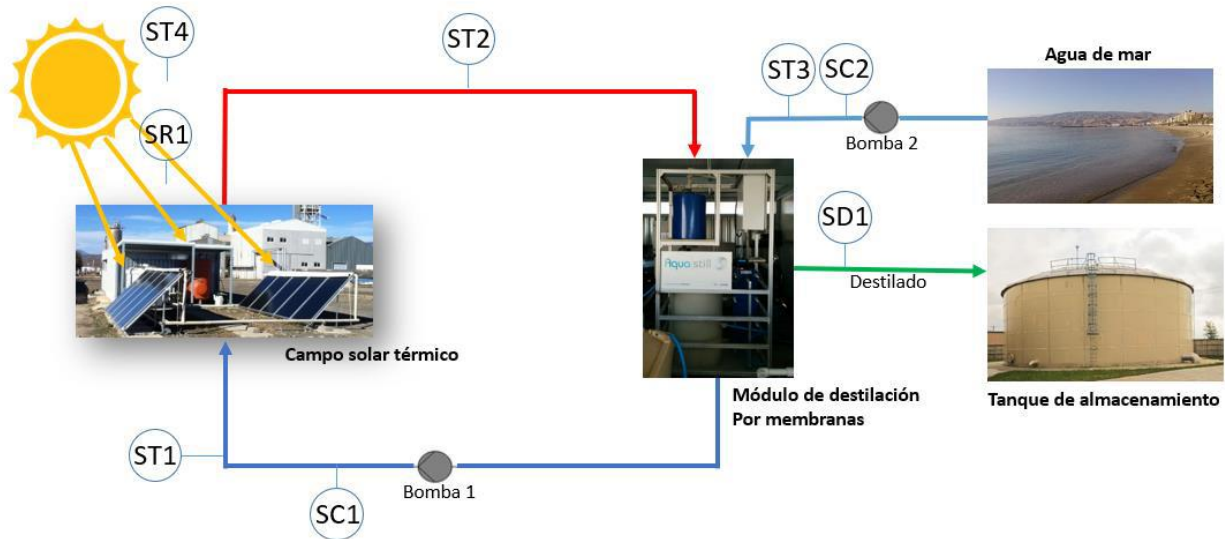


Figura 1. Diagrama esquemático de la instalación de destilación por membranas. ST son sensores de temperatura [°C], SC son sensores de caudal [L/h], SR es un sensor de radiación [W/m²], y SD es un sensor de destilado [L/h].

1.1 Descripción de los sensores

Sensores de temperatura:

- ST1: sensor de temperatura de entrada al campo solar [°C]
- ST2: sensor de temperatura de salida del campo solar [°C]
- ST3: sensor de temperatura del agua de mar [°C]
- ST4: sensor de temperatura ambiente [°C]

Sensores de control:

- SC1: sensor de caudal de entrada del campo solar [L/h] (señal de control)
- SC2: sensor de caudal de entrada al módulo de destilación [L/h] (señal de control)

Sensor de radiación:

- SR1: sensor de radiación [W/m²]

Sensor de destilado:

- SD1: sensor de destilado [L/h]

Valores óptimos:

- Para el primer proceso de control, el **valor óptimo de ST2** es de 82 °C.
- Para el segundo proceso de control, el **valor óptimo de SD1** es de 25 L/h.

1.2 Aspectos generales del programa ADA

Los datos de entrada del programa (**input.txt**) nos proporciona los conjuntos de valores que toman los sensores **SR1**, **ST4** y **ST3** en cada iteración **k** del sistema en tiempo real:

k	SR1 [W/m ²]	ST4 [°C]	ST3 [°C]
1	604	20	20
2	604	20	20.6
3	609	20.1	19.8
4	620	20	20.0
5	590	20	20
6	530	20	20.5
7	450	20	20
8	420	20.2	20
9	460	20	20.7
10	480	20	20
11	450	20	20.5
12	480	20.3	19.0
13	400	20	19.5
14	450	20	20
15	350	20	21

Tabla 1. input.txt

Valores iniciales de ST1 y ST2:

- ST1(0) = 50 °C
- ST2(0) = 60 °C

Se realizará un almacenamiento histórico de los datos de salida del Ejercicio 1 en **data.txt**, y los del Ejercicio 2 en **data2.txt**. Ambos conjuntos de datos también se imprimirán por pantalla. Además, atendiendo a las restricciones de seguridad, si en algún momento **ST2 > 98 °C**, se imprimirá un mensaje de alarma por pantalla, indicando el valor de la temperatura ST2 y la iteración **k** en la cual se ha producido dicha alarma. También se registrarán dichos mensajes de alarma en **alarm_log.txt** para el Ejercicio 1, y en **alarm_log2.txt** para el Ejercicio 2.

```

1  with Ada.Text_IO;           use Ada.Text_IO;
2  with Ada.Integer_Text_IO;   use Ada.Integer_Text_IO;
3  with Ada.Float_Text_IO;     use Ada.Float_Text_IO;
4  with Simulador;             use Simulador;
5
6  procedure Main is
7
8      -- Vectores de los valores de los sensores
9      ST3, ST4, SC1, SC2, SR1, SD1, Tt : Vector(1..15);
10     -- ST1 y ST2 deben aportar un valor inicial [0]
11     ST1, ST2                          : Vector(0..15);
12
13     -- Valor óptimo ST2 (Primer proceso de control)
14     ST2_Opt : Float := 82.0;
15     -- Valor óptimo SD1 (Segundo proceso de control)
16     SD1_Opt : Float := 25.0;
17
18     -- Variable de archivo de entrada (input.txt)
19     input : File_Type;
20     -- Variable de archivo de salida del Ejercicio 1 (data.txt)
21     data : File_Type;
22     -- Variable de archivo de salida del Ejercicio 2 (data2.txt)
23     data2 : File_Type;
24     -- Variable de archivo de alarma del Ejercicio 1(alarm_log.txt)
25     alarm : File_Type;
26     -- Variable de archivo de alarma del Ejercicio 2 (alarm_log2.txt)
27     alarm2 : File_Type;
28
29     -- Índice iterador para el bucle
30     k : Integer;
31
32     begin
33
34         -- Valores iniciales
35         ST1(0) := 50.0;
36         ST2(0) := 60.0;

```

Código 1. Código de main.adb con declaración de variables utilizadas.

Se utilizarán las siguientes constantes a lo largo de la práctica:

Constante	Valor
β	0.15 [m]
L_{eq}	15 [m]
H	4 [J/s K]
c	$9 \cdot 2 \cdot 6 \cdot 10000$ [s L/min m ³]
C_p	4190 [J/kg °C]
ρ	975 [kg/m ³]

Tabla 2. Constantes.

```

1  package Simulador is
2
3      -- Tipo Vector de Float
4      type Vector is Array (Integer range <>) of Float;
5
6      -- Constantes
7      b: constant float := 0.15;
8      Leq: constant float := 15.0;
9      H: constant float := 4.0;
10     c: constant float := 9.0*2.0*6.0*10000.0;
11     Cp: constant float := 4190.0;
12     p: constant float := 975.0;
13
14     -- Declaración de funciones Ejercicio 1
15     function calc_Tt (ST1, ST2:Float) return Float;
16     function calc_ST2 (SR1, ST4, ST1, SC1, Tt:Float) return Float;
17     function calc_ST1 (ST2:Float) return Float;
18     function calc_SD1 (SC2, ST2, ST3:Float) return Float;
19
20     -- Declaración de funciones Ejercicio 1
21     function calc_SC1 (SR1, Tt, ST4, ST2, ST1:Float) return Float;
22     function calc_SC2 (SD1, ST2, ST3:Float) return Float;
23
24 end Simulador;

```

Código 2. Simulador.ads con declaración de constantes y funciones utilizadas.

2. Ejercicio 1

Para realizar el Ejercicio 1, en primer lugar, se han inicializado los valores de los sensores de caudal, los cuales van a permanecer fijos a lo largo de la simulación:

- SC1 = 15 L/h
- SC2 = 450 L/h

Posteriormente, se ha abierto el archivo de entrada de datos, **input.txt**, y se han creado los archivos de salida de datos, **data.txt**, y de salida de mensajes de alarma, **alarm_log.txt**.

Se escribirán tanto por pantalla como en data.txt, la cabecera de la tabla que contendrá los datos de salida.

```

38  -- Ejercicio 1
39  Put_Line("Ejercicio 1:");
40  Put_Line("");
41
42  -- Valores fijos
43  SC1 := (others => 15.0);
44  SC2 := (others => 450.0);
45
46  -- Abrir input.txt (entrada)
47  Open(input, In_File, "input.txt");
48  -- Crear data.txt (salida Ejercicio 1)
49  Create(data, Out_File, "data.txt");
50  -- Crear alarm_log.txt (alarmas Ejercicio 1)
51  Create(alarm, Out_File, "alarm_log.txt");
52
53  -- Escribir cabecera de los datos en archivo data.txt (salida)
54  Put_Line(data,
55      " k" & ASCII.HT &
56      " ST1(k)" & ASCII.HT & ASCII.HT &
57      " ST2(k)" & ASCII.HT & ASCII.HT &
58      " ST3(k)" & ASCII.HT & ASCII.HT &
59      " ST4(k)" & ASCII.HT & ASCII.HT &
60      " SC1(k)" & ASCII.HT & ASCII.HT &
61      " SC2(k)" & ASCII.HT & ASCII.HT &
62      " SR1(k)" & ASCII.HT & ASCII.HT &
63      " SD1(k)");
64
65  -- Escribir cabecera de los datos por pantalla
66  Put_Line(" k" & ASCII.HT &
67      " ST1(k)" & ASCII.HT & ASCII.HT &
68      " ST2(k)" & ASCII.HT & ASCII.HT &
69      " ST3(k)" & ASCII.HT & ASCII.HT &
70      " ST4(k)" & ASCII.HT & ASCII.HT &
71      " SC1(k)" & ASCII.HT & ASCII.HT &
72      " SC2(k)" & ASCII.HT & ASCII.HT &
73      " SR1(k)" & ASCII.HT & ASCII.HT &
74      " SD1(k)");

```

Código 3. main.adb con el principio del Ejercicio 1.

La parte principal del Ejercicio 1 consta de un bucle for que iterará las 15 filas del archivo de entrada input.txt, cargando los valores k, SR1, ST4 y ST3 en cada iteración. Con estos valores y los valores fijos de ST1 y ST2, se llevarán a cabo las siguientes fórmulas para calcular el resto de valores:

$$ST2(k) = ST1(k-1) + \frac{(\beta \cdot L_{eq} \cdot SR1(k)) \cdot c}{SC1(k) \cdot C_p \cdot \rho} - \frac{(H \cdot (T_t(k) - ST4(k))) \cdot c}{SC1(k) \cdot C_p \cdot \rho}, \quad (1)$$

$$T_t(k) = \frac{ST1(k-1) + ST2(k-1)}{2}, \quad (2)$$

$$ST1(k) = ST2(k) - 10, \quad (3)$$

$$SD1(k) = 24 \cdot (0.135 + 0.003 \cdot ST2(k-1) - 0.0203 \cdot ST3(k) - 0.001 \cdot SC2(k) + 0.00004 \cdot ST2(k-1) \cdot SC2(k)). \quad (4)$$

Estas fórmulas se han implementado en el paquete Simulador.adb para su posterior uso en el programa principal:

```

1  package body Simulador is
2
3      -- Implementación de funciones Ejercicio 1
4
5  function calc_Tt (ST1, ST2 : Float) return Float is
6      Tt : Float;
7      begin
8          Tt := (ST1 + ST2) / 2.0;
9          return Tt;
10     end calc_Tt;
11
12 function calc_ST2 (SR1, ST4, ST1, SC1, Tt:Float) return Float is
13     calc : Float;
14     begin
15         calc := ST1 + ((b*Leq*SR1*C)/(SC1*Cp*p)) - ((H*(Tt-ST4)*C)/(SC1*Cp*p));
16         return calc;
17     end calc_ST2;
18
19 function calc_ST1 (ST2 : Float) return Float is
20     ST1 : Float;
21     begin
22         ST1 := ST2 - 10.0;
23         return ST1;
24     end calc_ST1;
25
26 function calc_SD1 (SC2, ST2, ST3 : Float) return Float is
27     SD1 : Float;
28     begin
29         SD1 := 24.0*(0.135+0.003*ST2-0.0203*ST3-0.001*SC2+0.00004*ST2*SC2);
30         return SD1;
31     end calc_SD1;

```

Código 4. Simulador.adb con la implementación de las fórmulas necesarias.

En cada iteración, tras realizar los cálculos, se comprobará la restricción de seguridad sobre ST2 mencionada anteriormente, escribiendo el mensaje de alarma cuando salte la restricción.

Por último, se escribirán los datos de salida calculados tanto por pantalla como en el archivo de salida data.txt.

Una vez realizados estos pasos en cada iteración, se cerrarán los archivos de texto utilizados (input.txt, data.txt, alarm_log.txt).


```

76 -- Bucle for que itera las 15 filas de input.txt (entrada)
77 for i in 1 .. 15 loop
78 -- Cargar valores de entrada
79 Get(input, k);
80 Get(input, SR1(k));
81 Get(input, ST4(k));
82 Get(input, ST3(k));
83
84 -- Realizar cálculos necesarios
85 Tt(k) := calc_Tt(ST1(k-1), ST2(k-1));
86 ST2(k) := calc_ST2(SR1(k), ST4(k), ST1(k-1), SC1(k), Tt(k));
87 ST1(k) := calc_ST1(ST2(k));
88 SD1(k) := calc_SD1(SC2(k), ST2(k-1), ST3(k));
89
90 -- Restricción de seguridad en ST2
91 if (ST2(k) > 98.0) then
92 -- Escribir restricción de seguridad en archivo alarm_log.txt
93 Put_Line(alarm, "k =" & k'Img & ASCII.HT & "Alarma: ST2 tiene una temperatura > 98 °C: " & ST2(k)'Img & "°C");
94 -- Escribir restricción de seguridad por pantalla
95 Put_Line("k =" & k'Img & ASCII.HT & "Alarma: ST2 tiene una temperatura > 98 °C: " & ST2(k)'Img & "°C");
96 end if;
97
98 -- Escribir todos los valores actuales en data.txt (salida)
99 Put_Line(data,
100 k'Img & ASCII.HT &
101 ST1(k)'Img & ASCII.HT &
102 ST2(k)'Img & ASCII.HT &
103 ST3(k)'Img & ASCII.HT &
104 ST4(k)'Img & ASCII.HT &
105 SC1(k)'Img & ASCII.HT &
106 SC2(k)'Img & ASCII.HT &
107 SR1(k)'Img & ASCII.HT &
108 SD1(k)'Img);
109
110 -- Escribir todos los valores actuales por pantalla
111 Put_Line(k'Img & ASCII.HT &
112 ST1(k)'Img & ASCII.HT &
113 ST2(k)'Img & ASCII.HT &
114 ST3(k)'Img & ASCII.HT &
115 ST4(k)'Img & ASCII.HT &
116 SC1(k)'Img & ASCII.HT &
117 SC2(k)'Img & ASCII.HT &
118 SR1(k)'Img & ASCII.HT &
119 SD1(k)'Img);
120
121 end loop;
122
123 -- Cerrar archivos de texto utilizados
124 Close(input);
125 Close(data);
126 Close(alarm);

```

Código 5. main.adb con bucle del Ejercicio 1.

2.1 Salida de resultados por pantalla

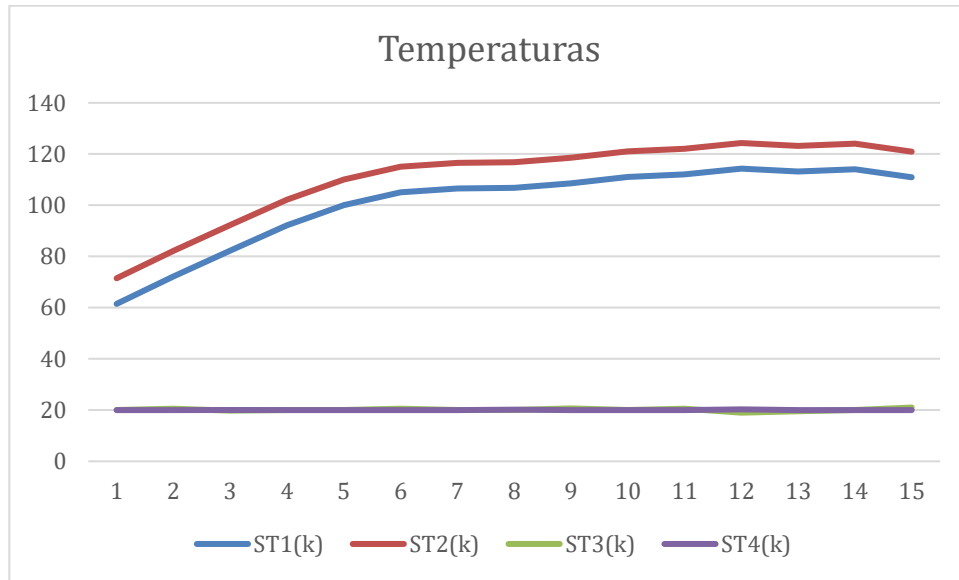
Ejercicio 1:

k	ST1(k)	ST2(k)	ST3(k)	ST4(k)	SC1(k)	SC2(k)	SR1(k)	SD1(k)
1	6.14841E+01	7.14841E+01	2.00000E+01	2.00000E+01	1.50000E+01	4.50000E+02	6.04000E+02	1.29360E+01
2	7.21586E+01	8.21586E+01	2.06000E+01	2.00000E+01	1.50000E+01	4.50000E+02	6.04000E+02	1.84317E+01
3	8.22859E+01	9.22859E+01	1.98000E+01	2.01000E+01	1.50000E+01	4.50000E+02	6.09000E+02	2.42014E+01
k = 4	Alarma: ST2 tiene una temperatura > 98 °C: 1.02128E+02°C							
4	9.21285E+01	1.02128E+02	2.00000E+01	2.00000E+01	1.50000E+01	4.50000E+02	6.20000E+02	2.92081E+01
k = 5	Alarma: ST2 tiene una temperatura > 98 °C: 1.10087E+02°C							
5	1.00087E+02	1.10087E+02	2.00000E+01	2.00000E+01	1.50000E+01	4.50000E+02	5.90000E+02	3.41687E+01
k = 6	Alarma: ST2 tiene una temperatura > 98 °C: 1.15106E+02°C							
6	1.05106E+02	1.15106E+02	2.05000E+01	2.00000E+01	1.50000E+01	4.50000E+02	5.30000E+02	3.79365E+01
k = 7	Alarma: ST2 tiene una temperatura > 98 °C: 1.16599E+02°C							
7	1.06599E+02	1.16599E+02	2.00000E+01	2.00000E+01	1.50000E+01	4.50000E+02	4.50000E+02	4.07095E+01
k = 8	Alarma: ST2 tiene una temperatura > 98 °C: 1.16810E+02°C							
8	1.06810E+02	1.16810E+02	2.00000E+01	2.02000E+01	1.50000E+01	4.50000E+02	4.20000E+02	4.14616E+01
k = 9	Alarma: ST2 tiene una temperatura > 98 °C: 1.18579E+02°C							
9	1.08579E+02	1.18579E+02	2.07000E+01	2.00000E+01	1.50000E+01	4.50000E+02	4.60000E+02	4.12273E+01
k = 10	Alarma: ST2 tiene una temperatura > 98 °C: 1.21016E+02°C							
10	1.11016E+02	1.21016E+02	2.00000E+01	2.00000E+01	1.50000E+01	4.50000E+02	4.80000E+02	4.24598E+01
k = 11	Alarma: ST2 tiene una temperatura > 98 °C: 1.22092E+02°C							
11	1.12092E+02	1.22092E+02	2.05000E+01	2.00000E+01	1.50000E+01	4.50000E+02	4.50000E+02	4.34446E+01
k = 12	Alarma: ST2 tiene una temperatura > 98 °C: 1.24303E+02°C							
12	1.14303E+02	1.24303E+02	1.90000E+01	2.03000E+01	1.50000E+01	4.50000E+02	4.80000E+02	4.47176E+01
k = 13	Alarma: ST2 tiene una temperatura > 98 °C: 1.23164E+02°C							
13	1.13164E+02	1.23164E+02	1.95000E+01	2.00000E+01	1.50000E+01	4.50000E+02	4.00000E+02	4.55882E+01
k = 14	Alarma: ST2 tiene una temperatura > 98 °C: 1.24088E+02°C							
14	1.14088E+02	1.24088E+02	2.00000E+01	2.00000E+01	1.50000E+01	4.50000E+02	4.50000E+02	4.47707E+01
k = 15	Alarma: ST2 tiene una temperatura > 98 °C: 1.20982E+02°C							
15	1.10982E+02	1.20982E+02	2.10000E+01	2.00000E+01	1.50000E+01	4.50000E+02	3.50000E+02	4.47494E+01

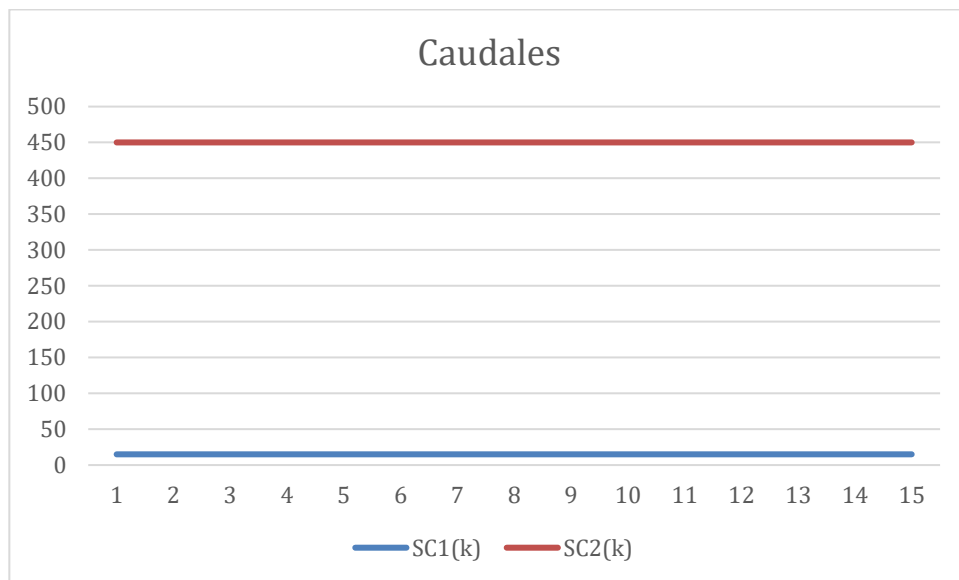
Imagen 1. Salida de resultados por pantalla Ejercicio 1.

2.2 Representación gráfica

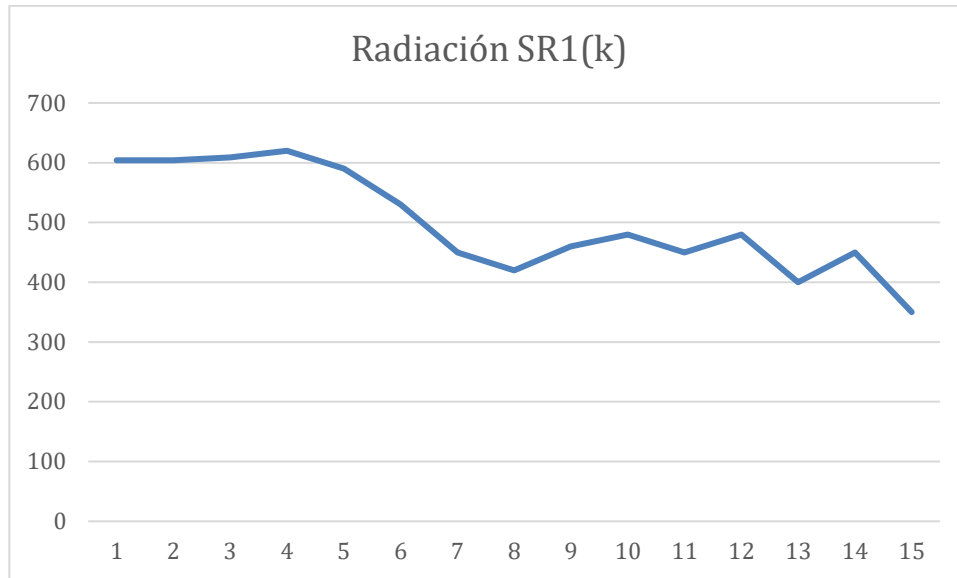
2.2.1 Representación gráfica temperaturas



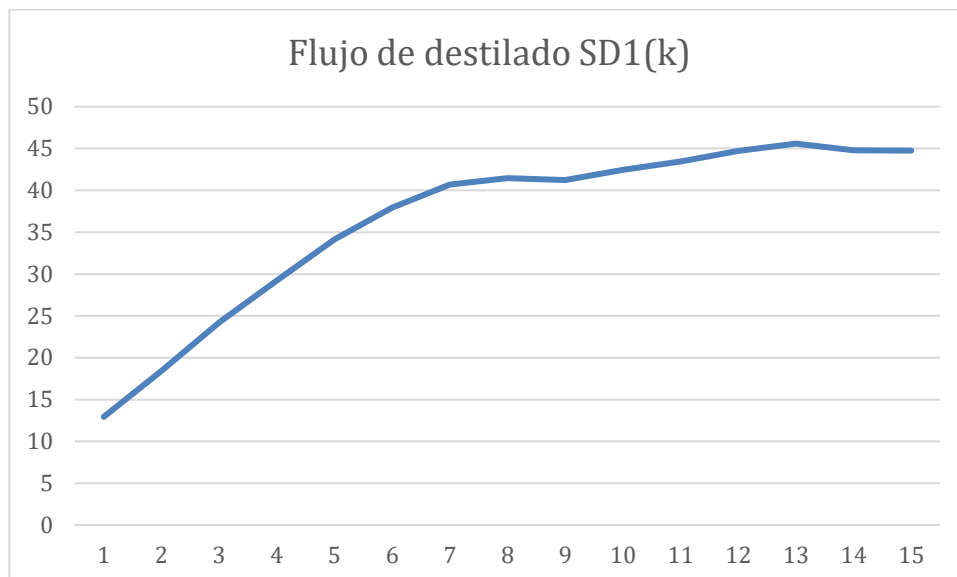
2.2.2 Representación gráfica caudales



2.2.3 Representación gráfica radiación



2.2.4 Representación gráfica flujo de destilado



2.3 Pregunta 1: ¿Se está operando el campo solar y el módulo de forma óptima?

Como podemos observar en las gráficas, los valores de la simulación de ST2 y los de SD1 no se acercan a sus valores óptimos, 82 °C y 25 L/h respectivamente, sino que los supera considerablemente. Por lo tanto, no se está operando de forma óptima.

2.4 Pregunta 2: ¿Se incumple las restricciones de seguridad en algún momento?

Sí, como podemos observar en la imagen de salida de los datos por pantalla, salta el mensaje de alarma de ST2, ya que supera los 98 °C, constantemente desde la iteración k=4, llegando incluso a los 124 °C en las iteraciones k=12 y k=14.

SD1 también supera su valor óptimo desde la iteración k=4, alcanzando incluso los 45 L/h en la iteración k=13.

3. Ejercicio 2

Al igual que para el Ejercicio 1, se abrirán o crearán los archivos de texto utilizados, y se escribirá tanto por pantalla como en data2.txt la cabecera de los datos resultantes:

```

130      -- Ejercicio 2
131      Put_Line("");
132      Put_Line("Ejercicio 2:");
133      Put_Line("");
134
135      -- Abrir input.txt (entrada)
136      Open(input, In_File, "input.txt");
137      -- Crear data2.txt (salida Ejercicio 2)
138      Create(data2, Out_File, "data2.txt");
139      -- Crear alarm_log2.txt (alarmas Ejercicio 2)
140      Create(alarm2, Out_File, "alarm_log2.txt");
141
142      -- Escribir cabecera de los datos en archivo data2.txt (salida)
143      Put_Line(data2,
144              " k" & ASCII.HT &
145              " ST1(k)" & ASCII.HT & ASCII.HT &
146              " ST2(k)" & ASCII.HT & ASCII.HT &
147              " ST3(k)" & ASCII.HT & ASCII.HT &
148              " ST4(k)" & ASCII.HT & ASCII.HT &
149              " SC1(k)" & ASCII.HT & ASCII.HT &
150              " SC2(k)" & ASCII.HT & ASCII.HT &
151              " SR1(k)" & ASCII.HT & ASCII.HT &
152              " SD1(k)");
153
154      -- Escribir cabecera de los datos por pantalla
155      Put_Line(" k" & ASCII.HT &
156              " ST1(k)" & ASCII.HT & ASCII.HT &
157              " ST2(k)" & ASCII.HT & ASCII.HT &
158              " ST3(k)" & ASCII.HT & ASCII.HT &
159              " ST4(k)" & ASCII.HT & ASCII.HT &
160              " SC1(k)" & ASCII.HT & ASCII.HT &
161              " SC2(k)" & ASCII.HT & ASCII.HT &
162              " SR1(k)" & ASCII.HT & ASCII.HT &
163              " SD1(k)");

```

Código 7. main.adb con el principio del Ejercicio 2.

Para realizar el Ejercicio 2, se debe calcular los controladores del campo solar y del módulo. Para ello, despejamos las variables de control (SC1 y SC2) de las ecuaciones (1) y (4), y sustituimos el valor de ST2(k) y SD1(k) por sus puntos de operación óptimos. Estas fórmulas se han implementado en el paquete Simulador.adb para su posterior uso en el programa principal:

```

33  -- Implementación de funciones Ejercicio 2
34
35  function calc_SC1(SR1, Tt, ST4, ST2, ST1:Float) return Float is
36      SC1: Float;
37      begin
38          SC1 := (((b*Leq*SR1) - (H*(Tt - ST4)))*c) / (Cp*p*(ST2 - ST1));
39          return SC1;
40      end calc_SC1;
41
42  function calc_SC2(SD1, ST2, ST3:Float) return Float is
43      SC2: Float;
44      begin
45          SC2 := ((SD1/24.0) - 0.135 - 0.003*ST2 + 0.0203*ST3) / (0.00004*ST2 - 0.001);
46          return SC2;
47      end calc_SC2;
48
49  end Simulador;

```

Código 6. main.adb con la implementación de las fórmulas necesarias.

Respecto al Ejercicio 1, en el Ejercicio 2 se añaden las nuevas fórmulas, y por último, el enunciado indica que se debe tener en cuenta que $7.5 \leq SC1(k) \leq 30$ L/h y que $400 \leq SC2(k) \leq 600$ L/h.

```

165  -- Bucle for que itera las 15 filas de input.txt (entrada)
166  for i in 1 .. 15 loop
167      -- Cargar valores de entrada
168      Get(input, k);
169      Get(input, SR1(k));
170      Get(input, ST4(k));
171      Get(input, ST3(k));
172
173      -- Realizar cálculos necesarios
174      Tt(k) := calc_Tt(ST1(k-1), ST2(k-1));
175      SC1(k) := calc_SC1(SR1(k), Tt(k), ST4(k), ST2_Opt, ST1(k-1));
176      SC2(k) := calc_SC2(SD1_Opt, ST2(k-1), ST3(k));
177      ST2(k) := calc_ST2(SR1(k), ST4(k), ST1(k-1), SC1(k), Tt(k));
178      ST1(k) := calc_ST1(ST2(k));
179      SD1(k) := calc_SD1(SC2(k), ST2(k-1), ST3(k));
180
181      -- Control de rangos seguros
182      if(SC1(k) < 7.5) then
183          SC1(k) := 7.5;
184      elsif(SC1(k) > 30.0) then
185          SC1(k) := 30.0;
186      end if;
187      if(SC2(k) < 400.0) then
188          SC2(k) := 400.0;
189      elsif(SC2(k) > 600.0) then
190          SC2(k) := 600.0;
191      end if;
192
193      -- Restricción de seguridad en ST2
194      if (ST2(k) > 98.0) then
195          -- Escribir restricción de seguridad en archivo alarm_log.txt
196          Put_Line(alarm2, "k =" & k'Img & ASCII.HT & "Alarma: ST2 tiene una temperatura > 98 ºC: " & ST2(k)'Img & "ºC");
197          -- Escribir restricción de seguridad por pantalla
198          Put_Line("k =" & k'Img & ASCII.HT & "Alarma: ST2 tiene una temperatura > 98 ºC: " & ST2(k)'Img & "ºC");
199      end if;
200
201      -- Escribir todos los valores actuales en data.txt (salida)
202      Put_Line(data2,
203          k'Img & ASCII.HT &
204          ST1(k)'Img & ASCII.HT &
205          ST2(k)'Img & ASCII.HT &
206          ST3(k)'Img & ASCII.HT &
207          ST4(k)'Img & ASCII.HT &
208          SC1(k)'Img & ASCII.HT &
209          SC2(k)'Img & ASCII.HT &
210          SR1(k)'Img & ASCII.HT &
211          SD1(k)'Img);
212
213      -- Escribir todos los valores actuales por pantalla
214      Put_Line(k'Img & ASCII.HT &
215          ST1(k)'Img & ASCII.HT &
216          ST2(k)'Img & ASCII.HT &
217          ST3(k)'Img & ASCII.HT &
218          ST4(k)'Img & ASCII.HT &
219          SC1(k)'Img & ASCII.HT &
220          SC2(k)'Img & ASCII.HT &
221          SR1(k)'Img & ASCII.HT &
222          SD1(k)'Img);
223
224  end loop;
225
226  -- Cerrar archivos de texto utilizados
227  Close(input);
228  Close(data2);
229  Close(alarm2);
230
231  end Main;

```

Código 8. main.adb con bucle del Ejercicio 2.

3.1 Salida de los datos

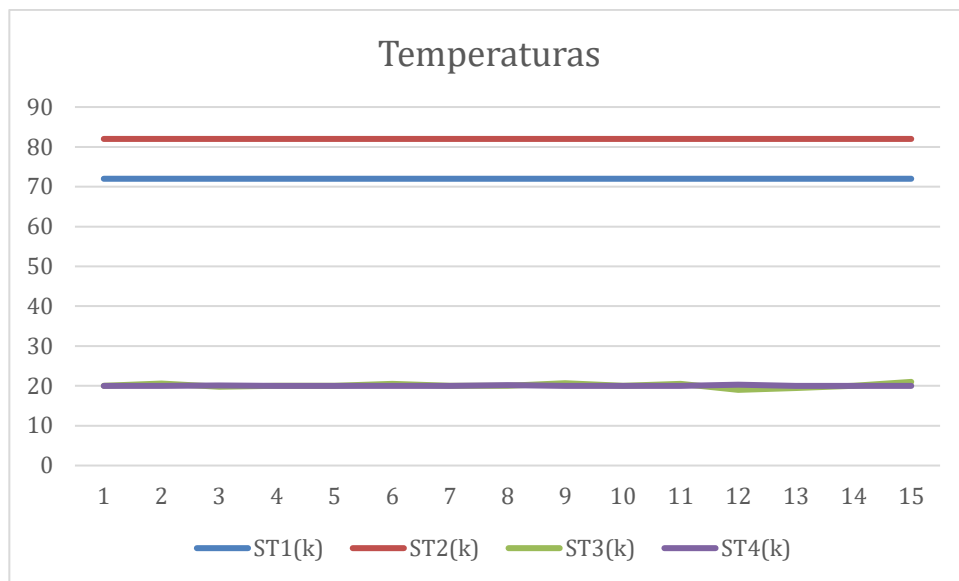
Ejercicio 2:

k	ST1(k)	ST2(k)	ST3(k)	ST4(k)	SC1(k)	SC2(k)	SR1(k)	SD1(k)
1	7.20000E+01	8.20000E+01	2.00000E+01	2.00000E+01	1.00707E+01	6.00000E+02	6.04000E+02	2.50000E+01
2	7.20000E+01	8.20000E+01	2.06000E+01	2.00000E+01	2.98998E+01	4.73178E+02	6.04000E+02	2.50000E+01
3	7.20000E+01	8.20000E+01	1.98000E+01	2.01000E+01	3.00000E+01	4.66056E+02	6.09000E+02	2.50000E+01
4	7.20000E+01	8.20000E+01	2.00000E+01	2.00000E+01	3.00000E+01	4.67836E+02	6.20000E+02	2.50000E+01
5	7.20000E+01	8.20000E+01	2.00000E+01	2.00000E+01	2.90670E+01	4.67836E+02	5.90000E+02	2.50000E+01
6	7.20000E+01	8.20000E+01	2.05000E+01	2.00000E+01	2.54981E+01	4.72288E+02	5.30000E+02	2.50000E+01
7	7.20000E+01	8.20000E+01	2.00000E+01	2.00000E+01	2.07395E+01	4.67836E+02	4.50000E+02	2.50000E+01
8	7.20000E+01	8.20000E+01	2.00000E+01	2.02000E+01	1.89762E+01	4.67836E+02	4.20000E+02	2.50000E+01
9	7.20000E+01	8.20000E+01	2.07000E+01	2.00000E+01	2.13343E+01	4.74069E+02	4.60000E+02	2.50000E+01
10	7.20000E+01	8.20000E+01	2.00000E+01	2.00000E+01	2.25240E+01	4.67836E+02	4.80000E+02	2.50000E+01
11	7.20000E+01	8.20000E+01	2.05000E+01	2.00000E+01	2.07395E+01	4.72288E+02	4.50000E+02	2.50000E+01
12	7.20000E+01	8.20000E+01	1.90000E+01	2.03000E+01	2.25557E+01	4.58933E+02	4.80000E+02	2.50000E+01
13	7.20000E+01	8.20000E+01	1.95000E+01	2.00000E+01	1.77654E+01	4.63385E+02	4.00000E+02	2.50000E+01
14	7.20000E+01	8.20000E+01	2.00000E+01	2.00000E+01	2.07395E+01	4.67836E+02	4.50000E+02	2.50000E+01
15	7.20000E+01	8.20000E+01	2.10000E+01	2.00000E+01	1.47913E+01	4.76740E+02	3.50000E+02	2.50000E+01

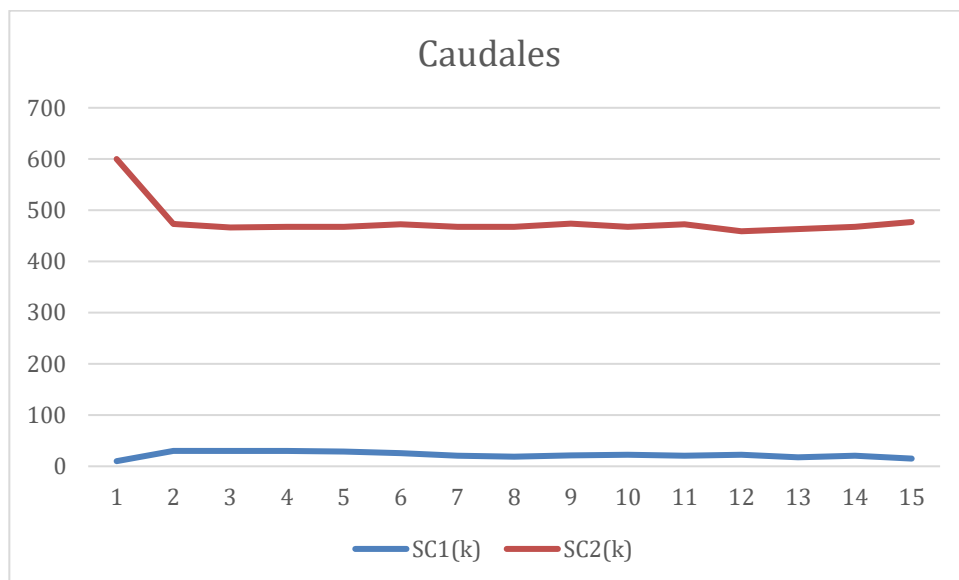
Imagen 3. Salida de resultados por pantalla Ejercicio 2.

3.2 Representación gráfica

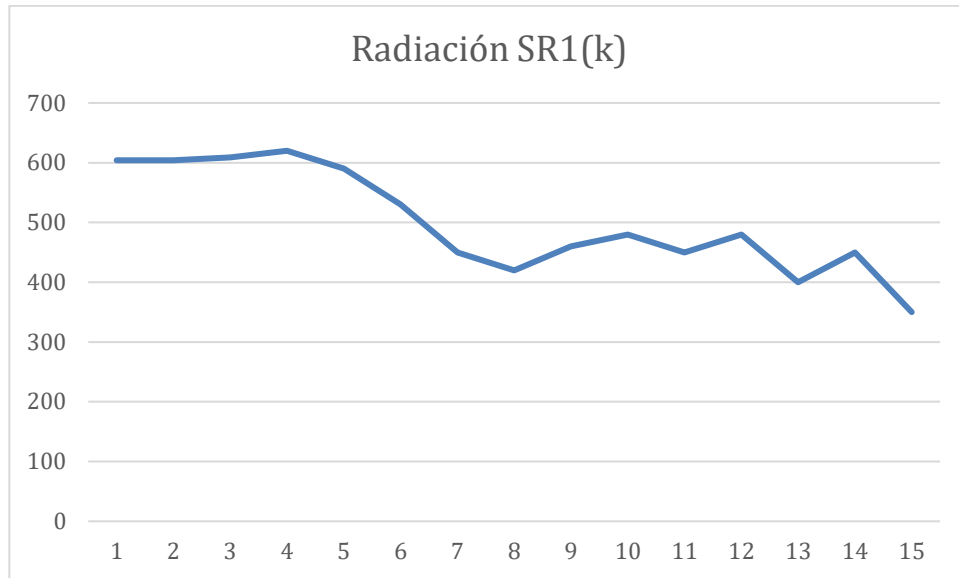
3.2.1 Representación gráfica temperaturas



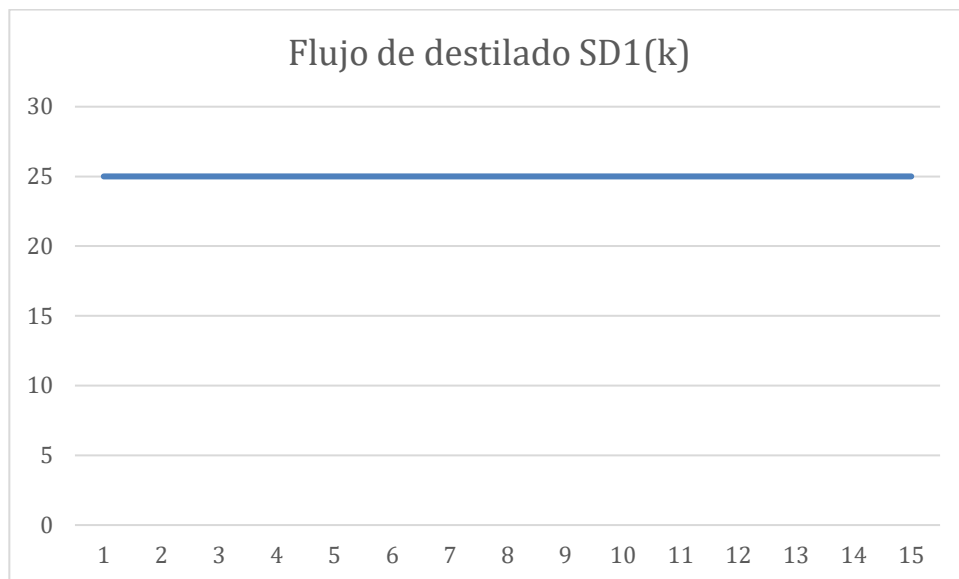
3.2.2 Representación gráfica caudales



3.2.3 Representación gráfica radiación



3.2.4 Representación gráfica flujo de destilado



3.3 Pregunta: ¿Se cumplen en este caso las condiciones de operación óptimas?

Sí. Como podemos observar en las gráficas, tanto ST2 como SD1 se mantienen en sus valores óptimos durante la simulación, 82 °C y 25 L/h respectivamente.