

5. (2 p) Un alumno de la asignatura Sistemas Operativos I ha escrito el pseudocódigo (ver Figura 1) de un programa de nombre *Prueba* que intenta resolver el problema de la exclusión mutua. Explicar **razonadamente** si el alumno ha conseguido su objetivo.

```

program/module Prueba;

var X: boolean;

process P1
begin
  loop
    while X = true do
      /* Espera a que el dispositivo se libere */
    end;
    X := true;
    /* Uso del recurso Sección Crítica */
    X := false;
    /* resto del proceso */
  end
end P1;

process P2
begin
  loop
    while X = true do
      /* Espera a que el dispositivo se libere */
    end;
    X := true;
    /* Uso del recurso Sección Crítica */
    X := false;
    /* resto del proceso */
  end
end P2;

begin /* Prueba*/
  X := false;
  cobegin
    P1;
    P2;
  coend
end Prueba;

```

Figura 1: Pseudocódigo del programa Prueba

Solución:

La solución que propone el alumno para resolver el problema de la exclusión mutua hace uso de una variable compartida X de tipo booleano que se suele denominar *indicador* (flag). Se asocia a cada recurso sobre el que se desea la exclusión mutua un indicador. Antes de acceder al recurso, un proceso debe examinar el indicador asociado que podrá tomar dos valores (`true` o `false`) que indicarán, respectivamente, si el recurso está siendo utilizado o que está disponible. La acción de bloqueo se realiza con la activación del indicador ($X := \text{true}$) y la de desbloqueo con su desactivación ($X := \text{false}$).

El uso de un único indicador, como propone el alumno, no resuelve el problema de la exclusión mutua ya que al ser la comprobación y la puesta del indicador a falso operaciones separadas, puede ocurrir que se entrelace el uso del recurso por ambos procesos.

Sólo si el computador dispusiera de una instrucción que permitiera comprobar el estado y modificarlo simultáneamente, el programa propuesto permitiría el uso del recurso sin el problema del entrelazado.