

Enrique Castilla Contreras

Tif: 661-247020 · E-mail: ecastillac@yahoo.com · Web: <http://www.geocities.com/ecastillac>

Es evidente que en un proyecto software es necesario el uso de multitud de herramientas, cada una de ellas de un tipo diferente, es decir, cada una de ellas especializada en realizar una tarea específica en una o varias de las fases del proyecto.

Mi objetivo ha sido siempre disponer de al menos, una herramienta de cada tipo, de forma que entre todas ellas cubran todas las fases y tareas más comunes de un proyecto software.

A continuación explico los tipos de herramientas que, en mi opinión, son necesarios en la mayoría de proyectos software, y que herramientas de cada uno de estos tipos utilizo yo:

- Una herramienta, basada en tecnología UML, para **especificar y modelizar software**.

[Rational Rose](#)

La herramienta de modelado UML por excelencia, como no podía ser de otra forma: los *inventores* del UML (Rumbaugh, Jacobson y Booch) colaboraron estrechamente con Rational (antes de ser adquirida por IBM), en el desarrollo de toda su familia de herramientas UML.

[UML Studio](#)

Admite todos los tipos de diagramas de UML, se pueden crear nuevos símbolos y genera código para Java y C++, y su relación precio/capacidades es inmejorable.

- Una herramienta que permita el **trabajo concurrente** de todas las personas que forman parte del equipo de desarrollo, sin importar el número de personas que lo formen ni su distribución geográfica, y que además, permita **manejar repositorios de códigos fuente**.

[CVS](#)

CVS (Concurrent Versions System) es sin ninguna duda el mejor software de este tipo.

Además de realizar a la perfección las tareas mencionadas (mas otras muchas que por espacio no puedo mencionar aquí), tiene tres importantes ventajas sobre otros productos similares: funciona en modo mezcla (otros funcionan en modo bloqueo), es multiplataforma (disponible en UNIX/Linux y Windows) y se distribuye bajo licencia GNU-GPL.

Ideal para subcontratar partes de un proyecto.

- Una herramienta que permita a los programadores **interaccionar con el repositorio**, independientemente del tipo de plataforma en la que éste resida.

[Cliente por defecto de CVS](#)

Cliente de línea de comando que se suministra en la distribución de CVS

[WinCVS](#)

Un cliente de CVS con una interfaz gráfica a base de ventanas. He traducido el manual de este producto al castellano.

Para mostrar las diferencias entre dos releases distintas del mismo fichero requiere una herramienta externa que realice esta tarea. Yo utilizo [WinDiff](#).

- Una herramienta para **generar la documentación** del software a partir de los códigos fuentes, a ser posible en varios formatos (RTF, PDF, HTML, ...).

[pod2html](#), [pod2pdf](#), [pod2latex](#), [pod2rtf](#), [pod2man](#), [pod2text](#), [pod2chm](#)

pod2html es una herramienta que se suministra en las distribuciones de Perl. Extrae de un archivo fragmentos de documentación escritos en lenguaje POD, y genera otro fichero con el resultado de traducir los fragmentos a HTML. Existen versiones para convertir de POD a cada uno de los formatos de archivo más comunes: pdf, LaTeX, rtf, man, texto, ...

[POD](#) (Plain Old Documentation) es un formato de documentación bastante *neutro* (los documentos escritos

Enrique Castilla Contreras

Tlf: 661-247020 · E-mail: ecastillac@yahoo.com · Web: <http://www.geocities.com/ecastillac>

con él se pueden generar en cualquier otro formato) y *sencillo* (POD es texto con algunas etiquetas). Cada una de estas utilidades está también disponible como un API de programación que permite generar la documentación programáticamente en el formato correspondiente, así es posible automatizar la generación de documentación en múltiples formatos.

[JavaDoc](#)

Extrae documentación de códigos fuente en Java.

Mi propia herramienta

Todavía en desarrollo. Integrará en una sola herramienta JavaDoc y las otras que se distribuyen con Perl (pod2html, pod2pdf, ...). Se distribuirá bajo licencia GNU-GPL.

- **Un servidor de bases de datos.**

[Teradata](#)

Sin duda el mejor RDBMS para la implantación de un Data Warehouse.

[Oracle](#)

Muy extendido, con muchas e interesantes extensiones al estandar SQL, y dispone de una familia de excelentes productos de desarrollo, basados en el servidor de bases de datos.

[MySQL](#)

Para aplicaciones de bajo coste.

- Una herramienta cliente que permita **ejecutar SQL**, a ser posible independientemente de cual sea el servidor de bases de datos.

[WinDDI](#)

Cliente de Teradata RDBMS. Para máquinas Windows.

[MySQL Control Center](#)

Cliente de MySQL. Es una herramienta que pueden usar tanto programadores como administradores: proporciona una interfaz cómoda para escribir y ejecutar código SQL, a la vez que permite administrar cualquier número de servidores de bases de datos MySQL. También para máquinas Windows.

[Emacs con módulo SQL](#)

El módulo SQL de Emacs colorea la sintaxis a la vez que uno escribe. Además, se puede ejecutar el código SQL a la vez que se escribe, y ver los resultados. Todo ello dentro del mismo entorno. En la configuración del módulo SQL se especifican los parámetros necesarios para establecer la conexión con el servidor de bases de datos. Este cliente se puede usar en cualquier plataforma donde funcione Emacs, que son prácticamente todas (todos los UNIXes, Windows, VMS, ...). Para mí esto es una característica muy importante porque me permite trabajar en varias plataformas sin necesidad de utilizar herramientas específicas en cada una de ellas.

[DBI Shell](#)

Es un cliente de línea de comando que proporciona conectividad con cualquier base de datos vía DBI (interfaz de conexión con bases de datos que se usa en el mundo Perl). Por estar basado en Perl, este cliente es también multiplataforma. Se puede usar con una interfaz más cómoda utilizándolo como cliente del módulo SQL de Emacs.

- **Lenguajes de programación.**

[Perl](#)

Enrique Castilla Contreras

Tif: 661-247020 · E-mail: ecastillac@yahoo.com · Web: <http://www.geocities.com/ecastillac>

Lo utilizo para todo, y en cualquier plataforma. No he encontrado nada que no se pueda programar con Perl. Desde administración de servidores hasta interfaces gráficas de usuario. La [CPAN](#) es el mayor repositorio de librerías (*modules* en terminología Perl), de uso público.

[Java](#)

Lo utilizo para aplicaciones en las que el cliente pide explícitamente el uso de este lenguaje y otras tecnologías relacionadas con él.

[C/C++](#)

Lo utilizo para desarrollar software embebido en dispositivos y aplicaciones o partes de aplicaciones donde la velocidad es un factor esencial.

- Un **entorno de desarrollo** que cumpla, al menos, con las funciones básicas: escribir código fuente y documentación, buscar cadenas de caracteres entre todos los ficheros de un proyecto, mostrar las diferencias entre dos ficheros, depurar el código, ejecutar makefiles, y si el lenguaje lo requiere, compilar y enlazar. Algunos entornos de desarrollo, además de realizar estas funciones básicas tienen integradas algunas herramientas mencionadas en puntos anteriores, por lo que en caso de utilizarlos, se reduce en número de herramientas necesarias.

[Emacs](#)

Emacs es, además de un potente editor de códigos fuente para desarrolladores, un completo entorno de desarrollo modular. Colorea la sintaxis de cualquier lenguaje (Java, Perl, C/C++, Ada, Python, SQL, HTML, SGML, XML, ...), existe un módulo que proporciona una interfaz de alto nivel con cualquier sistema de control de versiones (en particular CVS), y se pueden realizar multitud de funciones sin salir del entorno: compilar y enlazar, comparar ficheros, buscar cadenas de caracteres en el árbol de directorios de un proyecto, ... Uno instala los módulos que necesita, y cuando necesita nuevas funcionalidades puede instalar nuevos módulos que las realicen. De esta forma, uno se puede configurar su propio entorno de desarrollo. Es multiplataforma, lo que me permite cambiar de una plataforma a otra y seguir utilizando el mismo entorno de desarrollo.

[WebSphere Studio, Studio Site Developer](#)

Completo entorno de desarrollo para Java. Entre otros muchos componentes, tiene integrado un cliente de CVS, lo que permite trabajar con un repositorio de fuentes CVS sin abandonar el propio entorno de desarrollo. Puede trabajar con servidores de aplicaciones WebSphere (IBM) y Tomcat (este último se distribuye bajo licencia GNU).

- Una herramienta para **generar analizadores léxicos**.

[Perl y expresiones regulares](#)

Las expresiones regulares de Perl permiten codificar cualquier analizador léxico solo con este lenguaje, sin tener que recurrir a herramientas externas.

[Flex](#)

Flex es Lex (el generador de analizadores léxicos típico del mundo UNIX), pero con licencia GNU y multiplataforma. Genera código C o C++.

- Una herramienta para **generar analizadores sintácticos**.

[Bison](#)

Bison es Yacc (el generador de analizadores sintácticos típico del mundo UNIX), pero con licencia GNU y multiplataforma. Genera código C o C++.

Enrique Castilla Contreras

Tif: 661-247020 · E-mail: ecastillac@yahoo.com · Web: <http://www.geocities.com/ecastillac>

Yapp

Yapp (Yet Another Perl Parser compiler) es también un Yacc pero desarrollado como una librería de Perl (Parse::Yapp) por Francois Desarmenien. Por supuesto, genera código Perl, se distribuye bajo licencia GNU y funciona en cualquier plataforma donde funcione Perl.

- Una librería de componentes (**widgets**) para desarrollar Interfaces Gráficas de Usuario (GUI).

Perl/Tk

Tk es una excelente librería de componentes (controles, widgets) para desarrollar interfaces gráficas de usuario, creada por el MIT. Tradicionalmente, la única forma de usar Tk era programando en lenguaje Tcl (Tcl/Tk).

Hasta que Nick Ing-Simmons creó Perl/Tk (thanks Nick). El trabajo de Nick consistió en integrar Tk en Perl, es decir, desarrollar una interfaz de Perl hacia Tk.

Con Perl/Tk se pueden desarrollar aplicaciones a base de interfaces gráficas de usuario y orientación a objetos, así como desarrollar nuevos widgets a partir de otros.

Perl/Tk se distribuye como un conjunto de librerías de Perl bajo licencia GNU-GPL, y esta disponible para casi cualquier plataforma, incluido Windows.

- Una herramienta que permita **testear el software** desarrollado.

No hay una solución universal al problema de testear aplicaciones, entre otras razones porque el software de test depende del tipo de la aplicación y de la plataforma. Por ejemplo, los mecanismos de test para una aplicación web no sirven para una aplicación de escritorio, y la misma aplicación de escritorio en plataformas distintas se testea con software diferente.

Lo que si hay son excelentes soluciones parciales. Por ejemplo, existen excelentes productos para testear aplicaciones .NET, o aplicaciones web basadas en Java y WebSphere.

IBM Rational XDE Tester

Automatiza la ejecución de tests de regresión y funcionales para aplicaciones web desarrolladas en Java. Se integra con IBM WebSphere Studio.

Test::Builder, Test::Harness, Test::More, Test::Simple

Son 4 librerías (módulos) de Perl para escribir tests de aplicaciones.

Perl es un lenguaje especialmente adecuado para escribir tests de aplicaciones servidoras (aplicaciones web, aplicaciones de bases de datos, ...), y por supuesto aplicaciones Perl.

- Una herramienta para **generar distribuciones** del software y los manuales.

Wise Installer

Genera distribuciones de software para plataformas Windows.

ExtUtils::MakeMaker

Una librería Perl que genera distribuciones de software. Funciona en cualquier plataforma donde funcione Perl.

- Un sistema de **distribución de software** de forma remota.

CPAN (Comprehensive Perl Archive Network), Cliente de la CPAN

La CPAN es el sistema de distribución remota de software mas sencillo y eficiente que conozco. Un sistema de distribución remota de software necesita: un repositorio de paquetes software, un servidor que publique el material del repositorio y un cliente capaz de buscar, descargar e instalar paquetes.

La CPAN es esto mismo: el repositorio de paquetes es una estructura de carpetas en una máquina o en una base de datos, el servidor es el de FTP y HTTP, y el cliente se llama CPAN::Shell.

Instalando otros repositorios con la misma estructura de la CPAN, y permitiendo el acceso con los mismos protocolos, se puede usar el cliente de la CPAN aunque el repositorio no sea la propia CPAN sino otro similar. CPAN::Shell es un cliente de la CPAN basado en Perl. Entre otras funcionalidades permite buscar, examinar,

Enrique Castilla Contreras

Tlf: 661-247020 · E-mail: ecastillac@yahoo.com · Web: <http://www.geocities.com/ecastillac>

instalar y generar distribuciones de software.

Como muchas librerías Perl, además de un API de programación, dispone de una interfaz de usuario a base de línea de comando. El API permite desarrollar procesos de mirroring y de actualización automática de versiones.

- Una herramienta que permita hacer un **seguimiento de bugs** detectados en el software.
- Una herramienta que permita **publicar y asignar tareas pendientes**.
- Una herramienta que **extraiga información estadística acerca de un proyecto** (tiempo empleado en el desarrollo, número de personas o perfiles de cada tipo que han participado, número y tamaño de los ficheros desarrollados, ...).
- Una herramienta que permita **medir el esfuerzo necesario** en el desarrollo de un proyecto.
- Una herramienta para **planificar proyectos**, asignando personas y recursos.
- Una herramienta para **controlar proyectos**, que permita hacerse una idea del grado de cumplimiento respecto a la planificación.
- Herramientas adicionales: **ofimática, compresores, navegador web, cliente de correo, cliente FTP, cliente Telnet y visualizadores** de los principales tipos de archivos (PDF, Postscript, ...).

Además, para determinados tipos de proyectos puede requerirse el uso de herramientas adicionales. Por ejemplo, en el desarrollo de aplicaciones tipo web:

- Un **servidor de aplicaciones web** o al menos, un **servidor HTTP**.

[IBM WebSphere](#)

Un servidor de aplicaciones web basado en Java. El producto que hace las veces de servidor HTTP es Apache. Se integra perfectamente con el entorno de desarrollo IBM WebSphere Studio

[Tomcat](#)

Un servidor de aplicaciones web basado en Java. Tomcat es una ampliación de Apache. Licencia GNU.

[mod_perl](#)

mod_perl es un módulo (una expansión que añade nuevas funcionalidades) del servidor HTTP Apache. Con mod_perl instalado, un servidor Apache es un servidor de aplicaciones web basadas en Perl.

[Apache](#)

El mejor y más extendido servidor HTTP.

- Una herramienta para **diseñar y ensamblar componentes HTML**.

[Template Toolkit](#)

Es una herramienta de templates, válida para usar templates en cualquier tipo de aplicación, no solo HTML. Licencia GNU

- Una herramienta para **diseñar formularios**.

Enrique Castilla Contreras

Tlf: 661-247020 · E-mail: ecastillac@yahoo.com · Web: <http://www.geocities.com/ecastillac>

Adobe GoLive

Es una excelente herramienta de diseño web, con funciones mucho más amplias que el diseño de formularios HTML, aunque yo la utilizo principalmente para esto.

- Una herramienta para **diseñar y aplicar hojas de estilo**.

TopStyle

Aunque otras herramientas como GoLive también sirven para diseñar hojas de estilo, sin embargo, la herramienta más especializada en esta tarea es Top Style.

- Un **lenguaje de scripting** en el lado del cliente web.

JavaScript

- Una **herramienta XML que automatice la confección de listados y gráficos**.

Uno de mis proyectos (a proyectar), es el desarrollo de una herramienta de este tipo.

- Una **herramienta XML para implementar interfaces de usuario**.

Struts

- Una herramienta para **desarrollar portales sin tener que partir de cero**.

Slash

Un producto para implementar un tipo especializado de portal denominado

- Una herramienta para **tratamiento de imágenes**.

Por ahora, el tratamiento de imágenes no es un campo que llame mi atención.

Por supuesto, para manejar la mayoría de estas herramientas es necesario conocer la tecnología que aplican. Por ejemplo, de nada sirve disponer de la mejor herramienta para modelar sistemas con UML si uno no conoce la tecnología UML.

En la [página de las tecnologías que domino](#) encontrará la lista de tecnologías en las que se basan estas herramientas.

Finalmente, he aquí los criterios que siempre he seguido, para elegir que herramientas incorporar a mi repertorio:

- Potencia.
- Portabilidad.
- Tipo de licencia.
- Amplia aceptación.