<https://msdn.microsoft.com/es-es/library/ms229047(v=vs.110).aspx>

# Diseño de clases abstractas

[Otras versiones](javascript:void(0))

https://i-msdn.sec.s-msft.com/Areas/Epx/Content/Images/ImageSprite.png?v=636437933319989116

Publicado: octubre de 2016

**X no** definir constructores internos públicos o protegidos en tipos abstractos.

Los constructores deben ser públicos sólo si los usuarios necesitan crear instancias del tipo. Porque no se puede crear instancias de un tipo abstracto, un tipo abstracto con un constructor público está correctamente diseñado y confuso para los usuarios.

**✓ hacer** definir un constructor interno o protegidos en clases abstractas.

Un constructor protegido es más común y permite la clase base para realizar su propia inicialización cuando se crean subtipos.

Un constructor interno puede utilizarse para limitar las implementaciones concretas de la clase abstracta para el ensamblado que define la clase.

**✓ hacer** proporcionar al menos un tipo concreto que herede de cada clase abstracta que realiza el envío.

Haciendo esto ayuda a validar el diseño de la clase abstracta. Por ejemplo, [System.IO.FileStream](https://msdn.microsoft.com/es-es/library/system.io.filestream(v=vs.110).aspx) es una implementación de la [System.IO.Stream](https://msdn.microsoft.com/es-es/library/system.io.stream(v=vs.110).aspx) clase abstracta

<https://msdn.microsoft.com/es-es/library/ms229047(v=vs.100).aspx>

# Diseño de clases abstractas

**Visual Studio 2010**

[Otras versiones](javascript:void(0))

https://i-msdn.sec.s-msft.com/Areas/Epx/Content/Images/ImageSprite.png?v=636437933319989116

Dado que nunca se deberían crear instancias de las clases abstractas, es importante definir correctamente sus constructores. También es importante garantizar que la funcionalidad de su clase abstracta sea correcta y se pueda extender con facilidad. Las instrucciones siguientes ayudan a garantizar que las clases abstractas se diseñan correctamente y funcionan como se espera cuando se implementan.

**No defina constructores de tipo public o protected internal (Protected Friend en Visual Basic) en tipos abstractos.**

Los constructores con visibilidad **public** o **protected internal** son para los tipos de los que se pueden crear instancias. Nunca se pueden crear instancias de los tipos abstractos.

**Defina constructores protegidos o internos en clases abstractas.**

Si define un constructor protegido en una clase abstracta, la clase base puede realizar tareas de inicialización cuando se crean instancias de una clase derivada. Un constructor interno impide el uso de una clase abstracta como clase base de los tipos que no están en el mismo ensamblado que la clase abstracta.

**Proporcione por lo menos un tipo concreto que herede de cada una de las clases abstractas que distribuya.**

Esta práctica ayuda a los diseñadores de bibliotecas a localizar problemas o descuidos en el diseño de la clase abstracta. También significa que, para los escenarios de alto nivel en los que puede que los desarrolladores no comprendan las clases abstractas y la herencia, pueden utilizar la clase concreta sin necesidad de aprender estos conceptos. Por ejemplo, .NET Framework proporciona las clases abstractas [WebRequest](https://msdn.microsoft.com/es-es/library/system.net.webrequest(v=vs.100).aspx) y [WebResponse](https://msdn.microsoft.com/es-es/library/system.net.webresponse(v=vs.100).aspx) para controlar el envío de solicitudes a un identificador de recursos uniforme (URI) y recibir respuestas del mismo. Como una de varias implementaciones concretas de estas clases abstractas, Framework incluye las clases[HttpWebRequest](https://msdn.microsoft.com/es-es/library/system.net.httpwebrequest(v=vs.100).aspx) y [HttpWebResponse](https://msdn.microsoft.com/es-es/library/system.net.httpwebresponse(v=vs.100).aspx), que son implementaciones específicas de HTTP de las clases abstractas.