

Modificadores de acceso en C#

Por Gerardo Álvarez 01/04/2019



Los tipos y miembros de C# cuentan con un nivel de accesibilidad, debido a esto tenemos los modificadores de acceso.

Cada [modificador de acceso](#) establece el grado de acceso al tipo, desde el propio proyecto o desde otro proyecto de la solución.



```
·public·int·valor1;  
···private·int·valor2;  
····protected·int·valor3;  
······internal·int·valor4;
```

Esta administración permite dar mayor seguridad a los datos delimitando los accesos.



Todos los modificadores de acceso se anteponen en la [declaración del tipo](#) (métodos, variables, clases, etc.).

Public

El acceso a estos tipos está disponible desde el propio proyecto o desde otro sin ninguna restricción.

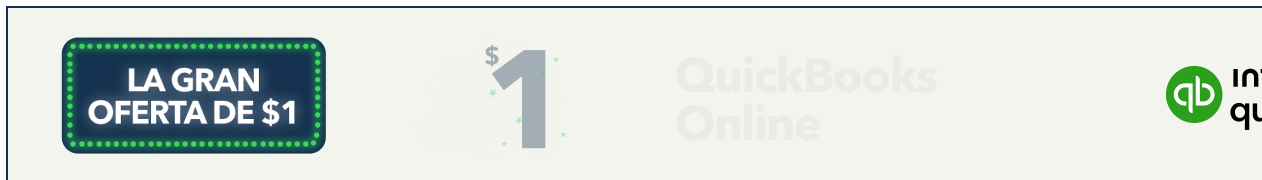
```
public void Sumar()  
{  
    //Operaciones  
}
```

Private

Permite el acceso solo a los miembros de la misma clase.

```
private void Resta()  
{  
    //Operaciones  
}
```

Protected



Accesible solo a los miembros de la misma clase y de las clases hijas de este.

```
protected void División()  
{  
    //Operaciones  
}
```

Internal

El acceso está disponible desde cualquier clase del mismo proyecto, pero no de otros proyectos de la misma solución.

```
internal void Multiplicacion()
{
    //Operaciones
}
```

Los anteriores modificadores de acceso son los más utilizados, aunque podemos encontrar dos más, Protected internal y Private protected.

Tipos y miembros

Entre los tipos más comunes que podemos agregar modificadores tenemos las clases, métodos, variables, propiedades, etc.

Clases

```
public class Persona
{
}
```

Variables

```
private int Precio;
```

Propiedades

```
public int Total { get; set; }
```

Para comprobar los niveles de acceso realiza las pruebas correspondientes, genera métodos con diferentes niveles de acceso.

Comprueba desde la misma clase, desde otra clase del mismo proyecto y desde otro proyecto la accesibilidad de los métodos.



#Csharp

#Visual Studio

[← PREVIOUS](#)[Colección List de una clase en C#](#)[NEXT →](#)[Arrays o Matrices en C#](#)

Deja un comentario

*Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con **

Comment *

Name *

Publicar comentario



CATEGORÍAS

Alimentos (16)

Cultura general (1)

Desarrollo de software (109)

News (1)

Notas de desarrollo (108)

Procesos organizacionales (32)

Tecnología (66)

ARCHIVOS



USAMOS ANUNCIOS PARA MANTENER
NUESTRO SITIO WEB ONLINE.

POR FAVOR, DESACTIVE EL ADBLOCK DE SU
NAVEGADOR PARA CONTINUAR NAVEGANDO.

MUCHAS GRACIAS

© 2023 Kyocode

[Inicio](#) [Acerca de](#) [Blog](#) [Contacto](#)

[Política de Privacidad](#)