

Nombre y Apellido:..... Curso: .....

TEORÍA					PRÁCTICA			NOTA
1	2	3	4	5	1	2	3	

**TEORÍA:** Responda brevemente las siguientes preguntas. Justifique.

1. Indique la relación que existe entre los términos de condición de carrera, sección crítica y exclusión mutua. Proponga dos técnicas para asegurar la exclusión mutua.

2. Responda Verdadero o Falso justificando:

a. La estrategia de Detección evita que se produzca Deadlock, analizando cada solicitud de recursos antes de que sea asignado.

b. Evitar la retención y espera puede producir una baja tasa de utilización de recursos.

3. Describa en detalle algún ejemplo del caso en el cual un proceso realiza una syscall y termina siendo bloqueado. ¿Qué implicancias tendría si dicha syscall hubiera sido ejecutada en forma no bloqueante?

4. ¿Qué utilidad tiene disponer de un máximo nivel de multiprogramación en un sistema? ¿Cuáles serían las consecuencias de configurarlo en niveles extremadamente bajos o altos?

5. Mencione alguna razón por la cual se podrían deshabilitar las interrupciones enmascarables temporalmente. ¿Debería el SO ser el único con la responsabilidad para realizar dicho cambio?

**PRÁCTICA:** Resuelva los siguientes ejercicios justificando las conclusiones obtenidas.

**Ejercicio 1**  
Se dispone de un sistema operativo con planificador de corto plazo RR con q=3, para la siguiente traza de ejecución:

	Llegada	CPU	I/O	CPU	I/O	CPU	$NTT = \frac{\sum \text{tiempos en Ready} + \sum \text{ráfagas cpu}}{\sum \text{ráfagas cpu}}$
P1	0	2	5	2	4	1	
P2	1	8	1	5	-	-	
P3	2	8	-	-	-	-	

- a) Realice el diagrama de gantt

b) Calcule la métrica “Normalized turnaround time” (NTT) para cada proceso y en base a dicha métrica mencione cuál proceso fue más perjudicado y por qué.

c) Proponga otro algoritmo de planificación que priorice al proceso afectado en el punto anterior, justifique conceptualmente, sin volver a realizar el gantt o el cálculo de la métrica.

**Ejercicio 2**  
Considerando las siguientes matrices:

Asignaciones					Peticiónes máximas				
	R1	R2	R3	R4		R1	R2	R3	R4
P1	0	3	1	3	P1	2	3	2	5
P2	1	1	3	2	P2	1	2	6	3
P3	0	2	1	0	P3	0	2	4	5
P4	2	0	2	0	P4	3	0	5	2
P5	1	3	5	2	P5	3	4	5	4

- a) Si se sabe que se cuenta con un vector de recursos disponibles = [1, X, 9, 4], utilice el algoritmo del banquero para determinar el mínimo X para que el sistema se encuentre en estado seguro.

b) Tomando el disponible calculado en a), indique qué acción tomaría el sistema si P2 solicitara 2 instancias de R2.

**Ejercicio 3**  
Luciano comenzó a trabajar en el área de medicina en la demanda espontánea del hospital británico. Cada paciente que llega debe esperar a que la recepción esté libre para poder iniciar los trámites. El trámite consiste en entregar la documentación y obtener un número con el cual será llamado por el médico clínico, quien comenzará a llamar siempre que haya algún paciente esperando. El médico escuchará su problema y luego de recetar un medicamento podrá el paciente irse a su casa.

La solución actual en pseudocódigo suele no funcionar apropiadamente, complete su sincronización agregando los semáforos que sean necesarios e indicando los valores de inicialización de los mismos para que cumpla con lo requerido sin causar deadlock ni starvation.

Paciente (N instancias)	Recepción (1 instancia)	Médico Clínico (1 instancia)
entregar_documentación() Id_paciente = recibir_número() <b>wait(atendido[id_paciente])</b> explicar_problema()  recibir_receta() irse_a_casa()	while(1){ recibir_documentación() nuevo_id = siguiente_id() entregar_número(nuevo_id)  add cola_pacientes, nuevo_id) }	while(1){ id_a_atender = get(cola_pacientes) <b>signal(atendido[id_a_atender])</b>  escuchar_problema() entregar_receta() }

**atendido[20] = {0, 0, ..., 0}**  
**Nota:** Sabemos que solo 20 personas son atendidas por día en la demanda espontánea. La función siguiente\_id() comienza en 0 y vuelve a 0 luego del 19.

**Condiciones de aprobación:** 3 preguntas correctamente respondidas y 1.5 ejercicios correctamente resueltos.

