

**Cuestiones (1'5 puntos)**

Se debe responder a cada cuestión rellenando los huecos dejados para ello.

**Cuestión 1: (0'75 pt)**

En la solución de Peterson mostrada a continuación, indicar la secuencia de acontecimientos tomando como referencia los de la tabla. Los procesos P0 y P1 ejecutan entrar\_en\_region, la región crítica y luego salir\_de\_region. Suponer que primero se ejecuta el proceso P0 hasta completar la instrucción turno=proceso; . Una vez completada esta instrucción el planificador pasaría a ejecutar el proceso P1, que consigue llegar hasta el bucle y luego le tocaría de nuevo el turno a P0 y así.

**SOLUCIÓN DE PETERSON:**

```
#define FALSE 0
#define TRUE 1
#define N 2
int turno=0;
int interesado[N]; /* iniciado a falso */
entrar_en_region(int proceso)
{
    int el_otro;
    el_otro= 1- proceso;
    interesado[proceso]=TRUE;
    turno=proceso;
    while(turno==proceso && interesado[el_otro]==TRUE);
}
salir_de_region(int proceso)
{
    interesado [proceso]=FALSE;
}
```

**TABLA DE ACONTECIMIENTOS DE REFERENCIA**

1. P0 pone turno a 0
2. P1 pone turno a 1
3. P0 entra en la región crítica
4. P1 entra en la región crítica
5. P0 espera de forma activa en el bucle
6. P1 espera de forma activa en el bucle
7. P0 ejecuta salir_de_region
8. P1 ejecuta salir_de_region

**SECUENCIA** (rellenar las casillas que se crean necesarias con los números correspondientes en la tabla anterior a los acontecimientos que se van produciendo. El primer acontecimiento que se produce es el 1):

<b>1</b>	<b>2</b>	<b>6</b>	<b>3</b>	<b>7</b>	<b>4</b>	<b>8</b>				
----------	----------	----------	----------	----------	----------	----------	--	--	--	--

**Cuestión 2: (0,75 pt)**

Dado el siguiente estado del sistema con respecto a la asignación de recursos:

Ra

	R1	R2	R3	R4
A	3	0	1	1
B	0	1	1	0
C	1	1	1	0
D	1	1	0	1
E	0	0	1	0

Rn

	R1	R2	R3	R4
A	1	1	2	1
B	0	1	1	1
C	3	1	0	0
D	1	0	0	0
E	2	1	0	0

E=(6, 3, 4, 2) P=(5, 3, 4, 2) D=(1, 0, 0, 0)

Aplicar el algoritmo del banquero para múltiples recursos y rellenar los siguientes datos:

¿El estado es Seguro? SI  NO

Posible secuencia de terminación de los procesos en caso de que sea seguro:

Proceso Que termina	Vector D una vez Terminado el proceso
<b>D</b>	D=( 2 , 1 , 0 , 1 )
<b>E</b>	D=( 2 , 1 , 1 , 1 )
<b>B</b>	D=( 2 , 2 , 2 , 1 )
<b>A</b>	D=( 5 , 2 , 3 , 2 )
<b>C</b>	D=( 6 , 3 , 4 , 2 )

El sistema se puede bloquear en este estado? SI  NO