PRÁCTICA 3 : *Gestión del Tiempo en Ada*

OBJETIVOS:

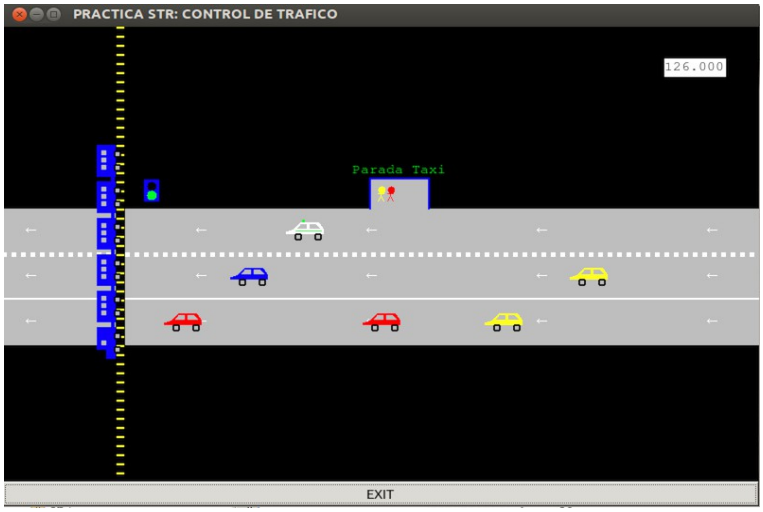
1. Conocer y saber utilizar **paquetes predefinidos** de Ada para la **gestión del tiempo**.
2. Implementar **tareas periódicas** en Ada y entender la diferencia que existe entre la ejecución de una tarea periódica utilizando la sentencia ***delay*** o utilizando ***delay until***.

PERIODO RECOMENDADO PARA SU REALIZACIÓN: 1 semana

ENUNCIADO: Cronómetro y tareas periódicas.

Como continuación de la práctica 2, se debe añadir el funcionamiento del cronómetro (contador de segundos) situado en la parte superior de la ventana de la aplicación, que deberá mostrar el tiempo transcurrido desde el inicio de la ejecución del programa.
También hay que modificar las tareas periódicas de forma adecuada para mejorar los instantes en los que se activan.

La siguiente figura muestra una instantánea del estado del sistema a los 126 segundos desde el inicio de su ejecución.



Aunque no está relacionado con la gestión del tiempo, en esta práctica vamos a realizar una pequeña modificación adicional a nuestro entorno simulado, que consiste en detectar colisiones de los taxis y coches con el tren, ya que de momento el paso del tren no está regulado por el semáforo.

PASOS A SEGUIR:

Antes de empezar tu implementación, primero debes sustituir el fichero ***pkg_graficos.adb*** por la nueva versión proporcionada. Si pruebas ahora tu proyecto puedes comprobar que si un taxi o coche colisiona con el tren, se quedan parados en la vía y se provocarán colisiones múltiples en los carriles de coches o no aparecerá un nuevo taxi. El por qué sucede esto y cómo solucionarlo lo veremos cuando implementes el paso 3. Lo primero que haremos será hacer funcionar el cronómetro.

PASO 1: Funcionamiento del cronómetro

Crea un nuevo paquete en tu proyecto denominado pkg_cronometro, 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 instantes se activan nuestras tareas periódicas, incluida la propia tarea cronómetro.

Puedes utilizar cualquiera de los dos paquetes predefinidos que Ada proporciona para la gestión del tiempo, aunque es recomendable que pruebes a utilizar ambos, en cuyo caso, implementa dos paquetes (pkg_cronometro_calendar y pkg_cronometro_realtime) y en cada uno de ellos utiliza el correspondiente paquete predefinido de Ada 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?

PASO 3: Detección de colisiones con el tren

En la nueva versión del fichero pkg_graficos.adb, el procedimiento ***pkg_graficos.Actualiza_Movimiento*** cuando se invoca con un taxi o un coche, puede lanzar la excepción ***pkg_tipos.DETECTADA_COLISION_TREN*** en caso de que detecte una colisión con el tren.
Lo que debes hacer es modificar el comportamiento de las tareas encargadas del comportamiento de un taxi y un coche, incorporando un manejador de excepciones cuando se invoca a este procedimiento para capturar dicha excepción, y asegurándote de que la tarea antes de terminar su ejecución invoque al procedimiento ***pkg_graficos.Desaparece***.
Con esto conseguiremos que en caso de colisión de un taxi o un coche con el tren, desaparezcan del sistema y no provoquen que se interrumpa la circulación de nuevos vehículos.

PASO 4: Corregir la implementación de tareas periódicas

Una vez demostrada la forma correcta de implementar tareas periódicas usando la sentencia *delay until* en lugar de la sentencia *delay*, corrige las tareas periódicas que **generan** taxis y trenes, para hacerlas más precisas en cuanto a sus instantes de activación. No cambies el delay del resto de tareas, ya que no se utiliza para que tengan una activación periódica.