

PRÁCTICA 1

Introducción a la programación en tiempo real.

SISTEMAS DE TIEMPO REAL

3º Ingeniería Informática

Lucas Barrientos Muñoz

Adrián Antequera Ramírez

­­

Contenido

[Objetivo. 3](#_Toc161791975)

[Antecedentes 4](#_Toc161791976)

[Paquete simulador. 4](#_Toc161791977)

[Estudio previo 4](#_Toc161791978)

# 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.

Escala de tiempo

Descripción generada automáticamente

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 irradiancia 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:

Texto

Descripción generada automáticamente

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

|  |  |
| --- | --- |
| **Constante** | **Valor** |
| β | 0.15 [m] |
| Leq | 15 [m] |
| H | 4 [J/s K] |
| C | 9∙2∙6∙10000 [s L/min m3] |
| Cp | 4190 [J/kg ºC] |
| p | 975 [kg/m3] |

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].