



1	2	3	test	extra	NOTA

Nombre y apellidos

DNI/NIE

**DURACIÓN:** Dispone de dos horas para realizar el examen.

Lea las instrucciones para el test en la hoja correspondiente.

**1 (1'25 puntos)** A un planificador de CPU llegan cuatro procesos, según el cuadro adjunto. Aplique las dos políticas SRTF y RR ( $q=2$ ) y, para cada una de ellas, obtenga 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
A	0	6
B	1	4
C	3	1
D	5	4

**2 (1 punto)** Los diseñadores de sistemas operativos, cuando idean políticas para gestionar el acceso a un recurso, no pueden en general encontrar un algoritmo que optimice al mismo tiempo el rendimiento, la seguridad y la justicia en el reparto del recurso: deben encontrar algoritmos que lleguen a un balance entre todos estos criterios.

**TAREA.** Ilustre este problema con un ejemplo concreto. No use más de 200 palabras en su exposición.

**3 (1'75 puntos)** En una agencia de meteorología tenemos un programa multihilo que se encarga de lanzar cinco hilos que intentan estimar por distintos métodos la temperatura que hará mañana en Las Palmas. En cuanto dos de los cinco hilos hayan terminado, otro hilo calculará la media de sus respectivas estimaciones y esa será la predicción definitiva.

Los hilos utilizan un vector global para anotar sus estimaciones, que serán leídas por el hilo predictor. El cuadro adjunto muestra el esquema general del código.

```
// variables globales
float estimaciones[5] = {∞,∞,∞,∞,∞};
float predicción = ∞;

hilo estimador(int i) { // i=0..4
    - Calcula la estimación por el método "i"
    estimaciones[i] = valor estimado
}

hilo predictor() {
    - Espera a que terminen dos estimaciones
    - Calcula la media de las dos estimaciones
    predicción = media calculada
}
```

**TAREA.** Tiene usted que añadir el código que sea necesario para sincronizar al hilo predictor con los hilos estimadores, utilizando semáforos como herramientas de sincronización.

Si lo considera necesario, puede crear nuevas variables o cambiar el código, siempre que se mantenga la arquitectura general aquí expuesta.