

Mono: mucho más que una implementación libre de .Net

Jordi Mas i Hernández, jordi@ximian.com

Septiembre 2004

El proyecto Mono¹ empezó de la mano de Miguel de Icaza² impulsor del proyecto Gnome³ y uno de sus principales programadores y arquitectos. Gnome es actualmente el escritorio más utilizado en entornos Linux, donde Icaza trabajó considerablemente en brindar su funcionalidad en diferentes lenguajes de programación y la integración entre ellos usando Corba. Aunque los esfuerzos fueron significativos, no lograron completamente los objetivos deseados.

Durante el año 2000 Microsoft presento la tecnología .Net. como una plataforma de software de última generación para simplificar el desarrollo de servicios web y aplicaciones de escritorio. Microsoft inició poco después el proceso de estandarización de la parte principal de la plataforma .Net en la asociación ECMA⁴ para su formalización y aceptación como estándar. Esta asociación ha estandarizado numerosas tecnologías, entre ellas, el lenguaje Javascript.

A principios del año 2001 se inició el proyecto Mono impulsado por la empresa Ximian (fundada por Icaza), actualmente parte de Novell⁵, como una implementación de código abierto de la plataforma .Net de Microsoft y del estándar ECMA. El objetivo original era proporcionar una plataforma de software libre que facilitara el desarrollo de aplicaciones de escritorio en entorno Linux reduciendo el tiempo y los costes de desarrollo. Con el tiempo también se ha posicionado como un entorno que permite ejecutar en Linux aplicaciones diseñadas para Microsoft .Net en entorno Windows, facilitando la migración de aplicaciones a Linux y aumentando su base de desarrolladores y usuarios. Actualmente trabajan en el proyecto Mono unos 20 ingenieros de Novell y han colaborado ya más de 300 voluntarios.

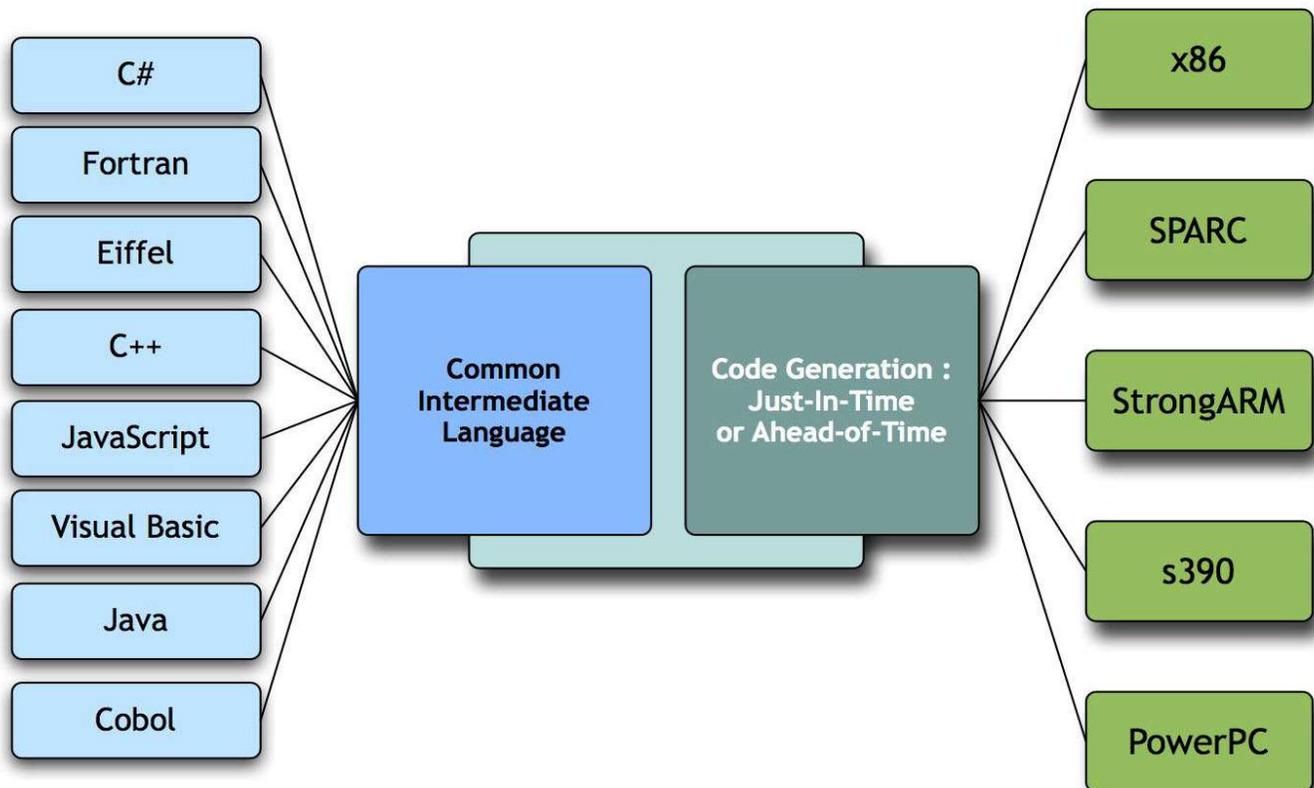
Respecto a los posibles problemas legales - una cuestión candente en todo el mundo del software libre - los componentes que se encuentran cubiertos por el estándar ECMA (parte de la librería de clases, el entorno de ejecución, el lenguaje C# y algunos componentes más) se encuentran libres de posibles problemas y está autorizada su implementación y uso para cualquier propósito⁶. En caso de que hubiera algún problema con la parte de la librería no cubierta por ECMA, Novell ha definido una estrategia⁷ que se basa en reimplementar las partes que pudieran tener problemas. Las librerías nativas de Linux (GTK#, Evolution#, etc) no deberían tener ningún problema.

Dentro de las tecnologías que introduce la plataforma .Net encontramos el lenguaje de programación orientado a objetos C# cuya sintaxis resultará familiar a los programadores de Java y C++. C# brinda las características más modernas como

gestión de memoria automática (recogida de objetos no usados), propiedades, eventos y atributos, entre otras.

Los paquetes que componen la distribución de la plataforma Mono comprenden un compilador C# , una máquina virtual (que permite ejecutar las aplicaciones), y un conjunto de librerías de clases que proporcionan miles de funciones listas para ser utilizadas con una completa documentación en línea.

Con Mono se pueden escribir aplicaciones en múltiples lenguajes de programación, incluyendo entre ellos Python, Object Pascal, Nermele, y C#. Una vez escritas las aplicaciones se traducen a CIL (*Common Intermediate Language*), que es un lenguaje intermedio que no tiene particularidades de ninguna arquitectura. Una vez compilado en CIL la aplicación se traduce al lenguaje específico de la arquitectura final donde será ejecutado. Este sistema permite distribuir un único programa binario para todas las arquitecturas en vez de un programa específico para cada plataforma. Pero no menos es importante es la libertad de escoger cualquier lenguaje de programación o combinación de ellos y poder ejecutar la aplicación en cualquiera de las plataformas en las que Mono se encuentra disponible, entre las que se incluyen Intel, AMD64, SPARC, StrongArm y S390x. Mono actualmente proporciona las herramientas para crear aplicaciones para Linux (diversas distribuciones), Solaris, Windows, Mac/OS, y *mainframes* de IBM.



A diferencia de los programas tradicionales que se ejecutan sobre el sistema directamente, los programas en la plataforma Mono se ejecutan sobre un entorno controlado de ejecución conocido como la máquina virtual. Este entorno proporciona numerosas ventajas sobre la ejecución tradicional directa: gestión de memoria automática (el sistema se encarga de recuperar automáticamente la memoria no usada por las aplicaciones simplificando la gestión a las aplicaciones), un entorno seguro de ejecución (donde podemos definir los recursos físicos y lógicos a los que la aplicación tiene acceso), y un sistema control de errores y ejecución que permite una gestión de errores avanzada.

Las aplicaciones Mono que se ejecutan en la máquina virtual se denominan *managed*, ya que son gestionadas completamente en la máquina virtual. Sin embargo, también disponemos del mecanismo *Pinvoke (platform invoke)* que nos permite acceder al mundo existente fuera de la máquina virtual, incluyendo el sistema operativo y las librerías que tengamos instaladas.

Mono proporciona las funciones necesarias para crear servicios web, esto incluye las tecnologías XML, SOAP, ASP.NET, y Remoting así como el acceso a bases de datos, a través ADO.NET, de tipo Oracle, MySQL, DB2, SQL Server, o Progress. Con estas librerías podemos portar servicios desarrollados en Windows o escribir los nuestros propios pudiendo utilizar Apache como servidor web.

Para el desarrollo de interfaces de usuario para aplicaciones cliente las opciones son múltiples. Por un lado se proporciona una implementación en código abierto de System.Windows.Forms (que es la opción que proporciona Microsoft) para que los desarrollos realizados en plataforma .Net sobre Windows puedan ser ejecutados en otras plataformas como Linux o Mac OS. También se proporciona GTK# una librería que expone toda la funcionalidad del entorno gráfico de Gnome y que permite realizar aplicaciones multiplataforma. Recientemente se ha materializado el proyecto Cocoa# que permite el desarrollo de aplicaciones nativas para Mac OS. Una parte importante de los esfuerzos del proyecto Mono se han centrado en proporcionar una completa librería de clases que facilite al máximo la reutilización de las tecnologías existentes en mundo del software libre, como Mozilla, Evolution o OpenOffice.

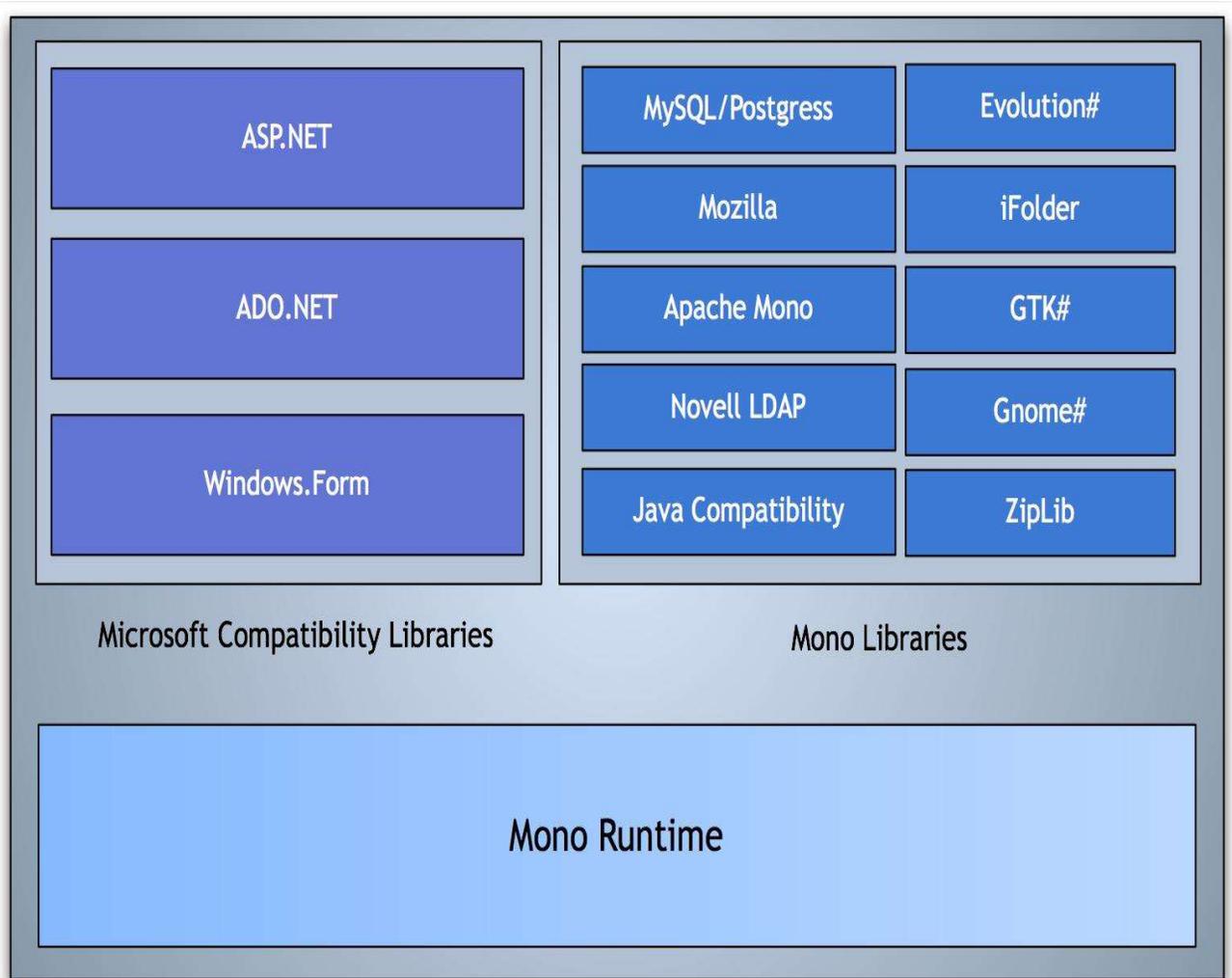
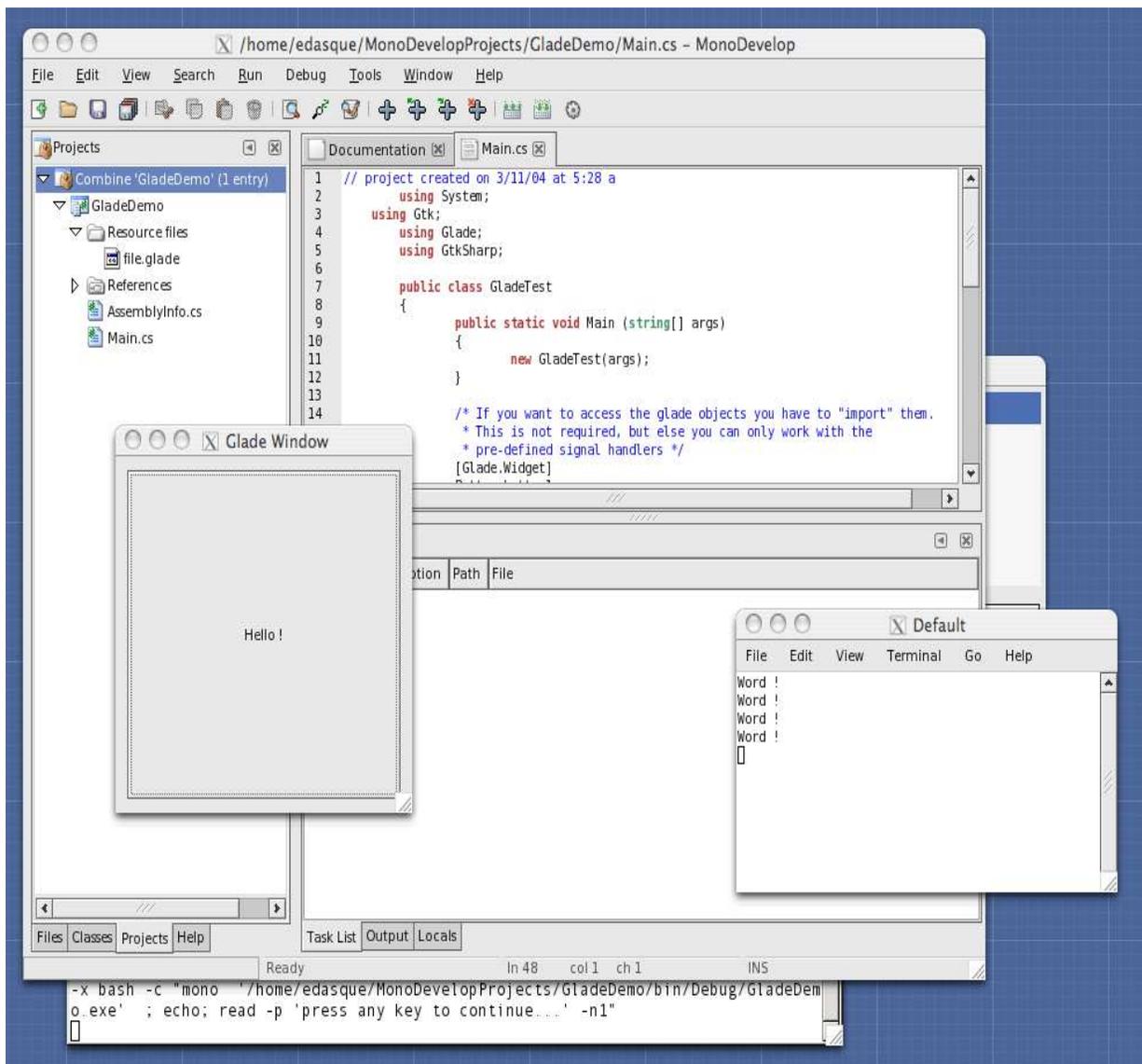


Gráfico que ilustra las principales librerías disponibles en Mono.

Dentro de las numerosas contribuciones por parte de voluntarios una especialmente significativa es Monodevelop⁸, una adaptación a Mono en plataforma Linux del entorno de desarrollo de código abierto SharpDevelop, que integra editor de textos, depurador, gestión de proyectos, y documentación en-línea. Monodevelop esta escrito usando Mono y GTK# y en sí mismo representa un ejemplo del tipo de aplicaciones que se pueden crear con Mono de forma rápida y sencilla.



Captura de pantalla del entorno de desarrollo Monodevelop

Hoy en día lo la plataforma Mono se esta utilizando para desarrollar aplicaciones Linux de última generación. Algunas de las más significativas son Dashboard⁹, un gestor de información personal e iFolder¹⁰, un sofisticado sistema de gestión de archivos. También está siendo utilizado para poder ejecutar en Linux servicios web diseñados en ASP.NET bajo plataforma Windows, como por ejemplo las numerosas aplicaciones ASP.NET que ha portado a Mono la consultora Voelcker Informatik AG para su uso en el Ayuntamiento de Munich funcionando en 350 servidores.

Conclusión

Mono es una plataforma eficaz y sencilla para el desarrollo de aplicaciones, para portar soluciones ya existentes para Windows al mundo Linux, y en general como una excelente herramienta para podernos focalizar en el desarrollo de programas

multiplataforma y servicios web. Es sin duda una plataforma que será crucial para que las empresas puedan migrar sus aplicaciones al mundo Linux y para que nuevas aplicaciones puedan ser desarrolladas con el menor coste posible.

Bibliografía

Sito web del proyecto Mono
<http://www.mono-project.com>

Standard EMCA-334: the C# Language Specification
<http://www.ecma-international.org/publications/standards/Ecma-334.htm>

Standard ECMA-335: Common Language Infrastructure (CLI)
<http://www.ecma-international.org/publications/standards/Ecma-335.htm>

Dumbill, Edd (2004) “Mono: A developer's Notebook”. O'Reilly

Schoenig, Hans (2003) “Mono Kick Start”. SAMS

© 2004 Jordi Mas i Hernández. Se garantiza el permiso para copiar y distribuir este documento completo en cualquier medio si se hace de forma literal y se mantiene esta nota.

- 1 <http://www.go-mono.com>
- 2 <http://primates.ximian.com/~miguel/>
- 3 <http://www.gnome.org>
- 4 <http://www.ecma-international.org/>
- 5 <http://www.novell.com>
- 6 <http://web.archive.org/web/20030609164123/http://mailserver.di.unipi.it/pipermail/dotnet-sscli/msg00218.html>
- 7 <http://www.mono-project.com/about/licensing.html>
- 8 <http://www.monodevelop.com>
- 9 <http://www.nat.org/dashboard/>
- 10 <http://www.novell.com/products/ifolder/index.html>