

Control de versiones CVS

AUGCyL - <http://www.augcyl.org>



QUÉ ES?

- Es un software que permite un control sobre distintas versiones de código fuente o documentos.
- Permite la colaboración entre distintas personas.
- Cuida que dos usuarios no modifiquen la misma parte del código
- Junta automáticamente los ficheros de varios desarrolladores
- Permite volver a cualquier punto del desarrollo

PROCESO DE DESARROLLO

1. Bajar del servidor cvs los ficheros a un directorio local (cvs checkout)
2. Modificar los archivos, según lo necesario
3. Mirar a ver si alguien ha cambiado la versión del servidor en el tiempo en que nosotros modificábamos la copia de trabajo. (cvs update)
4. Si el cvs ha actualizado nuestra copia local correctamente, nos lo indica. En caso contrario, produce un error de conflicto. Resolvemos el conflicto y volvemos al paso 3.
5. Subimos nuestros ficheros de nuevo al servidor (cvs

CONCEPTOS

-Copia de trabajo

-Revisión

-Informe de cambios

-Conflicto

-Repositorio

-Módulo

COMANDOS BÁSICOS

-Es necesario indicarle al comando cvs el origen de los
ficheros

- Con el parametro -d RUTA
- estableciendo la variable CVSROOT

-Formato:

- :TIPO:usuario@host:DIRECTORIO

Ej. 1º: cvs -d

:pserver:anonymous@anoncvs.gnome.org:/cvs/gnome login

2º: export

CVSROOT=':pserver:anonymous@anoncvs.gnome.org:/cvs/gnome
cvs login

MAS COMANDOS

-Para bajar las fuentes:

- ☐ cvs checkout módulo

-Para ver si ha cambiado algo en el servidor desde la última copia

- ☐ cvs update

-Para volver a subir el código cambiado

- ☐ cvs commit -m "Cambios realizados en la version"

OTROS COMANDOS INTERESANTES

- Con cvs diff podemos observar los cambios realizados
 - cvs diff (Compara la version actual con la del servidor)
 - cvs diff hello.c (Compara solamente el fichero hello.c con el del servidor)
 - cvs diff -C (Otro formato de salida)
 - cvs diff -r 1.3 -r 1.4 (Diferencias entre la revision 1.3 y 1.4)
- Con cvs import podemos subir ficheros que se han empezado a desarrollar fuera de un cvs
- Con cvs add podemos añadir nuevos ficheros

RECOMENDACIONES:

- Actualizar siempre nuestra copia de trabajo antes de empezar las modificaciones
- Probar bien que los cambios realizados son medianamente correctos.
- Subir los cambios al finalizar la sesión.
- Indicar con el flag -m '...' el informe de cambios cuando subimos al servidor los ficheros (commit)
- No pensar que el cvs hace todo el trabajo. Planificar bien la estructura del programa y cometido de cada desarrollador.