



## **PRÁCTICA 3 : Gestión del Tiempo en Ada**

### **OBJETIVOS:**

1. Conocer y saber utilizar **paquetes predefinidos** de Ada para la **gestión del tiempo**.
2. Implementar **tareas periódicas** en Ada y entender la diferencia que existe entre la ejecución de una tarea periódica utilizando la sentencia **delay** o utilizando **delay until**.

**PERIODO RECOMENDADO PARA SU REALIZACIÓN: ½ semana**

### **ENUNCIADO: Tarea periódica para un cronómetro.**

Al proyecto implementado en la práctica 2, hay que añadir el funcionamiento del cronómetro (contador de segundos) situado en la parte superior derecha de la ventana de la aplicación, que deberá mostrar el tiempo transcurrido desde el inicio de la ejecución del programa.

### **PASO 1: Funcionamiento del cronómetro**

Crea un nuevo paquete en tu proyecto e incluye en él una tarea periódica con una frecuencia de activación de un segundo. En cada activación de esta tarea, debes calcular el número de segundos transcurridos desde la hora de inicio de la ejecución de esta tarea periódica hasta el instante actual, y visualizar dicho tiempo transcurrido en el cronómetro. Para visualizar este tiempo calculado, debes utilizar el procedimiento **pkg\_graficos.Actualiza\_Cronometro**, que lo mostrará en el cronómetro con una precisión de milésimas de segundo.

Con el funcionamiento de este cronómetro pretendemos observar en qué instante se activa esta tarea periódica.

**Prueba el funcionamiento del cronómetro usando los dos paquetes predefinidos que Ada proporciona para la gestión del tiempo, para ello, implementa 2 paquetes y en cada uno de ellos utiliza un paquete predefinido de Ada distinto para la gestión del tiempo. De esta forma, aprenderás a manejar ambos, aunque no observarás diferencias en la ejecución. ¿Por qué?**

### **PASO 2: *delay until* vs *delay***

Vamos a utilizar la tarea cronómetro para entender la diferencia de implementar una tarea periódica usando la sentencia **delay until** frente a usar la sentencia **delay**. Para ello, ejecuta la tarea sustituyendo la sentencia **delay until** (puedes ponerla de momento entre comentarios) por la sentencia “**delay 1.0;**” y compara la diferencia que se aprecia en el cronómetro.

1. ¿Qué diferencia aprecias entre las ejecuciones que usan **delay until** y las que usan **delay**?
2. ¿A qué es debido esa diferencia?