

La iniciativa NET cuatro años después

Vicent-Ramon Palasí Lallana

La iniciativa NET cuatro años después

Vicent-Ramon Palasí Lallana
Director Académico
Universidad Francisco Gavidia
vpalasi@ufg.edu.sv

En junio del año 2000, Microsoft anunció la iniciativa NET, como la estrategia a seguir por la compañía de Redmond para crear un nuevo futuro en el área del software para computadoras. Más de tres años después de este anuncio, es un buen momento para detenerse y reflexionar qué es lo que ha sido de aquella iniciativa, cuál ha sido su nivel de éxito y qué implicaciones tiene en el área de desarrollo de sistemas.

En este artículo, se evalúan los diferentes aspectos de la iniciativa NET, con un énfasis en el área de herramientas de desarrollo. Por motivos didácticos, se comienza explicando lo más específico para acabar con lo más general. Así, primero se describe Visual Basic NET. Después se presenta la plataforma NET y sus lenguajes. A continuación, se describe la iniciativa NET y se evalúa el éxito de sus diversos aspectos. Finalmente, se incluye una serie de recomendaciones sobre la conveniencia o no de actualizar el desarrollo a NET.

1. EL LENGUAJE VISUAL BASIC NET

Comencemos con el lenguaje Visual Basic NET. La idea fundamental que debe tenerse en cuenta en relación a éste es que es un lenguaje de programación com-

pletamente nuevo. Al contrario de lo que indica su nombre, Visual Basic NET no es el mismo lenguaje que el Visual Basic conocido en el pasado, cuya última versión fue la 6. Es falsa la idea de que Visual Basic NET es la nueva versión de Visual Basic, pues se trata de lenguajes completamente diferentes. Así, los programas escritos en Visual Basic 6 no pueden ejecutarse en Visual Basic NET y tampoco a la inversa. Algo todavía más importante es que la filosofía y la forma de programar de ambos lenguajes es totalmente distinta. Por ello, los programadores en Visual Basic 6 no están capacitados para desarrollar en el nuevo lenguaje sino una capacitación adicional, incluso más extensa que la que se necesitaría para un lenguaje tradicional.

Dejando a un lado la confusión producida por el nombre, si Visual Basic NET se parece a algún otro lenguaje, no es al Visual Basic tradicional, sino al lenguaje Java. Visual Basic NET recoge las ideas fundamentales de Java y las implementa en un entorno de Windows, de forma que entre ambos lenguajes hay muchas más similitudes que diferencias.

Otra idea fundamental de Visual Basic NET es que es orientado a objetos. Sería imposible explicar en un espacio tan reducido como el del presente artículo en qué con-

interesado encontrará fácilmente información abundante sobre el tema). Baste decir que los programas orientados a objetos se componen de un conjunto de módulos de programación o subprogramas, llamados clases, que contienen tanto datos como procesos que se ejecutan sobre estos datos. La programación orientada a objetos se ha convertido en el estándar actual de programación debido a sus múltiples ventajas, entre las que podemos destacar una mayor robustez de los programas resultantes, mayor escalabilidad, flexibilidad y aumento en la reutilización de código ya programado.

2. EJECUCIÓN DE UN PROGRAMA EN VISUAL BASIC NET

En un lenguaje de programación tradicional, la ejecución de un programa es la que se muestra en la figura 1. En ella, llamamos "plataforma hardware" al conjunto del hardware y del sistema operativo. El programa se encuentra compilado, es decir, traducido en un lenguaje binario (el llamado "lenguaje máquina") que la plataforma hardware reconoce. Este programa puede ser ejecutado directamente por dicha plataforma, como puede observarse en la figura.

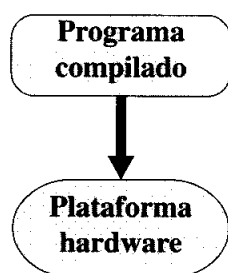


Figura 1. Ejecución de un programa tradicional

La ventaja de este enfoque tradicional es que es rápido, eficiente y directo. La desventaja es que, como cada plataforma hardware entiende un lenguaje máquina diferente, el programa compilado es completamente específico para esta plataforma. Esto hace imposible que un programa compilado para un PC pueda ejecutarse, por ejemplo, en una máquina Windows CE.

Por el contrario, en Visual Basic NET, la ejecución de un programa es la que aparece en la figura 2 (nótese que en las figuras de la 1 a la 4 del presente artículo, los rectángulos representan archivos con código de programación y las figuras redondeadas simbolizan programas —o plataformas— en ejecución). El programa compilado es ahora un conjunto de clases (pues se trata de un lenguaje orientado a objetos). Pero estas clases no están en lenguaje máquina, sino en un lenguaje intermedio llamado MSIL ("MicroSoft Intermediate Language"). Como este lenguaje no es entendido por la plataforma hardware (la cual, como sabemos, sólo conoce el lenguaje máquina), se necesita un programa para ejecutarlo. Este programa se llama CLR ("Common Language Runtime") y se ejecuta directamente sobre la plataforma. Microsoft ofrece de forma gratuita un CLR para cada plataforma Windows.

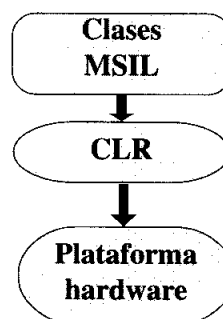


Figura 2. Ejecución de un programa en Visual Basic NET

Como se ve en la figura, este enfoque es más complicado que el anterior y, por ello, su utilidad no es evidente a primera vista. Sin embargo, hay que notar que el lenguaje MSIL es independiente de la plataforma hardware y, por ello, puede ejecutarse en cualquier plataforma (siempre que exista un CLR para ella, lo que, como hemos visto, sucede para todas las plataformas Windows, aunque no para el resto). Así, el programa compilado en un PC se ejecutará en una Windows CE, sin más necesidad que copiarlo.

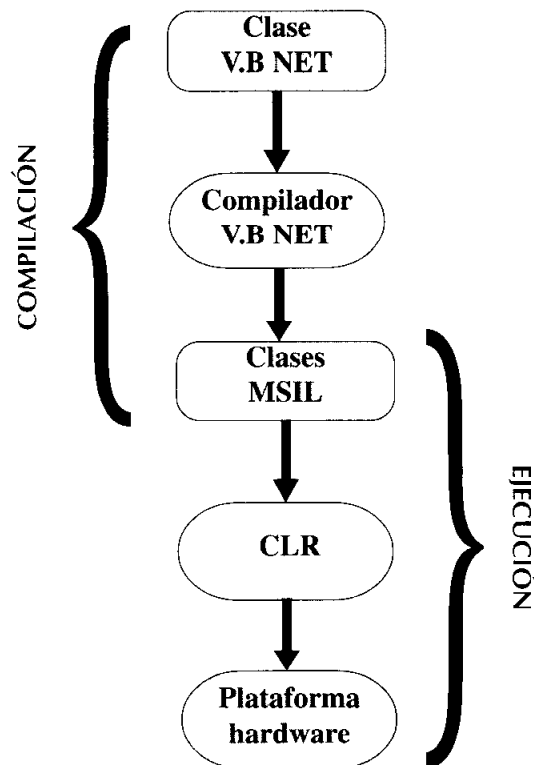
Esto da independencia de la plataforma, siempre que tenga Windows como sistema

operativo. Dicho de otra manera, da independencia del hardware. (Aunque existe un proyecto de código abierto, llamado Mono, que intenta implementar el CLR sobre Linux, es dudoso que esto permita que en el futuro que los programas NET sean independientes del sistema operativo, por razones que sería muy largo explicar aquí).

3. COMPILACIÓN DE UN PROGRAMA EN VISUAL BASIC NET

Hasta ahora, nos hemos centrado en la ejecución de un programa, pero, ¿qué hay de su compilación? Ésta se representa en la figura 3.

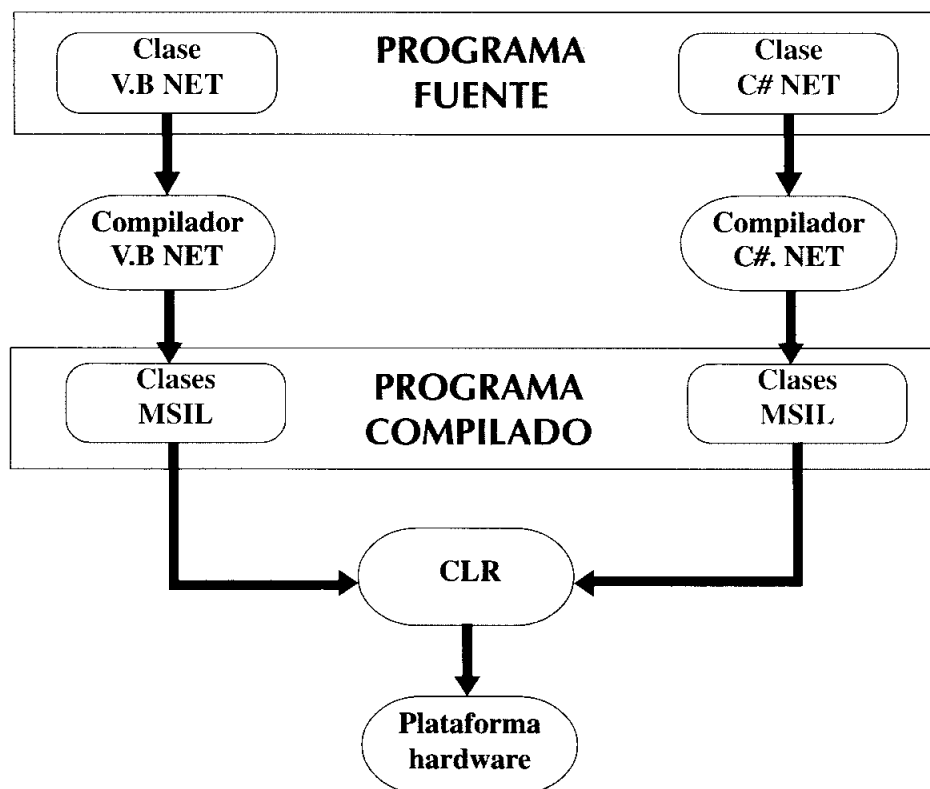
Figura 3. Compilación y ejecución de un programa en Visual Basic NET



Como se observa en la figura, a partir del programa fuente en Visual Basic NET, el compilador crea un programa compilado en clases MSIL, el cual es ejecutado por el CLR. Hasta aquí no hay conceptos adicionales a los que se han presentado anteriormente.

Sin embargo, como ahora el programa no se ejecuta sobre la plataforma hardware sino sobre el CLR, esto nos permite programar algunos aspectos adicionales en este último. Uno de los más interesantes es permitir la ejecución de un programa escrito en más de un lenguaje de programación. Esto se muestra en la figura 4.

Figura 4. Compilación y ejecución de un programa en varios lenguajes NET



Como se ve en la figura, el programa contiene clases escritas en Visual Basic NET y en C# NET. Estos son dos lenguajes diferentes, pero, al ser compilados, los dos producen clases MSIL, las cuales pueden ser ejecutadas conjuntamente por el CLR. De esta forma, podemos tener programas realizados en varios lenguajes simultáneamente (siempre que éstos sean NET) y estos funcionan tan correctamente como un programa tradicional con un único lenguaje.

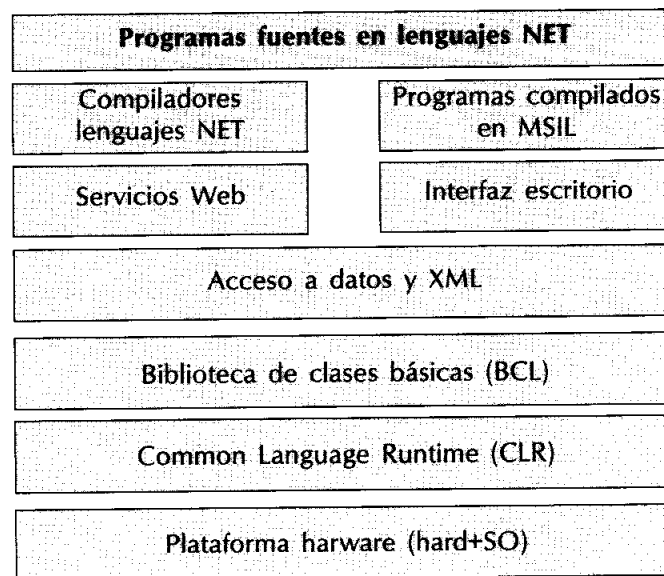
De esta forma, la programación en NET no se reduce a un único lenguaje de programación, sino que abarca un conjunto de lenguajes que pueden combinarse libremente con el fin de utilizar “la herramienta correcta para cada tarea”. Entre los lenguajes NET podemos destacar C# NET, C++ NET, J# NET, F# NET (ofrecidos por Microsoft) y COBOL NET, RPG NET y Fortran NET (ofrecidos por terceras partes).

4. LA PLATAFORMA NET

Hasta ahora hemos hablado de los lenguajes de programación NET. Sin embargo, la programación en estos lenguajes está intrínsecamente ligada a la llamada plataforma NET (o “NET framework”, en inglés) que vamos a presentar brevemente a continuación.

La plataforma NET es un conjunto de componentes software (programas y librerías) que se usa para compilar y ejecutar programas escritos en los lenguajes NET. Su estructura se refleja en la figura 5. En esta figura, los componentes de la plataforma están en color (los componentes en blanco no forman parte de la plataforma NET, pero se incluyen para indicar cómo se relacionan con ella).

Figura 5. Arquitectura de la plataforma NET («NET framework»)



Como se ve en la figura, la plataforma NET se ejecuta sobre la plataforma hardware. Uno de sus componentes es el CLR, que ya hemos visto. Todos los otros componentes se ejecutan sobre el CLR y de esta forma son independientes de la plataforma hardware (siempre que ésta sea Windows).

Sobre el CLR se ejecutan una serie de librerías (en amarillo en la figura) que son utilizadas por los programas, simplificando el desarrollo al ofrecer una serie de servicios ya programados, listos para reutilizar. Hay una librería básica que contiene soporte para estructuras de datos, interoperabilidad con el código NET, entre otros servicios. Hay librerías para acceso a bases de datos y XML así como para interfaz gráfica, tanto para el escritorio como para el Web.

Estas librerías son usadas por los programas compilados (escritos, como hemos visto, en lenguaje MSIL). Entre estos programas podemos destacar a los compiladores para los diferentes lenguajes NET, que traducen los programas en cada uno de estos lenguajes a programas compilados MSIL, como se ha visto anteriormente.

La plataforma NET es gratuita y libremente distribuible. Es un requisito imprescindible para desarrollar y ejecutar los programas NET y su curva de aprendizaje es lenta, debido a la gran cantidad de clases existentes en las diferentes librerías.

5. LA INICIATIVA NET

Hasta ahora nos hemos centrado en el desarrollo en NET, lo cual es lógico, ya que el presente artículo está dirigido a desarrolladores. En el presente apartado,

nos centraremos en el conjunto de la iniciativa NET, en sus causas, consecuencias y estado actual.

La iniciativa NET surge como intento de solución a varios problemas técnicos y comerciales planteados a Microsoft en los últimos años. En el área técnica, la tardanza de la compañía en reconocer la importancia de Internet produjo que otras tecnologías, como Apache y Java, tomaran el liderazgo de esta nueva área. Ello convirtió a J2EE (un conjunto de tecnologías derivadas del lenguaje Java) en el estándar para aplicaciones empresariales robustas y escalables y dejó a las herramientas de programación de Microsoft en el área de desarrollos pequeños y medianos.

Comercialmente, después del crecimiento espectacular de los años ochenta y noventa, Microsoft entró en una fase de estancamiento debida a la reducción de ventas de computadoras, a la ausencia de características interesantes que impulsarían a los usuarios a actualizarse a las nuevas versiones y, en definitiva, a la madurez de su mercado tradicional (los sistemas operativos y aplicaciones de escritorio). Además, nuevas leyes de declaración de impuestos hicieron que los ingresos variaran enormemente entre los trimestres en que se lanzaba alguna nueva versión importante y los que no, produciendo una indeseable fluctuación en el precio de las acciones de la compañía. Como solución a estos problemas de estancamiento y fluctuación, Microsoft intentó encaminarse a un modelo de suscripción con el que, en vez de que el usuario comprara una licencia permanente de un programa, pagara una suscripción por usarlo durante un periodo de tiempo. Aunque este modelo de suscripción ha ido imponiéndose con los

clientes corporativos, los consumidores se han resistido notablemente a él.

Como solución a todos estos problemas técnicos y comerciales, surge la iniciativa NET, que es un nombre único con el que Microsoft agrupó a un variado conjunto de tecnologías y estrategias empresariales con los siguientes objetivos:

1. Introducirse en el área de desarrollos empresariales de gran tamaño que, en la actualidad, domina J2EE. Para ello, se crearon los lenguajes y plataforma NET que se han explicado en este artículo.
2. Evolucionar hacia un modelo ASP ("Application Service Provider") de ejecución de aplicaciones. Con este modelo, los programas no se encontrarían en el disco duro de una máquina del cliente, sino se ubicarían en un servidor de Internet de la compañía desarrolladora, la cual vendería el derecho de utilizarlos a través de Internet por un periodo de tiempo, implementando así un modelo de suscripción. Así, por ejemplo, Microsoft podría cobrar una cuota mensual por usar Office, ya que éste se encontraría en los servidores de Microsoft y no en el disco duro del cliente.
3. Convertirse en la plataforma de elección para los servicios Web. Los primeros años desde la aparición de la iniciativa NET se caracterizaron por el auge del mercadeo de los servicios Web, que fueron presentados de forma exagerada como la tecnología que iba a dominar el mundo del desarrollo. Como estrategia publicitaria, la plataforma NET se presentó como una forma de implementar los servicios Web, aunque era más que eso.

4. Implementar una serie de servicios Web estándar ofrecidos por Microsoft y hacerlos tan ubicuos como el Windows. A los servicios principales se les dio el nombre de "Hailstorm" y, más adelante, de "My Services". Uno de los más destacados es Microsoft Passport, un servicio cuyo objetivo era guardar toda la información personal y de tarjetas de crédito del usuario para que éste no debiera autenticarse en cada página Web.

El éxito de estas tecnologías ha sido desigual. Por una parte, el modelo ASP y los servicios Hailstorm han enfrentado una fuerte oposición de los usuarios por razones obvias, por lo que no han prosperado y han sido prácticamente abandonados. Los servicios Web siguen siendo importantes, pero no son la solución mágica que se creía y la plataforma NET sólo es una de varias alternativas para su implementación. Microsoft ha dejado de añadir la expresión "NET" a todos sus nuevos productos (así, lo que una vez fue "Windows NET Server" hoy es "Windows Server 2003"). Asimismo, la incorporación de código NET a los productos de Microsoft es muy limitada.

Por otra parte, en el área de desarrollo, NET no sólo no ha sido abandonado, sino que goza de buena salud y de perspectivas para el futuro. De hecho, la plataforma NET es un paso positivo para el desarrollo en Windows, ya que lo dota de mayor escalabilidad y robustez, aunque se trata todavía de una plataforma que no está madura. Las librerías de la plataforma pueden considerarse como un reemplazo orientado a objetos de la API Win32, del acceso a datos y a Internet, con todas las ventajas que ello supone.

6. LA CONVENIENCIA DE ACTUALIZARSE A LA PLATAFORMA NET

¿Hasta qué punto es conveniente actualizarse a la plataforma NET? Como en cualquier decisión de esta naturaleza, la respuesta no es sencilla y dependerá de las diferentes circunstancias y contextos en el que se produzcan los proyectos de desarrollo. Por su importancia, destacamos los siguientes factores:

- **Tamaño del proyecto.** La plataforma NET (igual que J2EE) es adecuada para proyectos de gran tamaño, que requieren robustez y escalabilidad. Sin embargo, es inconveniente para proyectos pequeños o medianos. Por ello, si su proyecto de programación es de tamaño moderado, le aconsejamos que evalúe otros lenguajes de programación, como PHP, Python, Perl, Delphi o Powerbuilder. En el caso de que su proyecto sea de gran tamaño, la plataforma NET puede ser una elección adecuada, aunque le recomendamos mejor J2EE, la cual está más madura para el desarrollo de este tipo de aplicaciones.
- **Novedad del proyecto.** Se debe analizar si el proyecto consiste en crear un

programa nuevo o bien mantener uno existente. En el primer caso, se aplican las consideraciones que se hicieron en el último párrafo. En el segundo caso, le recomendamos que evalúe la vida útil futura del programa. Si esta vida útil futura no pasa de un par de años, le recomendamos que siga manteniéndolo en el mismo lenguaje en que se desarrolló. Sin embargo, si la vida útil futura excede el periodo de dos años, nuestra recomendación depende del lenguaje en que está programado. Si está programado en Visual Basic 6 (o anteriores), le recomendamos que actualice a Visual Basic NET (recuerde que esto implica una reprogramación, es decir, una inversión considerable). En caso de que esté programado en un lenguaje diferente a Visual Basic, le recomendamos que evalúe otros lenguajes hasta que la plataforma NET esté más madura.

Estas recomendaciones se resumen en la figura 6 (en esta figura, donde pone "Otros lenguajes", se refiere a los lenguajes que se han mencionado en el párrafo titulado "Tamaño del proyecto").

Figura 6. Conveniencia o no de actualizarse a Visual Basic NET

LENGUAJE RECOMENDADO SEGUN PROGRAMA	Programa de gran tamaño	Programa de tamaño pequeño
Programa nuevo	J2EE (VB NET)	Otros lenguajes
No nuevo. Menos de 2 años de vida útil futura.	Continuar con el mismo lenguaje	Continuar con el mismo lenguaje
No nuevo. Más de 2 años. En Visual Basic 6.	VB NET	Otros lenguajes
No nuevo. Más de 2 años. En otro lenguaje	J2EE (VB NET)	Otros lenguajes

Todo lo anterior se refiere a los proyectos de desarrollo de sistemas. En cuanto a los recursos humanos, si usted tiene en su empresa desarrolladores en el ambiente Visual Basic, le aconsejamos que los capacite para que puedan programar en Visual Basic NET. Si estos desarrolladores conocen J2EE, esta capacitación es relativamente sencilla. En caso contrario, no es trivial y puede llevar de 6 meses a 2 años hasta que sus programadores puedan programar en NET con calidad, por lo que le

aconsejamos que lo plantee como una inversión de futuro.

CONCLUSIÓN

Aunque muchos aspectos de la iniciativa NET no han prosperado, el desarrollo en NET ha venido para quedarse. Por ello, ser consciente de la plataforma NET y del desarrollo usando la misma nos ayudará a estar mejor posicionados y nos preparará para las novedades que vengan en el futuro.