



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## Entorno de Trabajo Concurrente (II) (11719 lectures)

Per Tomeu Borrás, [tborras](http://www.iglu.es) (<http://www.iglu.es>)

Creado el 02/11/2006 13:19 modificado el 02/11/2006 13:57



Hoy día se cuenta con buenas herramientas de sincronización y coordinación que permiten trabajar a mucha gente en el mismo proyecto. Pero siempre bajo la perspectiva de **una persona sola frente a su ordenador**.

Intentaremos cambiar esa perspectiva y probar un sistema que fomente el desarrollo en equipo. Usando, como no, **100% Software Libre**

[Ampliar](#)<sup>(1)</sup>

## Objetivos

Vamos a experimentar un poco con sistemas de desarrollo diferentes donde se de mucha más importancia al grupo de desarrollo

El "experimento" consiste en crear un entorno de trabajo adecuado para el trabajo colaborativo y en intentar trabajar cambiando la filosofía de trabajo.

Para ello estableceremos un centro de trabajo similar a un centro de control donde todos los usuarios, a parte de acceder a sus respectivos ordenadores tengan acceso a una pantalla común a todos ellos donde puedan testear, probar y explicar puntos concretos a tratar. Y todo esto lo realizaremos concurrentemente y facilitando al máximo la interacción de tal modo que no haya que levantarse, pedir turno, o establecer algún mecanismo de control.

## Requisitos Hardware

- Un proyector de video. Preferentemente uno con lámparas de larga duración (Los proyectores caseros son los mejores candidatos) y con, como mínimo, 2000 lumens para tener una pantalla compartida fácil de ver en entornos iluminados. También es aconsejable que soporte una resolución grande (1024x768 o 1280x1024).
- Un ordenador de proyección con Debian Unstable para soportar el software necesario.
- Varios ordenadores clientes (Linux) con entorno gráfico.
- Todos los equipos conectados por una LAN.

## Requisitos Software

- Gobby o cualquier editor concurrente. A pesar de tener un escritorio común es interesante que el equipo haga los progresos en común. Además con uno de éstos editores es más sencillo una división rápida de tareas.
- Multicursor-WM. Este software permite la interacción de múltiples usuarios en un mismo escritorio. Su funcionamiento se basa en hacer cambios de contexto del ratón y del teclado para cada uno de los usuarios. De esta forma se puede usar con software no diseñado para ser concurrente. Multicursor es básicamente un cliente IceWM donde se ha permitido la gestión de colas para múltiples cursores y se ha hecho una representación gráfica de éstos. Para ponerlo en marcha se usa el comando `icewm` y como parámetro se puede pasar el tamaño del cursor

```
icewm --cursor=[small | medium | large]
```



- **x2x-mp**. Este Software, incluido en el paquete de multicursor, permite que la pantalla proyectada y común sea compartida por todos los usuarios mediante un desplazamiento a la derecha o izquierda de la pantalla de cada uno de sus ordenadores. Su sintaxis es muy similar a la del programa x2x (en el que esta basado) con la salvedad de que incluye el parámetro **ncursor** para indicar el puntero que se utilizara. Multicursor Soporta hasta 8 punteros simultaneos.

```
ssh -X proyector@proyector x2x-mp -to :0 -[east | west] -ncursor [1..8]
```

## Selección del personal

- Es importante que los equipos de desarrollo sean bien avenidos y formados por grupos reducidos (2, 3 o 4 personas).
- El equipo puede, o no, ser liderado por un coordinador. Pero lo más lógico es que, con equipos multidisciplinares, asuma el mando la persona más capacitada para cada una de las tareas que surjan.



## Metodología de Trabajo

- Las sesiones deben ser cortas (no más de 3 horas) ya que aunque no lo parezca, cansa mucho.
- No se precisa de una gran estructuración de tareas, ya que éstas van surgiendo a lo largo de la sesión. Pero siempre es conveniente tener fijados los objetivos de la sesión y evaluar al final de la misma si éstos objetivos se cumplen o no.
- Al ser trabajo **puramente** en equipo todos los miembros deben estar centrados en el trabajo. No se deben producir interrupciones por parte de ninguno de los miembros. Es mejor apartar el teléfono y dar aviso de que no se moleste. También es impensable el uso de elementos que impidan la comunicación entre miembros (como pudieran ser unos auriculares de música).

## Ventajas

- Nivelación del equipo que adquiere una base común de conocimiento a medida que va trabajando con este método.
- No hay tiempos muertos ya que el equipo tiene objetivos comunes y un ritmo de trabajo auto-establecido.
- Mayor calidad ya que el código resultante debe ser validado y aprobado por todos los miembros.
- Permite valorar de forma eficaz al personal y su integración dentro del equipo de desarrollo (útil para selección de personal).
- Permite mejorar la planificación de trabajos. Las fechas y ajustes son más fiables. Sobre todo en equipos mejor entrenados donde existe una constancia de desarrollo.
- Software sin dependencias unipersonales. Tras una sesión de trabajo todos los miembros conocen el código modificado y pueden seguir trabajando con él de forma individual.
- Estandarización del código. Uniformidad de estilo. Se acaban adquiriendo formas comunes por mimetización.
- Ausencia de barrera psicológica al empezar el trabajo. Ya que la incorporación de nuevos miembros es directa y revisada por los miembros más expertos.
- Mayor involucración por parte de los miembros. (Por lo menos en las pruebas realizadas así ha sido, la programación cobra un renovado interés basado en la interacción).

## Inconvenientes

- Mayor inversión en recursos. Debe disponerse de un equipo mayor para programar.
- Fatiga si se utiliza durante mucho tiempo. Es un sistema que, aunque no lo parezca, acaba cansando mucho al personal ya que exige mayor concentración y un ritmo de trabajo constante. Es preferible realizar sesiones cortas de 3 horas como máximo.
- Posibilidad de disputas. En equipos poco avenidos o recientes puede haber disputas en la toma de decisiones. En dicho caso es mejor introducir un personaje con capacidad de liderazgo que guíe al grupo. En equipos más entrenados las disputas disminuyen.



## Conclusiones

En las primeras pruebas ha demostrado ser un sistema eficaz en rendimiento y agradable al equipo de desarrollo

Se notan carencias en algunas de las herramientas utilizadas (principalmente en el editor concurrente que no permite hacer que cada individuo pueda guardar el documento sino que tiene que destinarse al creador de la sesión para tal efecto

A pesar de todo es un sistema que debería ser utilizado, con relativa frecuencia, en todo equipo de programación.

## Enlaces de interés

[Multicursor WM -- Escritorio Concurrente](#)<sup>(3)</sup>

[Fotos adicionales](#)<sup>(4)</sup>

---

### Lista de enlaces de este artículo:

1. <http://bulma.net/~tborras/1.JPG>
  2. <http://bulma.net/~tborras/2.JPG>
  3. <http://multicursor-wm.sourceforge.net>
  4. <http://bulma.net/~tborras/fproyector/>
- 

E-mail del autor: tborras \_ARROBA\_ conetxia.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2351>