



1	2	3	4	test	extra	NOTA

Nombre y apellidos

DNI/NIE

DURACIÓN: Dispones de dos horas para realizar el examen.

Lee las instrucciones para el test en la hoja correspondiente.

1 (1'25 puntos) A un planificador de CPU llegan tres procesos, según el cuadro adjunto. Aplica las dos políticas SRTF (SJF expulsivo) y RR (Q=3) y, para cada una de ellas, obtén lo siguiente:

- Diagrama de Gantt o similar con la planificación.
- Tiempo de espera y de retorno de cada uno de los procesos.
- Número de cambios de contexto realizados durante la planificación.

proceso	llegada	duración
Pa	0	10
Pb	2	6
Pc	5	2

2 (0'50 puntos) ¿Qué diferencias hay entre un micronúcleo y un núcleo monolítico?

3 (1 punto) Muchos informáticos sostienen que uno de los beneficios del sistema operativo es que los desarrolladores de aplicaciones no tienen que estar modificando su *software* cada vez que hay un cambio en las tecnologías *hardware*, especialmente cuando aparece una nueva tecnología de almacenamiento (ej. discos SSD). Elabora unos argumentos que apoyen esa afirmación. No uses más de 150 palabras en tu exposición.

4 (1'25 puntos) En una tienda de pájaros tenemos una jaula con canarios que revolotean en su interior. Regularmente cada canario quiere comer de un comedero de alpiste, en el que solamente puede haber tres pájaros al mismo tiempo. Si un pájaro quiere comer y el comedero está lleno, se debe esperar a que haya hueco. Por su parte, el encargado de la tienda de vez en cuando repone el alpiste del comedero. Mientras el encargado está reponiendo, ningún pájaro puede estar comiendo: el encargado debe esperar a que se quede vacío el comedero y, una vez vacío, ningún pájaro entra a comer hasta que el encargado termina de reponer.

El algoritmo general de un pajarito y del encargado se muestra en el siguiente cuadro.

```
void pajarito () {  
    while ( pajaritoVivo() ) {  
        ... revolotear por la jaula  
        ... esperar a que haya hueco y que  
        el encargado no esté reponiendo el  
        alpiste  
        COMER();  
    }  
}
```

```
void encargado() {  
    while ( ! jubilado() ) {  
        ... hacer otras cosas  
        ... esperar a que no haya pájaros  
        comiendo  
        REPONER_ALPISTE();  
    }  
}
```

TAREA. Tienes que arreglar el algoritmo para que los pajaritos y el encargado se sincronicen entre todos ellos conforme al enunciado expuesto. Puedes utilizar variables de estado y operaciones básicas de sincronización (entradas y salidas en sección crítica, colas de espera, etc.). Si ya conoces el uso de semáforos, puedes resolverlo con ellos si lo prefieres (no habrá diferencia en la calificación de la pregunta).

NOTA. Escribe tu intento aunque no tengas la solución completa y perfecta para todos los escenarios. En ese caso, describe las limitaciones que sepas que tiene tu propuesta.