Clases:

* Es una construcción que permite crear tipos personalizados propios mediante la agrupación de [variables](https://www.ecured.cu/Variables) de otros tipos
* Define los datos y el comportamiento de un tipo

Tipos de clase

* Public: Son muy comunes, accesibles desde cualquier otra clase en la misma librería (de otro modo hay que importarlas).
* Abstract: Aquellas que tienen por lo menos un método abstracto. No implementan sus [métodos](https://www.ecured.cu/M%C3%A9todos), sino que dan las bases para que sean implementados en la herencia.
* Final: Son las que terminan la cadena de herencia. Útiles por motivos de seguridad y eficiencia de un programa, ya que no permiten crear más sub-divisiones por debajo de esta clase.
* Synchronizable: Especifica que sus métodos son sincronizados, evitando problemas con los thread (hilo de ejecución), de forma que estos no pueden empezar a correr un método si no ha acabado el otro.

Características

* Nombre de la clase. Sirve para identificar a todos los objetos que tengan unas determinadas características.
* Conjunto de atributos. Datos miembros. El valor de los atributos representan el estado de cada objeto.
* Conjunto de métodos. [Funciones](https://www.ecured.cu/index.php?title=Funciones&action=edit&redlink=1) miembro. Permite que los objetos cambien de estado, dependiendo del estado anterior que tuviera el objeto.
* Niveles de acceso para proteger ciertos miembros de la clase. Normalmente, se definirán como ocultos (privados) los [atributos](https://www.ecured.cu/index.php?title=Atributos&action=edit&redlink=1) y visibles (públicos) los métodos.

Herencia

La herencia se realiza a través de una derivación, lo que significa que una clase se declara utilizando una clase base de la cual hereda los datos y el comportamiento. Una clase base se especifica anexando dos puntos y el nombre de la clase base

 Cuando una clase declara una clase base, hereda todos los miembros de la clase base excepto los constructores

A diferencia de C++, una clase en C# solo puede heredar directamente de una clase base

dado que una clase base puede heredar de otra clase, una clase puede heredar indirectamente de varias clases base