Guía de Estudio : Clase 06

**Formularios:**

1. ¿Los formularios son objetos?

Los formularios son objetos que exponen propiedades, métodos que definen su comportamiento y eventos que definen la interacción con el usuario.

Se instancia en el main.

1. ¿De qué clase heredan todos los formularios?

: Form

1. ¿Qué es una **partial class** o clase parcial?

El concepto de Partial Class, que se incorpora en .NET 2.0, permite separar el código de una clase en dos archivos fuentes diferentes.

4. ¿Puedo agregar parámetros de entrada a la clase del formulario? ¿Y

sobrecargar el constructor? ¿Y declarar nuevos campos/propiedades?

Si,si,si.

1. ¿Cuál es la diferencia entre **Show()** y **ShowDialog()** ?

Show ()

Visualiza el formulario. Puede especificarse su formulario Owner (dueño o propietario).

Showdialog()

Puede utilizar este método para mostrar un cuadro de diálogo modal en la aplicación.

6. ¿Qué es un **formulario MDI** ? ¿Con qué propiedad indico que un

formulario es un contenedor MDI? ¿Con qué propiedad del formulario

hijo indico cuál es el formulario MDI padre?

Mdi =

interfaces de múltiples documentos (MDI),

permite mostrar varias ventanas dentro de una ventana padra,

(SDI) seria solo una ventana como el notepad

Propiedad : isMdiContainer (true o false )

MdiParent = formPadre;

7. Explique el **ciclo de vida** de los formularios asociándolo a sus eventos

correspondientes.

New (Cuando se crea la instancia )

Load ( cuando carga antes que sea visible los controles)

Paint ( pasa a ser visible )

Activated ( se activa cada q/ hacemos foco) .

Form.closing ( se esta cerrando )

Form.closed (ya se cerro , no es visible)

Disposed ( se libero memoria)

**Arrays:**

8. ¿Qué es un **array “jagged”** ? ¿En qué valor se inicializan sus elementos?

Jagged = anidado ;

Un Arrayanidado es un Array de Arrays, por lo tanto, al ser tipos por referencia se inicializan en null.

El valor por defecto de Arrayde elementos numéricos (value types) se establece en cero, mientras que los objetos (reference types) se establece en null.

9. ¿Los arrays son objetos?

Aquí los Arraysson objetos derivados de la clase System.Array.

10. ¿Qué significa que en C# los arrays son de **“base-cero”** ?

Inicia en 0 , un Arraycon ‘n’ elementos está indexado de 0 a n-1.

11. Los arrays implementan la **interfaz IEnumerable** , ¿qué es lo que esto

les permite hacer?

Usar foreach

12. ¿Cómo se declara e instancia un **array multidimensional** ?

[acceso] tipo[ , ] identificador;

13. Considerando que la **propiedad Length** devuelve la cantidad TOTAL de

elementos de TODAS las dimensiones del array, ¿qué valor vamos a

mostrar por consola en el siguiente código?:

int[ , , ] a = new int[ 3, 2, 3 ];

int[ , ] b = new int[ 5, 4 ];

Console.WriteLine(“Array Length = {0}”, a.Length + b.Length);

3\*2\*3 + 5\*4

38

**Strings:**

14. ¿Es lo mismo declarar una variable como string (en minúscula) o como

String (con la primera letra en mayúscula)? ¿Por qué?

Si es lo mismo , porque es un alias de la clase String

15. ¿Se puede recorrer un string con un foreach? ¿Por qué?

Supongo que si , porque si son arrays, deberían heredar el IEnumerable

16. ¿Qué significa que los strings son **inmutables** ? ¿qué sucede en

realidad cuando usamos métodos u operadores para modificar un

string?

En [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) y [programación funcional](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional) un «objeto inmutable» es un [objeto](https://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)) cuyo estado no puede ser modificado una vez creado. Cuando se modifica , en realidad se crea uno nuevo y se elimina el anterior.