Guía de Estudio : Clase 02.

**Programación orientada a objetos:**

1. ¿Qué propone el **paradigma orientado a objetos** ? ¿Qué es un

paradigma?

Un **paradigma** es una teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento

2. Nombre a los **pilares** del paradigma orientado a objetos.

-Abstraccion , -Encapsulamiento , -Herencia , - Polimorfismo

3. ¿Qué es una **clase** ?

Una clase es un conjunto de objetos que tienen características comunes . Clasificadas en base a comportamientos, y atributos escenciales de los objetos.

4. ¿Qué es un **objeto** ? ¿En qué se relacionan con las clases?

Un objeto es un conjunto de atributos y procedimientos que representan algo.

-Atributos . Metodos. Operaciones , Relaciones , Identificador

5. ¿Qué significa **instanciar** un objeto?

Inicializar un objeto con x valores.

6. Explique qué es la **abstracción** en el contexto de programación

orientada a objetos y cuál es su relación con las clases.

En programación , consiste en seleccionar las características mas relevantes y comportamientos en común de un conjunto de objetos , definiendo nuevos tipos de entidades , que usamos para armar nuevas clases que se adecuen a lo que estemos trabajando.

7. ¿Qué son los **atributos** o estado de un objeto?

Los atributos son variables que contienen los valores específicos de cada objeto. Cada objeto tiene su propio conjunto de datos.

**[modificador] tipo Identificador;**

8. ¿Qué son los **métodos** de un objeto?

Los métodos son funciones o algoritmos que producirán cambios en los atributos del 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)?

Los métodos y atributos estáticos son atributos asociados a la clase y no a una instancia particular ( pertenecen a todo el conjunto , no a uno en especial).

-Pueden ser accedidos sin necesidad de crear un objeto de la clase.

-No pueden acceder a los métodos no estáticos de la clase.

-Los métodos estáticos se utilizan para procesar datos sin necesidad de acceder a los atributos de un objeto.

Los metodos de instancia son utilizados sobre el objeto en que se llamen , no sobre toda la clase.

-Es necesario crear un objeto para utilizarlos.

10. ¿Qué es una **clase estática** ? ¿En qué se diferencia de las clases no

estáticas?

Una [static class](https://msdn.microsoft.com/en-us/library/79b3xss3.aspx" \t "_blank) es aquella clase que se usa sin necesidad de realizar una instanciación de la misma. Se utiliza como una unidad de organización para métodos no asociados a objetos particulares y separa datos y comportamientos que son independientes de cualquier identidad del objeto.  
Las clases estáticas son adecuadas cuando no tienen que almacenar información, sino sólo realizar cálculos o algún proceso que no cambie.

Sólo puede contener [constructores estáticos](https://msdn.microsoft.com/en-us/library/k9x6w0hc.aspx) y privados.

11. ¿Puedo tener miembros estáticos en clases no-estáticas? ¿Puedo

tener miembros no-estáticos en clases estáticas?

-Si puedo tener miembros estáticos en clases NO-STATIC,

-No puedo tener miembros NO estáticos en clases STATIC.

12. ¿Necesito instanciar un objeto de la clase para llamar a sus métodos

estáticos?

Nop.

13. ¿Cuántos valores distintos pueden existir para un campo estático y

cuántos de uno no-estático cuando se crearon 5 instancias de la clase?

Valor estatico -> 1

Valor no-estatico -> 5

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?

a->No podes usar this , porque los métodos estáticos no necesitan un objeto , y se podría dar el caso que no referencies a nada.

b->Nop.

c->Sip.

15. ¿Se pueden declarar variables estáticas dentro de un método

(locales)?

Como poder se puede , pero un static pertenece a toda la clase , asi que no tiene sentido declararla adentro de un método , hay que declararla en el scope de la clase .

16. De dos ejemplos de métodos estáticos que pertenezcan a las clases de

.NET Framework.

Math.Abs();

Console.WriteLine();

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

de .NET Framework.

Yo que se alguno que cree yo.

Random.next

**Namespaces y directivas:**

18. ¿Qué es un **namespace** y cuál es su función principal?

Es una agrupación de clases y otros elementos.

Proporciona un marco de trabajo jerarquico sobre el cual se construye y organiza el código.

Sirve para organizar el código y reducir los conflictos entre nombres.Hace posible que en un programa se puedan utilizar componentes de distinta procedencia.

19. ¿Puedo tener distintos namespaces dentro de un mismo proyecto o

ensamblado de .NET?

Sep.

20. ¿Para qué se usa la **directiva using** ?

Para “incluir” todo un namespace y nos evita de utilizar los nombres completos de ubicación a la hora de llamar métodos.

21. ¿Para qué se usa la **directiva alias** ?

Para referirse a los namespace de una manera diferente, se utiliza para nombres largos.

22. ¿Puedo declarar dos clases independientes/distintas con el mismo

identificador dentro del mismo namespace ? ¿Y en namespaces

distintos?

En el mismo namespace no , en diferentes si np.