

4. (2 p) Dos procesos A y B se ejecutan concurrentemente en un determinado sistema. El proceso A ejecuta unas tareas (“Tareas 1”) y alcanza un punto de encuentro. Posteriormente realiza otras tareas (“Tareas 2”) y finaliza. Por su parte el proceso B ejecuta unas tareas (“Tareas 3”) y llega al punto de encuentro. Posteriormente realiza otras tareas (“Tareas 4”) y finaliza. El primer proceso que llega al punto de encuentro no puede continuar su ejecución hasta que no llegue el otro proceso. No se sabe que proceso comienza a ejecutarse primero o cual es el primero que termina. Escribir en pseudocódigo un programa de nombre *coordinación* que usando semáforos coordine la actividad de los procesos A y B. Dicho programa debe tener cuatro partes: declaración de variables y semáforos, código del proceso A, código del proceso B y código para inicializar los semáforos y lanzar la ejecución concurrente de ambos procesos.

**Solución:**

La solución que se propone para modelar el punto de encuentro entre los procesos A y B utiliza dos semáforos binarios S1 y S2. Ambos semáforos son inicializados al valor 0 ya que se utilizan para sincronizar.

```

program/module coordinación;
var
    S1, S2: semáforo;

process proceso_A;
begin
    /* Tareas 1*/
    señal(S2);
    espera(S1);
    /* Tareas 2*/
end

process proceso_B;
begin
    /* Tareas 3*/
    señal(S1);
    espera(S2);
    /* Tareas 4*/
end

begin
    inicializa (S1,0);
    inicializa (S2,0);
cobegin
    proceso_A, proceso_B;
coend
end coordinación;

```