

Guía de Estudio: Clase 02.

Programación orientada a objetos:

1. ¿Qué propone el **paradigma orientado a objetos**? ¿Qué es un paradigma?
2. Nombre a los **pilares** del paradigma orientado a objetos.
3. ¿Qué es una **clase**?
4. ¿Qué es un **objeto**? ¿En qué se relacionan con las clases?
5. ¿Qué significa **instanciar** un objeto?
6. Explique qué es la **abstracción** en el contexto de programación orientada a objetos y cuál es su relación con las clases.
7. ¿Qué son los **atributos** o estado de un objeto?
8. ¿Qué son los **métodos** de un objeto?

Miembros estáticos y de instancia (no estáticos):

9. ¿Qué es un **método estático**? ¿En qué se diferencia de los métodos de instancia (no estáticos)?
10. ¿Qué es una **clase estática**? ¿En qué se diferencia de las clases no estáticas?
11. ¿Puedo tener miembros estáticos en clases no-estáticas? ¿Puedo tener miembros no-estáticos en clases estáticas?
12. ¿Necesito instanciar un objeto de la clase para llamar a sus métodos estáticos?
13. ¿Cuántos valores distintos pueden existir para un campo estático y cuántos de uno no-estático cuando se crearon 5 instancias de la clase?
14. ¿Se puede usar el **operador "this"** dentro de un método estático? ¿Se puede acceder a miembros no-estáticos desde un método estático de la misma clase? ¿Se puede acceder a miembros estáticos desde un método de instancia?
15. ¿Se pueden declarar variables estáticas dentro de un método (locales)?
16. De dos ejemplos de métodos estáticos que pertenezcan a las clases de .NET Framework.

17. De un ejemplo de un método de instancia que pertenezca a las clases de .NET Framework.

Namespaces y directivas:

18. ¿Qué es un **namespace** y cuál es su función principal?
19. ¿Puedo tener distintos namespaces dentro de un mismo proyecto o ensamblado de .NET?
20. ¿Para qué se usa la **directiva using**?
21. ¿Para qué se usa la **directiva alias**?
22. ¿Puedo declarar dos clases independientes/distintas con el mismo identificador dentro del mismo namespace? ¿Y en namespaces distintos?