



Framework .NET



Marzo de 2017

Introducción .NET

- ▶ Microsoft. NET puede ser considerado una visión de cómo el software debe ser por escrito, y un conjunto de herramientas para desarrollar software que se da cuenta de esta visión [Microsoft].
- ▶ Vistazo a los problemas comerciales comunes de conectividad e interoperabilidad:
 - ▶ La mayoría de las empresas cooperen con otras empresas, sin embargo, sus sistemas de información funcionan de manera aislada.
 - ▶ Cadenas no están integradas. Esto es a menudo visto como un obstáculo para mejorar la productividad.
 - ▶ Cuando las empresas están mejor conectadas, pueden lograr una mayor eficiencia.

Introducción al Framework .NET

- ▶ Los programas de C# se ejecutan en .NET Framework, un componente que forma parte de Windows y que incluye un sistema de ejecución virtual denominado Common Language Runtime (CLR) y un conjunto unificado de bibliotecas de clases.

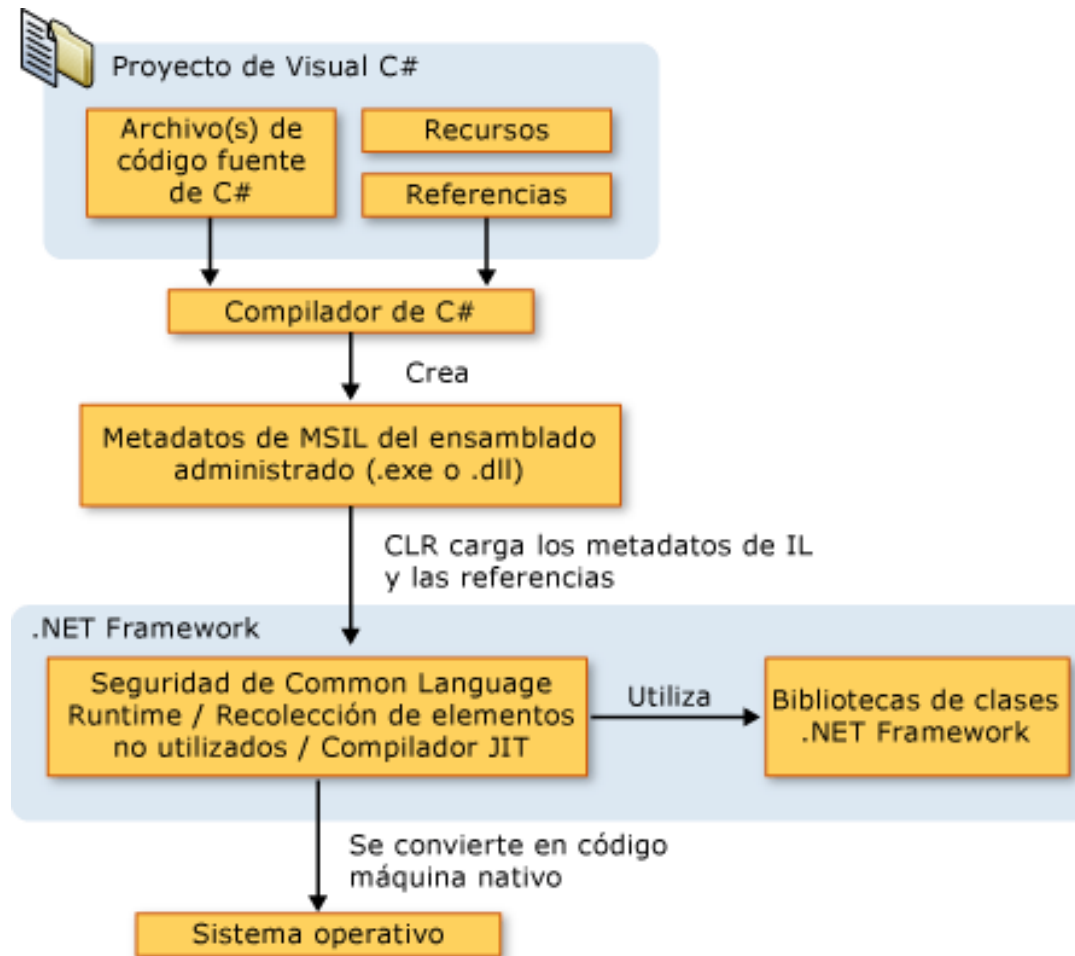
Introducción al Framework .NET

- ▶ CLR es la implementación comercial de Microsoft de CLI (Common Language Infrastructure), un estándar internacional que constituye la base para crear entornos de ejecución y desarrollo en los que los lenguajes y las bibliotecas trabajan juntos sin ningún problema.

Introducción

- ▶ El código fuente escrito en C# se compila en un lenguaje intermedio (IL) conforme con la especificación CLI.
- ▶ El código de lenguaje intermedio y recursos se almacenan en disco en un archivo ejecutable denominado ensamblado, cuya extensión es .exe o .dll generalmente.
- ▶ Un ensamblado contiene un manifiesto que proporciona información sobre los tipos, la versión, la referencia cultural y los requisitos de seguridad del ensamblado.

Compilación y ejecución de un programa C#



Compilación y ejecución de un programa C#

- ▶ El código ejecutado por CLR se denomina algunas veces "código administrado", en contraposición al "código no administrado" que se compila en lenguaje máquina nativo destinado a un sistema específico.
- ▶ En el diagrama anterior muestra las relaciones en tiempo de compilación y tiempo de ejecución de los archivos de código fuente de C#, las bibliotecas de clases de .NET Framework, los ensamblados y CLR.

Componentes del Framework .NET

ASP.NET, Windows Forms, ADO.NET and XML

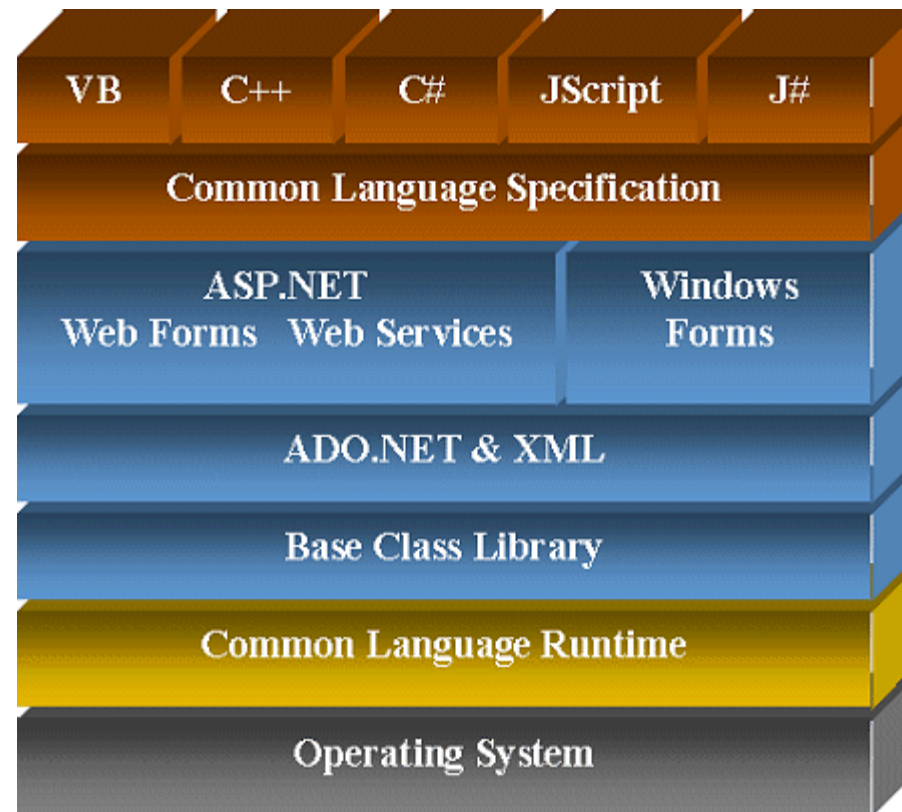
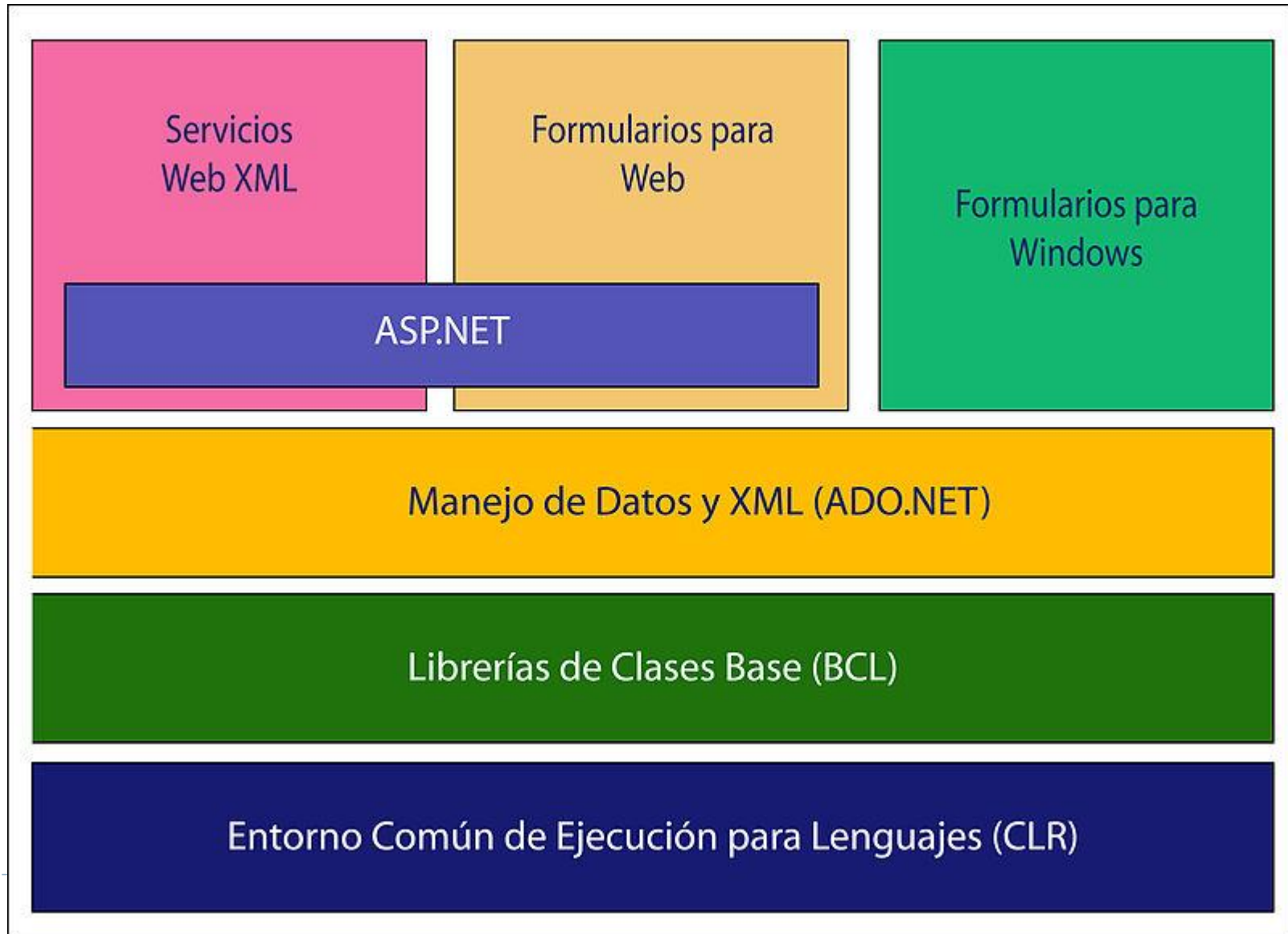


Diagrama CLR



Microsoft
.net
Development
Series

A selection of new types and namespaces



0077-7462/91

Microsoft
.net
Development
Series



.NET 4 Poster

Complemented by the MIT Development Tools and  **Addison-Wesley**
Pearson Education

What's new in .NET Framework 4.5?

Windows Presentation Foundation

Built-in Ribbon controls ★
 Databinding improvements
 Ability to add breakpoints to databindings
 Data source change aware views (Live Shaping) ★
 Validation improvements ⚡
 Improved legacy UI integration
 Dispatcher improvements ⚡
 Speed-up of large data sets 📈

Windows 8 support ★

Support for
Windows Runtime (WinRT)

.NET Profile
for Metro-style apps

Improved support for sharing
DLLs between .NET profiles

ASP.NET

Built-in providers now
natively support SQL Azure

Support for implementing
WebSocket receivers ★

Built-in JavaScript + CSS
combining and minification ★

ASP.NET MVC 4

Async controllers ⚡
 Built-in mobile templates +
jQuery.Mobile support

Alternate views (e.g.
print version, mobile site)

Support for Recipes:
intelligent codegen ★

ASP.NET Web Pages 2

New site templates
 Versatile validation support
 Support for OAuth and OpenID ★
 Built-in map embedding tools;
supports Google, Bing and others

Did you know? ASP.NET Web Pages is yet
another way to work the web besides Web
Forms and MVC. WP sites use Razor and
are typically developed with WebMatrix.

Web Forms

Strongly typed data binding
 MVC-like support for Models ★
 HTML encoded binding expressions

HTML5 support ★

Control support for
new semantic
elements
 Validator and UpdatePanel now
support new HTML5 elements

Multifile support
for FileUpload
control

Tooling

IntelliSense
improvements
 More JavaScript
support
 IIS Express
used by default

New templates
and snippets

Asynchronous pipeline support ⚡
(Response, Request, HttpHandlers)

Performance improvements: Multicore JIT, 35 % faster startup,
memory optimizations, assembly sharing between sites, pre-fetch support 📈

Request validation improvements:
AntiXSS built-in, validation usable per field

Windows Communication Foundation

Support for UDP multicast channels
 TCP channels now work with partial trust
 Asynchronous operations ⚡
 Streaming improvements

Simplified configuration (again!)
 Can now generate service stubs from WSDL
 WebSocket support ★

Windows Workflow Foundation

C# Expressions
 State machine workflows are back! ★
 Workflow versioning ★
 Code-first activity design
 Faster execution 📈
 Designer usability improvements

Managed Extensibility Framework 2.0

Debugging improvements
 Support for explicit bindings between objects
 Support for binding POCOs: no more attribute requirements

ADO.NET

Sparse columns support improved (SQL Server)
 Passwords are now stored encrypted
 Asynchronous operations ⚡

SQL Express LocalDB

New light version of SQL
Express for developer
use. Supported in .NET
4.5. separate patch for
4.0 is coming.

SQL Server 2012 ("Denali") Support

High Availability support on connection string level
 Fast failover across multiple subnets
 Support for new spatial data types (polygons, arcs etc.) ★

Entity Framework 4.5

Enumeration support
 Migrations for schema changes ★
 Designer improvements

Spatial data type support ★
 Table-valued function support ★
 Multi-result sproc support

Multiple diagrams per model
 Code-first support ★
 Auto-compiled LINQ queries 📈

Base Class Library

Usability improved for WeakReferences and Streams
 Key interfaces now have async versions ⚡
 New ArraySegment and ReadOnlyDictionary classes
 Support for CLR objects over 2 GB in size
 Resource file management performance improved 📈
 Unicode support for console applications

Task Parallel Library ⚡

Task thread controls improved:
Task.WaitAll/WaitAny, various
timeout primitives available

TPL Dataflow: Tools for parallel
data flow processing ★

C# 5.0

Support for async programming:
async and await keywords ⚡
 Methods can access call site info
as parameters (CallerInfo)

Visual Basic 11

Iterator implementations (Yield)
 Async ja Await equal to C# ⚡
 Global keyword for namespace
handling
 Call Hierarchy view available

Visual C++ 11

C++11 standard support improved
 Auto-vectorization and
parallelization of loops ⚡
 Support for controlling
GPU-driven processing
(C++ AMP) 📈

Wait, there's more!

Visual Studio "11" is also coming, including
support for .NET 4.5, Windows 8 and more.
There are plenty of other new goodies, too.

Also, .NET team is working on Project Roslyn,
which enables interesting scenarios for
integrating the C#/VB compilers to your own
apps. Roslyn is in CTP, but will ship post-4.5.

Legend:

⚡ = Asynchrony support
 📈 = Performance improvement
 ★ = Significant new feature

Information based on public sources available
in October 2011. Changes before RTM are likely.

Composed by Jouni Heikkinen
Thanks to Sami Poimata and Jani Järvinen for the data.

www.heikkinen.net/hardcoded

01/03/2017