

[Directory](#) y [DirectoryInfo](#)**Qué diferencias hay:**

La forma de utilizarlas, de poder trabajar con sus métodos...

En Directory SIEMPRE que se utilizan sus métodos hay que pasar como argumento la ruta (el path) mientras que en DirectoryInfo creamos ya un objeto con esa ruta, lo guardará y luego sobre él aplicaremos los métodos que queramos

Ejem.: Obtener los archivos del directorio C:\pruebas

- Directory
`Directory.GetFiles("c:\pruebas")` Devuelve los archivos de esa ruta
- DirectoryInfo
`Dim dir As New DirectoryInfo("c:\pruebas")`
`dir.GetFiles()`

Primero creamos un objeto de la clase DirectoryInfo que estará asociado a esa ruta
Y obtenemos los archivos de ese objeto

Métodos "iguales" en las dos clases:

Las dos clases tienen muchos métodos iguales o parecidos, por lo que podemos utilizar el que más nos guste; aunque siempre teniendo en cuenta que los métodos, aunque sean "iguales", puede que no obtengan los mismos tipos de datos, que no tengan los mismos argumentos, que no se escriban igual, ...

Ejemplo1: Obtener los directorios del directorio c:\pruebas

- Directory
`Dim dirs() as String`
`dirs= directory.getDirectories("c:\pruebas")`
En este caso GetGirectories devuelve un array de Strings
- DirectoryInfo
`Dim dirPri As DirectoryInfo("c:\pruebas")` *Una instancia de DirectoryInfo hace referencia a un directorio en particular (en este caso a c:\pruebas)*
`Dim dirs() As DirectoryInfo`
`Dirs = dimPri.getDirectories`
En este otro caso, getDirectories devuelve un array de DirectoryInfo

Ejemplo2: Comprobar si c:\pruebas existe

- Directory

```
Dim ex As Boolean = Directory.Exists("c:\pruebas")
```

La clase directory tiene un método exists

- DirectoryInfo

```
Dim dir As New DirectoryInfo("c:\tareass")
```

```
Dim ex As Boolean = dir.Exists
```

La clase DirectoryInfo tiene una propiedad llamada exists (de ahí que no lleve paréntesis)

Ejemplo3: mover el directorio dMover que está en el directorio dOrigen al directorio dDestino

- Directory

```
Directory.Move("c:\dOrigen\dMover", "c:\dDestino")
```

En directory tenemos el método Move que mueve la carpeta al destino

- DirectoryInfo

```
Dim dir As New DirectoryInfo("c:\dOrigen\dMover")
```

```
dir.ToMove("c:\dDestino")
```

En DirectoryInfo hay un metodo llamado ToMove que mueve la carpeta al destino

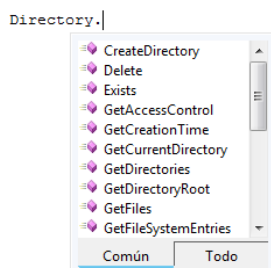
Métodos que sólo están en una clase:

Algunos métodos sólo están en una de estas dos clases.

¿Cómo saber si están en una o en otra?

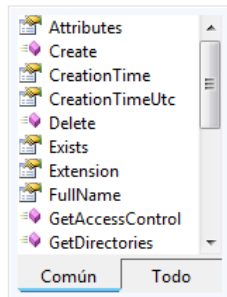
- O bien mirando en la ayuda del msds ([Directory](#) y [DirectoryInfo](#))
- O intentando que el propio Visual Basic nos lo diga escribiendo un poquito de código...

Para saber si esta en Directory:



Para saber si esta en DirectoryInfo:

```
Dim dir As New DirectoryInfo("asdasd")
dir.
```



Saber qué parámetros precisa y/o qué devuelve un método...

Otro problema que solemos tener a la hora de utilizar los métodos es saber qué devuelven o qué tenemos que pasar como parámetros.

Aquí volvemos a lo mismo:

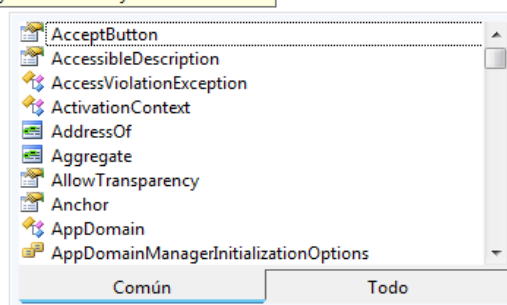
- O miramos la ayuda en Internet ([Directory](#) y [DirectoryInfo](#))
- O intentamos que Visual Basic nos eche una mano:

Para ello escribimos más o menos el método que queremos y situamos el cursor encima, aparecerá una pantalla de lo que traduciremos a grandes rasgos cuales son los argumentos y lo que devuelve (si devuelve algo, que puede que no)

Ejemplo:

```
Directory.GetDirectories( )
```

1 de 3 GetDirectories (path As String) As String()
path: The path for which an array of subdirectory names is returned.



Nos dice que hay 3 maneras de hacerlo la primera pasando un path(ruta) como Stringy que nos devuelve un array de String(String()).

.... Con la ayuda de las flechitas veremos las otras dos maneras...



Ejemplo:

```
Dim dir As New DirectoryInfo("asd")  
dir.Parent( )
```

Parent () As System.IO.DirectoryInfo
Gets the parent directory of a specified subdirectory.

Aquí nos dice que no hay que pasar parámetros y nos devuelve un DirectoryInfo