

Nombre: _____

D.N.I: _____

1	2	3	4	5	6	7	8	9	10	11	Suma
(4)	(3)	(2)	(8)	(3)	(8)	(6)	(4)	(4)	(6)	(2)	(50)

Pregunta 1: [4 Puntos]

Asumimos que queremos implementar un sistema de venta por internet donde muchos clientes pueden acceder al sistema simultaneamente. Los productos en venta también están distribuidos por varias tiendas con gestión propia. Describe muy brevemente donde piensas que un programa concurrente y distribuido mejoraría el rendimiento, explica también en qué tipo de rendimiento te estás enfocando en cada argumento.

Pregunta 2: [3 Puntos]

¿Cuál es la diferencia entre los términos “seguridad” y “vivacidad” en el contexto de la programación concurrente? Explica también la diferencia en el contexto del bloqueo de procesos.

Pregunta 3: [2 Puntos]

Razona sobre diferencias entre un sistema distribuido tipo servidor/clientes y un sistema con entidades equivalentes.

Pregunta 4: [8 Puntos]

¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos? y ¿Cuáles son las técnicas que se pueden aplicar para que no se produzcan interbloqueos infinitos?

Pregunta 5: [3 Puntos]

¿Qué se entiende por “justicia” de un planificador que controla el acceso en los recursos?

Pregunta 6: [8 Puntos]

Asumimos que existan tres tipos de procesos con diferentes prioridades (digamos *A*, *B* y *C*) que quieren acceder a un recurso.

1. ¿Cómo implementarías el control del planificador para que todos los procesos tengan acceso al recurso con la siguiente forma de justicia: dentro de la misma prioridad el acceso se realiza en orden de pedida y entre los diferentes prioridades se distribuye los accesos para que a lo largo del tiempo por lo menos unos 60 % de los accesos son para los procesos de tipo *A*, 30 % para los del tipo *B* y 10 % para los del tipo *C*? (Ayuda: un planificador sabe contar).
2. Razona si tu solución garantiza una espera *finita* para todos los procesos pidiendo acceso al recurso.

Pregunta 7: [6 Puntos]

Describe brevemente el patrón de diseño *guardián* y su “presencia” en Java.

Pregunta 8: [4 Puntos]

Explica la semántica del modificador `volatile` de Java y los casos en los que usamos `volatile` en la implementación de la lista concurrente.

Pregunta 9: [4 Puntos]

Razona brevemente sobre una posible *ampliación* de la solución para el ejemplo pingPONG de las prácticas, es decir, un pingPONGpang, donde queremos que los cuatro hilos involucrados (el “arbitro”, el “ping”, el “pong” y el “pang”) produzcan la siguiente salida:

```
arbitro: empezamos
ping:    ping
pong:    PONG
pang:    pang
...      (y asi alternando durante 10 segundos)
arbitro: vamos a terminar
...      (como mucho un ping y/o un PONG y/o un pang)
arbitro: juego termindo
```

Pregunta 10: [6 Puntos]

¿Cuál fue la estrategia que aplicamos para que no se produjese ningún bloqueo en la lista concurrente? Razona brevemente como implementar con dicha estrategia las operaciones borrar, insertar (delante) y iterar.

Pregunta 11: [2 Puntos]

¿Cómo resolvimos el problema de la terminación prematura de una iteración sobre la lista concurrente (es decir, un hilo no itera hasta el final de la lista)?