

Nombre:

DNI:

Grupo:

Sobre 10, cada respuesta vale 2 si es correcta, 0 si está en blanco o claramente tachada, y -2/3 si es errónea.  
Anotar las respuestas (**a, b, c o d**) en la siguiente tabla.

1	2	3	4	5
c	d	b	d	b

- Parecidos y diferencias entre los métodos de E/S: alguna de las siguientes afirmaciones es **incorrecta**: T5 tr.29-32,37,58,43,44
  - La consulta del estado del dispositivo por parte de la CPU se suele hacer con E/S programada (salvo con dispositivos que siempre están listos para transferir) y con E/S por IRQ (cuando se usa *polling* para determinar el origen de la IRQ)
  - Se suele avisar a la CPU (mediante una IRQ) de que debe realizar alguna tarea, tanto en E/S por IRQ (obligatoriamente, la tarea es la transferencia) como en E/S por DMA (optativamente, el controlador DMA puede avisar de que acabó)
  - Sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo de E/S **E/S IRQ compartida disptvs**
  - Sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S
- Tipos de interrupción que suelen contemplar las CPUs comerciales actuales: alguno de los siguientes **\*no\*** lo es:
  - Internas (excepciones o *traps*): generadas internamente por la CPU para indicar una condición que requiere atención (división por cero, fallo de página, etc)
  - Externas (IRQs hardware): generadas por un dispositivo externo a la CPU, activan la línea #INTR (o equivalente)
  - Software: generadas al ejecutar la instrucción INT (o equivalente)
  - Firmware (*faults*): generadas por el microcódigo de la CPU (*segm. fault, page fault...*) **ver Tema5 tr.50-51**
- Alguna de las siguientes técnicas **\*no\*** es de utilidad para determinar la causa de una interrupción **ver Tema5 tr.56-58**
  - Múltiples líneas de interrupción INT1#, INT2#...
  - Línea de reconocimiento INTA# **básica para retirar IRQ, evitar redisparo, enviar vector, T5 tr.46,55**
  - Consulta de estado, o *polling*
  - Interrupciones vectorizadas
- Alguna de las siguientes técnicas **\*no\*** está relacionada con la gestión por prioridad de IRQs simultáneas:
  - Centralizada : un único circuito recibe las diversas peticiones y reconoce la prioritaria
  - Sondeo, o *polling*: un software (la ISR) sondea el estado, la prioridad viene establecida por el orden de consulta
  - Encadenamiento, o *daisy-chain*: del reconocimiento INTA#, la prioridad viene establecida por el orden en la cadena
  - Inhibición: un registro de máscara retiene las peticiones menos prioritarias hasta que les llegue el turno **T5 tr.60-62,68**
- Respecto a salvaguardar los registros de la CPU al inicio de una rutina de servicio de interrupción (ISR):
  - No es necesario salvar ninguno más, el contador de programa y los flags de estado ya los salva la propia CPU como parte del mecanismo de interrupción
  - Se deben guardar los registros que se modifiquen en la propia ISR. Eso es posible hacerlo porque el propio programador de la ISR conoce qué registros va a modificar **ver Tema5, tr.46**
  - Se deben guardar los registros salva-invocado (p.ej. EBX, ESI, EDI en el caso de una CPU IA-32), los registros salva-invocante ya los guarda el programa interrumpido
  - Se deben guardar todos los registros, para restaurarlos a la salida y así garantizar que el programa interrumpido no sufre ninguna modificación (salvo el inevitable retraso temporal) debido a la interrupción