

Función para la entrada de datos en programas escritos utilizando el compilador QuickBasic

*José R. Lebrón González, BSEE, PE
Profesor Asociado*

Sinopsis

En este trabajo se presenta una rutina de entrada de datos que puede ser útil para desarrollar programas de ingeniería utilizando el lenguaje de programación QUICKBASIC. Para escribir la rutina se utilizaron algunas de las funciones avanzadas que ofrece esta versión del lenguaje BASIC, que permiten el desarrollo de programas complejos. Además, se presenta una porción de un programa que sirve como ejemplo de la utilización de la misma.

Abstract

This document shows a data entry function written in the QUICKBASIC version of the BASIC computer programming language that can be used for the development of engineering application programs. The function uses many of the advanced features offered in this version of the BASIC programming language. An example that shows how to use the function is also presented.

Introducción

Los programas escritos para operar en un computador necesitan alguna información para procesar. En los sistemas interactivos la información se suministra a través del teclado del computador. Para que un programa sea efectivo, el proceso de entrar la información debe permitir corregir errores y verificar la validez de la información antes de que se acepte. Los errores introducidos en la información suministrada pueden atraparse efectivamente si se diseña una rutina de entrada de información eficiente.

Descripción de la función

Los lenguajes de programación proveen métodos para introducir información. En BASIC, por ejemplo, se puede utilizar la función INPUT. En otros lenguajes existen funciones similares, como SCANF en lenguaje C, READ en PASCAL y otros. Estas funciones no proveen mucho control al momento de introducir la información y poca tolerancia a errores. La función que se presenta en este trabajo se denominó ENTRADA y se diseñó para sustituir la función INPUT del lenguaje BASIC y para el compilador QUICKBASIC de Microsoft Corporation. Esta función también puede usarse con en el nuevo interpretador QBasic que se incluye en la versión 5.0 del sistema operativo MS-DOS. Las características principales de la función ENTRADA son:

1. Funciona como un editor de línea, en el cual es posible mover el cursor a cualquier parte del campo y modificar la información. También permite eliminar, añadir o modificar un símbolo.
2. Al llamar la función se especifica el largo del campo y el número de símbolos (letras, números, etc.).
3. Permite especificar el tipo de símbolos o teclas que aceptará, como cualquier tecla, letras mayúsculas y dígitos, números enteros o números reales. En el caso de los números enteros, solamente acepta un signo (más o menos) en el primer espacio del campo. Para los números reales, además de aceptar los signos en el primer espacio, solamente acepta un punto decimal y un exponente.
4. Permite especificar el color a utilizarse al presentar el campo en la pantalla.
5. Puede aceptar letras acentuadas o la ñe utilizando la combinación de la tecla ALT simultáneamente con la tecla de la letra sin acentuar.
6. Puede detectar las teclas de función (F1 - F10) y devolverle al programa el valor de la que se oprimió sin necesidad de utilizar las funciones que provee BASIC para atrapar estas teclas.

La función ENTRADA utiliza muchas de las características avanzadas que provee el lenguaje QUICKBASIC. La versatilidad de la función permite que pueda usarse como tutor en el uso de estas características avanzadas de

QUICKBASIC.

Las versiones de BASIC interpretado (GW BASIC o BASICA) permiten que el programador defina funciones que están limitadas a una línea, lo que limita su funcionalidad. QUICKBASIC y QBASIC proveen dos métodos para hacer el mismo trabajo, funciones y subprogramas. Aunque las funciones (rutinas) y los sub-programas pueden modificar el contenido de los parámetros, las funciones devuelven un valor. Durante el resto del artículo se usa el término rutina para referirse a las funciones.

Las rutinas pueden ser de cualquier tamaño. Las variables definidas en las rutinas son locales, esto es ni el programa principal ni otras rutinas las reconocen. Las variables que se deseen pasar a una rutina deben identificarse como parámetros o definirse como variables globales usando la definición COMMON SHARED.

Algunas diferencias que notarán los que están familiarizados solamente con las versiones del BASIC interpretado son la ausencia de número de líneas, construcciones decisionales IF-THEN-ELSE que contienen más de una línea y no hacen referencia a un número de línea, la construcción de decisión múltiple SELECT CASE y otras. Estas funciones adicionales hacen que el lenguaje QUICKBASIC sea tan poderoso como los lenguajes C y PASCAL, pero manteniendo la facilidad de aplicación que caracteriza el BASIC.

Para permitir el grado de control deseado, la función ENTRADA toma una tecla a la vez y la procesa. Cuando se oprime una tecla válida, ésta se añade a una variable literal ("STRING"). Cuando se especifica un número la rutina lo procesa como una variable literal y es necesario convertirlo de número a literal antes de llamar la rutina y de literal a número después de procesarlo.

Parámetros que usa la función ENTRADA

La función ENTRADA recibe siete parámetros y es importante que éstos sean del tipo especificado (ENTEROS, REALES, LITERALES, etc.). Los parámetros son los siguientes:

- col -** Número entero que contiene la coordenada de la columna donde comienza el campo. Los valores aceptables están entre 1 y 80.

- fila** - Número entero que contiene la coordenada de la fila donde comienza el campo. Los valores aceptables están entre 1 y 25.
- largo** - Número entero que define el largo del campo. Aunque los números enteros pueden tener valores entre -32767 y +32768, los valores negativos no tienen sentido y deben evitarse. QUICKBASIC acepta variables literales de hasta 128 espacios. El valor de "largo" debe limitarse entre 1 y 79 espacios para que quepa en una línea de la pantalla típica. La rutina no verifica los límites de esta variable por lo que se debe tener precaución de no especificar un largo inaceptable.
- tipo** - Número entero que define el tipo de campo a aceptarse. Los valores reconocidos son los siguientes;
- 0 - acepta cualquier símbolo con valor ASCII entre 32 y 255. (Cualquier tecla menos las de control)
 - 1 - igual que 0 pero convierte todas las letras en mayúsculas.
 - 2 - acepta solamente números reales. El signo, si existe, solamente se acepta en la primera posición. El símbolo de exponente ("e" o "E") se acepta solo una vez pero no en la primera posición. El punto decimal se acepta solo una vez.
 - 3 - similar a la 2 pero solamente acepta números enteros.
- fclr** - Número entero que define el color del fondo de la pantalla. Los valores aceptables están entre 0 y 7 y corresponden a los colores definidos en BASIC.
- lclr** - Número entero que define el color que se utilizará para representar los símbolos en la pantalla. Los valores aceptables están entre 0 y 15.
- TEXTO** - Variable literal que contiene la información a editarse o que se entrará por el teclado. Si no se desea un valor inicial esta variable debe iniciarse a cero.

La función ENTRADA devuelve un número entero que contiene el valor de la combinación de las teclas utilizadas para terminar la rutina. En el caso de las combinaciones de teclas que el sistema DOS define con dos dígitos, (como las teclas de función F-1 a F10 y la combinación de ALT o CNTRL con otras teclas), la función devuelve solamente el valor significativo de la combinación de teclas. Por ejemplo, si se terminó la rutina utilizando la tecla <ENTER>, la función devuelve 13, que es el valor ASCII de esta tecla. La tecla F1 devuelve 0 en el primer dígito y 59 en el segundo, en este caso la función devuelve 59.

Funcionamiento de la función ENTRADA

(Favor de referirse al Anejo #1 para el listado del programa)

La rutina comienza iniciando algunas variables internas necesarias.

Los sistemas MSDOS/PCDOS pueden configurarse con distintos tipos de pantallas como monocromática, CGA, EGA, VGA, etc. (las últimas tres son a colores). Para procesar texto no hace diferencia si la pantalla es del tipo CGA, EGA o VGA, solamente es necesario determinar si es a colores o monocromática. Los sistemas a colores utilizan 7 líneas para representar los símbolos y las monocromáticas utilizan 13. Para que el cursor se localize correctamente bajo el símbolo, es necesario definir el tamaño del mismo.

La rutina invoca la función VIDEO, la cual busca en el área de parámetros del sistema operativo para detectar el tipo de pantalla activo. La función VIDEO es útil para que un programa opere correctamente en cualquier tipo de configuración de pantalla. Si se definen colores en un programa, es posible que algunos no se puedan distinguir en un sistema monocromático. Además, algunas combinaciones de colores proveen un efecto agradable a colores pero en pantallas monocromáticas no proveen suficiente contraste. Una técnica que permite resolver este problema es definir los colores como variables, a las cuales se les asignan valores iniciales distintos para color y monocromático. En un bloque de decisión (puede ser un IF-THEN-ELSE) se pueden ajustar los colores de acuerdo al tipo de pantalla que se detecte.

Luego comienza el lazo principal de la función y usa la construcción DO - LOOP. Existen dos lazos de este tipo: el primero mantiene control hasta que se oprima una tecla y el segundo mantiene control hasta que se finalice de editar el campo.

Una vez se oprime una tecla, comienza su proceso de identificación. El primer grupo de construcción SELECT - CASE identifica el tipo de símbolos que se aceptarán. De acuerdo al valor del parámetro "tipo" la función interpreta los caracteres que se entran através del teclado. La tabla 1 define la acción de la función de acuerdo al valor del parámetro tipo.

Tabla 1. Acción de la función de acuerdo al parámetro tipo

tipo	Acción de la función
0	Acepta cualquier valor y convierte la combinación ALT + letra en las letras acentuadas, eñe o signo de interrogación.
1	Capitaliza todas las letras
2	Pasa por alto cualquier tecla que no sea número o una tecla de función o número. Aquí se asegura que solo se introduzca un signo en la primera posición del campo, un exponente o un punto decimal.
3	Igual que para el valor 2, pero los números enteros no pueden tener exponente ni punto decimal.

Si se oprime una de las teclas de signo, exponente o punto decimal, afirmamos las variables "PDecimal", "Exponente" o "Signo" a 1 para recordar que ya se tiene uno de estos símbolos.

La función ENTRADA también contiene otra construcción SELECT - CASE que es responsable de identificar las teclas de control y procesarlas. Además, si se oprime una tecla válida, la letra o símbolo se añade a la variable literal que contiene la información (TEXTO).

Si la tecla que se presiona es ESC o ENTER, entonces se termina la entrada de datos. Cuando se oprime una de las teclas de función (F-1 a F-10) también se concluye la modificación del campo. En ambos casos se devuelve al programa que llama la función el valor de la tecla que se utilizó para terminar.

Si se presiona la tecla BACKSPACE, entonces se elimina el último símbolo de la variable literal y se ajusta la posición del cursor. También

verificamos si no se ha eliminado uno de los símbolos de exponente, signo o punto decimal para permitir que se puedan añadir en otra posición.

Si se detecta una de las teclas de control del cursor, se procede a ajustar el valor de la variable "cursor", que lleva cuenta de la posición del cursor dentro del campo. END mueve el cursor al último símbolo del campo, HOME al primero. Lo mismo ocurre con las teclas de flechas.

Si se presiona la tecla de INS, ocurre un cambio entre los modos de edición. Cuando comienza el lazo principal, se coloca la función en el modo de sustitución. En este modo, cuando se oprime una tecla de un símbolo válido, el símbolo sustituye al que ocupa la posición indicada por el cursor. Por otro lado, en el modo de inserción, donde no ocurre la sustitución, cuando se introduce un símbolo, los demás símbolos se desplazan hacia la derecha para hacer espacio. El valor de la variable "insert" contiene el modo de edición activo. Para que el usuario sepa el modo activo, se modifica el tamaño del cursor de uno normal (debajo del símbolo) a uno en forma de bloque.

La función de DEL es similar a la de BACKSPACE, excepto que solamente elimina el símbolo indicado por el cursor.

Las demás teclas se consideran válidas y se añaden a la variable literal.

Ejemplo

(Favor de referirse al Anejo #2 para el listado del programa)

Como ejemplo de la posible utilización de la función ENTRADA se incluye una porción de un programa que acepta entrada de datos. En el ejemplo se introduce una rutina que permite validar los números que se entran en algunos campos de acuerdo a unos límites establecidos.

El programa comienza dibujando la pantalla de entrada de datos utilizando símbolos de línea y dejando espacio para introducir la información.

Para ilustrar la forma en que se pueden definir valores iniciales se le asignan valores literales a las variables Av\$ y Ai\$. Luego se le asignan valores a las variables último%, que contiene el número de campos en la pantalla; control%, que lleva cuenta del movimiento entre campos y activo, que contiene el campo activo%.

El lazo principal consiste de un lazo del tipo DO-LOOP. Se verifica la variable control% y se actúa de acuerdo a su contenido de la siguiente forma:

- Si es ENTER se pasa al próximo campo si no es el último.
- Si es "CURSOR HACIA ABAJO" se pasa al próximo campo.
- Si es "CURSOR HACIA ARRIBA" se pasa al campo anterior.
- Si es "F-1" se invoca el subprograma de ayuda.

Se verifica la variable activo% y se actúa de acuerdo a su contenido. Dependiendo del campo activo se invoca la función ENTRADA. Al terminar de editar el campo se invoca la función VALIDAR. Si el valor está entre los límites especificados se permite progresar al próximo campo, de lo contrario no se permite.

El lazo continúa hasta que se edite el último campo o se oprima la tecla ESC.

Subprograma AYUDA

Este subprograma no recibe parámetros. Al invocarse imprime el texto de ayuda en la pantalla.

Subprograma BORRAR

Este subprograma no recibe parámetros. Al invocarse borra el espacio dispuesto para imprimir mensajes.

Función VALIDAR

Esta función permite validar el contenido de los campos. Los parámetros que recibe son los siguientes:

- valor - variable literal que contiene lo que se quiere verificar.
- max - valor real que contiene el valor máximo a aceptarse.
- min - valor real que contiene el valor mínimo a aceptarse.
- Fclr - valor entero que contiene el color del fondo.
- Lclr - valor entero que contiene el color de las letras.

La función comienza convirtiendo el contenido de la variable valor en un número. Con una construcción IF-THEN-ELSE compara este valor con los indicados para máximo y mínimo. De no encontrarse dentro de los límites establecidos, imprime un mensaje de error y devuelve un valor de 1. De lo contrario devuelve un valor de 0.

Conclusión

Las rutinas que se presentan en este artículo pueden ser de utilidad para el desarrollo de programas de aplicación escritos en lenguaje QUICKBASIC. Para escribir las rutinas se han utilizado algunas de las funciones que este compilador ofrece y no están disponibles en la versión interpretada GWBASIC o BASICA. Con estas funciones es posible desarrollar programas con buena presentación que permitan buen control de la entrada de información.

Anejo #1

```

FUNCION ENTRADA% (col AS INTEGER, fila AS INTEGER, largo AS INTEGER, tipo AS
INTEGER, Fclr AS INTEGER, Lclr AS INTEGER, TEXTO AS STRING)
*****
!*Función ENTRADA.SUB.
!*Esta función permite la entrada de información desde el teclado
!*Funciona como un editor de línea
!*por: José R. Lebrón
*****
!*Trabaja solamente con QuickBasic, versiones 4.0 y 4.5
!* Parámetros:
!* col - columna donde se escribirá el campo - entero
!* fila - fila donde se escribirá el campo - entero
!* largo - largo del campo - entero
!* tipo - tipo de campo: - entero
!*     0 - literales
!*     1 - literales, sólo mayúsculas
!*     2 - números reales
!*     3 - números enteros
!* fclr - color del fondo - entero
!* lclr - color de las letras - entero
!* TEXTO - variable literal que contiene la información editada
!*
!* Devuelve el valor de la última tecla de control oprimida
*****

DEFINT A-Z
cursor = 1
PDecimal = 0
Exponente = 0
Signo = 0

IF VIDEO% = 0 THEN
    curap = 12
    COLOR 0, 7
ELSE
    curap = 7
    COLOR Fclr, Lclr
END IF

!*Le añade espacios al final del campo
TEXTO = TEXTO + STRING$(largo - LEN(TEXTO), 32)

LOCATE fila, col: PRINT TEXTO
LOCATE fila, col, 1, curap
DO
    kp$ = INKEY$
SELECT CASE tipo
    CASE 0
        SELECT CASE kp$
            CASE CHR$(0) + CHR$(30) 'ALT + a

```

```

kp$ = "á"
CASE CHR$(0) + CHR$(18) 'ALT + e
kp$ = "é"
CASE CHR$(0) + CHR$(24) 'ALT + o
kp$ = "ó"
CASE CHR$(0) + CHR$(23) 'ALT + i
kp$ = "í"
CASE CHR$(0) + CHR$(22) 'ALT + u
kp$ = "ú"
CASE CHR$(0) + CHR$(49) 'ALT + ñ
kp$ = "ñ"
CASE CHR$(0) + CHR$(50) 'ALT + m
kp$ = "ñ"
CASE CHR$(0) + CHR$(16) 'ALT + q
END SELECT

```

CASE 1 'Acepta solamente mayúsculas

```
kp$ = UCASE$(kp$)
```

```
IF kp$ = CHR$(0) + CHR$(50) THEN kp$ = "ñ"
```

CASE 2 'Acepta solamente números.

```
SELECT CASE kp$
```

```
  CASE CHR$(48) TO CHR$(57) '¿Son números
```

```
  CASE CHR$(46) '¿Es un punto decimal?
```

```
    IF PDecimal=1 THEN kp$ = "" 'Ya existe el punto decimal
```

```
  CASE CHR$(69), CHR$(101) '¿Es un exponente (E)?
```

```
    IF cursor = 1 THEN kp$ = "" 'Los exponentes no están
```

```
    'permitidos en el primer
```

```
    'espacio
```

```
    IF Exponente=1 THEN kp$ = "" 'Lo rechaza
```

```
  CASE CHR$(43), CHR$(45) '¿Es un signo?
```

```
    IF cursor <> 1 THEN kp$ = "" 'Los signos solamente se
```

```
    'permiten en el primer
```

```
    'espacio
```

```
    IF Signo = 1 THEN kp$ = ""
```

```
  CASE CHR$(0) TO CHR$(31) 'Se aceptan los símbolos de
```

```
  'control
```

```
CASE ELSE
```

```
  kp$ = ""
```

```
  'símbolos 'No se aceptan otros
```

```
END SELECT
```

CASE 3 'Acepta solamente números enteros

```
SELECT CASE kp$
```

```
  CASE CHR$(48) TO CHR$(57)
```

```
  CASE CHR$(43), CHR$(45)
```

```
    IF cursor <> 1 THEN kp$ = ""
```

```
    IF Signo = 1 THEN kp$ = ""
```

```
  CASE CHR$(0) TO CHR$(31)
```

```
CASE ELSE
```

```
  kp$ = ""
```

```
END SELECT
```

CASE ELSE

```
END SELECT
```

```
LOOP WHILE kp$ = "" 'Sigue leyendo el teclado mientras no se entre algo
```

Lebrón/Entrada datos para QuickBasic

```

'Si se ha entrado un punto decimal, exponente, o signo hay que recordarlo
IF kp$ = "." THEN PDecimal = 1
IF kp$ = "e" OR kp$ = "E" THEN Exponente = 1
IF kp$ = "+" OR kp$ = "-" THEN Signo = 1

SELECT CASE kp$
    'Ahora comenzamos a procesar la tecla

    CASE CHR$(13), CHR$(27)
        ENTRADA = ASC(kp$)
        EXIT DO
        'Se oprimió ENTER(13) o ESC(27)
        'Devuelve el valor ASCII
        'Sale del lazo

    CASE CHR$(0) + CHR$(59) TO CHR$(0) + CHR$(68), CHR$(0) + CHR$(72),
        CHR$(0) + CHR$(80)
        'Se oprimió una de las teclas de función (F1 a F10)
        ENTRADA = ASC(RIGHT$(kp$, 1))
        EXIT DO
        'Devuelve el valor
        'Sale del lazo

    CASE CHR$(8) 'Tecla de retroceso (BACKSPACE)
        cursor = cursor - 1
        'Mueve el cursor un espacio a la
        'izquierda
        IF cursor < 1 THEN cursor = 1
        'No permite movimiento si llegó
        'al principio del campo
        temp$ = MID$(TEXTO, cursor, 1)
        'Hace un copia del símbolo
        IF temp$ = "." THEN PDecimal = 0
        'Si es un punto, aclara la
        'bandera de punto

    'Si es una E, aclara la bandera de exponente
    IF temp$ = "e" OR temp$ = "E" THEN Exponente = 0

    'Si es un signo, aclara la bandera de signo
    IF temp$ = "+" OR temp$ = "-" THEN Signo = 0

    k = largo - cursor
    'k contiene el número de
    'espacios que quedan
    MID$(TEXTO, cursor, k) = MID$(TEXTO, cursor + 1, k)
    MID$(TEXTO, largo, 1) = " "
    'Borra el símbolo
    LOCATE fila, col: PRINT TEXTO
    'Imprime el campo

    CASE CHR$(0) + CHR$(79)
        cursor = LEN(RTRIM$(TEXTO))
        'Tecla END
        'Mueve el cursor al último
        'símbolo del campo

    CASE CHR$(0) + CHR$(71) 'Tecla HOME
        cursor = 1 'Mueve el cursor al primer espacio del campo

    CASE CHR$(0) + CHR$(75) 'Tecla del cursor a la izquierda
        cursor = cursor - 1 'Mueve el cursor un espacio a la izquierda
        IF cursor < 1 THEN cursor = largo 'A menos que estemos en el
        'principio.

    CASE CHR$(0) + CHR$(77) 'Tecla del cursor a la derecha
        cursor = cursor + 1 'Mueve el cursor un espacio a la derecha
        IF cursor > largo THEN cursor = 1 'A menos que estemos en el final

    CASE CHR$(0) + CHR$(82)
        IF insert = 0 THEN
            'Tecla INS, cambia al modo de inserción
            'Si no está en el modo de inserción

```

```

        insert = 1           'Lo coloca en este modo
    LOCATE , , , 1, curap 'Cambia el tamaño del cursor
ELSE
        insert = 0           'De lo contrario
        LOCATE , , , curap  'Lo coloca en el modo de sustitución
END IF

```

```

CASE CHR$(0) + CHR$(83) 'Tecla de DEL
    k = largo - cursor
    temp$ = MID$(TEXTO, cursor, 1)
    IF temp$ = "." THEN PDecimal = 0
    IF temp$ = "e" OR temp$ = "E" THEN Exponente = 0
    IF temp$ = "+" OR temp$ = "-" THEN Signo = 0
    MID$(TEXTO, cursor, k) = MID$(TEXTO, cursor + 1, k)
    IF cursor > largo THEN cursor = 1: cursor = largo
    MID$(TEXTO, largo, 1) = " "
    LOCATE fila, col: PRINT TEXTO
    IF cursor > largo THEN cursor = largo: SOUND 1000, 1

```

```

CASE CHR$(32) TO CHR$(255) 'Se oprimió cualquier otra tecla

```

```

    IF insert = 1 THEN 'Añadir un espacio si está en el modo de
        'inserción
        G = largo - cursor
        MID$(TEXTO, cursor, G) = " " + MID$(TEXTO, cursor, G)
        TEXTO = LEFT$(TEXTO, largo)
    END IF

```

```

    MID$(TEXTO, cursor, 1) = kp$ 'Añade el símbolo a la variable TEXTO
    LOCATE fila, col: PRINT TEXTO 'La escribe a la pantalla
    cursor = cursor + 1           'Incrementa la posición del cursor
    IF cursor > largo THEN cursor = cursor - 1

```

```

CASE ELSE

```

```

END SELECT
LOCATE fila, col + cursor - 1
LOOP

```

```

END FUNCTION

```

```

*****
* Esta función verifica el tipo de pantalla activo *
*****

```

```

DEFINT A-Z
FUNCTION VIDEO%
DEF SEG = 0           'Cambia al segmento base
IF PEEK(&H463) = &HB4 THEN 'Verifica si la pantalla es monocromática
    VIDEO = 1
ELSE
    VIDEO = 0
END IF
END FUNCTION
'Termina la función VIDEO

```

Lebrón/Entrada datos para QuickBasic

Anejo # 2

Programa de ejemplo para demostrar algunas de las posibilidades de la función ENTRADA. El programa dibuja una pantalla de entrada de información y permite al usuario el entrar o modificar la misma.

```
DECLARE FUNCTION VIDEO% ( )
DECLARE FUNCTION ENTRADA% (col%, fila%, largo%, tipo%, Fclr%, Lclr%, TEXTOS)

DECLARE SUB BORRAR ( ) ' Declaración del subprograma BORRAR
DECLARE SUB AYUDA ( ) ' Declaración del subprograma para imprimir
                        ' la pantalla de ayuda

' Declaración de una función de validación de la información.
DECLARE FUNCTION VALIDAR% (valor AS STRING, max AS ANY, min AS ANY,
                          Fclr AS INTEGER, Lclr AS INTEGER)

DECLARE FUNCTION ENTRADA% (col AS INTEGER, fila AS INTEGER, largo AS INTEGER,
                          tipo AS INTEGER, Fclr AS INTEGER, Lclr AS INTEGER, TEXTO AS STRING)

'EL PROGRAMA COMIENZA AQUI

COLOR 15, 1 ' Coloca la pantalla con fondo azul y letras blancas brillantes
CLS        ' Borra la pantalla

'Las siguientes líneas dibujan una caja en la pantalla, con el título del
'programa y el nombre de los campos en sitio adecuado
FOR x% = 2 TO 23
    LOCATE x%, 1: PRINT "|";
    LOCATE x%, 80: PRINT "|";
NEXT x%

LOCATE 1, 1: PRINT "┌" + STRING$(78, "=") + "┐";
LOCATE 3, 1: PRINT "│" + STRING$(78, "=") + "│";
LOCATE 2, 12: PRINT "Diseño de Amplificadores Utilizando Transistores
Bipolares";
LOCATE 20, 1: PRINT "└" + STRING$(78, "=") + "┘";
LOCATE 24, 1: PRINT "┌" + STRING$(78, "=") + "┐";
LOCATE 6, 14: PRINT "Ganancia de Voltaje (Av):";
LOCATE 8, 14: PRINT "Ganancia de Corriente (Ai):";
LOCATE 10, 14: PRINT "Impedancia de Entrada (Zi):";
LOCATE 12, 14: PRINT "Impedancia de Salida (Zo):";
LOCATE 14, 14: PRINT "Frecuencia de Operación:
Ohmios";
LOCATE 16, 14: PRINT "Frecuencia Máxima:
Ohmios";
LOCATE 18, 14: PRINT "Temperatura Máxima de Operación:
Hertz";
LOCATE 21, 4: PRINT "Utilice las teclas del cursor para moverse entre los
Hertz";
campos.";
LOCATE 22, 4: PRINT "Oprima <ENTER> para terminar su entrada. Oprima F-1 para
Grados C";
obtener ayuda.";

'A las variables Av$ y Ai$ se les ha asignado un valor inicial
Av$ = "100"
Ai$ = "1"

'Las siguientes variables mantienen la información sobre los campos
ultimo% = 7 'Número de campos en la pantalla
control% = 0 'Variable de control del movimiento entre campos
```

activo% = 1 'Contiene el número del campo activo

'Comienza el lazo principal

DO

SELECT CASE control%

CASE 13

' se oprimió ENTER

IF activo% = ultimo% THEN

EXIT DO

ELSE

activo% = activo% + 1

END IF

CASE 80

' se oprimió ↑, próximo campo

IF activo% = ultimo% THEN

activo% = ultimo%

ELSE

activo% = activo% + 1

END IF

CASE 72

' se oprimió ↓, campo anterior

IF activo% <> 1 THEN

activo% = activo% - 1

END IF

CASE 59

' se oprimió la tecla F-1

CALL AYUDA

END SELECT

SELECT CASE activo%

'Activamos los campos

CASE 1

'Primer campo

TEXTOS\$ = Av\$

'Invocamos al editor de líneas

control% = ENTRADA(47, 6, 7, 3, 0, 7, TEXTOS\$)

'Si el valor del campo está fuera de los límites especificados,

'no permitimos avanzar al próximo.

IF VALIDAR(TEXTOS\$, 500, 0, 15, 1) = 1 THEN activo% = activo% - 1

Av\$ = TEXTOS\$

CASE 2

TEXTOS\$ = Ai\$

control% = ENTRADA(47, 8, 7, 3, 0, 7, TEXTOS\$)

IF VALIDAR(TEXTOS\$, 500, 0, 15, 1) = 1 THEN activo% = activo% - 1

Ai\$ = TEXTOS\$

CASE 3

TEXTOS\$ = Zi\$

control% = ENTRADA(47, 10, 7, 3, 0, 7, TEXTOS\$)

IF VALIDAR(TEXTOS\$, 1000000, 50, 15, 1) = 1 THEN activo%=activo% - 1

Zi\$ = TEXTOS\$

CASE 4

TEXTOS\$ = Zo\$

control% = ENTRADA(47, 12, 7, 3, 0, 7, TEXTOS\$)

Zo\$ = TEXTOS\$

IF VALIDAR(TEXTOS\$, 10000, 100, 15, 1) = 1 THEN activo%=activo% - 1

Lebrón/Entrada datos para QuickBasic

```

CASE 5
TEXTOS$ = frec$
control% = ENTRADA(47, 14, 7, 2, 0, 7, TEXTOS$)
frec$ = TEXTOS$
IF VALIDAR(TEXTOS$, 100000, 10, 15, 1) = 1 THEN activo% = activo% - 1

```

```

CASE 6
TEXTOS$ = frmax$
control% = ENTRADA(47, 16, 7, 2, 0, 7, TEXTOS$)
frmax$ = TEXTOS$
IF VALIDAR(TEXTOS$, 100000, 10, 15, 1) = 1 THEN activo% = activo% - 1

```

```

CASE 7
TEXTOS$ = temp$
control% = ENTRADA(47, 18, 7, 2, 0, 7, TEXTOS$)
temp$ = TEXTOS$
IF VALIDAR(TEXTOS$, 200, -30, 15, 1) = 1 THEN activo% = activo% - 1

```

```

CASE ELSE
END SELECT

```

```

LOOP WHILE control% <> 27

```

'Termina el lazo principal. De aquí en adelante debe comenzar el resto del programa que utilizará los valores entrados utilizando la rutina anterior.

```

*****
'Este es el subprograma de ayuda. No recibe parámetros ni devuelve valor *
*****

```

```

SUB AYUDA

```

```

CALL BORRAR

```

```

COLOR 15, 1

```

```

LOCATE 21, 5: PRINT CHR$(24); " Campo anterior. "; CHR$(25); " Próximo
campo. <- Cursor hacia la izquierda.";

```

```

LOCATE 22, 5: PRINT "-> Cursor hacia la derecha. INS Cambiar
de modo de edición.";

```

```

LOCATE 23, 4: PRINT "DEL Eliminar símbolo. BACKSPACE Eliminar
símbolo anterior.";

```

```

a$ = INPUT$(1)

```

```

END SUB

```

```

'Termina el subprograma AYUDA

```

```

*****
'Este subprograma borra el pedazo de pantalla que se utiliza para escribir *
'los mensajes
*****

```

```

SUB BORRAR

```

```

FOR x% = 21 TO 23

```

```

LOCATE x%, 2: PRINT STRING$(78, 32)

```

```

NEXT x%

```

```

END SUB

```

```

'Termina el subprograma BORRAR

```



```

*****
!* Esta es la función de validación de información que utilizamos en este *
!* ejemplo *
*****
!* Los parámetros son: *
!* valor - variable literal que contiene lo que se quiere verificar *
!* max - valor real que contiene el valor máximo a aceptarse *
!* min - valor real que contiene el valor mínimo a aceptarse *
!* Fclr - valor entero que contiene el color del fondo *
!* Lclr - valor entero que contiene el color de las letras *
!* Devuelve 0 si la prueba es exitosa y 1 si no lo es *
*****
FUNCION VALIDAR% (valor AS STRING, max AS SINGLE, min AS SINGLE,
Fclr AS INTEGER, Lclr AS INTEGER)

COLOR Fclr, Lclr
LOCATE 1, 1, 0
numero = VAL(valor) 'Convierte la variable literal a un número

'Si está fuera de los límites especificados
IF (numero > max) OR (numero < min) THEN
    BEEP 'Suenan las campanas
    CALL BORRAR 'Borra el área de mensajes
    'Imprime un mensaje de error con los límites
    'aceptables
    LOCATE 21, 17: PRINT "Valores aceptables:Máximo=";max;" Mínimo="; min
    LOCATE 22, 21: PRINT "Oprima cualquier tecla para continuar..."
    a$ = INPUT$(1) 'Espera que se oprima cualquier tecla
    CALL BORRAR 'Vuelve a borrar el área de mensajes
    VALIDAR = 1 'Devuelve el valor de 1 al programa principal
ELSE
    VALIDAR = 0 'Si el valor es aceptable devuelve 0 al
    'programa principal
END IF
END FUNCTION
'Termina la función VALIDAR

```