



SECCIONES:

- ◆ **PASCAL:**
OOO, un nuevo enfoque
- ◆ **OS/2:**
Programación Visual con REXX
- ◆ **UNIX:**
Mecanismos IPC (y II)
- ◆ **FORUM C:**
Applications Frameworks

LOS SPRITES

REPRODUCCIÓN

DE **VIDEO DIGITAL**

LA **TABLA**

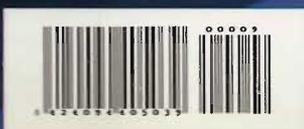
DE **PARTICIONES**

NOVEDADES

EN BORLAND **C++ 4.5**

¿QUÉ ES **PLUG & PLAY?**

ANAYA
MULTIMEDIA

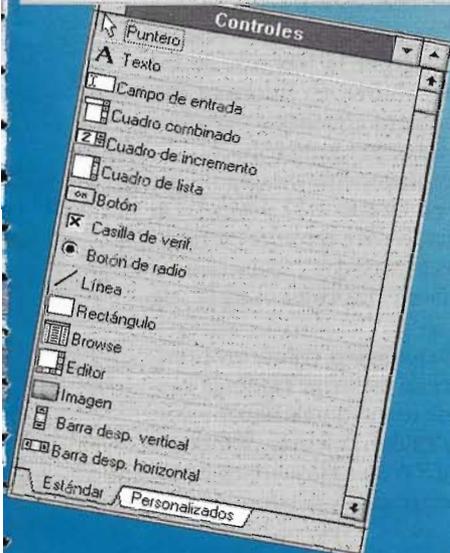
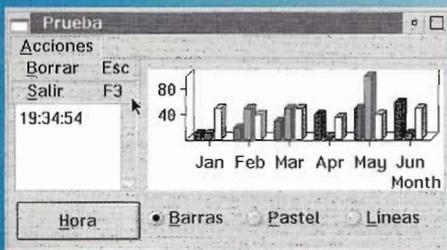
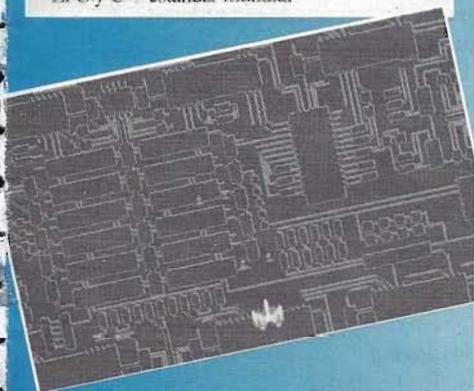
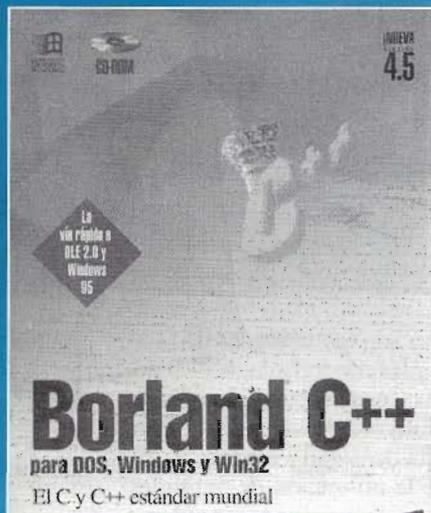


**Disco en el interior con:
Todos los fuentes y dos
fantásticos programas.**

EN ESTE NUMERO ...

Julio-Agosto 1995

Número 9



- 1** Carta del director
- 3** En este número ...
- 4** Novedades

A FONDO

- 9** Analizamos el compilador Borland C++ 4.5

INVESTIGACIÓN

- 15** ¿Qué es Plug&Play?
- 22** Programación en modo protegido sobre MS-DOS (y II)
- 29** La tabla de Particiones

FORUM C

- 38** C++ & Applications Frameworks para Windows
- 46** Roles y Objetos en C++ (y II)
- 53** Programación Gráfica. Ventanas de visión - Clipping
- 60** Programación de drivers para Windows (y II)

CLÁSICOS DE LA PROGRAMACIÓN

- 66** Introducción a la programación visual con REXX en OS/2

BASES DE DATOS

- 72** La sensatez de la integridad

PASCAL

- 76** OOP, un nuevo enfoque. Las herramientas gráficas de Borland
- 81** Reproducción de vídeo digital desde Borland Pascal

PROGRÁMATE TU OCIO

- 89** Los Sprites

UNIX

- 97** Mecanismos IPC del UNIX SYSTEM V (y II)
- 105** Preguntas y respuestas
- 109** Trucos del lector
- 112** Este mes nuestro disco incluye ...

Introducción a la programación visual con REXX en OS/2 (I)

Francesc Roses Albiol

*Nacido en Barcelona en 1958.
Licenciado en Filología Catalana por la Universidad de Barcelona.*

Desde 1985 desarrolla en el Centre d'Informàtica de la Universitat de Barcelona (CIUB) la coordinación informática de diversos proyectos de lexicografía computacional. Profesor del Máster de Lingüística Aplicada de la Universitat de Barcelona. Desde 1994 es el responsable de la sección de Microinformàtica del CIUB.

Escribir programas para OS/2 en C utilizando directamente las APIs del sistema es una tarea formidable, además de ser tediosa. Lo que es demasiado costoso no se utiliza, o sólo se utiliza cuando la necesidad es muy fuerte. Es la razón por la cual en general sólo se acomete un proyecto de programación en un entorno gráfico cuando se dispone de una cantidad considerable de recursos humanos y de tiempo. Nos encontramos así con la paradoja de que un sistema gráfico como OS/2, que es mucho más sencillo y ergonómico que todos sus predecesores, encuentra menos programadores de utilidades sencillas que un sistema como DOS, en el que casi cualquiera se anima a escribir un pequeño programa batch para automatizar sus tareas.

SE debe a que no contamos con las herramientas adecuadas. Programar cualquier cosa en C es muy laborioso (aparte de que C es un lenguaje de programación espantoso y pasado de moda); pasar a C++, que está un poco mejor, requiere un reaprendizaje extenso, y manejar un segundo conjunto de APIs, similares pero distintas de las de C. En los dos casos hay que manejar un número desmesuradamente grande de APIs y dominar una buena serie de conceptos antes de llegar al más mínimo atisbo de productividad. Los sistemas visuales de programación ocultan al programador la complejidad de las APIs del sistema, ofreciéndole un modelo de programación mucho más sencillo, sin dejar de hacerle accesible por ello toda la riqueza y complejidad del entorno gráfico.

En Windows se ha hecho popular Visual Basic; en OS/2, que dispone de un lenguaje de programación nativo mucho mejor que Basic como es REXX; es lógico que hayan aparecido productos que permitan la programación visual en REXX. La compañía que vende VisPro/REXX, que es el producto que utilizaremos como base para nuestras explicaciones, vende también un par de productos (VisPro/C y VisPro/C++) que están en la línea de Visual C++ para Windows y tienen una filosofía similar al que analizamos, aunque, naturalmente, para utilizarlos haya que programar en C (o C++), e, inevitablemente, disponer de un conocimiento mucho más profundo de las APIs del sistema. No nos ocuparemos de ellos aquí.

En esta serie de artículos vamos a introducir los conceptos básicos de la programación visual en REXX. En el que nos ocupa mostraremos cómo los conceptos y las herramientas de programación visual nos permiten escribir una aplicación con una funcionalidad

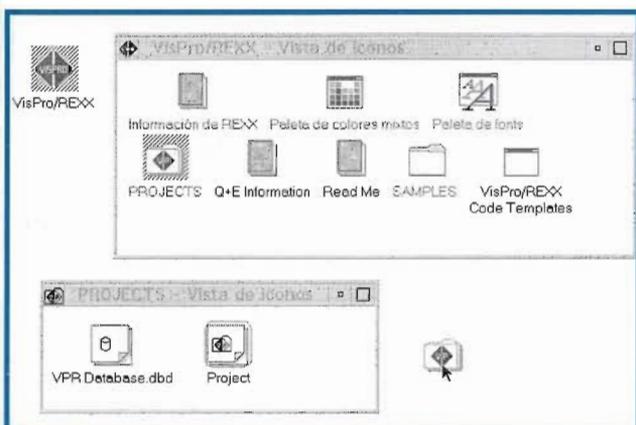


Figura A. Creación de un nuevo proyecto VisPro/REXX



Figura B. Contenido inicial de un proyecto

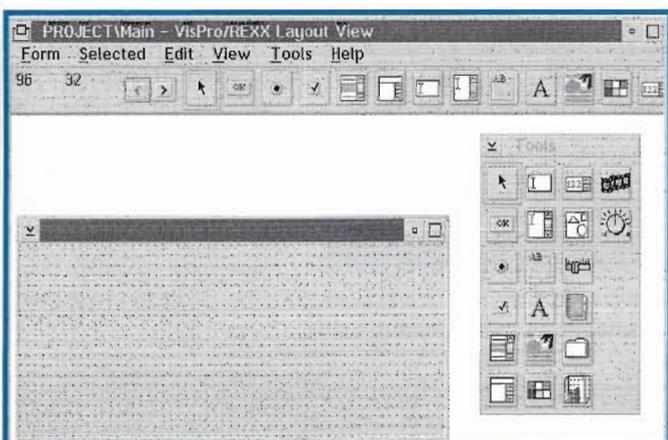


Figura C. La pantalla Main antes de modificarla. Puede apreciarse la sencillez y elegancia de la interfaz de usuario

VisPro/REXX para escribir aplicaciones que gestionen bases de datos en SQL.

En la actualidad existen dos productos similares para escribir programas OS/2 visualmente en REXX: VisPro/REXX de HockWare y VX-REXX de Watcom. He elegido dar los ejemplos en VisPro/REXX, ya que además de ser el producto que utilizo con más frecuencia (y por tanto domino mejor), posee una interfaz de usuario más clara y sencilla que VX-REXX, aunque este último producto (mucho más extenso que VisPro) ofrece algunas opciones de las que el que mostramos carece; las explicaciones que damos podrán aplicarse con muy pocas modificaciones a VX-REXX.

Nuestro primer programa visual en OS/2

Vamos a seguir paso a paso la construcción de un programa sencillo en VisPro/REXX. La instalación estándar de VisPro crea una carpeta llamada VisPro/REXX. Si la abrimos, encontramos una serie de objetos que nos serán útiles a la hora de programar; varios de ellos son sombras de objetos del sistema, que la instalación ha creado para hacerlos más accesibles. Para crear un nuevo programa, debemos abrir la carpeta *Projects* (proyectos) (Figura A).

Ésta contiene modelos para la interface con las bases de datos en DB2/2 (que no comentaremos), y para los programas en VisPro/REXX, o proyectos. Si arrastramos una copia desde el modelo y lo abrimos, nos encontraremos con los componentes de un proyecto estándar (Figura B).

En Windows se ha hecho popular Visual Basic; en OS/2, que dispone de un lenguaje de programación nativo mucho mejor que Basic como es REXX; es lógico que hayan aparecido productos que permitan la programación visual en REXX

Un programa VisPro/REXX se compone de una pantalla principal (*Main*), que no debe ser cambiada de nombre; posiblemente de más pantallas, para las que cada proyecto nos proporciona un modelo, desde el cual podemos crearlas (*Form*), y que pueden tener cualquier nombre; una serie de recursos (fundamentalmente iconos), que podemos manipular mediante

significativa en un tiempo record (el programa de ejemplo que mostramos se escribe en menos de 15 minutos); los siguientes estarán dedicados a explorar el uso de controles avanzados de OS/2, como los contenedores (*containers*) y libros de notas (*notebooks*) desde VisPro/REXX, a discutir algunos aspectos de la programación visual (como las cuestiones de sincronismo) que hacen a esta disciplina distinta de la programación clásica, y a mostrar cómo utilizar

Resource.VPR; una carpeta para las subrutinas de nuestro programa, y otra carpeta para los diversos hilos (*threads*) de nuestro programa, si éste utiliza esta posibilidad del OS/2. Lo más sencillo si estamos usando VisPro por primera vez es abrir la pantalla *Main* (**Figura C**)

Al abrirse, *Main* muestra su contenido, en estilo OS/2. La pantalla que estamos diseñando está inicialmente vacía; disponemos de una caja de herramientas flotante (*Tools*, que se puede cerrar), y de una barra de herramientas equivalente en la parte superior de la ventana de VisPro, desde las cuales podemos seleccionar elementos gráficos del OS/2 con los que podemos *pintar* nuestra aplicación sobre la ventana que estamos diseñando. Por ejemplo, podemos pulsar con el ratón el botón de la barra de herramientas que contiene un dibujo de un botón (con la inscripción "OK"): el cursor cambia automáticamente para representar que estamos pintando un botón, y, allí donde pulsemos, un botón quedará representado (**Figura D**).

El botón que acabamos de pintar puede ser cambiado de posición o de tamaño mediante las téc-

En esta serie de artículos vamos a introducir los conceptos básicos de la programación visual en REXX

nicas usuales en OS/2. Más adelante veremos cómo hacer para que nuestra aplicación reaccione a la acción del usuario de pulsar ese botón.

Igualmente, podemos cambiar el texto que aparece en el botón mediante el método habitual para los objetos en OS/2, esto es, pul-

sando con el ratón sobre el botón mientras pulsamos a la vez la tecla *Alt* (**Figura E**).

Del mismo modo, podemos escoger una *List Box* de la barra de herramientas y *pintarla* sobre nuestra ventana (**Figura F**).

Observaremos que la *List Box* escogida por defecto por VisPro contiene una barra de desplazamiento horizontal. Si no deseamos que eso ocurra, recurriremos al libro de valores del objeto (**Figura G**). En efecto, cada objeto que pintamos en VisPro/REXX tiene un libro de valores, como si fuera un objeto del *WorkPlace Shell*; veremos que las similitudes entre VisPro y OS/2, o, si se quiere, su integración con este último, no acaban aquí, lo cual hace que el tiempo de aprendizaje del producto sea mucho menor que si esa integración no se diese.

Desde el libro de valores de cualquier objeto podemos cambiar un gran número de propiedades de ese objeto, como la fuente (*font*) utilizada, los colores del primer plano y el fondo, su posición exacta, o asignarle un nombre simbólico mediante el cual podemos referirnos al objeto desde el programa, en vez de utilizar el número que VisPro asigna-

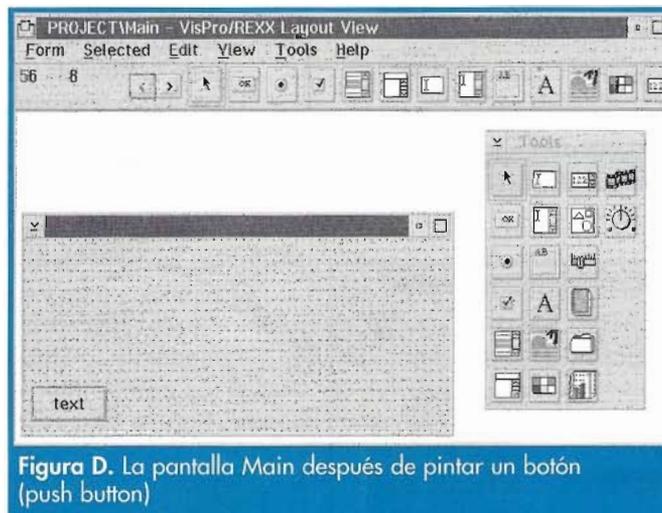


Figura D. La pantalla *Main* después de pintar un botón (push button)

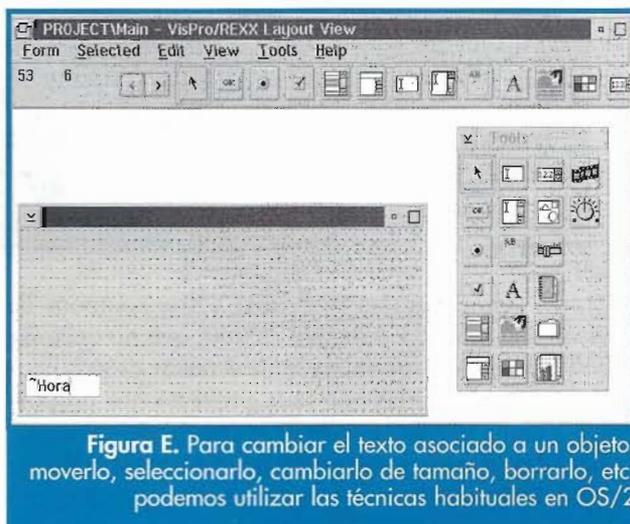


Figura E. Para cambiar el texto asociado a un objeto, moverlo, seleccionarlo, cambiarlo de tamaño, borrarlo, etc, podemos utilizar las técnicas habituales en OS/2

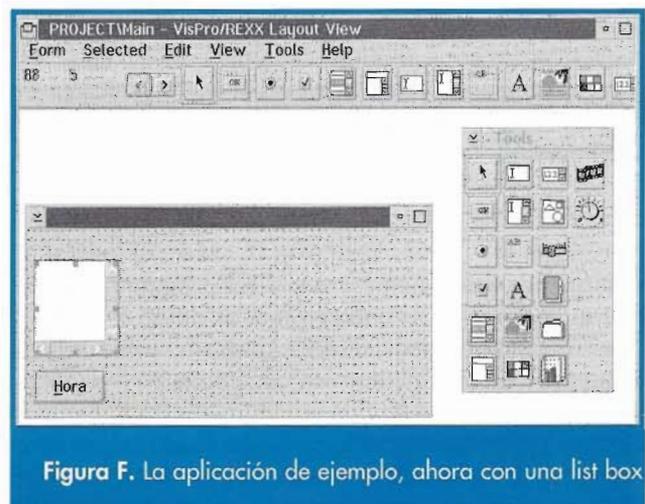


Figura F. La aplicación de ejemplo, ahora con una list box

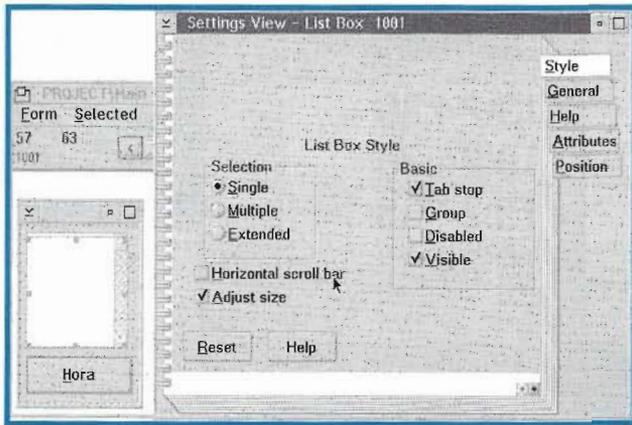


Figura G. Cada elemento visual de nuestra aplicación es un objeto, que, como los del Workplace Shell de OS/2, tiene su correspondiente libro de valores

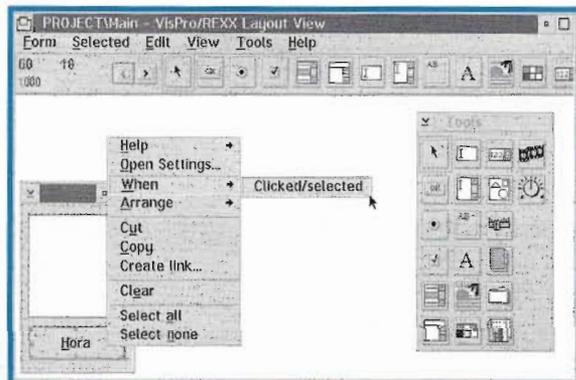


Figura H. Para asignar comportamiento a los elementos gráficos de nuestra aplicación basta con pulsar el botón derecho del ratón

(Figura H). Deseamos, por ejemplo, que cada vez que el usuario pulse el botón, la hora actual aparezca en la *List Box*; nada más fácil: pulsamos encima del botón *Hora* con el botón derecho del ratón, como en OS/2

na automáticamente al crear el objeto, para mayor claridad. Nosotros no utilizaremos esta última posibilidad, ya que no nos será necesario para este programa de ejemplo.

Una vez pintados el botón y la *List Box*, podemos ya empezar a pensar en asignar comportamiento a la ventana que hemos diseñado

—nunca salimos de él—, para obtener la lista de cosas que el botón *sabe hacer*; entre éstas encontramos una opción *When*, que se despliega a *Clicked/Selected*. Quiere decir que podemos decirle al botón qué ocurrirá cuando lo pulsemos.

Veremos que aparece un editor, en el cual podemos escribir código uti-

lizando REXX para definir el comportamiento del botón. Sin embargo, en muchos casos no será necesario escribir el código a mano, o, al menos, gran parte del código lo generará automáticamente VisPro, según nuestras indicaciones.

Hemos dicho que deseábamos que la *List Box* añadiese a su contenido la hora actual. Para conseguir ese efecto, no tenemos más que estirar la *List Box* sobre la ventana del código asociado a pulsar el botón (Figura I).

Al dejar caer la *List Box* sobre el editor de código, aparece automáticamente la lista de las acciones que es posible realizar sobre la *List Box*. Escogemos la de "añadir un ítem al final" (Figura J).

Vispro añade automáticamente el código para añadir algo al final, pero como no sabe *qué* queremos añadir, pone *value* en su lugar; nosotros no tenemos más que cambiar ese valor por lo que, desde nuestro programa, deseamos que aparezca; en nuestro caso, utilizaremos la función incorporada *Time()* de REXX, y suprimiremos *value* (Figura K).

Nuestra mini-aplicación ya está escrita. Podemos ejecutarla pulsando Control+R (Figura L), o seleccionando *Form --> Run*. Aparecerá una ventana mostrando la aplicación, y, efectivamente, cada vez que pulsamos el botón *Hora* aparece la hora actual en la *List Box*. ¡Y todo esto en un menos de cinco minutos!

En una aplicación real, probablemente deseamos que aparezca

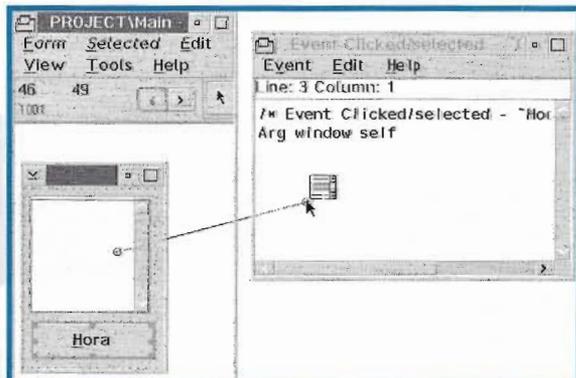


Figura I. Programación visual significa que parte del código lo escribe el entorno, respondiendo a nuestras acciones de tipo arrastrar y soltar

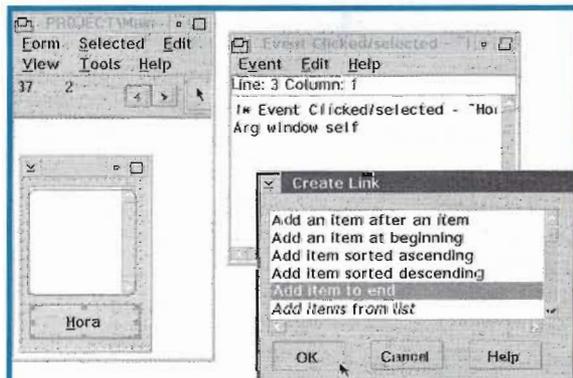


Figura J. Soltar un objeto sobre el código hace aparecer una lista de opciones, entre las cuales podemos elegir la que nos convenga

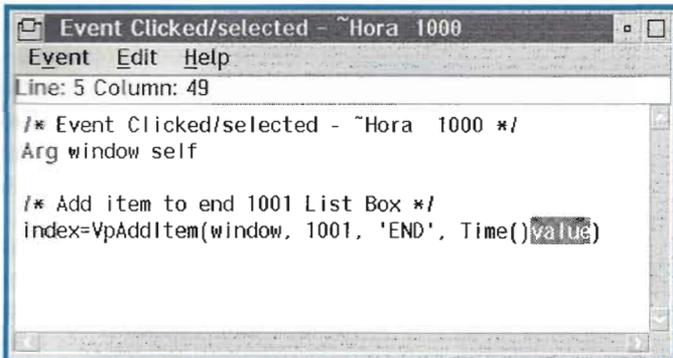


Figura K. Añadimos nuestro código al generado automáticamente por VisPro/REXX

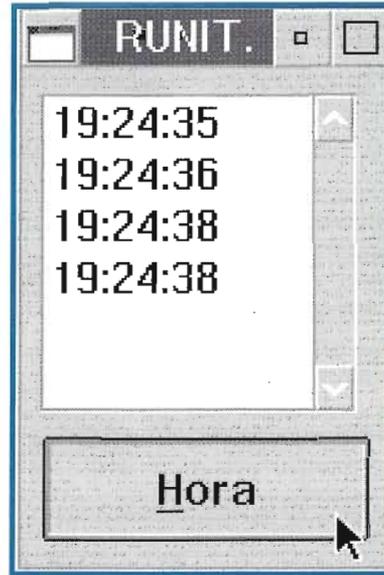


Figura L. Ya disponemos de una mini-aplicación, perfectamente operativa

algo significativo en la barra de título de la ventana. Esto se puede conseguir programándolo, de un modo muy sencillo, desde VisPro, si el texto de la barra de título debe cambiar según el momento de la ejecución del programa, o definiendo estáticamente cuál va a ser el título de la ventana desde el libro de valores de la aplicación (Figura M). Para hacer ésto último, podemos utilizar la opción *View* del editor de ventanas de VisPro, que nos permite conmutar entre diversas *vistas*, o, alternativamente, utilizar *Form->Open->Settings*, que nos permite tener dos vistas abiertas simultáneamente.

Desde la vista de valores del editor de ventanas podemos cambiar no sólo el título de la ventana, sino también el icono de la aplicación, y toda una serie de otros parámetros de la aplicación. En particular, desde la página *Style* podemos asignarle a nuestra aplicación un aspecto *Scrollable*.

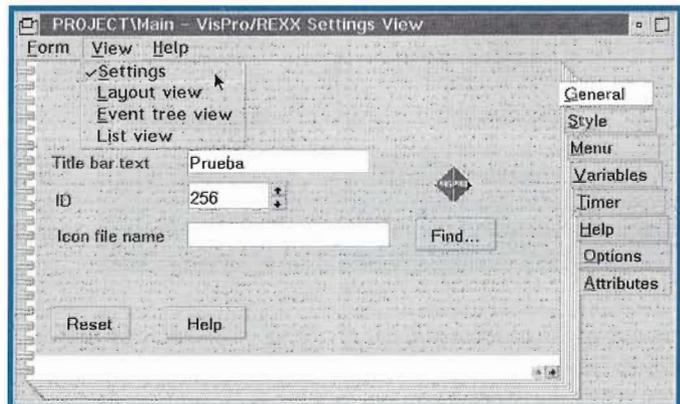


Figura M. El libro de valores de la aplicación

Una vez pintados el boton y la List Box, podemos ya empezar a pensar en asignar comportamiento a la ventana que hemos diseñado

Ya hemos generado nuestra primera aplicación. Veamos ahora cómo mejorarla con nuevos elementos gráficos. En primer lugar, añadiremos un objeto de gráficos de negocios desde la paleta de herramientas, y, después, tres botones de radio, mediante los cuales decidiremos cual va a ser el aspecto de nuestro gráfico (los valores que se muestran son los que VisPro/REXX dispone por defecto al añadir un objeto de gráfico de negocios, pero, obviamente, los valores y

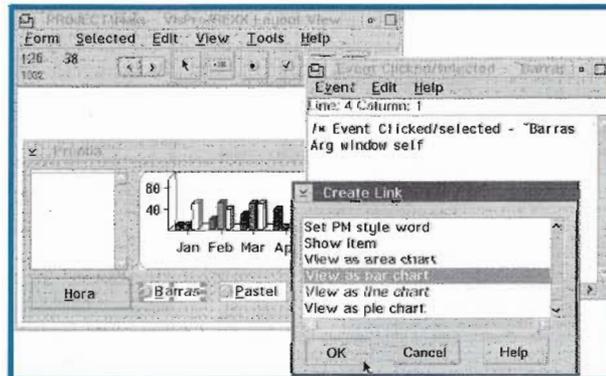


Figura N. Ampliamos la aplicación para que incluya un gráfico tridimensional

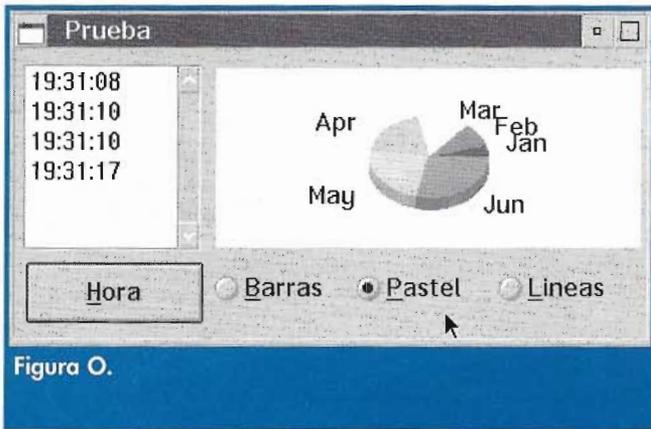


Figura O.

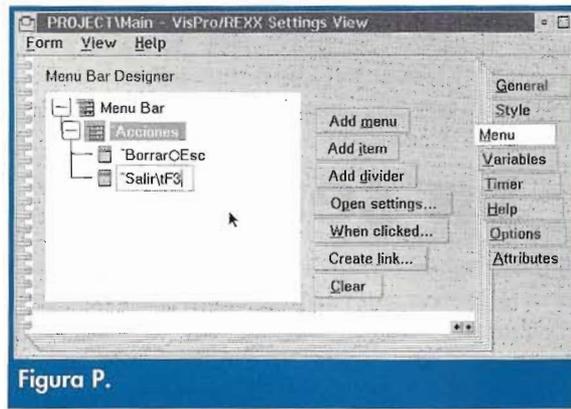


Figura P.

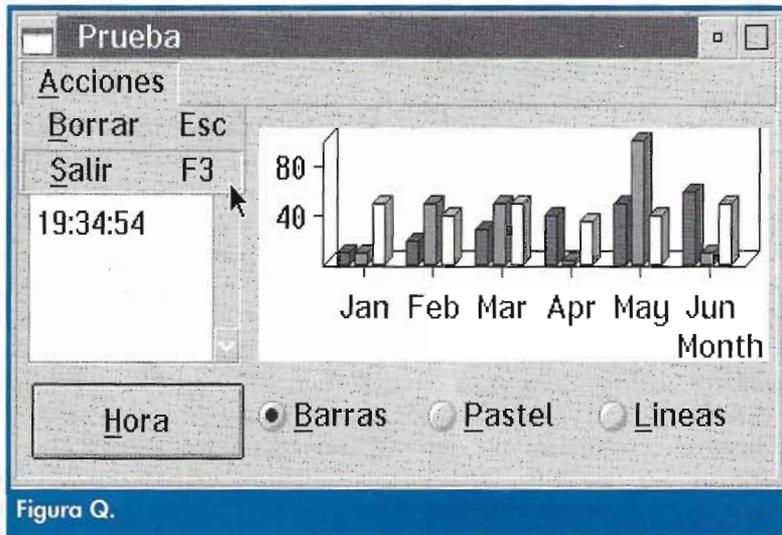


Figura Q.

formato estandar (p. ej. *Barras*), y *Salir*, para salir del programa. Queremos además que esas funciones estén accesibles directamente utilizando las teclas aceleradoras *Esc* y *F3*, respectivamente. Utilizaremos para ello la página *Menu* del libro de valores del editor de ventanas. *Add menu* nos permitirá crear un menu, *Add Item* los items necesarios; los valores de cada *item* nos permitirán asignar las teclas aceleradoras, y el comportamiento correspondiente lo programaremos al pulsar *When clicked*....

Esta es la aplicación, después de los cambios que hemos realizado en ella. La operación, en su conjunto, ha llevado menos de quince minutos, incluyendo las explicaciones (si somos programadores expertos en VisPro/REXX podemos escribir el programa completo en menos de cinco minutos). Ahora sólo nos queda generar un programa ejecutable, que ocupará alrededor de 100K, y funcionará autónomamente en cualquier ordenador que disponga de OS/2, sin necesidad de DLLs extras, ni pago de *royalties*, ni nada.○

las etiquetas, y, en general, cada aspecto del gráfico, es controlable desde la aplicación). Deseamos que al pulsar cada uno de los botones de radio cambie el aspecto del gráfico; para ello, procedemos exactamente igual que antes, mediante la técnica de *arrastrar y soltar*, y, en este caso, se nos ofrece la posibilidad de determinar el aspecto del gráfico (la lista de opciones que aparece al terminar la acción de *arrastrar y soltar* es, como es lógico, dependiente del objeto arrastrado).

Podemos probar de nuevo nuestra aplicación, que ahora es mucho más vistosa, ¡y sin que hayamos escrito ninguna línea de código!

Para terminar con nuestra aplicación de ejemplo, vamos a añadirle una barra de menus, con dos opciones: *Borrar*, que limpiará la *List Box* y devolverá el gráfico y los botones correspondientes a un

Direcciones de Madrid y Barcelona donde se puede adquirir VISPRO/REXX

Addlink Software Científico
C/ Rosellón, 205 4º 1º
08008 Barcelona
+34 3 415 49 04

Bitmailer, S.L.
Alcántara, 57 2-A
28006 Madrid
+34 1 402 15 51