

C de Cerca: CAPÍTULO PRIMERO

Primeros pasos

Programar el Mac no es fácil, pero tampoco es tan difícil. En esta primera entrega explicamos cómo instalar el compilador y hacemos una primera aproximación al C.

Para mucha gente, el Mac es una especie de «puzzle». Es una caja cerrada en cuyo interior ocurren fenómenos mágicos. La mayoría de usuarios de Mac están contentos mientras el software que emplean funcione.

Sin embargo, hay otro tipo de personas (seguramente los que están leyendo esta sección) que son diferentes. Quieren resolver el «puzzle», saber qué pasa dentro de esta caja mágica llamada Mac, escribir su propio software para Macintosh. Hemos preparado una serie de artículos que pretenden acercarlos al Mac a través de una introducción básica a la forma más común de programación en el entorno Macintosh.

Todos conocemos una calculadora de bolsillo. La ponemos en marcha... ¡y no hace nada! Si no introducimos algunas cifras e indicamos qué operación queremos hacer podríamos pasarnos la vida mirando la calculadora y esperando a que se le agoten las pilas... Las cifras que introducimos en la calculadora son los datos con los que debe trabajar, y la operación que tecleamos (suma, resta, etcétera) constituye la orden o instrucción que deseamos que ejecute. Una vez la calculadora «sabe» qué es lo que tiene que hacer, lo ejecuta.

Como la calculadora de nuestro ejemplo, el Mac necesita que alguien le diga qué tiene que hacer. Sin órdenes o instrucciones, un ordenador no es más que una calculadora (y no de bolsillo precisamente).

Un **programa**, como el que desarrollaremos en esta sección, no es más que una serie de instrucciones que podemos dar al Mac para que éste las ejecute una tras otra.

El Mac, como todos los ordenadores, sólo es capaz de entender instrucciones codificadas en su lenguaje nativo –que es seguramente lo más parecido al idioma que deben hablar en Venus...– Afortunadamente, los humanos (como la mayoría de nosotros, supongo) podemos crear **programas** para el Mac en

lenguajes más «terricolas» que están formados por un reducido vocabulario y se ajustan a unas simples normas de sintaxis. Una vez redactado el **programa**, el propio Mac se encargará de traducirlo a su idioma interno.

El lenguaje de programación C

Uno de los lenguajes más populares para crear **programas** para el Mac es el llamado **C**. En esta sección aprenderemos a conocer el vocabulario y la sintaxis del **C**, desarrollando un simple programa para el Mac.

En resumen, los pasos necesarios para crear un programa para el Mac son:

A: Conocer el idioma en el que vamos a escribirlo (obvio, ¿no?). En nuestro caso, se trata del lenguaje **C**.

B: Redactar (y salvar en disco, por cierto) un documento de texto, conteniendo las instrucciones que debe ejecutar el Mac, empleando el lenguaje **C**.

C: Conseguir que el Mac traduzca nuestro documento a su propio lenguaje interno. (Este proceso se llama compilación, como veremos más adelante.)

D: Si el paso **C** ha ido bien, el Mac habrá generado un fichero conteniendo las instrucciones en su propio lenguaje. (Este fichero puede ser ejecutado y, por cierto, también recibe el nombre de... ¡Programa!)

¿Qué es CodeWarrior?

CodeWarrior es un conjunto de herramientas (programas y documentos) con las que podemos crear nuestras propias aplicaciones, tanto para Mac (68K) como para PowerMac (PPC).

CodeWarrior fue el primer entorno de desarrollo para el sistema operativo (Mac OS) en la plataforma Power PC y es, actualmente, el producto más usado en el mundo para desarrollar para Macintosh.

Instalación del CodeWarrior Lite

En el CDROM de **MACFORMAT** de este mes encontraremos una carpeta denominada «C de Cerca» que contiene una versión reducida de CodeWarrior y el primer capítulo de C de Cerca. Hacen falta entre 5 y 6 MB libres en el disco duro para su instalación, y para ello bastará con arrastrar la carpeta directamente sobre el disco duro. La figura 1 muestra los ítems que debe contener la carpeta una vez ya copiada al disco duro. A continuación, y para terminar con la instalación, debemos arrastrar el icono «MetroNub», que se encuentra dentro de la carpeta de Code Warrior Lite, sobre la Carpeta del Sistema y seleccionar OK en el cuadro de diálogo que nos indica que las extensiones deben colocarse en la Carpeta del Sistema. Después de esto, deberíamos reiniciar el Macintosh, pero dado que no vamos a usar el **debugger** en este capítulo, no es necesario hacerlo.

¿Qué significa «Lite»?

Significa que es una versión reducida del verdadero Code Warrior, y que si bien podremos usar su editor-compilador y debugger para ver y compilar los proyectos que iremos incluyendo en C de Cerca, no se podrán crear, con esta versión, nuevos proyectos.

Manos a la obra: El editor-compilador

No es el propósito de esta sección entrar en el funcionamiento a fondo ni de las opciones del editor, ni del compilador, ni del debugger, así que únicamente se mencionarán las necesarias para la comprensión y para la utilización de los distintos ejemplos de la sección. Por si le sirve de ayuda, CodeWarrior Lite soporta globos de ayuda en algunas de sus ventanas y menús, pero además, es una aplicación muy Mac, en el sentido de que es fácil de usar y de deducir para qué sirve cada una de sus opciones.

Como suponemos que estará ansioso para empezar, vamos a ello. Abrimos la carpeta **Capítulo 1** que se encuentra dentro de la carpeta **Ejemplos** y hacemos doble clic sobre el icono **CdeCerca_1.µ**. Se abrirá Code Warrior Lite 1.4, y, con él, la ventana del proyecto CdeCerca_1 (figura 2). Seguro que alguno se ha preguntado ya, ¿y qué es un proyecto? Bueno, pues, para crear una aplicación o programa en CodeWarrior, lo primero que debemos hacer es crear un proyecto, que no es más que un documento en el que apuntamos todo lo necesario para compilar el programa. Como por ejemplo los documentos que contienen el código (las fuentes), las librerías, y los recursos como los gráficos, sonidos, iconos, etcétera, que vamos a necesitar en nuestro programa.

El documento proyecto puede tener cualquier nombre, pero es costumbre que termine con una extensión «.µ» (opción-m).

En la figura 2, vemos los dos documentos que contiene este proyecto y que son los únicos que vamos a necesitar para este primer programa: **MacOS.lib** que es una librería que viene con CodeWarrior, y nuestro código fuente **CdeCerca.c**.

Es usual que todos los documentos incluidos en un proyecto tengan un nombre con una extensión al final del mismo. Esto es especialmente útil, sobre todo, cuando estos proyectos incluyen muchos documentos. De esta forma, es mucho más fácil reconocer por su extensión, cuál es su función dentro del proyecto. Así, las librerías terminan en «.lib», los documentos con el código en «.c», los recursos en «.rsrc», etcétera.

El código fuente

Vamos a ver el código que hemos preparado para este mes. Para ello deberemos hacer doble-clic sobre el nombre **CdeCerca.c** de la ventana que muestra el proyecto. Se abrirá un documento de texto, figura 5.

Y podremos entonces observar que el código es un simple documento de texto, y que CodeWarrior es un simple editor en apariencia, sólo que con algunas prestaciones destinadas a ayudarnos a escribir y compilar este código.

Lo primero que vamos a aprender es a escribir comentarios, que es algo muy (MUY) útil en C o en cualquier otro lenguaje de programación. El editor de Code Warrior muestra, por defecto, los comentarios en color rojo, el código en negro, y las palabras reservadas o identificativos de color azul (estos colores pueden cambiarse en las preferencias del editor).

Code Warrior *entiende* que lo que escribimos es un comentario si empieza con «//» y considera que este comentario continúa hasta el final de la línea (una línea acaba cuando pulsamos Retorno).

```
// esto es un comentario
```

```
// esto es otro comentario en otra línea
```

```
esto ya no sería un comentario. //pero esto sí
```

También podemos usar el sistema más tradicional de comentar el código en C y que se indica con los signos «/*» que abren el comentario y «*/» que lo terminan. Con este sistema podremos hacer comentarios de varias líneas.

```
/*
```

```
esto es un comentario.
```

```
esto es el mismo comentario pero que continúa en otra línea
```

```
*/ atención, esto ya no sería un comentario
```

Quizá pueda parecer una nadería que los comentarios sean inmediatamente convertidos en texto de color al igual que las palabras reservadas, pero no se puede imaginar la de errores que esto evita.

Si en los ejemplos de esta sección hemos usado siempre este último sistema de comentario ha sido porque de esta forma mantenemos la compatibilidad con compiladores más antiguos. Así, si algún lector dispone de ellos podrá compilar nuestros ejemplos sin demasiados retoques.

El capítulo de este mes es una pura introducción más que a la programación en C, al entorno que vamos a usar. Así que, nos vamos a limitar a compilar el proyecto y a poner el programa resultado en marcha.

Sin embargo, el código está muy comentado y permite comprender bastante bien para qué sirve cada línea. Poco a poco, en los próximos capítulos, iremos explicando cada una de estas líneas, y añadiéndole de nuevas para poder dar mayores prestaciones a la aplicación que vamos a ir construyendo mes a mes.

Compilador en marcha

Para compilar y ejecutar este proyecto basta con pulsar la opción **Ejecutar** del editor CodeWarrior Lite tal y como se indica en la figura 3. El programa resultante se llama «Mínima expresión», irá a parar a la carpeta **Capítulo 1**, figura 4, y se pondrá en marcha automáticamente.

El programa no hace nada, bueno o casi nada, básicamente abre una ventana, dibuja un texto y se queda esperando que hagamos clic para cerrarse: figura 6. No es mucho, pero creemos que suficiente para el primer día.

Para los más intrépidos del lugar, hemos de advertir que se pueden realizar cambios al código y después volverlo a compilar, de forma que podremos obtener distintas aplicaciones. Algunos cambios pueden ser tan inocentes como por ejemplo, cambiar el mensaje que aparece en la ventana del programa «La mínima expresión de un ...» (línea 193 del código), pero otros, pueden FÁCILMENTE desembocar en errores que si no son detectables por el compilador, pueden colgar el Mac en su ejecución posterior. Así que, y sin asustar a nadie, no deje de investigar, ni de probar lo que quiera, pero, eso sí, aceptando que el Mac se puede (seguro) colgar. ¡Ah!, por cierto, no deje de leer el documento Metrowerks Warranty Disclaimer. Para los que no sepan inglés les podemos indicar que más o menos dice que ni Metrowerk, ni Ediciones Zinco, ni nosotros, nos hacemos responsables de las destrozas que pueda hacer con el CodeWarrior.

Como Pilatos, vaya.

Enric Herrera 100437.2651@compuserve.com

Rafael Escoté 100072.501@compuserve.com

¿NECESITA MÁS AYUDA?

Si quiere hacer alguna consulta especial puede dirigir sus preguntas a Rafael Escoté o Enric Herrera para la Sección **C de Cerca** a:

MACFORMAT, Conde Borrell 208, segundo, cuarta, 08029 Barcelona

O bien enviar un mensaje electrónico a:

100072.501@compuserve.com

Capítulo aparecido en el número 11 de MACFORMAT