

Apellidos:

Nombre:

DNI:

La duración máxima del examen será de 1:00 horas.

Examen 1

Ejercicio 1) Preguntas cortas de programación concurrente (6p)

1.a) ¿Qué son las propiedades de corrección en un programa concurrente? Describe cada una de ellas e indica los dos tipos en los que se categorizan (3p)

1.b) ¿Qué es una condición de carrera en un programa concurrente? Pon un ejemplo de código en el que se pueda producir una condición de carrera y explica el tipo de condición. (2p)

1.c) ¿Qué es la espera activa? ¿Qué efectos tiene la espera activa en la multiprogramación y el multiproceso? ¿Por qué debería evitarse para la implementación de programas concurrentes? (1p)

Ejercicio 2) Programa concurrente (4p)

Se pide implementar completamente la clase Buffer para comunicar varios hilos en un esquema productor / consumidor. La clase Buffer debe tener el siguiente interfaz:

```
public class Buffer {  
  
    //Crea un buffer con el tamaño indicado  
    public Buffer(int tamaño){ ... }  
  
    //Inserta el dato en el buffer si hay espacio.  
    //Si no, queda bloqueado hasta que haya espacio.  
    public void insertar(int dato){ ... }  
  
    //Saca un dato del buffer. Si no hay datos, queda bloqueado hasta que haya datos  
    public int sacar(){ ... }  
}
```

La clase Buffer debe cumplir los siguientes requisitos:

- Tiene que ser *Thread safe*, es decir, varios hilos pueden compartir objetos de esta clase de forma segura.
- No se puede usar la espera activa para la implementación.
- Se tiene que implementar usando semáforos. El alumno tendrá que explicar razonadamente para qué sirve cada uno de los semáforos.
- Se puede usar SimpleConcurrent o la API estándar de Java.