

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
d	c	b	c	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
  - En E/S programada la CPU debe encargarse tanto de consultar el estado del dispositivo de E/S como de transferir el dato deseado. También hay dispositivos que siempre están listos para transferir y no requieren consulta de estado.
  - En E/S por IRQ la CPU se ahorra la consulta de estado, pero sigue teniendo que transferir el dato. También hay formas de conexión de las líneas IRQ que requieren que la CPU haga consulta de estado.
  - En E/S por DMA la CPU se ahorra tanto la consulta de estado como la transferencia de datos. Pero se usan instrucciones de E/S para programar el controlador DMA, y la CPU suele recibir peticiones de interrupción (IRQ) del propio DMA.
  - Todos los métodos de E/S pueden terminar usando (opcionalmente) alguna característica de otro método (E/S DMA usando IRQs, E/S IRQs usando *polling*, etc) **E/S prog. no usa IRQ-DMA**
- Tipos de interrupción que suelen contemplar las CPUs comerciales actuales: alguno de los siguientes **\*no\*** lo es:
  - Software: generadas al ejecutar la instrucción INT (o equivalente)
  - Hardware (IRQs externas): generadas por un dispositivo externo a la CPU, activan la línea #INTR (o equivalente)
  - Faltas o *faults*: generadas por el SO (*segment. fault, page fault, mem.prot. fault...*) **ver Tema5 tr.50-51**
  - Excepciones o *traps*: generadas internamente por la CPU para indicar una condición que requiere atención (división por cero, *overflow*, etc)
- Alguna de las siguientes técnicas **\*no\*** es de utilidad para determinar la causa de una interrupción **ver Tema5 tr.56-58**
  - Consulta de estado, o *polling*
  - Daisy-chain* (encadenamiento) de ACK# **útil para prioridad entre IRQs simultáneas, ver Tema5 tr.65-67**
  - Interrupciones vectorizadas
  - Múltiples líneas de interrupción INT1#, INT2#...
- En un sistema computador con varios dispositivos que hagan IRQ aparecen algunos problemas que se suelen resolver mediante mecanismos de prioridad. Alguno de los siguientes problemas o técnicas **\*no\*** está relacionado con este escenario:
  - Interrupciones simultáneas
  - Interrupciones anidadas
  - Redisparo de interrupciones **ver Tema5 tr.59**
  - Inhibición de interrupciones
- Alguno de los siguientes **\*no\*** es un método de inhibición de interrupciones que hayamos estudiado en clase:
  - Mecanismos automáticos: como no atender interrupciones durante la primera instrucción de la ISR, o durante toda la ISR, o como no atender interrupciones de nivel igual o inferior a la prioridad que se está atendiendo actualmente
  - Mecanismos híbridos: combinación de esquemas centralizado y distribuido (*daisy-chain*) **ver Tema5 tr.70-72, 62**
  - Registro de máscara: la CPU (o controlador de IRQ) posee un registro en el que cada bit bloquea un nivel de interrupción, pudiendo desactivar selectivamente los niveles que no se desee atender
  - Flag general de inhibición: la CPU posee un flag y un par de instrucciones para activarlo y desactivarlo