

Programación con C# .NET

Tema 1:
Introducción a .NET



Introducción

- Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años
- .NET ofrece una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados *remotamente* y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.
- www.microsoft.com/net
- www.microsoft.com/latam/net/



Características de .NET

- .NET es una nueva plataforma para el desarrollo y explotación de aplicaciones “gestionadas” (managed) modernas y orientadas a objetos.
- Las aplicaciones .NET se pueden desarrollar en cualquier lenguaje de programación que se ajusta a .NET
- .NET soporta una extensa colección de librerías de clases independientes del lenguaje de programación.
- .NET soporta la creación de componentes.
- .NET ofrece integración multi-lenguaje, reutilización de componentes, y herencia entre componentes desarrollados en diferentes lenguajes.



Características de .NET (cont)

- .NET ofrece una nueva manera de desarrollar aplicaciones de sobremesa usando las clases Windows Forms
- .NET ofrece una nueva manera de desarrollar aplicaciones basadas en navegador Web a través de ASP.NET
- Las clases ADO.NET proveen una arquitectura desconectada para acceso a datos a través de Internet
- .NET soporta la creación de Servicios Web XML independientes de la plataforma, a través de SOAP (Simple Object Access Protocol) y WSDL (Web Services Description Language)
- .NET ofrece una nueva arquitectura para el desarrollo y explotación de objetos remotos
- .NET convierte a varias tecnologías y técnicas Windows en obsoletas



Componentes de .NET

- Microsoft.NET está compuesto de:
 - Plataforma .NET
 - .NET Framework SDK
 - Visual Studio .NET
 - Servicios Web (Microsoft .NET myServices)
 - Servidores para empresas (SQL Server.NET...)



Plataforma .NET

- El concepto en el que se basa .NET no es nuevo.
- Java y su entorno de ejecución (JVM) ya utilizan el concepto de encapsulamiento del sistema operativo para permitir la interoperabilidad entre diferentes sistemas operativos.

Plataforma .NET (cont)

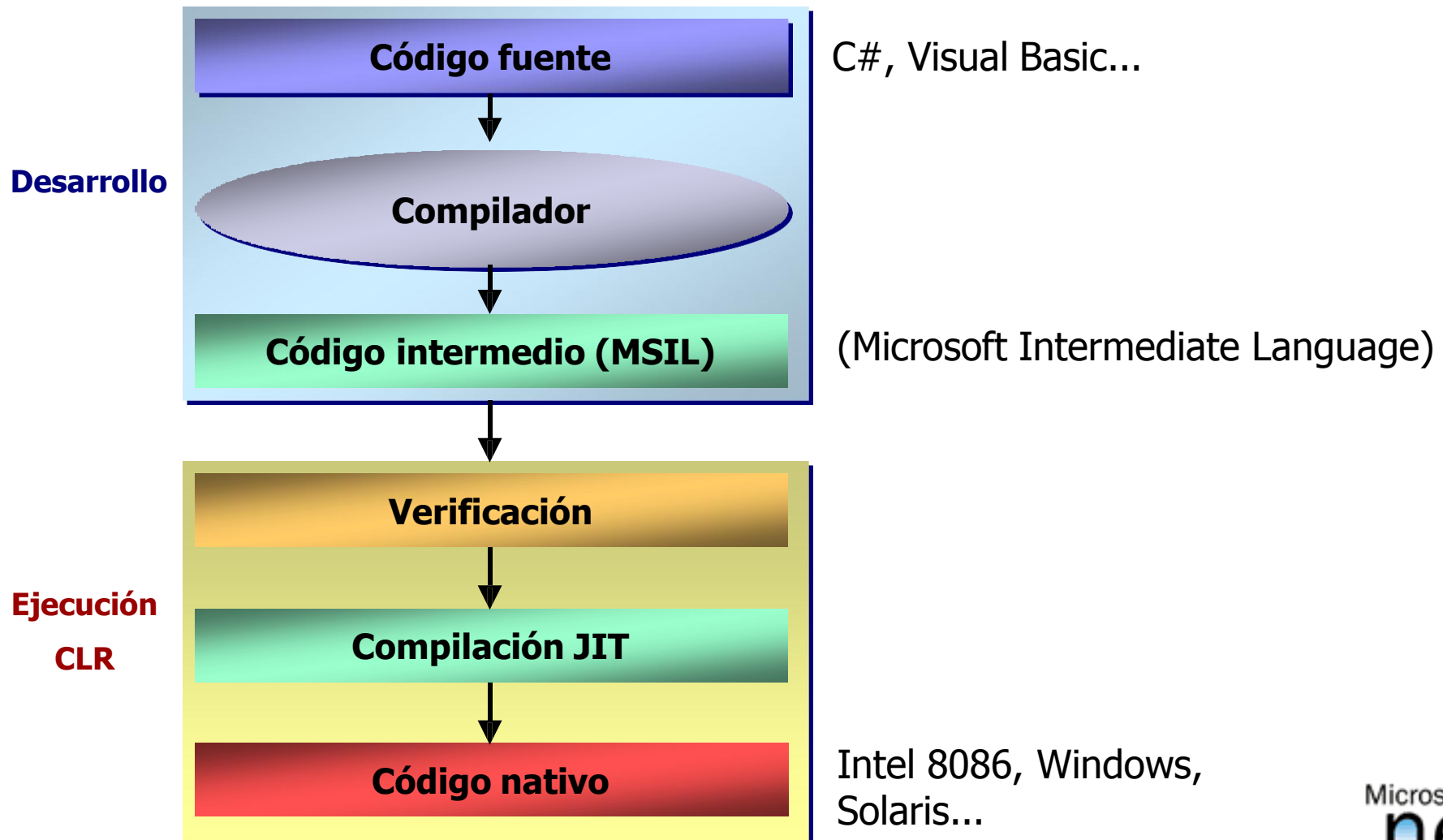




Common Language Runtime (CLR)

- Un **Runtime** es un entorno en el que se ejecutan los programas.
- El **CLR** se encarga de **gestionar** la ejecución de las aplicaciones .NET.
- Al código escrito para ejecutarse en la plataforma .NET se le llama **código gestionado**.
- Al código escrito para ejecutarse directamente se le llama **código no gestionado** o **código nativo**.
- El CLR realiza la compilación *Just in Time* (**JIT**) que traduce el código gestionado en código nativo sobre la arquitectura de hardware que esté corriendo.

Common Language Runtime (cont)





Common Language Runtime (cont)

- Características y servicios que ofrece:
 - Ejecución multiplataforma
 - Integración de lenguajes
 - Gestión de memoria (Recolector de basura)
 - Tratamiento de excepciones
 - Soporte multi-hilo (multi-threading)
 - Distribución transparente
 - Interoperabilidad con código antiguo



Clases de la plataforma .NET (BCL)

- La librería de clases (**Base Class Library**) es una librería formada por cientos de tipos que permiten acceder a los servicios ofrecidos por el CLR y a sus funcionalidades más frecuentemente usadas.
- Además el programador puede crear nuevas clases que extiendan su funcionalidad y se integren perfectamente con el resto de las clases de la BCL.



Clases de la plataforma .NET (cont)

- Esta librería de clases está escrita en MSIL
- Cualquier lenguaje cuyo compilador genere MSIL podrá usarla.
- Con esta librería podemos crear todo tipo de aplicaciones: aplicaciones de consola, de ventanas, servicios Web, ASP.NET...



Clases de la plataforma .NET (cont)

- Dada la amplitud de la BCL, ha sido necesario organizar las clases en ella incluida en **espacios de nombres (namespace)** que agrupen clases con funcionalidades similares.
- El espacio de nombres **System** es el espacio raíz del que cuelgan todos los demás.



Clases de la plataforma .NET (cont)

■ Espacios de nombres más usados:

- **System:** Contiene tipos de datos muy usados, datos básicos, tablas, excepciones... Es el raíz.
- **System.Object:** Soporta toda la jerarquía de las clases de .NET.
- **System.Collections:** Colecciones de datos de uso común como pilas, colas, listas...
- **System.Data:** Manipulación de bases de datos. Forma la denominada arquitectura ADO.NET.
- **System.IO:** Manipulación de ficheros.
- **System.Reflection:** Acceso a los metadatos que acompañan a los módulos de código.
- **System.Math:** Funciones logarítmicas, trigonométricas...
- **System.XML:** Acceso a datos en formato XML.
- **System.Web:** Aplicaciones Web.



ASP.NET

- ASP.NET es la versión para la plataforma .NET de la tecnología ASP (Active Server Pages).
- Dentro de esta capa podemos distinguir tres partes muy diferenciadas:
 - Web Forms
 - Server Controls
 - Web Services



Windows Forms

- Son una forma avanzada e integrada de crear aplicaciones de escritorio Win32 estándar.
- Descienden de las Windows Foundation Classes (WFC) de Microsoft.
- Cualquier lenguaje de la plataforma .Net puede utilizar WinForms.
- Forman parte de las clases de la plataforma .NET en el espacio de nombres:
`System.Windows.Forms`



.NET Framework SDK

- Microsoft distribuye este kit de desarrollo como parte del paquete .NET. Disponible en:

<http://msdn.microsoft.com/netframework/technologyinfo/howtoget/default.aspx>

- Este kit contiene documentación sobre la plataforma, ejemplos y código fuente, y una serie de utilidades que sirven para desarrollo y prueba de aplicaciones .NET.



Visual Studio.NET

- Es un entorno gráfico que permite a los desarrolladores crear, probar y depurar aplicaciones desarrolladas o no para la plataforma .NET.
- Inicialmente se llamó Visual Studio 7.0 ya que es la evolución del Visual Studio 6.0.
- La última versión a Octubre 2005, es el Visual Studio 2003.
- SharpDevelop es un IDE *open source* para .NET que representa una buena alternativa a Visual Studio.NET:

<http://www.icsharpcode.net/opensource/sd/>



Visual Studio.NET

- Este entorno incorpora los siguientes lenguajes de programación:
 - Visual Basic.NET: Versión .NET de Visual Basic 6.0.
 - Visual C++.net: Evolución de Visual C++.
 - Visual J#.NET: Adaptación de Visual J++.
 - Visual C#.NET: Un nuevo lenguaje de programación OO.
- Empresas ajenas han creado compiladores de sus lenguajes para la plataforma .NET:
Eiffel, Perl, Python, Haskell, Pascal, Oberon...



Desarrollando para la plataforma .NET

- .NET es independiente del lenguaje de programación, no estamos restringidos a desarrollar en Visual C++ y Basic:
 - Existen compiladores para C#, Visual Basic.NET, Jscript.NET, COBOL, Python, Perl, etc.
 - .NET soporta estos lenguajes no soportando realmente ninguno de ellos: .NET sólo entiende Microsoft Intermediate Language (MSIL)



Una plataforma independiente del lenguaje

- En .NET se traduce código fuente a IL (Intermediate Language) + Metadatos
 - IL es compilado antes de ser ejecutado y no está diseñado para un lenguaje en particular como en Java
 - Las sentencias IL manipulan tipos comunes compartidos por todos los lenguajes .NET (Common Type System – CTS)
 - El Common Language Runtime (CLR) es responsable de cargar y ejecutar una aplicación .NET
 - Usa compilación **JIT** (Just-In-Time) para traducir **IL** a código nativo
 - IL es siempre compilado y nunca interpretado
 - Cada método del código IL sólo es compilado una vez cuando es indicado.



Hola Mundo en .NET c#

// fichero: hola.cs

```
using System;
```

```
class Hola {
```

```
    public static void Main() {
```

```
        Console.WriteLine("Hola Món");
```

```
    }
```

```
}
```

- Todo programa en C# contiene al menos una clase
- Main es el punto de entrada del programa
- `Console.WriteLine("...")`, visualiza un mensaje, invocando método `WriteLine` de la clase `Console`
- El espacio de nombres `System` pertenece a .NET FCL (Framework Class Library)
- BCL contiene muchas clases útiles, que pueden usarse desde nuestras aplicaciones .NET



Enlaces de interés

- GotDotNet: The Microsoft .NET Framework Community:
 - <http://www.gotdotnet.com/>
- .NET home page:
 - <http://www.microsoft.com/net/>
- MSDN .NET Framework site:
 - <http://msdn.microsoft.com/netframework/>
- Website con ejemplos y ayuda en programación C#:
 - <http://www.c-sharpcorner.com/>
 - <http://www.csharp-help.com/>
 - <http://www.csharp-station.com/>
- Implementación Open Source .NET (Proyecto mono):
 - <http://www.go-mono.com/>
- Fundamental: Class Library en la sección Reference de .NET Framework SDK Documentation



Bibliografía

- **C#. Curso de Programación.**
 - Autor: Fco. Javier Ceballos Sierra.
 - Editorial:
 - RA-MA en España.
 - Alfaomega Grupo Editor en América.
- **Enciclopedia de Microsoft Visual C#.**
 - Autor: Fco. Javier Ceballos Sierra.
 - Editorial:
 - RA-MA en España.
 - Alfaomega Grupo Editor en América.