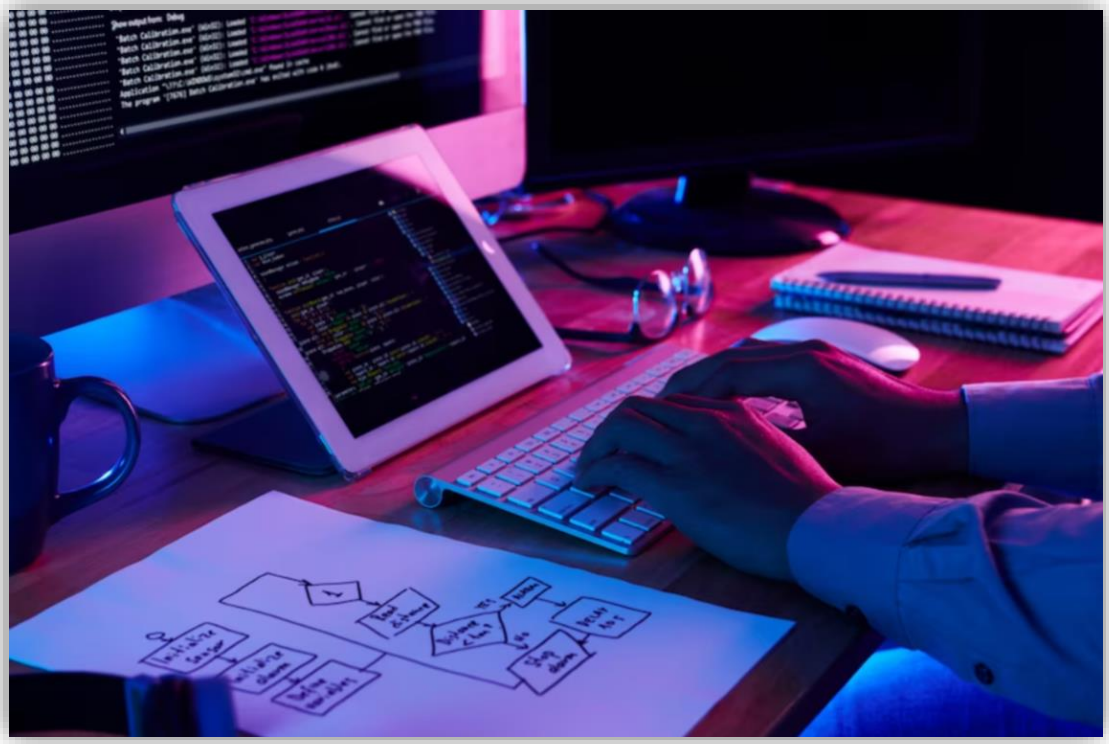
UNIVERSIDAD DE ALMERÍA

PRÁCTICA 1

Introducción a la programación en tiempo real.



SISTEMAS DE TIEMPO REAL

3º Ingeniería Informática

Lucas Barrientos Muñoz

Adrián Antequera Ramírez

Contenido

Objetivo.

El lenguaje de programación **Ada** fue diseñado en un esfuerzo de colaboración, patrocinado por el **Departamento de Defensa** de los **EE. UU.** con la participación de la industria, mundo académico y comunidad internacional. Su propósito principal fue proporcionar un **lenguaje de alto nivel** en el que pudieran expresarse, desarrollarse y mantenerse los problemas de programación de sistemas. **Ada** contiene **mecanismos especiales** para la gestión de **sucesos concurrentes** en un entorno de **tiempo real**, desarrollando paquetes específicos de la aplicación y definiendo **operadores** y **procedimientos genéricos**. Fue a principios de los 70s cuando el Departamento de Defensa de los EE. UU., identificó un grave problema en el crecimiento del coste del software en los sistemas de computadoras "empotradas", es decir, sistemas que están incluidos en distintos vehículos militares.

Los **objetivos** de esta práctica son:

- **Aprendizaje** de los **aspectos básicos** de la creación de programas en el lenguaje de programación **Ada**.
- **Analizar** los conceptos más importantes relacionados con la planificación en **sistemas de tiempo real**.
- **Definir** un **esquema** de la **planificación** basado en un ejecutivo cíclico.
- **Evaluar** las **ventajas** e **inconvenientes** de este método.

#	Tipo	Tarea	Peso	
1	Implementación	Paquete simulador	70%	70%
2	Implementación	Procedimientos de lectura y escritura de .txt		
3	Implementación	Procedimientos de seguridad		
4	Implementación	Procedimientos de almacenamiento y visualización		
5	Implementación	Procedimientos caudales óptimos		
6	Documentación	Estudio de la implementación y comparativas	15%	30%
7	Documentación	Estudio teórico	10%	
8	Documentación	Anexos y Bibliografía	5%	

- Lucas Barrientos Muñoz ➞ #1, #2, #5, #7, #8.
- Adrián Antequera Ramírez ➞ #3, #4, #6, #8.

Si hay alguna tarea asignada a ambos, esta se hace de manera conjunta. Una vez finalizadas todas las tareas, se procede a revisar la memoria entre ambos para corregir posibles errores.

Antecedentes

Esta memoria responde al requerimiento de implementar un sistema de control para el control de una instalación de destilación por membranas, en la cual, a través de un proceso de evaporación se produce agua desalada usando como alimentación agua de mar. Para ello, se ha hecho uso de librerías como **Ada.Text_IO** y **Ada.Float_Text_IO** las cuales se utilizan para la lectura y escritura de documentos de texto plano, como para la impresión en pantalla de los resultados.

Paquete simulador.

Estudio previo

En la figura 1, se presenta el diagrama esquemático de una instalación de destilación por membranas alimentada por energía solar.

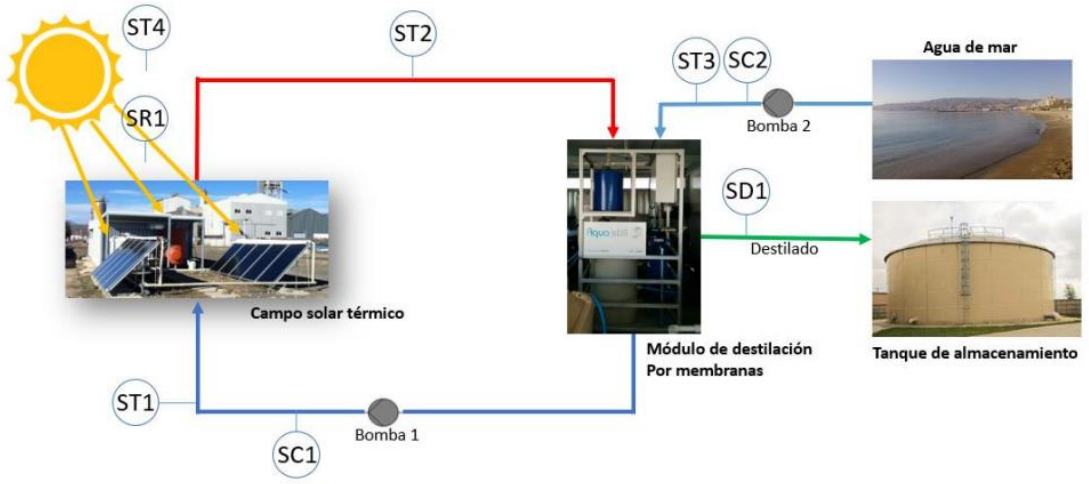


Figura 1: Diagrama esquemático de la instalación de destilación por membranas. ***ST*** *son sensores de temperatura,* ***SC*** *son sensores de caudal,* ***SR*** es un sensor de radiación y **SD** es un sensor de destilado.

Para **asegurar** su **eficacia**, se requieren **dos procesos de control esenciales**. En primer lugar, se debe regular la temperatura de salida

del campo solar ajustando el caudal de entrada de este, en función de la radiación solar medida y las condiciones ambientales. En segundo lugar, se controla el caudal de entrada al módulo de destilación para mantener un nivel de producción específico, considerando la temperatura del agua de mar y la del campo solar.

Por **motivos de seguridad**, se establece un **límite** de **temperatura** en la **salida del campo solar**, evitando que supere los **98 °C**. En caso de exceder este límite, se activará una alarma visual y se registrará la incidencia junto con el número de iteración en un archivo de historial de alarmas denominado "**alarm_log.txt**".

Además, se implementará una **interfaz de visualización** para supervisar los actuadores y alarmas en tiempo real, así como un **sistema de registro histórico** para almacenar y analizar los datos de operación de la instalación.

Se pide montar un simulador de la planta real en base a las siguientes ecuaciones:

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

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

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

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

Donde el valor de cada una de las constantes se define en la siguiente tabla:

Constante	Valor
β	0.15 [m]
L_{eq}	15 [m]
H	4 [J/s K]
C	9·2·6·10000 [s L/min m^3]
C_p	4190 [J/kg °C]
p	975 [kg/m^3]

Para ello, se deberán encapsular en un paquete llamado "**Simulador**". Una vez montado, se debe simular la planta usando como datos de entrada los contenidos en el archivo "**input.txt**".

Los valores para los caudales (**SC1** y **SC2**) se tomarán fijos en toda la simulación, con un valor de 15 [L/min] y 450 [L/h] para **SC1** y **SC2**,

respectivamente. Además, se debe tomar como valores iniciales para la simulación **ST2(0) = 60** [ºC] y **ST1(0) = 50** [ºC].

Entonces, el sistema de control estará compuesto por los siguientes elementos:

Sensores de temperatura:

- **ST1**: Mide la temperatura de entrada al campo.
- **ST2**: Mide la temperatura de salida al campo.
- **ST3**: Mide la temperatura del mar.
- **ST4**: Mide la temperatura ambiente.

Sensores de control:

- **SC1**: Señal de control (caudal de entrada del campo solar).
- **SC2**: Señal de control (caudal de entrada al módulo de destilación).

Sensor de destilado:

- **SD1**: Mide el nivel de producción de destilado.

Sensor de radiación:

- **SR1**: Mide los niveles de radiación solar.

Los datos de entrada del archivo "**input.txt**", se presentan a continuación en la siguiente tabla:

k	SR1 [W/m²]	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

Código

Para la realización de esta práctica, crearemos un paquete llamado "**Simulador**" y otro llamado "**ProcesamientoDatos**" que contendrán los correspondientes archivos con extensión "**.adb"** y "**.ads"**. También, se utilizará el main para llamar a los procedimientos de **ProcesamientoDatos**.

Simulador.ads

En este archivo se encapsularán las funciones y constantes que utilizaremos en el **body** del paquete. Las declararemos como **Float** para poder operar con ellas correctamente.

```
package Simulador is

   b : constant Float := 0.15;
   Leq : constant Float := 15.0;
   c : constant Float := 9.0*2.0*6.0*10000.0;
   Cp : constant Float := 4190.0;
   p : constant Float := 975.0;
   H : constant Float := 4.0;

   function ST1_f(ST2: Float) return Float;
   function ST2_f(SR1, ST4, ST1, SC1, Tt : Float) return Float;
   function Tt_f(ST1, ST2: Float) return Float;
   function SD1_f(SC2, ST2, ST3 : Float) return Float;

   function SC1_f(SR1, Tt, ST4, ST2, ST1 : Float) return Float;
   function SC2_f(SD1, ST2, ST3 : Float) return Float;
end Simulador;
```

Simulador.adb

En este archivo, desarrollaremos los procedimientos necesarios para llevar a cabo la simulación, es decir, introduciremos las fórmulas dadas en el guión de la práctica para la correcta simulación del sistema de control.

```
package body Simulador is

   function ST1_f(ST2 : Float) return Float is
      ST1 : Float;
   begin
      ST1 := ST2 - 10.0;
      return ST1;
```

```
   end ST1_f;

   function ST2_f(SR1 : Float; ST4 : Float; ST1 : Float; SC1 : Float; Tt
: Float) return Float is
      sol : Float;
   begin
      sol := ST1 + ((b*Leq*SR1*c)/(SC1*Cp*p)) - ((H*(Tt-
ST4)*c)/(SC1*Cp*p));
      return sol;
   end ST2_f;

   function Tt_f(ST1, ST2 : Float) return Float is
      Tt : Float;
   begin
      Tt := (ST1 + ST2) / 2.0;
      return Tt;
   end Tt_f;

   function SD1_f(SC2, ST2, ST3 :Float) return Float is
      SD1 : Float;
   begin
      SD1 := 24.0*(0.135+0.003*ST2-0.0203*ST3-0.001*SC2+0.0004*ST2*SC2);
      return SD1;
   end SD1_f;

   -- Procedimientos Ejercicio 2

   function SC1_f(SR1, Tt, ST4, ST2, ST1 : Float) return Float is
      SC1 : Float;
   begin
      SC1 := (((b*Leq*SR1) - (H*(Tt - ST4)))*c) / (Cp*p*(ST2 - ST1));
      return SC1;
   end SC1_f;

   function SC2_f(SD1, ST2, ST3 : Float) return Float is
      SC2 : Float;
   begin
      SC2 := ((SD1/24.0) - 0.135 - 0.003*ST2 + 0.0203*ST3) / (0.00004*ST2
- 0.001);
      return SC2;
   end SC2_f;

end Simulador;
```

Paquete ProcesamientoDatos.

Código.

Primero, declararemos los valores óptimos predefinidos e inicializaremos los **arrays** que se utilizarán tanto para leer como para escribir los datos y resultados. También asignaremos los valores fijos utilizados en el **Ejercicio 1**.

```
procedure Procesamiento_Datos is
   -- Variables utilizadas para el contador
   i : Integer;
   k : Float;

   -- Archivos de texto
   input : File_Type;
   output : File_Type;
   alarm_log : File_Type; -- Nuevo archivo para el registro de alarmas

   -- Temperaturas Optimas
   ST2_Opt : Float := 82.0;
   SD1_Opt : Float := 26.5;

   type Vector is Array(Integer range<>) of Float;
   ST3, ST4, SR1, SD1, Tt : Vector(1..15);
   ST1, ST2, SC1, SC2 : Vector(0..15);

begin
   -- Valores Fijos Ejercicio 1
   SC1(0) := 15.0;
   SC2(0) := 450.0;

   -- Estados iniciales de ST1 y ST2
   ST1(0) := 50.0;
   ST2(0) := 60.0;
```

Ahora, utilizaremos los procedimientos **Open** y **Create** para abrir y crear los diferentes archivos que utilizaremos, en este caso, **Open** lo utilizaremos para los datos de entrada del archivo "**input.txt**", y **Create** lo utilizaremos para los archivos de salida "**output.txt**" y "**alarm_log.txt**".

```
   -- Lectura del archivo de entrada
   Open(input, In_File, "input.txt");

   -- Creación del archivo de salida
   Create(output, Out_File, "output.txt");
```

```
-- Apertura del archivo de registro de alarmas
Create(alarm_log, Out_File, "alarm_log.txt");
```

Para el archivo de salida "**output.txt**" le pondremos el encabezado con las siguientes líneas, el atributo "**ASCII.HT**", añade un espacio tabulado.

```
-- Encabezado para el archivo de salida
Put(output, "k"); Put(output, ASCII.HT);
Put(output, "ST1"); Put(output, ASCII.HT);
Put(output, "ST2"); Put(output, ASCII.HT);
Put(output, "ST3"); Put(output, ASCII.HT);
Put(output, "ST4"); Put(output, ASCII.HT);
Put(output, "SC1"); Put(output, ASCII.HT);
Put(output, "SC2"); Put(output, ASCII.HT);
Put(output, "SR1"); Put(output, ASCII.HT);
Put(output, "SD1"); New_Line(output);
```

Haremos lo mismo para la salida por consola, pero en vez de pasarle al **Put** dos parámetros, que son, el archivo destino y lo que tiene que escribir, le pasaremos solo lo que necesitamos que escriba.

```
-- Encabezado para la salida en consola
Put("k"); Put(ASCII.HT);
Put("ST1"); Put(ASCII.HT);
Put("ST2"); Put(ASCII.HT);
Put("ST3"); Put(ASCII.HT);
Put("ST4"); Put(ASCII.HT);
Put("SC1"); Put(ASCII.HT);
Put("SC2"); Put(ASCII.HT);
Put("SR1"); Put(ASCII.HT);
Put("SD1"); New_Line;
```

Ahora, crearemos un bucle **while**, que no acabará hasta que se lea el archivo completo, antes del bucle, inicializaremos el contador **i** a 1, y al final de cada vuelta, se le añadirá 1, así podemos controlar las iteraciones **k**.

```
i := 1;
while not End_Of_File(input) loop

-- Resto del código

   i := i + 1;
end loop;
```

También llamaremos a los procedimientos de las fórmulas definidos en el paquete "**Simulador**" y le daremos los valores necesarios a cada una de las variables o constantes.

```
      Tt(i) := Tt_f(ST1(i-1), ST2(i-1));
      ST2(i) := ST2_f(SR1(i), ST4(i), ST1(i-1), SC1(0), Tt(i));
      ST1(i) := ST1_f(ST2(i));
      SD1(i) := SD1_f(SC2(0), ST2(i-1), ST3(i));
```

Para implementar las medidas de seguridad, haremos un condicional **if**, que controle la temperatura del sensor **ST2**, en caso de que en alguna iteración sobrepase los **98 ºC**, se activará el procedimiento de seguridad, el cual escribirá un aviso en consola de la iteración en la que sobrepasa la temperatura y registrará el aviso en el archivo "**alarm_log.txt**".

```
      if ST2(i) > 98.0 then
         Put_Line("¡Cuidado! ST2 tiene una temperatura de: " &
Float'Image(ST2(i)) & "ºC");
         Put_Line("Mensaje de alarma registrado en el archivo
alarm_log.txt.");
         Put(alarm_log, "Iteración: "); Put(alarm_log, Integer'Image(i));
Put(alarm_log, ", Temperatura: "); Put(alarm_log, Float'Image(ST2(i)));
Put(alarm_log, "ºC");
         New_Line(alarm_log);
      end if;
```

Por último escribiremos el resultado en el archivo "**output.txt**" e imprimiremos por consola los resultados junto con los mensajes de alarma, de la misma forma que antes, utilizando **Put**, y asignándole uno o dos parámetros en función de los que sea necesario.

```
      Put(output, Integer'Image(i)); Put(output, ASCII.HT);
      Put(output, Float'Image(ST1(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(ST2(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(ST3(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(ST4(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(SC1(0))); Put(output, ASCII.HT);
      Put(output, Float'Image(SC2(0))); Put(output, ASCII.HT);
      Put(output, Float'Image(SR1(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(SD1(i))); New_Line(output);

      Put(Integer'Image(i)); Put(ASCII.HT);
      Put(Float'Image(ST1(i))); Put(ASCII.HT);
      Put(Float'Image(ST2(i))); Put(ASCII.HT);
      Put(Float'Image(ST3(i))); Put(ASCII.HT);
      Put(Float'Image(ST4(i))); Put(ASCII.HT);
```

```
      Put(Float'Image(SC1(0))); Put(ASCII.HT);
      Put(Float'Image(SC2(0))); Put(ASCII.HT);
      Put(Float'Image(SR1(i))); Put(ASCII.HT);
      Put(Float'Image(SD1(i))); New_Line;
```

Fuera del bucle **while**, cerraremos los archivos que hemos abierto o creado, haciendo uso del procedimiento **Close**.

```
   Close(input);
   Close(output);
   Close(alarm_log);
```

Aquí está el código completo:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Float_Text_IO; use Ada.Float_Text_IO;
with Simulador; use Simulador;

procedure Procesamiento_Datos is
   -- Variables utilizadas para el contador
   i : Integer;
   k : Float;

   -- Archivos de texto
   input : File_Type;
   output : File_Type;
   alarm_log : File_Type; -- Nuevo archivo para el registro de alarmas

   -- Temperaturas Óptimas
   ST2_Opt : Float := 82.0;
   SD1_Opt : Float := 26.5;

   type Vector is Array(Integer range<>) of Float;
   ST3, ST4, SR1, SD1, Tt : Vector(1..15);
   ST1, ST2, SC1, SC2 : Vector(0..15);

begin
   -- Valores Fijos Ejercicio 1
   SC1(0) := 15.0;
   SC2(0) := 450.0;

   -- Estados iniciales de ST1 y ST2
   ST1(0) := 50.0;
   ST2(0) := 60.0;

   -- Lectura del archivo de entrada
   Open(input, In_File, "input.txt");
```

```
-- Creaci?n del archivo de salida
Create(output, Out_File, "output.txt");

-- Apertura del archivo de registro de alarmas
Create(alarm_log, Out_File, "alarm_log.txt");

-- Encabezado para el archivo de salida
Put(output, "k"); Put(output, ASCII.HT);
Put(output, "ST1"); Put(output, ASCII.HT);
Put(output, "ST2"); Put(output, ASCII.HT);
Put(output, "ST3"); Put(output, ASCII.HT);
Put(output, "ST4"); Put(output, ASCII.HT);
Put(output, "SC1"); Put(output, ASCII.HT);
Put(output, "SC2"); Put(output, ASCII.HT);
Put(output, "SR1"); Put(output, ASCII.HT);
Put(output, "SD1"); New_Line(output);

-- Resultados
Put_Line("Resultados del Ejercicio 1:");
New_Line;

-- Encabezado para la salida en consola
Put("k"); Put(ASCII.HT);
Put("ST1"); Put(ASCII.HT);
Put("ST2"); Put(ASCII.HT);
Put("ST3"); Put(ASCII.HT);
Put("ST4"); Put(ASCII.HT);
Put("SC1"); Put(ASCII.HT);
Put("SC2"); Put(ASCII.HT);
Put("SR1"); Put(ASCII.HT);
Put("SD1"); New_Line;

-- Procesamiento del archivo de entrada y escritura de resultados
i := 1;
while not End_Of_File(input) loop
   Get(input, k);
   Get(input, SR1(i));
   Get(input, ST4(i));
   Get(input, ST3(i));

   Tt(i) := Tt_f(ST1(i-1), ST2(i-1));
   ST2(i) := ST2_f(SR1(i), ST4(i), ST1(i-1), SC1(0), Tt(i));
   ST1(i) := ST1_f(ST2(i));
   SD1(i) := SD1_f(SC2(0), ST2(i-1), ST3(i));

   -- Advertencia de seguridad
   if ST2(i) > 98.0 then
```

```
         Put_Line("¡Cuidado! ST2 tiene una temperatura de: " &
Float'Image(ST2(i)) & "ºC");
         Put_Line("Mensaje de alarma registrado en el archivo
alarm_log.txt.");
         Put(alarm_log, "Iteración: "); Put(alarm_log, Integer'Image(i));
Put(alarm_log, ", Temperatura: "); Put(alarm_log, Float'Image(ST2(i)));
Put(alarm_log, "ºC");
         New_Line(alarm_log);
      end if;

      Put(output, Integer'Image(i)); Put(output, ASCII.HT);
      Put(output, Float'Image(ST1(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(ST2(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(ST3(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(ST4(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(SC1(0))); Put(output, ASCII.HT);
      Put(output, Float'Image(SC2(0))); Put(output, ASCII.HT);
      Put(output, Float'Image(SR1(i))); Put(output, ASCII.HT);
      Put(output, Float'Image(SD1(i))); New_Line(output);

      Put(Integer'Image(i)); Put(ASCII.HT);
      Put(Float'Image(ST1(i))); Put(ASCII.HT);
      Put(Float'Image(ST2(i))); Put(ASCII.HT);
      Put(Float'Image(ST3(i))); Put(ASCII.HT);
      Put(Float'Image(ST4(i))); Put(ASCII.HT);
      Put(Float'Image(SC1(0))); Put(ASCII.HT);
      Put(Float'Image(SC2(0))); Put(ASCII.HT);
      Put(Float'Image(SR1(i))); Put(ASCII.HT);
      Put(Float'Image(SD1(i))); New_Line;

      i := i + 1;
   end loop;

   -- Cierre de archivos
   Close(input);
   Close(output);
   Close(alarm_log); -- Cerrar el archivo de registro de alarmas

   end Procesamiento_Datos;

end ProcesamientoDatos;
```

Ejecución y resultados.

Cuando ejecutamos el programa en el compilador **GNAT Studio**, el resultado imprimido en consola es el siguiente:

```
Resultados del Ejercicio 1:

k       ST1     ST2     ST3     ST4     SC1     SC2     SR1     SD1
 1       6.14841E+01     7.14841E+01     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     6.04000E+02     2.46216E+02
 2       7.21586E+01     8.21586E+01     2.06000E+01     2.00000E+01     1.50000E+01     4.50000E+02     6.04000E+02     2.96362E+02
 3       8.22859E+01     9.22859E+01     1.98000E+01     2.01000E+01     1.50000E+01     4.50000E+02     6.09000E+02     3.43634E+02
¡Cuidado! ST2 tiene una temperatura de:  1.02128E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 4       9.21285E+01     1.02128E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     6.20000E+02     3.88016E+02
¡Cuidado! ST2 tiene una temperatura de:  1.10087E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 5       1.00087E+02     1.10087E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     5.90000E+02     4.31244E+02
¡Cuidado! ST2 tiene una temperatura de:  1.15106E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 6       1.05106E+02     1.15106E+02     2.05000E+01     2.00000E+01     1.50000E+01     4.50000E+02     5.30000E+02     4.65956E+02
¡Cuidado! ST2 tiene una temperatura de:  1.16599E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 7       1.06599E+02     1.16599E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.50000E+02     4.88242E+02
¡Cuidado! ST2 tiene una temperatura de:  1.16810E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 8       1.06810E+02     1.16810E+02     2.00000E+01     2.02000E+01     1.50000E+01     4.50000E+02     4.20000E+02     4.94797E+02
¡Cuidado! ST2 tiene una temperatura de:  1.18579E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 9       1.08579E+02     1.18579E+02     2.07000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.60000E+02     4.95385E+02
¡Cuidado! ST2 tiene una temperatura de:  1.21016E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 10      1.11016E+02     1.21016E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.80000E+02     5.03495E+02
¡Cuidado! ST2 tiene una temperatura de:  1.22092E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 11      1.12092E+02     1.22092E+02     2.05000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.50000E+02     5.13956E+02
¡Cuidado! ST2 tiene una temperatura de:  1.24303E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 12      1.14303E+02     1.24303E+02     1.90000E+01     2.03000E+01     1.50000E+01     4.50000E+02     4.80000E+02     5.19411E+02
¡Cuidado! ST2 tiene una temperatura de:  1.23164E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 13      1.13164E+02     1.23164E+02     1.95000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.00000E+02     5.28877E+02
¡Cuidado! ST2 tiene una temperatura de:  1.24088E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 14      1.14088E+02     1.24088E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.50000E+02     5.23633E+02
¡Cuidado! ST2 tiene una temperatura de:  1.20982E+02ºC
Mensaje de alarma registrado en el archivo alarm_log.txt.
 15      1.10982E+02     1.20982E+02     2.10000E+01     2.00000E+01     1.50000E+01     4.50000E+02     3.50000E+02     5.27205E+02
```

Figura 2 Salida por consola

Como podemos ver, hay avisos de la temperatura del sensor **ST2** en las iteraciones **4**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **13**, **14** y **15**.

En el archivo de salida "**output.txt**", los resultados se verán de la siguiente forma:

```
k       ST1     ST2     ST3     ST4     SC1     SC2     SR1     SD1
 1       6.14841E+01     7.14841E+01     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     6.04000E+02     2.46216E+02
 2       7.21586E+01     8.21586E+01     2.06000E+01     2.00000E+01     1.50000E+01     4.50000E+02     6.04000E+02     2.96362E+02
 3       8.22859E+01     9.22859E+01     1.98000E+01     2.01000E+01     1.50000E+01     4.50000E+02     6.09000E+02     3.43634E+02
 4       9.21285E+01     1.02128E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     6.20000E+02     3.88016E+02
 5       1.00087E+02     1.10087E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     5.90000E+02     4.31244E+02
 6       1.05106E+02     1.15106E+02     2.05000E+01     2.00000E+01     1.50000E+01     4.50000E+02     5.30000E+02     4.65956E+02
 7       1.06599E+02     1.16599E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.50000E+02     4.88242E+02
 8       1.06810E+02     1.16810E+02     2.00000E+01     2.02000E+01     1.50000E+01     4.50000E+02     4.20000E+02     4.94797E+02
 9       1.08579E+02     1.18579E+02     2.07000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.60000E+02     4.95385E+02
 10      1.11016E+02     1.21016E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.80000E+02     5.03495E+02
 11      1.12092E+02     1.22092E+02     2.05000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.50000E+02     5.13956E+02
 12      1.14303E+02     1.24303E+02     1.90000E+01     2.03000E+01     1.50000E+01     4.50000E+02     4.80000E+02     5.19411E+02
 13      1.13164E+02     1.23164E+02     1.95000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.00000E+02     5.28877E+02
 14      1.14088E+02     1.24088E+02     2.00000E+01     2.00000E+01     1.50000E+01     4.50000E+02     4.50000E+02     5.23633E+02
 15      1.10982E+02     1.20982E+02     2.10000E+01     2.00000E+01     1.50000E+01     4.50000E+02     3.50000E+02     5.27205E+02
```

Figura 3 Archivo Output.txt

La salida es la misma que por consola, solo que omitiendo los avisos de alarma, que estarán registrados de la siguiente forma en el archivo "**alarm_log.txt**":

```
Iteración:  4, Temperatura:  1.02128E+02ºC
Iteración:  5, Temperatura:  1.10087E+02ºC
Iteración:  6, Temperatura:  1.15106E+02ºC
Iteración:  7, Temperatura:  1.16599E+02ºC
Iteración:  8, Temperatura:  1.16810E+02ºC
Iteración:  9, Temperatura:  1.18579E+02ºC
Iteración:  10, Temperatura:  1.21016E+02ºC
Iteración:  11, Temperatura:  1.22092E+02ºC
Iteración:  12, Temperatura:  1.24303E+02ºC
Iteración:  13, Temperatura:  1.23164E+02ºC
Iteración:  14, Temperatura:  1.24088E+02ºC
Iteración:  15, Temperatura:  1.20982E+02ºC
```

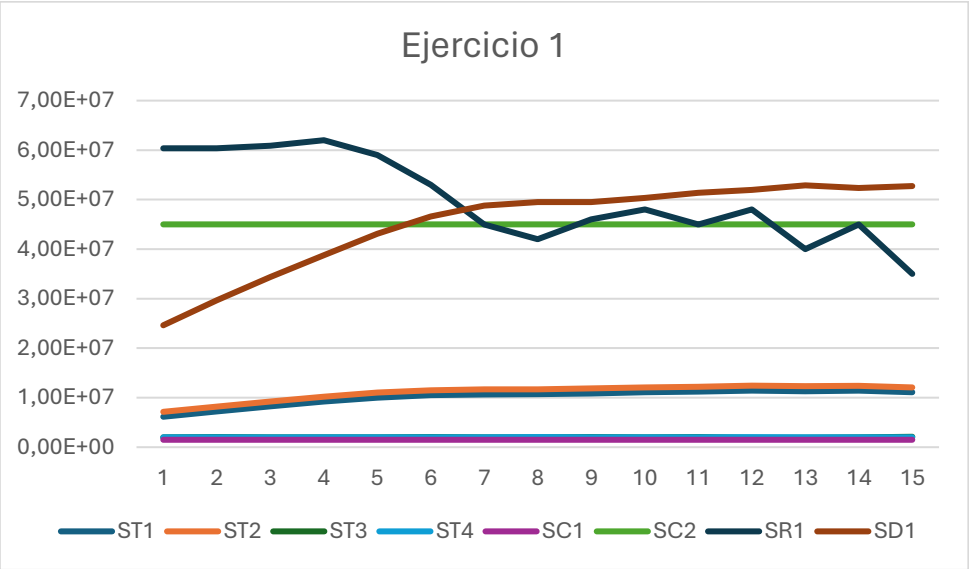
Figura 4 Archivo alarm_log.txt

Estudio de la implementación del modelo con caudales fijos.

Los resultados obtenidos en el archivo "**output.txt**", los pegaremos en **Excel** y el editor los colocará de manera automática en una tabla.

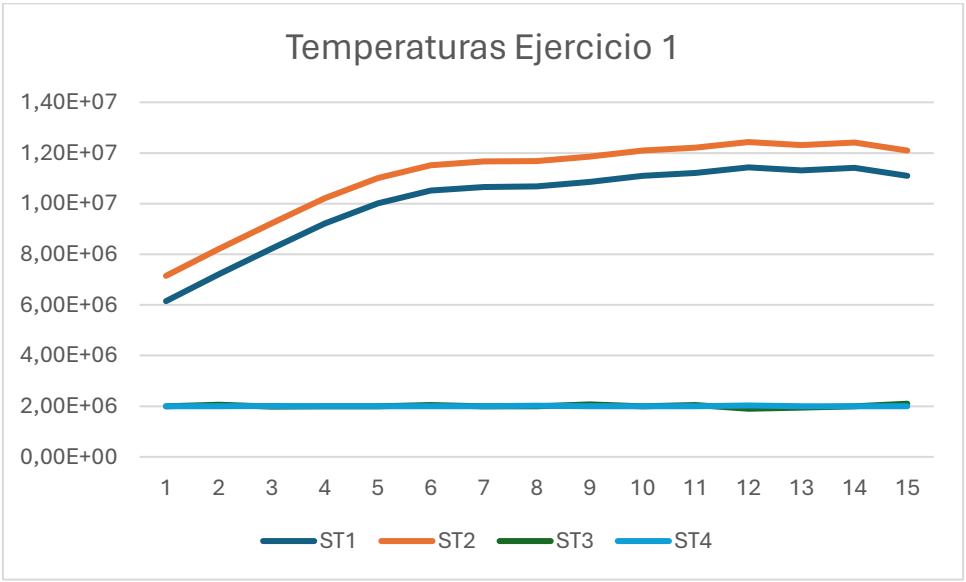
k	ST1	ST2	ST3	ST4	SC1	SC2	SR1	SD1
1	6,15E+06	7,15E+06	2,00E+06	2,00E+06	1,50E+06	4,50E+07	6,04E+07	2,46E+07
2	7,22E+06	8,22E+06	2,06E+06	2,00E+06	1,50E+06	4,50E+07	6,04E+07	2,96E+07
3	8,23E+06	9,23E+06	1,98E+06	2,01E+06	1,50E+06	4,50E+07	6,09E+07	3,44E+07
4	9,21E+06	1,02E+07	2,00E+06	2,00E+06	1,50E+06	4,50E+07	6,20E+07	3,88E+07
5	1,00E+07	1,10E+07	2,00E+06	2,00E+06	1,50E+06	4,50E+07	5,90E+07	4,31E+07
6	1,05E+07	1,15E+07	2,05E+06	2,00E+06	1,50E+06	4,50E+07	5,30E+07	4,66E+07
7	1,07E+07	1,17E+07	2,00E+06	2,00E+06	1,50E+06	4,50E+07	4,50E+07	4,88E+07
8	1,07E+07	1,17E+07	2,00E+06	2,02E+06	1,50E+06	4,50E+07	4,20E+07	4,95E+07
9	1,09E+07	1,19E+07	2,07E+06	2,00E+06	1,50E+06	4,50E+07	4,60E+07	4,95E+07
10	1,11E+07	1,21E+07	2,00E+06	2,00E+06	1,50E+06	4,50E+07	4,80E+07	5,03E+07
11	1,12E+07	1,22E+07	2,05E+06	2,00E+06	1,50E+06	4,50E+07	4,50E+07	5,14E+07
12	1,14E+07	1,24E+07	1,90E+06	2,03E+06	1,50E+06	4,50E+07	4,80E+07	5,19E+07
13	1,13E+07	1,23E+07	1,95E+06	2,00E+06	1,50E+06	4,50E+07	4,00E+07	5,29E+07
14	1,14E+07	1,24E+07	2,00E+06	2,00E+06	1,50E+06	4,50E+07	4,50E+07	5,24E+07
15	1,11E+07	1,21E+07	2,10E+06	2,00E+06	1,50E+06	4,50E+07	3,50E+07	5,27E+07

A partir de esta tabla, podemos hacer una gráfica de líneas automática para así poder comparar y poder estudiar el funcionamiento del modelo.



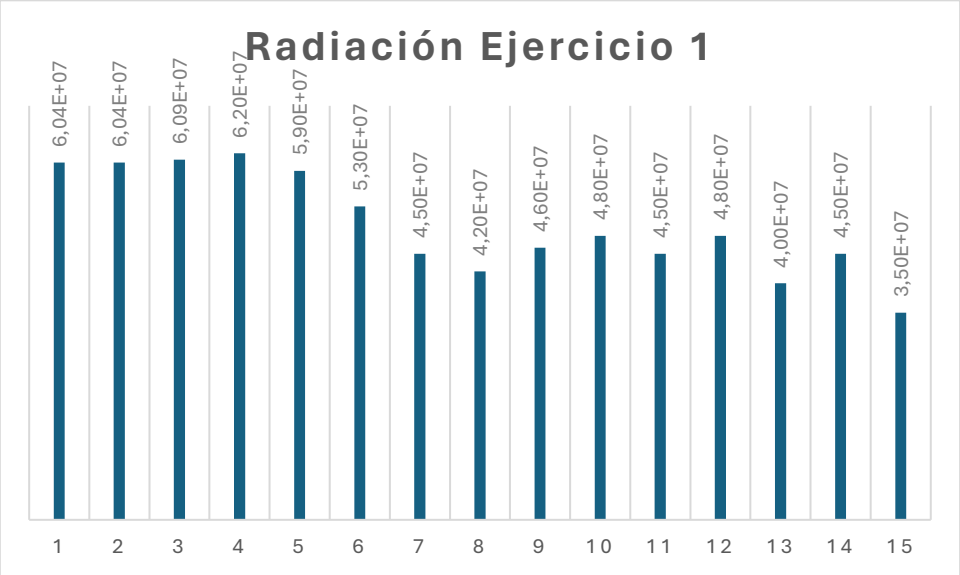
De esta gráfica, podemos interpretar una leve subida de las temperaturas en los sensores **ST1** y **ST2** que luego se estabilizan y se mantienen constantes. A su vez, podemos observar que la temperatura de **ST2** se mantiene constantemente por encima de los **98 °C**, cosa que incumple el mecanismo de seguridad del simulador.

Las temperaturas de **ST3** y **ST4**, en cambio, se han mantenido constantemente cerca de los **20 °C**. A continuación, un gráfico de la evolución de las temperaturas del modelo del **ejercicio 1**.

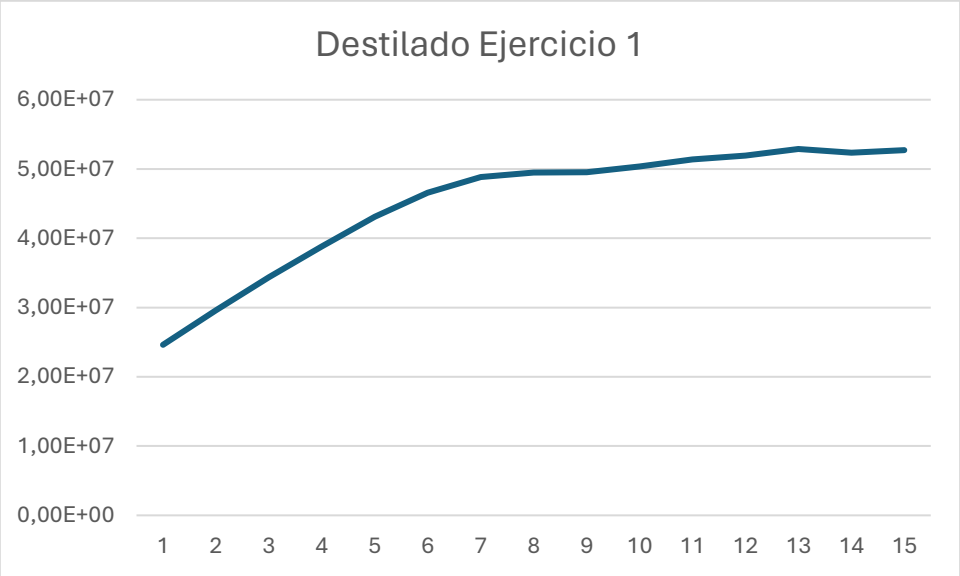


En el **ejercicio 1**, los caudales se mantendrán constantes, por lo tanto, no es necesario estudiarlos. Anteriormente se declararon los valores de ambos caudales a **15** y **450** para **SC1** y **SC2**, respectivamente.

Si estudiamos los valores de la radiación solar medidos por el sensor **SR1**, podemos observar un decremento de estos conforme van avanzando las iteraciones.



Por último, vamos a estudiar el comportamiento en el sensor de destilado **SD1**, como sabemos, este debe alcanzar los **26.5 L/h**, al menos, para que haya un correcto funcionamiento de la planta. Como podemos ver, a partir de la segunda iteración, superamos este valor.



Implementación con caudales óptimos.

Código.

En el paquete "**Simulador**", introducimos las fórmulas para despejar las variables de control de las ecuaciones dadas.

```
SC1(i) := SC1_f(SR1(i), Tt(i), ST4(i), ST2_Opt, ST1(i-1));
SC2(i) := SC2_f(SD1_Opt, ST2(i-1), ST3(i));
```

Añadimos los **rangos seguros** para comprobar que los valores de los **caudales** no pasen de los valores establecidos y se mantengan en un **rango óptimo**.

```
if(SC1(i) < 7.5) then
   SC1(i) := 7.5;
elsif(SC1(i) > 30.0) then
   SC1(i) := 30.0;
end if;

if(SC2(i) < 400.0) then
   SC2(i) := 400.0;
elsif(SC2(i) > 600.0) then
   SC2(i) := 600.0;
end if;
```

Por último, en vez de escribir el archivo de salida "**output.txt**", crearemos un archivo nuevo que se llamará "**output_ejercicio2.txt**", donde guardaremos los resultados de la simulación con los caudales óptimos.

```
-- Creación del archivo de salida
Create(output, Out_File, "output_ejercicio2.txt");
```

Ejecución y resultados.

Cuando ejecutamos el programa en el compilador **GNAT Studio**, el resultado imprimido en consola es el siguiente:

```
Resultados del Ejercicio 2:

k       ST1     ST2     ST3     ST4     SC1     SC2     SR1     SD1
 1      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     1.00707E+01     6.00000E+02     6.04000E+02     4.69053E+02
 2      7.20000E+01     8.20000E+01     2.06000E+01     2.00000E+01     2.98998E+01     5.00591E+02     6.04000E+02     3.81159E+02
 3      7.20000E+01     8.20000E+01     1.98000E+01     2.01000E+01     3.00000E+01     4.93468E+02     6.09000E+02     3.76112E+02
 4      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     3.00000E+01     4.95249E+02     6.20000E+02     3.77374E+02
 5      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.90670E+01     4.95249E+02     5.90000E+02     3.77374E+02
 6      7.20000E+01     8.20000E+01     2.05000E+01     2.00000E+01     2.54981E+01     4.99700E+02     5.30000E+02     3.80528E+02
 7      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.07395E+01     4.95249E+02     4.50000E+02     3.77374E+02
 8      7.20000E+01     8.20000E+01     2.00000E+01     2.02000E+01     1.89762E+01     4.95249E+02     4.20000E+02     3.77374E+02
 9      7.20000E+01     8.20000E+01     2.07000E+01     2.00000E+01     2.13343E+01     5.01481E+02     4.60000E+02     3.81789E+02
 10     7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.25240E+01     4.95249E+02     4.80000E+02     3.77374E+02
 11     7.20000E+01     8.20000E+01     2.05000E+01     2.00000E+01     2.07395E+01     4.99700E+02     4.50000E+02     3.80528E+02
 12     7.20000E+01     8.20000E+01     1.90000E+01     2.03000E+01     2.25557E+01     4.86345E+02     4.80000E+02     3.71066E+02
 13     7.20000E+01     8.20000E+01     1.95000E+01     2.00000E+01     1.77654E+01     4.90797E+02     4.00000E+02     3.74220E+02
 14     7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.07395E+01     4.95249E+02     4.50000E+02     3.77374E+02
 15     7.20000E+01     8.20000E+01     2.10000E+01     2.00000E+01     1.47913E+01     5.04152E+02     3.50000E+02     3.83682E+02
```

Figura 5 Salida por consola

Como podemos observar, en este caso, no salta ningún aviso de temperatura.

En el archivo de salida "**output_ejercicio2.txt**", los resultados se verán de la siguiente forma:

```
k       ST1     ST2     ST3     ST4     SC1     SC2     SR1     SD1
 1      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     1.00707E+01     6.00000E+02     6.04000E+02     4.69053E+02
 2      7.20000E+01     8.20000E+01     2.06000E+01     2.00000E+01     2.98998E+01     5.00591E+02     6.04000E+02     3.81159E+02
 3      7.20000E+01     8.20000E+01     1.98000E+01     2.01000E+01     3.00000E+01     4.93468E+02     6.09000E+02     3.76112E+02
 4      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     3.00000E+01     4.95249E+02     6.20000E+02     3.77374E+02
 5      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.90670E+01     4.95249E+02     5.90000E+02     3.77374E+02
 6      7.20000E+01     8.20000E+01     2.05000E+01     2.00000E+01     2.54981E+01     4.99700E+02     5.30000E+02     3.80528E+02
 7      7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.07395E+01     4.95249E+02     4.50000E+02     3.77374E+02
 8      7.20000E+01     8.20000E+01     2.00000E+01     2.02000E+01     1.89762E+01     4.95249E+02     4.20000E+02     3.77374E+02
 9      7.20000E+01     8.20000E+01     2.07000E+01     2.00000E+01     2.13343E+01     5.01481E+02     4.60000E+02     3.81789E+02
 10     7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.25240E+01     4.95249E+02     4.80000E+02     3.77374E+02
 11     7.20000E+01     8.20000E+01     2.05000E+01     2.00000E+01     2.07395E+01     4.99700E+02     4.50000E+02     3.80528E+02
 12     7.20000E+01     8.20000E+01     1.90000E+01     2.03000E+01     2.25557E+01     4.86345E+02     4.80000E+02     3.71066E+02
 13     7.20000E+01     8.20000E+01     1.95000E+01     2.00000E+01     1.77654E+01     4.90797E+02     4.00000E+02     3.74220E+02
 14     7.20000E+01     8.20000E+01     2.00000E+01     2.00000E+01     2.07395E+01     4.95249E+02     4.50000E+02     3.77374E+02
 15     7.20000E+01     8.20000E+01     2.10000E+01     2.00000E+01     1.47913E+01     5.04152E+02     3.50000E+02     3.83682E+02
```

Figura 6 Archivo output_ejercicio2.txt

El archivo "**alarm_log.txt**" se mantendrá vacío en este caso.

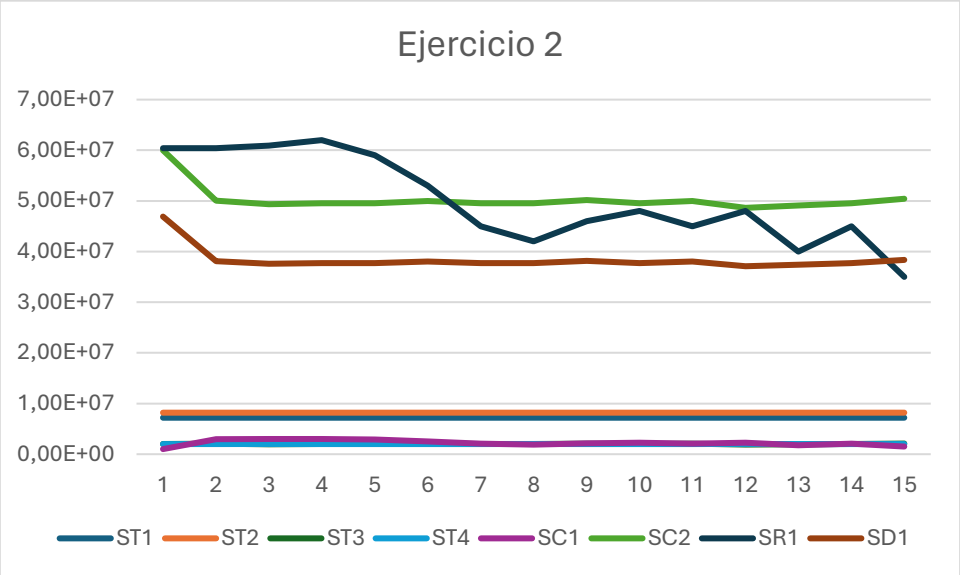
Estudio de la implementación del modelo con caudales óptimos.

Si introducimos los datos en **Excel**, el editor los coloca de forma automática en una tabla, que queda así:

k	ST1	ST2	ST3	ST4	SC1	SC2	SR1	SD1
1	7,20E+06	8,20E+06	2,00E+06	2,00E+06	1,01E+06	6,00E+07	6,04E+07	4,69E+07
2	7,20E+06	8,20E+06	2,06E+06	2,00E+06	2,99E+06	5,01E+07	6,04E+07	3,81E+07
3	7,20E+06	8,20E+06	1,98E+06	2,01E+06	3,00E+06	4,93E+07	6,09E+07	3,76E+07
4	7,20E+06	8,20E+06	2,00E+06	2,00E+06	3,00E+06	4,95E+07	6,20E+07	3,77E+07
5	7,20E+06	8,20E+06	2,00E+06	2,00E+06	2,91E+06	4,95E+07	5,90E+07	3,77E+07
6	7,20E+06	8,20E+06	2,05E+06	2,00E+06	2,55E+06	5,00E+07	5,30E+07	3,81E+07
7	7,20E+06	8,20E+06	2,00E+06	2,00E+06	2,07E+06	4,95E+07	4,50E+07	3,77E+07

8	7,20E+06	8,20E+06	2,00E+06	2,02E+06	1,90E+06	4,95E+07	4,20E+07	3,77E+07
9	7,20E+06	8,20E+06	2,07E+06	2,00E+06	2,13E+06	5,01E+07	4,60E+07	3,82E+07
10	7,20E+06	8,20E+06	2,00E+06	2,00E+06	2,25E+06	4,95E+07	4,80E+07	3,77E+07
11	7,20E+06	8,20E+06	2,05E+06	2,00E+06	2,07E+06	5,00E+07	4,50E+07	3,81E+07
12	7,20E+06	8,20E+06	1,90E+06	2,03E+06	2,26E+06	4,86E+07	4,80E+07	3,71E+07
13	7,20E+06	8,20E+06	1,95E+06	2,00E+06	1,78E+06	4,91E+07	4,00E+07	3,74E+07
14	7,20E+06	8,20E+06	2,00E+06	2,00E+06	2,07E+06	4,95E+07	4,50E+07	3,77E+07
15	7,20E+06	8,20E+06	2,10E+06	2,00E+06	1,48E+06	5,04E+07	3,50E+07	3,84E+07

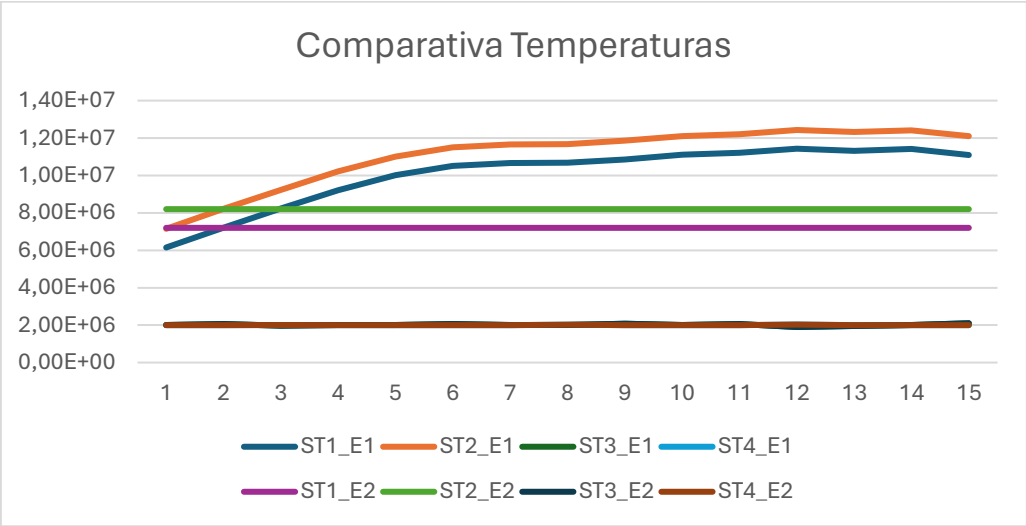
A partir de esta tabla, de la misma forma que con el modelo con caudal fijo, podemos hacer una gráfica de líneas automática para así poder comparar y poder estudiar el funcionamiento del modelo.



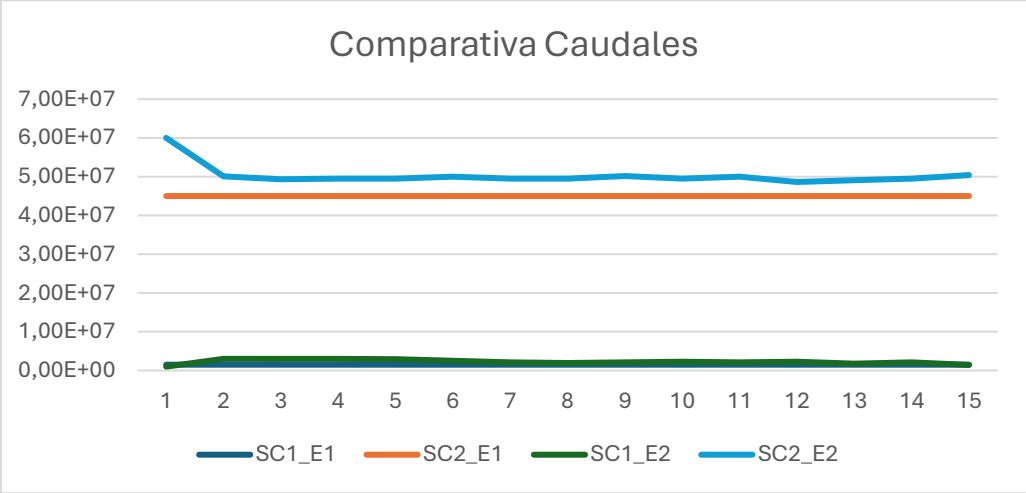
De esta gráfica, podemos interpretar la bajada de temperatura escalonada de el sensor **ST1**. Una bajada inicial del caudal que luego se mantiene en el tiempo y un funcionamiento constante y progresivo del resto de sensores.

Comparativa entre caudal fijo y caudal óptimo.

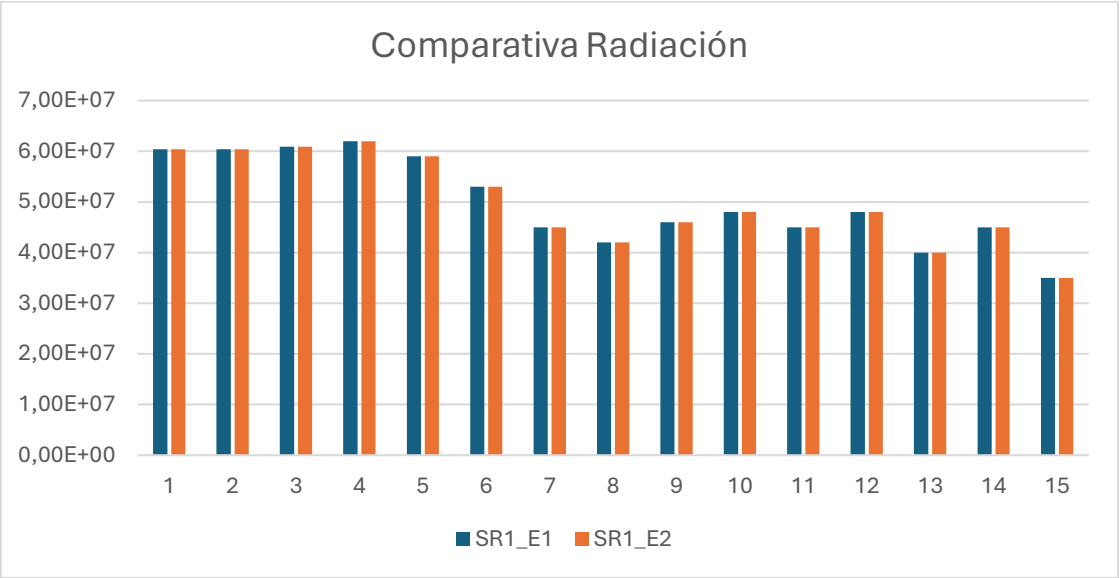
Como podemos observar la siguiente gráfica, los valores de **ST3** y **ST4** se mantienen a **20 °C** durante toda la simulación. **ST1** y **ST2** no varían durante toda la simulación en el modelo con caudal fijo, en cambio, podemos ver una variación ascendente constante que se estabiliza al final en la simulación con caudales óptimos.



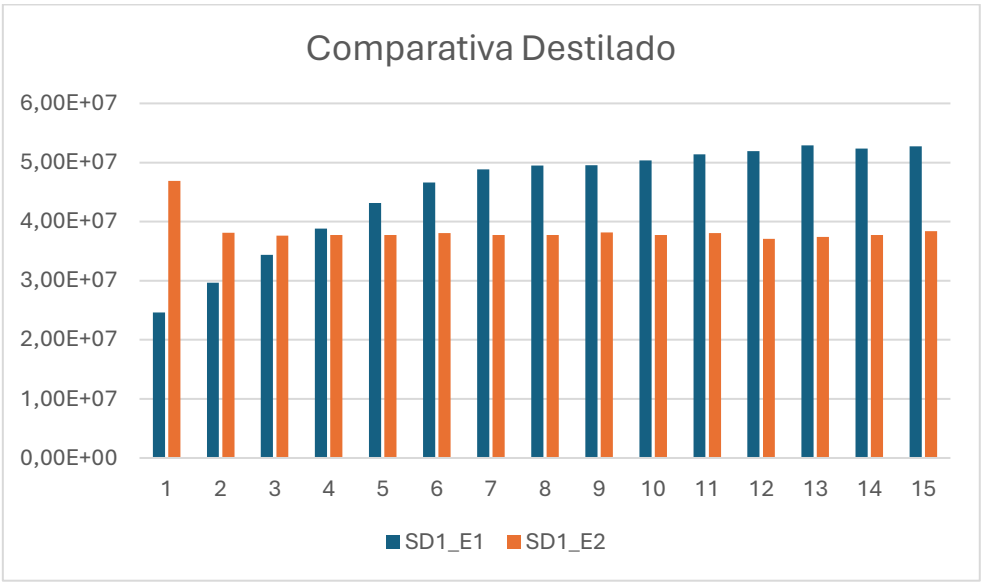
En cuanto a los sensores de caudal, podemos observar que, a pesar de variar el valor de estos de una simulación a la otra, **SC1** se mantiene en ambos casos entorno a los **15 L/h**. El sensor de caudal **SC2** se mantiene siempre un poco por encima de los **450 L/h** del anterior ejercicio.



Los niveles de **radiación** medidos en ambos ejercicios son iguales ya que son valores constantes.



Como podemos apreciar en la siguiente gráfica, el sensor de **destilado** del ejercicio 2, capta una **medición** más **cercana** a los niveles **óptimos** de producción.



Preguntas teóricas.

Ejercicio 1.

¿Se está operando el campo solar y el módulo de forma correcta?

Si tenemos en cuenta las **temperaturas** del campo solar, **no** se trabaja de forma óptima. Se mantiene a lo largo de la simulación considerablemente superior a la temperatura óptima (**82 °C**). En cuanto

al **destilado**, a partir de la segunda iteración se mantiene por encima del nivel óptimo (**26.5 L/h**). Esto **no** es **indicativo** de un buen funcionamiento, ya que, una **sobreproducción** puede llevar al **deterioro** de la **planta de destilado** y a la consiguiente **bajada de calidad** del **producto final**.

¿Se incumple las restricciones de seguridad en algún momento?

Sí, se incumple la restricción de los **98 °C** de temperatura de la iteración 4 a la 15, ambas incluidas.

```
Iteración:  4, Temperatura:  1.02128E+02ºC
Iteración:  5, Temperatura:  1.10087E+02ºC
Iteración:  6, Temperatura:  1.15106E+02ºC
Iteración:  7, Temperatura:  1.16599E+02ºC
Iteración:  8, Temperatura:  1.16810E+02ºC
Iteración:  9, Temperatura:  1.18579E+02ºC
Iteración:  10, Temperatura:  1.21016E+02ºC
Iteración:  11, Temperatura:  1.22092E+02ºC
Iteración:  12, Temperatura:  1.24303E+02ºC
Iteración:  13, Temperatura:  1.23164E+02ºC
Iteración:  14, Temperatura:  1.24088E+02ºC
Iteración:  15, Temperatura:  1.20982E+02ºC
```

Figura 7 Archivo alarm_log.txt

Ejercicio 2.

¿Se cumplen las condiciones de operación óptimas?

Sí, se cumplen las condiciones óptimas debido a que el nivel de destilado medido con el sensor **SD1** se mantiene cercano a los **26.5 L/h**, la condición de operación óptima de la planta de destilado.

Anexos y bibliografía

Anexo

Para la realización de esta práctica, hemos creado un repositorio en la plataforma de control de versiones **GitHub**, en la cual, a través del siguiente enlace, podrá seguir nuestro trabajo en la asignatura, así como comprobar el trabajo y la periodicidad de las subidas de código y del trabajo continuo de la asignatura.

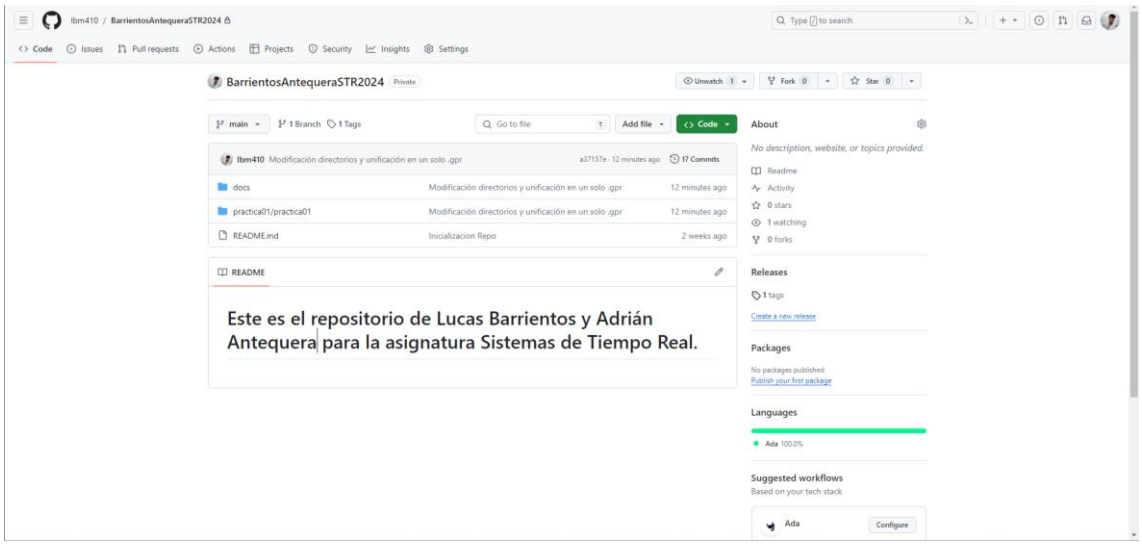


Figura 8 Captura del repositorio de la asignatura

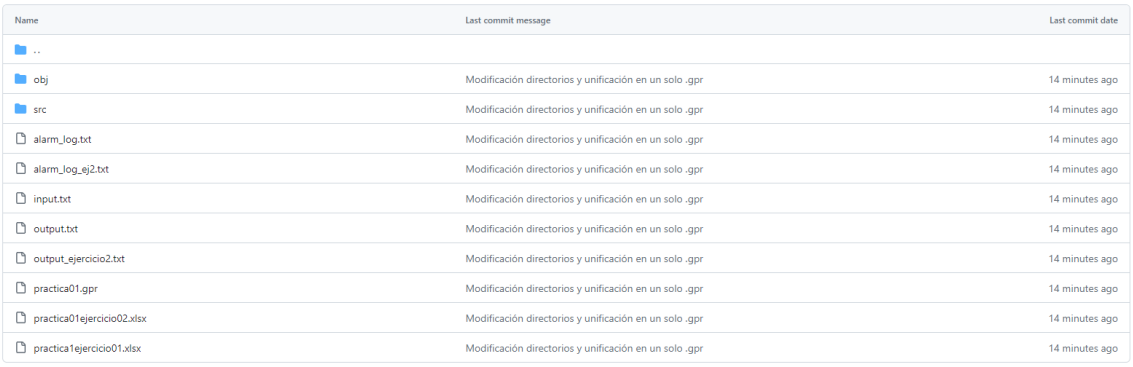


Figura 9 Carpeta practica01 ***GitHub***

En la carpeta **docs** del repositorio, se encuentra tanto el archivo **PDF** de esta memoria como el archivo **.docx** de Microsoft Word.

Bibliografía.

Ada-Europe. (2022). *Ada reference manual*. Ada Reference Manual.

https://docs.adacore.com/live/wave/arm22/html/arm22/arm22.html