

Cuestionario Teórico

Recomendación sobre cómo usar el cuestionario:

- ✓ Repase la clase en cuestión.
- ✓ Realice la ejercitación práctica.
- ✓ Estudie como si fuera a dar un parcial sobre ese tema específico.
- ✓ Responda el cuestionario tratando de no ayudarse con ninguna otra fuente que sus propios conocimientos.
- ✓ Si lo intentó y aun así no puede responder la pregunta, puede ayudarse del material teórico o investigar.
- ✓ Verifique sus respuestas contrastando con los apuntes y el material teórico.
- ✓ Refuerce sus conocimientos y respuestas con la puesta en común en clase, de ser necesario consulte sus dudas con el profesor.
- ✓ Repase lo aprendido antes del parcial.

Módulo 24 – Excepciones

- (001) ¿Qué es una excepción? ¿Qué es el manejo de excepciones (exception handling)?
- (002) ¿Qué son las excepciones en C#? ¿Qué tienen en común todas las excepciones?
- (003) ¿Qué es el Stack Trace o pila de llamadas? ¿En qué orden se lee?
- (004) ¿Cómo capturo una excepción? ¿Cuál es la función de cada bloque?
- (005) En el caso de una estructura try-catch se tiene más de un tipo de bloque catch, ¿se podría ejecutar más de un bloque catch que forme parte de la misma estructura?
- (006) ¿En qué parte del código continúa la ejecución del programa una vez manejada una excepción?
- (007) ¿Existe una forma de capturar cualquier excepción sin importar su tipo? ¿Qué habría que considerar si se tiene más de un tipo de bloque catch?
- (008) ¿Qué sucede cuando se lanza una excepción? ¿Qué sucede si no la manejo/controlo?
- (009) ¿Cómo lanzo una excepción?
- (010) Dentro de un bloque catch, ¿cuál es la diferencia entre “throw;” y “throw ex;” (ex es un identificador para una excepción capturada)?
- (011) ¿Se lanzan en tiempo de compilación o de ejecución? ¿Por qué?
- (012) ¿Cómo creo una excepción propia?
- (013) ¿Qué es la propiedad InnerException? Describa a qué clase pertenece, su contenido y cómo se carga. ¿Qué sucede si no se le proporciona un valor?
- (014) Bloque finally: ¿En qué condiciones se ejecutará el código que contiene? ¿Cómo se ubica dentro de una estructura de manejo de excepciones? ¿Para qué es útil?

Módulo 25 – Pruebas de calidad

(015) Indique el orden y describa brevemente qué sucede en cada etapa del ciclo de vida general de los sistemas. (mantenimiento – diseño – desarrollo/construcción – implementación – análisis – pruebas)

(016) ¿Qué es una prueba unitaria?

(017) ¿Qué es una prueba integral?

(018) ¿Qué es una prueba funcional?

(019) ¿Qué es el patrón AAA? Describa cada una de sus etapas.

(020) ¿Qué es la clase Assert? ¿Para qué sirve?

Módulo 26 – Tipos genéricos

(021) ¿Qué es una clase genérica? ¿Qué permite?

(022) ¿Se puede tener más de un parámetro genérico en una clase o un método?

(023) ¿Qué es una restricción o constraint?

(024) ¿Qué sucede si no hay restricciones?

(025) ¿Qué sucede si intentamos instanciar una clase genérica pasando como argumento un tipo que no cumple con las restricciones?

(026) ¿Puedo declarar métodos genéricos en clases no-genéricas?

(027) ¿Sólo se puede aplicar una sola restricción por parámetro?

(028) ¿Qué nombre pueden tener los comodines o parámetros genéricos?

(029) Complete la siguiente tabla:

Restricción	Descripción
where T : struct	
where T : class	
where T : notnull	
where T : new()	
where T : <clase base >	
where T : <interface >	

where $T : U$	
---------------	--

Módulo 27 – Interfaces

- (030) ¿Qué es una interfaz?
- (031) ¿Qué puede especificar una interfaz (atributos, métodos, propiedades, etc)?
- (032) ¿Qué nivel de visibilidad/acceso pueden tener los miembros especificados?
- (033) ¿Las clases sólo pueden implementar una interfaz?
- (034) ¿Se puede elegir cuáles de las operaciones definidas en la interfaz implementar en una clase?
- (035) ¿Una clase puede heredar de otra y al mismo tiempo implementar una o más interfaces?
- (036) ¿Las interfaces se heredan?
- (037) ¿Qué significa implementar una interfaz de forma explícita? ¿Qué utilidad tiene? ¿Qué consecuencias o efectos tiene?

Módulo 28 – Archivos y serialización

- (038) ¿Qué es serializar?
- (039) ¿Para qué sirve serializar?
- (040) ¿En qué formatos se puede serializar (vistos en clase)? Indique qué miembros de la clase se incluyen en cada formato.
- (041) Indique qué características debe tener la clase para ser serializable en cada formato.

Módulo 29 – Bases de datos

Módulo 30 – Hilos

Módulo 31 – Eventos

Módulo 32 – Métodos de extensión