

# HOWTO: "Desarrollo de TOL"

---

## Contents

<b>1</b>	<b>Código Fuente e Información General</b>	<b>2</b>
1.1	URLs . . . . .	3
1.2	Listas de Correo . . . . .	3
<b>2</b>	<b>CVS de Tol</b>	<b>3</b>
2.1	Introducción . . . . .	3
2.2	Cómo obtener una cuenta del CVS . . . . .	4
2.3	Uso del CVS . . . . .	4
2.4	Acceso al CVS a través de la cuenta de usuario . . . . .	5
2.5	Acceso al CVS a través del usuario anónimo . . . . .	5
2.6	Descargar un módulo . . . . .	5
2.7	Descargar un directorio contenido en un módulo . . . . .	5
2.8	Actualización del código previamente descargado . . . . .	5
2.9	Entregar los cambios al repositorio . . . . .	6
2.10	Añadir un fichero . . . . .	6
2.11	Borrar un fichero . . . . .	6
2.12	Añadir un directorio . . . . .	7
2.13	Borrar un directorio . . . . .	7
2.14	Almacenar las opción habituales . . . . .	7
<b>3</b>	<b>Software adicional y compilación</b>	<b>7</b>
3.1	Compilación del Código Fuente . . . . .	8
<b>4</b>	<b>Documentación</b>	<b>8</b>
4.1	Documentando el Código Fuente . . . . .	8
4.1.1	Tipos de comentarios de documentación . . . . .	9
4.1.2	Comentar Atributos . . . . .	9
4.1.3	Comentar Métodos y Funciones . . . . .	9
<b>5</b>	<b>Envío de código a TOL-project</b>	<b>9</b>

<b>6</b>	<b>Acerca de éste documento</b>	<b>10</b>
6.1	Dónde encontrarlo . . . . .	10
6.2	Otros Formatos . . . . .	10
6.3	Cambios . . . . .	10

Copyright (c) 2003 Daniel Rus <daniel@uponeits.com> y Jorge Suit Pérez <jsperez@bayesinf.com>

Concedidos los derechos de copia, distribución y/o modificación de éste documento bajo los términos de la "GNU Free Documentation License", Versión 1.2 o posteriores publicadas por la Free Software Foundation.

Para involucrarse en el desarrollo de TOL es necesario saber cómo: encontrar la información y el código de TOL, usar el CVS, y compilar. El objetivo de éste documento ayuda a realizar estas tareas sin perder demasiado tiempo.

Éste documento esta dirigido a nuevos desarrolladores, a desarrolladores ocasionales, y a todo aquel que este considerando contribuir al desarrollo del Lenguaje TOL. El documento resuelve las cuestiones básicas con las que todo desarrollador se encuentra al implicarse en el desarrollo del lenguaje.

## 1 Código Fuente e Información General

TOL es un lenguaje de programación interpretado, funcional, declarativo, orientado al manejo de estructuras temporales, desarrollado desde 1995 para XXXXX. Hasta 2002 en el desarrollo del lenguaje estuvieron implicados varias personas, entre las que destacan, Victor de Buen, Antonio Salmerón, XXXXX... A partir de 2002 el proyecto dió un giro hacia el Software Libre y nos incorporamos a él otros desarrolladores.

En la fecha en la que se escribe este documento el Software esta dividido en varias partes:

- **libTol**: la biblioteca tol
- **tolcon**: un intérprete y entorno interactivo usado para ejecutar ficheros con código tol
- **libTolTcl**: la biblioteca de enlaza de Tol con el lenguaje Tcl
- **tolBase**: la Iinterfaz Gráfica de Usuario de Tol, escrita como una extensión de Tcl.

Hay muchos modos de colaborar con estos proyectos. Se puede escribir nuevo código para el lenguaje, mejorar el que ya hay, escribir documentación, traducir Tol a otros idiomas, reportar errores y sugerir nuevas prestaciones.

Si quieres involucrarte en el desarrollo de código, o estas pensando en ello, continua leyendo.

Si puedes contribuir de alguna otra manera, generando documentación o traduciendo Tol a otros idiomas, por favor, visita éstos otros documentos: XXXXX.

Tol esta bajo la licencia GNU GPL de la Free Software Foundation.

El código fuente de Tol esta estructurado en:

- clases básicas
- núcleo (parsing, tipos básicos y entorno de ejecución)
- operadores y funciones soportadas por los tipos básicos
- algoritmos matemáticos

Tol utiliza algunos algoritmos de la GNU Scientific Library (GSL), y de la implementación que hace ésta de los algoritmos básicos del álgebra lineal (BLAS) por lo que será necesario disponer del paquete GSL en el entorno de desarrollo. Para obtener GNU GSL visita <http://www.gnu.org>.

## 1.1 URLs

Lista de URLs relacionadas con tol.

Página Web principal de tol: <http://www.tol-project.org>

Página Web de los desarrolladores: <http://devel.tol-project.org>

Página Web sobre los informes de errores: <http://bugs.tol-project.org>

Acceso al Código Fuente y Binarios: <ftp://ftp.tol-project.org>

Acceso al árbol cvs de tol: <http://webcvs.tol-project.org>

Página Web de Bayes Decisión SL, la empresa que desarrolla tol: <http://www.bayesinf.com>

## 1.2 Listas de Correo

Los proyectos de tol tienen varias listas de correo.

- ***tol-devel***, donde los desarrolladores de tol hablan sobre las cuestiones relacionadas con el desarrollo de tol y las aplicaciones que le rodean.
- ***tol-tcl-devel***, igual que la anterior pero orientada a la biblioteca TolTcl.
- ***tolbase-devel***, igual que la anterior pero orientada a la Interfaz Gráfica TolBase.
- ***tol-cvs***, es la lista que recoge los cambios que se van aplicando al repositorio de código fuente.
- ***tol-users***, donde los usuarios de tol hablan sobre el uso del lenguaje y la resolución de problemas con tol.
- ***tol-announce***, donde se anuncian las novedades relativas a tol.

Para suscribirse hay que enviar un mensaje a [<nombredelalistadecorreo-request@tol-project.org>](mailto:nombredelalistadecorreo-request@tol-project.org) con el mensaje "*subscribe myid@mydomainname*", donde *myid@mydomainname* es la cuenta de correo electrónico de quien se suscribe. Por ejemplo, para suscribirse a ***tol-devel*** hay que enviar un correo electrónico a [<tol-devel-request@tol-project.org>](mailto:tol-devel-request@tol-project.org).

## 2 CVS de Tol

### 2.1 Introducción

El CVS de Tol es el repositorio de código fuente de todos los proyectos relacionados con Tol. Se puede acceder a su contenido a través de:

1. World Wide Web: [webcvs.tol-project.org](http://webcvs.tol-project.org)
2. la descarga de los tarball (tar.gz y tar.bz2) por FTP
3. la utilidad de línea de comandos "cvs"

#### 4. una interfaz gráfica de cvs

Los métodos 1 y 2 son métodos de solo-lectura de acceso al código fuente y están disponibles públicamente, en cambio, los métodos 3 y 4 permiten también el acceso de lectura y escrita.

El método 1 permite navegar por los directorios de los proyectos y acceder al código fuente. Es recomendable solamente para acceder a determinados ficheros, pero no para trabajar con todo el árbol de código.

Los tarball reúnen el código del cvs en momentos determinados y bajo números de versión de manera que se van liberando a medida que los proyectos van evolucionando.

El repositorio cvs recoge los cambios que los desarrolladores van haciendo al código fuente. Cada vez que un desarrollador realiza un cambio en el cvs incluye un comentario, de manera que el desarrollo del código puede ser seguido más fácilmente. Estos cambios son enviados a la lista de correo ***tol-cvs@tol-project.org***.

El código en el CVS puede ser marcado en cualquier momento de manera que se pueda volver siempre a la posición fijada por la marca establecida. El código en el CVS también puede ser ramificado en diferentes líneas de desarrollo. Las ramificaciones comienzan siempre en las marcas establecidas. La rama marcada como "HEAD" se emplea como rama principal del árbol de código, donde se van incorporando los desarrollos. Cuando llega el momento de liberar nuevas versiones de los proyectos, estas se marcan en el cvs con nombres distintos, lo que permitirá tenerlas organizadas.

El acceso al árbol de código fuente a través del método 3 y 4 se explica con más detalle en la siguiente subsección.

## 2.2 Cómo obtener una cuenta del CVS

Se puede obtener una cuenta de acceso de escritura al CVS de TOL siempre que la persona que la solicite este vinculada con el desarrollo del lenguaje o con alguno de los módulos que lo componen. Para solicitar la cuenta es necesario enviar un email indicándolo al administrador del sistema a ***<sysadmin@tol-project.org>***.

Hay dos métodos de acceso al CVS: **pserver** o **ssh**.

Para obtener acceso a través del primer método es necesario incluir en el email la password del usuario. Ésta puede ser generada mediante el siguiente comando perl:

```
perl -e print\ crypt\('password','xx')\.\\"\\n\"
```

En dicho comando **password** es la clave con la que accederá al CVS, y **xx** son dos caracteres del conjunto (a-zA-Z0-9./). La salida del comando es la password cifrada que debe adjuntar en el email.

Para acceder al CVS a través del método **ssh** es necesario generar un par de claves, una pública y otra privada. Para crear el par de claves puede ejecutar la siguiente utilidad que probablemente tenga instalada en su sistema:

```
ssh-keygen -t dsa
```

A continuación siga las instrucciones que se le muestran. Una vez finalizado tendrá dos nuevos ficheros generados con una clave cada uno. El fichero generado con extensión **.pub** contiene la clave pública y el otro la clave privada. Junto al email que nos envíe no olvide adjuntar el fichero con la clave pública. El fichero con la clave privada lo necesitará en todas aquellas máquinas desde las que quiera acceder al CVS.

## 2.3 Uso del CVS

Probablemente la utilidad **"cvs"** este ya instalada en su sistema. Si no es así, debería consultar <http://www.cvshome.org>, donde podrá obtener la mejor información sobre la misma.

Para aprender a usar el CVS es muy recomendable leer el manual que esta disponible en la página web mencionada. No obstante aqui se recogen los comandos más típicos.

## 2.4 Acceso al CVS a través de la cuenta de usuario

En éste apartado vamos a asumir que ya tiene una cuenta de usuario para acceder al CVS. Supongamos que el nombre de usuario de su cuenta de CVS es `cvsuser`. Para poder acceder al CVS necesitará tener creada la variable de entorno `CVSR00T` y establecida a `:pserver:cvsuser@cvs.tol-project.org:/home/tolcvs"`

Si esta realizando el acceso a través de `ssh` tendrá que cambiar, en la variable `CVSR00T` la palabra `pserver` por la palabra `ext`, así como establecer otra variable de entorno denominada `CVS_RSH` con valor `"ssh"`.

## 2.5 Acceso al CVS a través del usuario anónimo

En caso de que no tenga una cuenta de usuario para acceder al CVS, puede emplear el usuario `"anonymous"`. En ese caso tiene que establecer la variable de entorno `CVSR00T` al siguiente valor `:pserver:anonymous@cvs.tol-project.org:/home/tolcvs"`.

## 2.6 Descargar un módulo

Actualmente el CVS cuenta con los siguientes módulos: `tol`, `tolctl` y `tolbase`.

Para descargar alguno de ellos es necesario usar el comando `checkout` del cvs:

```
cvs -z4 checkout -r HEAD nombredelmódulo
```

La opción `-r` le dice al cvs que rama del módulo quiere descargar. Si no se indica ninguna rama con la opción `-r` se descarga el módulo de la rama principal, `HEAD`.

```
cvs -z4 checkout nombredelmódulo
```

Para descargar la rama `Tol_1_1_10` del módulo `tol` utilizaremos el comando:

```
cvs -z4 checkout -r Tol_1_1_10 tol
```

## 2.7 Descargar un directorio contenido en un módulo

En algunas circunstancias puede que solo le interese trabajar con determinados directorios de un módulo. Para descargar solo determinados directorio de un módulo podemos emplear la opción `-l` del cvs que indica que no descargue recursivamente todos los subdirectorios contenidos en el módulo indicado:

```
cvs -z4 co -l tol
```

```
cvs -z4 co -l tol/bdat
```

El primer comando descarga el contenido del directorio raíz del módulo `tol`, es decir, sus ficheros, sin descargar ninguno de sus directorios hijos. El segundo comando descargará el directorio `bdat` contenido en el módulo `tol` y lo ubicará dentro del anterior directorio.

## 2.8 Actualización del código previamente descargado

El código que ha desargado del CVS puede estar siendo modificado por otros desarrolladores que trabajan con sus copias descargadas en sus sistemas. El código modificado, después de ser probado, es enviado al repositorio. Dispone de otro comando que le permite actualizar la copia de código que descargó y así

permanecer actualizado a los cambios que se realizan en el CVS. Dicho comando es el comando `update` del cvs:

```
cvs -z4 update -P -d tol
```

Si ya se encuentra dentro del directorio quiere actualizar no es necesario especificarlo:

```
cvs -z4 update -P -d
```

La opción `-P` indica que sean eliminados los directorios vacíos, la opción `-d` indica que sean construidos aquellos directorios que se hayan añadido al CVS desde la última descarga.

## 2.9 Entregar los cambios al repositorio

Para enviar los cambios al repositorio y que estos estén disponibles para el resto de usuarios del CVS tenemos que emplear el comando `commit` del cvs:

```
cvs -z4 commit nomarch
```

`nomarch` es el nombre del archivo que estamos entregando al CVS. Podemos entregar una lista de archivos con una sola sentencia `cvs commit` simplemente indicando el camino y el nombre a los mismos:

```
cvs -z4 commit nomarch1 camino2/nomarch2 camino3/nomarch3 ...
```

Una vez ejecutado el comando se nos solicitará una explicación de los cambios que estamos entregando al CVS. Todos los cambios que se realicen en el código deben ir acompañados, en el `commit`, de éste comentario explicativo de los cambios que se realizan, algo escueto, pero lo suficientemente descriptivo. Podemos indicar la opción `-m` a `commit` para incluir directamente el comentario en la línea de comando:

```
cvs -z4 commit -m "set macros *_C_DECLS to use C functions with C++ code" tol_bcommon.h
```

Puedes establecer la variable de entorno `EDITOR` o `CVSEDITOR` a tu editor favorito para que el cvs lo invoque al hacer el `commit` sin la opción `-m`.

## 2.10 Añadir un fichero

Para añadir un fichero `nomarch.cpp` a un directorio `dir`:

```
cd dir
```

```
cvs add nomarch.cpp
```

```
cvs commit -m "explanatory comment" nomarch.cpp
```

Hasta que no hagamos el `cvs commit` el nuevo fichero no pasará a formar parte del directorio `dir` en el CVS.

## 2.11 Borrar un fichero

Para borrar un fichero `nomarch.cpp` de un directorio `dir`:

```
cd dir
```

```
rm nomarch.cpp
```

```
cvs remove nomarch.cpp
```

```
cvs commit -m "explanatory comment"
```

### 2.12 Añadir un directorio

Si necesitamos añadir código al proyecto y tenemos que hacerlo en un nuevo directorio, hay que seguir los siguientes pasos. Suponiendo que el directorio que queremos crear se llama `nomdir` y que vamos a añadir el fichero `nomarch.cpp` al directorio `nomdir`, todo ello bajo el directorio del proyecto `tol`:

```
cd tol
mkdir nomdir
ahora escribimos el nuevo fichero (nomarch.cpp)
cvs add nomdir
cvs add nomarch.cpp
cvs commit -m "explanatory comment"
```

El fichero y el directorio no se añadirá al repositorio hasta que no ejecute el `commit`. Para poder añadir un directorio al repositorio es necesario que éste contenga ficheros.

### 2.13 Borrar un directorio

Para borrar un directorio es necesario primero borrar todos el contenido de éste. Por ejemplo, para borrar el directorio `nomdir`, que contiene el fichero `nomarch.cpp`:

```
cd tol/nomdir
ahora borramos el fichero nomarch.cpp siguiendo el método explicado en 2.11 (), y continuamos borrando el directorio:
cd ..
cvs -P update
```

El último comando eliminará el directorio `nomdir` localmente.

### 2.14 Almacenar las opción habituales

El sistema CVS ofrece la posibilidad de almacenar las opciones más usadas en un fichero de recurso, `~/cvsrc`. Recomendamos emplear los siguientes valores en dicho fichero:

```
cvs -z4
diff -u
update -dP
checkout -P
```

La primera línea indica que la opción `-z4` se aplicará cada vez que aparezca el comando `cvs` en la línea de comandos. La opción `-z4` aplica el nivel 4 de compresión. La segunda línea indica que la salida del comando `cvs diff` sea conforme al formato `unidiff`, más sencilla de interpretar que la salida normal. Las opciones `-dP` dadas a `update` indican que se obtengan los nuevos directorios del árbol y que se borren aquellos que hayan quedado vacíos. La opción `-P` de `checkout` indica que no se descarguen los directorios vacíos.

## 3 Software adicional y compilación

Existen actualmente tres proyectos englobados bajo el nombre TOL-project:

La biblioteca libTol, que contiene el lenguaje, la biblioteca libTolTcl que sirve de enlace entre Tol y el lenguaje Tcl, y la extensión de Tcl denominada tolBase (interfaz gráfica de usuario para el manejo del lenguaje Tol).

Cada una de estas piezas se construyen por separado y tienen unos requerimientos distintos de compilación.

La biblioteca libTol requiere la presencia de otras bibliotecas para su compilación. Éstas son la GSL (GNU Scientific Library) y una implementación de ODBC, que en el caso de estar compilando en Unix o Linux puede ser la implementación UnixODBC o la que realiza Open Link llamada iODBC.

Por lo que respecta a la biblioteca libTolTcl de enlace entre Tol y Tcl, es necesario tener instalado las bibliotecas de código de Tcl/Tk del lenguaje así como los ficheros include que lo componen.

El programa TolBase requiere para su correcto funcionamiento de Tcl/tk 8.4, Tktable2.7, TclODBC, BLT2.4 y Tol.

### 3.1 Compilación del Código Fuente

Tanto si se accede al código a través del CVS como a través de los tarball, cada uno de los proyectos dispone de la información para construir el software en sistemas Unix y gnu/Linux empleando las herramientas comúnmente conocidas como GNU Autotools.

Cada proyecto cuenta con un guión bootstrap que toma el fichero configure.in para crear un shell script configure. El shell script configure creará los ficheros Makefile necesarios para construir cada proyecto. En cada proyecto se incluye un fichero INSTALL con instrucciones que explican cuáles son las opciones que podemos emplear para invocar configure adecuadamente.

Bajo el directorio win se incluyen los ficheros que necesita el entorno de compilación Visual C de Microsoft. Aunque bajo dicho sistema operativo, previa instalación de la implementación de Shell Cygwin (<http://www.cygwin.com>), se pueden utilizar las GNU Autotools.

## 4 Documentación

Es necesario documentar el trabajo de desarrollo. En dicho trabajo existen dos actividades de documentación: la documentación del código, y la documentación general.

La documentación del código de la biblioteca de tol, libTol, se analiza por Doxygen, una herramienta de generación automática de documentación, y se generan las páginas HTML de la API de Tol, disponibles en <http://devel.tol-project.org/tolapi/index.html>. Con apenas aplicar ciertas normas de estilo en los comentarios del código fuente, Doxygen generará una atractiva documentación muy útil para navegar por el código.

La documentación general del trabajo de desarrollo es aquella relacionada con la explicación, en un formato humanizado, del trabajo realizado. Explicar para que sirven las nuevas funciones creadas, o algún nuevo tipo de utilidad que se haya añadido al lenguaje, o explicar los cambios que se han realizado y como hay que emplear las nuevas características. Esta documentación esta disponible en <http://devel.tol-project.org/doc/index.html>, en los formatos: html, ps, dvi y pdf. Para generar la nueva documentación se pueden usar editores como LyX (<http://www.lyx.org>).

### 4.1 Documentando el Código Fuente

Las reglas básicas que se emplean para documentar el código empleando Doxygen se explican en el documento "Source Code Organization" (<http://devel.tol-project.org/doc/srcCodeOrg.html>), pero aqui hacemos una breve explicación de las mismas.

Siempre que sea n



#### 4.1.1 Tipos de comentarios de documentación

Para que Doxygen genere documentación a partir de los comentarios en una línea de C++ o los comentarios en múltiples líneas comunes a C y C++, es necesario acompañarlos del signo de admiración. Los comentarios en una línea tienen el siguiente aspecto:

```
//! comentario en una línea para que sea captado por Doxygen
```

Los comentarios en múltiples líneas tienen éste otro aspecto:

```
/*! Esto es un comentario en  
 * múltiples líneas para que lo  
 * capte Doxygen  
*/
```

#### 4.1.2 Comentar Atributos

Los comentarios de los atributos de una clase aparecen comentados así:

```
Any* buffer_; //!< Storage pointer  
BInt maxSize_; //!< Maximum number of items that has been stored
```

#### 4.1.3 Comentar Métodos y Funciones

```
//! Add a new obj to buffer_  
/*! Verify if \a obj exist in buffer_ using a function of type  
 * BorderCriterium and calling Find method. Then insert \a obj  
 * in buffer_ with Add method.  
*/  
BBool AddUnique(const Any@ obj, BOrderCriterium order)  
{  
    BBool unique = (Find(obj, order)<0);  
    if(unique) { Add(obj); }  
    return (unique);  
}
```

El primero comentario, el de una sola línea, aparecerá como comentario breve en la documentación junto a la declaración del método. El comentario de múltiples líneas aparecerá en la descripción detallada del método. Doxygen tiene elementos que permiten resaltar los parámetros de los métodos y funciones, y los tipos devueltos. También permite crear enlaces html a otras funciones mencionadas en el comentario.

## 5 Envío de código a TOL-project

Si esta participando en el desarrollo de alguno de los proyectos de TOL pero no tiene una cuenta en el CVS (algo frecuente si las colaboraciones son esporádicas o se realizan solo en determinadas partes del código) el código se envía por correo electrónico a alguno de los mantenedores de los proyectos con los que tiene contacto. Las direcciones de correo electrónico de estas personas se encuentran a menudo en los ficheros fuente de los proyectos o bien en el fichero AUTHORS de cada proyecto.

## 6 Acerca de éste documento

### 6.1 Dónde encontrarlo

La versión actualizada de éste documento se puede encontrar en <http://devel.tol-project.org/doc/comoDesarrolloDeTol/>

### 6.2 Otros Formatos

Se puede encontrar una copia de éste documento en formato html, ps, dvi y pdf en el URL anterior.

### 6.3 Cambios

Cambios desde la versión <14 de Noviembre de 2003> a <fijar\_fecha>

- Cambio 1
- Cambio 2
- ...