

Los mejores trucos salidos de los teclados de los expertos

# LISTAS Y JUEGOS



En el día a día del usuario y desarrollador de Linux, nos topamos con problemas de toda suerte e índole, que se resuelven con o mucha, poca o ninguna ayuda de la comunidad, con Google, Dogpile, o Usenet, pero siempre con dosis de imaginación e intuición. Esta sección reúne los mejores soluciones para el beneficio de todos. Soluciones que no se encuentran googleando o que se encuentran, pero o que son insatisfactorias, incompletas o directamente mal planteadas. **POR PAUL C. BROWN**

**É**ste mes, nuestro director, Paul C. Brown nos cuenta como crear una lista sin repeticiones a partir de una columna de Calc, la hoja de cálculo de OpenOffice, y como correr esos imprescindibles programas DOS (como "Monkey Island") sin tener que recurrir a un emulador de grandes proporciones tipo gemu o bochs.

He de explicar que mi distro es de esas eclécticas (léase: un horrible híbrido). Empezó su vida como una Mandrake 9.2 (no os riáis los de Debian, que os oigo). Ahora mismo es una especie de monstruo de Frankenstein que comprende paquetes de MDK 9.2, 10.0, 10.1, 10.2,

PLF y cosas compiladas por mi mismo con sus respectivas dependencias y ajustes. Más que evolucionar, se ha desparramado y cada nueva "mejora" ha supuesto romper una docena de cosas que ya funcionaban satisfactoriamente.

Lo último ha sido el sonido.

## Pasando Lista

¿Qué es lo que suele ocurrir? Pues que lo que empieza su vida como un "sucio y rápido hack", después no hay manera de librarse de él. Algo así como lo que le paso a MS-DOS, que empezó su vida como QDOS, que no irán las siglas de Quick Disk Operating System ni nada por

el estilo, sino de Quick and *Dirty* Operating System, y fíjate tú: más de 15 años con un parche.

En este sentido, tomamos decisiones todos los días que son provisionales pero se quedan para siempre. Tomemos el caso de las hojas de cálculo: con demasiada frecuencia hacen de sustitutos de bases de datos.

Imaginemos que administras un pequeña empresa con tres programadores y quieres seguirle la pista a las diferentes tareas y las horas que les dedican a cada una cada uno de los desarrolladores. ¿Vas a crear una base de datos para tan poca cosa? Las respuesta a eso



es: Pues sí, por que, para cuando te quieres dar cuenta, los tres desarrolladores se han convertido en diez o veinte y el número de proyectos se ha multiplicado por cinco.

Enhorabuena por la expansión. Pero si el documento que utilizaste para controlar al personal empezó siendo un hoja de cálculo, probablemente siga siendo una hoja de cálculo.

A lo hecho, pecho, como suele decirse. Es final de mes y ahora tienes que sacar una lista de todas las horas trabajadas para saber lo que se les debe a cada uno. Tienes una hoja como el que se ve en la Figura 1. ¿Qué vas a hacer? ¿Contar a mano a cada programador a mano e ir apuntando en un post-it los datos? Aparte de lame, esa solución es ineficiente, ya que, si algo nos enseñan las máquinas es que, si bien programadas y configuradas, los ordenadores cometen bastantes menos errores que los humanos ¿y si te olvidas de incluir ese nuevo programador finlandés? Todo tu presupuesto al garete. Así que, lo suyo, es dejar este tipo de trabajo a la hoja de cálculo.

Lo primero que necesitamos es poder reproducir la lista de programadores sin repeticiones. Esto, que se hace muy fácil con un *GROUP BY* con una consulta para bases de datos SQL (¿ves? Te lo dije), no es tan sencillo de hacer con una hoja de cálculo. De hecho, no hay ninguna función que nos facilite esta tarea en Calc... pero tenemos Basic. Sí, señoras y señores, es relativamente sencillo con este lenguaje, similar al VisualBasic de MSOffice, escribir una función para Calc utilizando lo que OpenOffice.org conoce como "macro".

Lo primero es, con nuestra hoja de cálculo abierto, escoger *Tools > Macros > Macro* de la barra de menús. En la ventana de diálogo, en el cuadro de texto del ángulo superior izquierdo, escribimos el

	A	B	C
Programador	Proyecto		Horas
2	Manuel	Reconstruir Windows	300
3	Ferns	Reconstruir Programa Solicitud JS	12
4	Diego	Reconstruir Ethernet	600
5	Diego	Crear API Base Datos	10
6	Federico	Organizar Drivers	13
7	Manuel	Crear Interfaz del VBI	20
8	Manuel	Crear API Base Datos	40
9	Luis	Organizar Drivers	7
10	Manuel	Organizar Drivers	10
11	Ferns	Reconstruir Ethernet	9
12			0

Figura 1: La lista de programadores, tareas y horas.

nombre de nuestro macro (ver Figura 2). Yo lo he llamado *newname*, por que devuelve un nuevo nombre y sólo un nuevo nombre cuando lo encuentra en una lista sobre el que itera.

Pulsamos en el *New* para habilitar el macro en la lista. Después escogemos el nuevo macro de la lista y pulsamos en *Edit*. Abriremos una IDE bastante chula (ver Figura 3) que nos permite escribir scripts en BASIC. Nuestro macro es en realidad muy corto (véase Listado 1). BÁSICamente, lo que hace es iterar a través de un matriz de entrada (*search\_list*), comparándola con un matriz destino (*in\_target\_list*) y devuelve el primer nombre del matriz de entrada que no se encuentra en el matriz de destino. Pero ¿de dónde salen esas matrices? Esos matrices son un grupo de celdas de la hoja de cálculo. Si la lista de la que queremos sacar los nombres se halla ocupando las celdas *A1* a la *A10*, el BASIC de OOo interpreta eso como una matriz bidimensional donde, en nuestro caso, *search\_list(A,1)* es el primer elemento, *search\_list(A,2)* es el segundo, y así sucesivamente.

La matriz de destino es la columna donde ya se han insertado nombres. Por tanto, si vamos a poner los nombres de nuestros empleados en la columna *E*, tendremos que comprobar que el nombre buscado no esté en las celdas de la columna *E* por encima de la celda donde se encuentra la función, ya que esas celdas ya contendrán otros nombres de la lista que hemos encontrado.

¿Confuso? Yo también. Veámoslo con un ejemplo. Guardemos nuestra función y salgamos del editor pulsando en el icono del disco y cerrando la ventana del editor. Ahora, en la celda *E2* escribamos:

```
=NEWNAME($A$2:$A$50;$E$1:$E1)
```

(Me imagino que ya sabréis que hace el dólar ¿no? Evita que la dirección que le sucede cambie cuando copiamos la fórmula a otra celda, zoquete). Veremos como aparece el primer nombre de la lista contenida en *A2..A50* en la celda. Esto, por supuesto, se podría haber conseguido simplemente con

```
=A1
```

Pero ahora copiemos el contenido de *E2* al portapapeles, escojamos a continuación las celdas *E3..E50*. ¡Ooooooh! Nos aparece la lista sin repeticiones... y sin omisiones (ver Figura 4).

Ahora es pan comido sumar las hora de cada programador individual para calcular el misérrimo salario que les vamos a pagar con una formula del tipo

```
{=SUM((($A$2:$A$50=$E2)*$C$2:$C$50)}
```

(suponiendo que las horas se hallan en la columna *Cx*). Esto es una fórmula de matriz (ver las llaves que la encierran). Es decir, contiene como argumentos elementos que han de tomarse como

### Listado 1: Macro "newname"

```
01 function
   newname(search_list,in_target_list)
02
03 dim i,j as integer
04 dim found as boolean
05
06 found=True
07 newname=""
08
09 for i=lbound(search_list,1) to
   ubound(search_list,1)
10   found=False
11   for
     j=lbound(in_target_list,1) to
     ubound(in_target_list,1)
12     if
       search_list(i,1)=in_target_list(j,1) then found=True
13   next
14   if found=False and
     search_list(i,1)<>"" then
15     newname=search_list(i,1)
16   exit for
17   endif
18 next
```

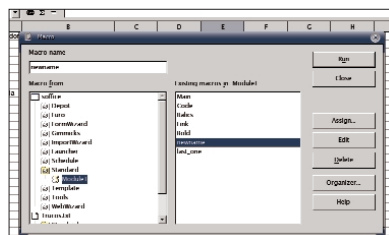


Figura 2: Ventana de diálogo de creación de macros.

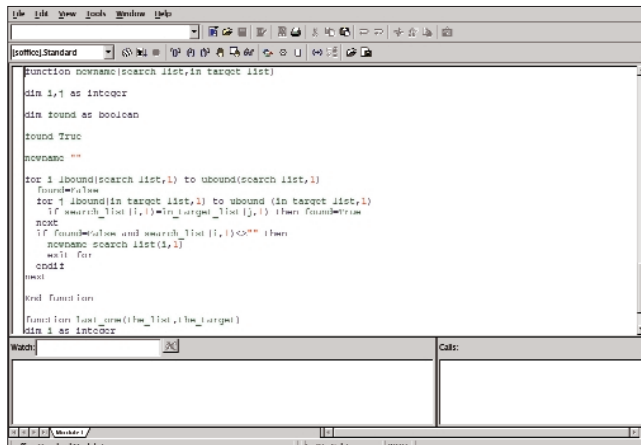


Figura 3: La IDE de desarrollo de OpenOffice.org.

matriz. Para editar uno de estos monstruos y hacer que se comporte como una fórmula de matriz, hemos de pulsar [May] + [Ctrl] + [Enter] al terminar la edición. En este caso, la fórmula busca en el rango comprendido entre las celdas A2...A50 valores que sean iguales a E2. Si encuentra una igualdad, se devuelve 1, si no 0. El resultado se multiplica por los valores correspondientes sacados de las celdas C2...C50, teniendo como efecto final, que sólo se sumen los valores

correspondientes al programador cuyo nombre aparece en E2. Una vez escrita la primera fórmula para el primer programador, podemos copiar y pegar la fórmula de la manera habitual para el resto. El resultado es que se suman las horas por cada uno de los programadores (ver Figura 4). ¡Ja! Además es dinámico: Si añadimos

nuevos programadores a la lista o asignamos nuevas tareas a programadores existentes, la lista se automodifica como tiene que ser.

	A	E	C	D	I	J
1	Programadores	Programas	Horas			
2	Manuel	Programas Monitores	350	Manuel	3h	
3	Manuel	Programas Monitores	50	Manuel	2h	
4	Manuel	Programas Monitores	500	Manuel	3h	
5	Manuel	Programas Monitores	10	Manuel	3h	
6	Manuel	Programas Monitores	20	Manuel	3h	
7	Manuel	Programas Monitores	20	Manuel	3h	
8	Manuel	Programas Monitores	40	Manuel	3h	
9	Manuel	Programas Monitores	20	Manuel	3h	
10	Manuel	Programas Monitores	50	Manuel	3h	
11	Manuel	Programas Monitores	50	Manuel	3h	
12	Manuel	Programas Monitores	50	Manuel	3h	
13	Manuel	Programas Monitores	50	Manuel	3h	
14	Manuel	Programas Monitores	50	Manuel	3h	
15	Manuel	Programas Monitores	50	Manuel	3h	

Figura 4: A la izquierda, la lista original de programadores, tareas y horas. A la derecha la lista generada de programadores y sus horas sumadas.

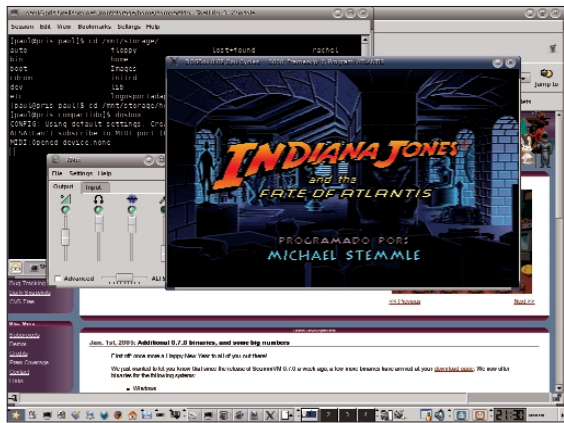


Figura 5: Dosbox ejecutando Indiana Jones and The Fate of Atlantis.

### Guybrush Rulez Threepwood

Ya veíamos en el artículo dedicado a Qemu en el número 5 de Linux Magazine, lo importante que se está haciendo el poder correr múltiples sistemas operativos bajo uno "padre". En ese artículo se citaba el ejemplo de un programa de la empresa, importantísimo, que nadie puede (ni tiene tiempo para) reescribir y que, vaya por dios, sólo funciona bajo DOS.

Todos sabemos que programa es ese: ¡The Curse of Monkey Island, por supuesto! Pero, seamos realistas, señores, correr todo un señor emulador como Qemu, Bochs o VMWare para seguir las peripecias de Guybrush es, como diríamos en inglés, un poco "overkill", es decir, exagerado. Incluso dosemu, que ni siquiera se comporta muy bien con estos juegos, y que una vez conseguí configurar y hacer correr, aunque me sale urticaria cada vez que pienso que lo voy a tener que volver a utilizar, vale la pena.

Para estos casos de tan imperiosa necesidad, de hecho, tenemos dos herramientas a medida: Dosbox [3] y Scummvm [4]. El primero no es, ni pretende ser, a diferencia de dosemu y Wine, un verdadero emulador de otro sistema operativo, sino que incluye mejoras y "transportes" a medios (emulando, por ejemplo, antiguas tar-

jetas de sonido a través del driver de sonido) al sistema Linux que permiten correr con un mínimo de problemas... ¡JUEGOS!

No hay que crear discos virtuales ni zarandajas en Dosbox. Basta "montar" un directorio C: con cualquiera de nuestros directorios existentes. Por ejemplo: si tenemos un montón de juegos (comprados legalmente, claro) en el directorio /home/mi\_dir/dosgames/, bastaría con hacer

```
mount c /home/mi_dir/dosgames/
```

en el shell de Dosbox para disponer de una flamante unidad C: donde hacer nuestros pinitos.

Para cambiarnos a la unidad C: (Dosbox arranca en un directorio virtual, Z:) haríamos:

```
c:
```

Si tienes problemas para conseguir los dos puntos (:), prueba a mantener pulsado la tecla Alt y pulsando 5 y 8 en el teclado numérico. Cuando sueltes Alt, los dos puntos deberían aparecer.

Ya podemos correr cualquier juego que se halle en los directorios y subdirectorios de C: (ver Figura 5).

Otra alternativa para estos programas es ScummVM. Pero ya hablamos de ScummVM en otro artículo en éste número (ver la sección de Linux User) y, por tanto, no me explayaré.

Pero, vamos, que no todo va a ser programar interfaces para bases de datos, digo yo. Otro día os explico como hago mi Cous-cous de gambas y pescado. ■