**Dotnet** - Es una herramienta para administrar código fuente .NET. De cierta manera es un un marco de trabajo que brinda una arquitectura básica para realizar programas utilizando en su principio tecnologías de Microsoft en entornos de Microsoft, con muchas facilidades para los desarrolladores.

En el caso de windows desde PowerShell en la consola de comandos se digitan los siguientes comandos:

**dotnet** , verificas que todo está instalado correctamente.

**dotnet new** , Verificas que la herramienta está correctamente instalada y te muestra una serie de instrucciones que puedes utilizar para crear aplicaciones con .Net.

**dotnet new console** , creas primera aplicación de consola, te generará tres archivos:  
obj

platzi.csproj (proyecto de C-Sharp)

Es un archivo xml donde te indica la versión de “netcorespp”

Program.cs (código fuente de C-Sharp)

Nota-.Con el comando cat mas el nombre del archivo puedes visualizar que contiene cada uno de los archivos creado

**dotnet new console --output** , le indicas donde quieres que cree la aplicación

**dotnet run** , ejecuta el programa, inicia el proceso de compilación y analiza

**dotnet run --projetc**, cual programa deseo ejecutar d

**dotnet build** , compila, genera un ejecutable.  
Se abre la carpeta bin hasta llegar al ejecutable.dll, es el ejecutable que ya tiene la lógica del programa para ejecutar se utiliza la siguiente instrucción: dotnet ejecutable.dll

El punto dll es universal para cualquier plataforma, ya está analizado y [compilado.Es](http://compilado.es/" \t "_blank) propia de netcore

**dotnet build -c Release -r win10-x64**, creas un .exe , lo optimizas para windows. Un ejecutable especial para Windows. En la carpeta de Release la abres hasta llegar al ejecutable.exe

**A. CLASES, OBJETOS, MÉTODOS, ACCIONES Y ATRIBUTOS**

**1. ¿Qué puede ser descrito como un objeto?**  
Todo puede ser descrito como un objeto, con ciertas características y métodos, por ejemplo:  
Toda escuela tiene un nombre, una ubicación, un tipo, esas serian sus características. Mientras que sus métodos pueden ser iniciar clases, terminar clases, sonar el timbre, etc.

**2. Un objeto NO es una clase.**  
La clase es la descripción de un objeto, pero no el objeto, más bien es una plantilla para crear objetos.

**3. Objetos**  
Un objeto es una cosa tangible, algo a que se puede aprehender intelectualmente o algo hacia lo que se puede dirigir una acción o pensamiento.  
Un objeto representa un item individual e identificable, o una entidad real o abstracta, con un papel definido en el dominio del problema.  
Un objeto tiene:  
− Estado  
− Comportamiento  
− Identidad  
La estructura y el comportamiento de objetos similares se definen en sus clases comunes.  
El término objeto y ejemplo (instance) de una clase son intercambiables.

**4. Clases**  
Una clase es un conjunto de objetos que comparten una estructura y comportamiento comunes.  
Clase representa una abstracción, la esencia que comparten los objetos.  
Un objeto es un ejemplo de una clase.  
Un objeto no es una clase, y una clase no es un objeto (aunque puede serlo, p.e. en Smalltalk).  
Las clases actúan como intermediarias entre una abstracción y los clientes que pretenden utilizar la abstracción.  
De esta forma, la clase muestra:  
visión externa de comportamiento (interface), que enfatiza la abstracción escondiendo su estructura y secretos de comportamiento.  
visión interna (implementación), que abarca el código que se ofrece en la interface de la clase.