

Clases

Una clase es una estructura de datos que combina estados (campos) y acciones (métodos y otros miembros de función) en una sola unidad.

De una clase se pueden hacer instancias y objetos para usar sus funciones etc y permitir la reutilización de código

Las clases admiten herencia y polimorfismo, mecanismos por los que las clases derivadas pueden extender y especializar clases base.

```
//[access modifier] - [class] - [identifier]
public class Customer {
    // Fields, properties, methods and events go here...
}
```

Clases anónimas

Las clases anónimas son una manera cómoda de encapsular un conjunto de propiedades en un único objeto sin tener que definir primero un tipo explícitamente

Los tipos anónimos contienen una o varias propiedades públicas de solo lectura. No es válido ningún otro tipo de miembros de clase, como métodos o eventos. La expresión que se usa para inicializar una propiedad no puede ser `null`, una función anónima o un tipo de puntero.

```
var v = new {
    Nombre = "asier",
    apellido = "garcia",
    edad = 22
};
Console.WriteLine(v.Nombre + v.apellido + v.edad);
```

System.Object

`Object` en .Net es una clase de la que hereda absolutamente todo, `object` es la clase mas base que se va a heredar siempre con el fin de disponer siempre los métodos:

- `ToString()` : nos va a permitir devolver un **string** referente al tipo de dato o la clase que estamos usando.
- `Equals()` : es el método esencial para comparar entre diferentes objetos
- `GetHashCode()` : nos permite obtener el identificador único que posee el objeto en cuestión

Estos métodos al ser heredados se pueden sobrescribir, tratar, etc. Por ejemplo

```
public override bool Equals (object obj) {  
    // Se puede borrar la linea  
    // Se programa el contenido de comparación  
    return base.Equals(obj);  
}  
public override int GetHashCode () {  
    // Se puede borrar la linea  
    // Se programa la obtención del identificador del objeto  
    return base.GetHashCode();  
}  
public override string ToString () {  
    // Se puede borrar la linea  
    // Se realiza el paso del objeto a ToString, también se puede  
    // retornar con un format diferente etc.  
    return base.ToString();  
}
```