

**installer**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> installer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 31, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>installer</b>	<b>1</b>
1.1	installer.guide	1
1.2	installer.guide/Derechos de Copyright	1
1.3	installer.guide/Historia	2
1.4	installer.guide/Presentación	3
1.5	installer.guide/Invocación estándar	3
1.6	installer.guide/Operaciones iniciales	5
1.7	installer.guide/Ventanas de inicio	5
1.8	installer.guide/Operaciones de instalación	5
1.9	installer.guide/Guía de estilo	6
1.10	installer.guide/Tutorial del lenguaje de scripts	7
1.11	installer.guide/Elementos básicos	8
1.12	installer.guide/Caracteres de escape	8
1.13	installer.guide/Símbolos (Variables)	9
1.14	installer.guide/Tipos de símbolos	10
1.15	installer.guide/Órdenes	10
1.16	installer.guide/Tipos de datos	11
1.17	installer.guide/Características especiales	13
1.18	installer.guide/Miscelánea	14
1.19	installer.guide/Referencia del Lenguaje de Installer	14
1.20	installer.guide/NOTAS MUY IMPORTANTES	15
1.21	installer.guide/Órdenes	16
1.22	installer.guide/set	17
1.23	installer.guide/symbolset	17
1.24	installer.guide/makedir	18
1.25	installer.guide/copyfiles	18
1.26	installer.guide/copylib	20
1.27	installer.guide/startup	22
1.28	installer.guide/tooltype	23
1.29	installer.guide/textfile	24

1.30	installer.guide/execute . . . . .	25
1.31	installer.guide/run . . . . .	25
1.32	installer.guide/rexx . . . . .	26
1.33	installer.guide/makeassign . . . . .	26
1.34	installer.guide/rename . . . . .	26
1.35	installer.guide/delete . . . . .	27
1.36	installer.guide/protect . . . . .	28
1.37	installer.guide/complete . . . . .	29
1.38	installer.guide/message . . . . .	29
1.39	installer.guide/working . . . . .	30
1.40	installer.guide/welcome . . . . .	30
1.41	installer.guide/effect . . . . .	30
1.42	installer.guide/showmedia . . . . .	32
1.43	installer.guide/setmedia . . . . .	34
1.44	installer.guide/closemedia . . . . .	35
1.45	installer.guide/Órdenes de Control . . . . .	35
1.46	installer.guide/if . . . . .	36
1.47	installer.guide/select . . . . .	36
1.48	installer.guide/while . . . . .	36
1.49	installer.guide/until . . . . .	36
1.50	installer.guide/foreach . . . . .	37
1.51	installer.guide/secuencia . . . . .	37
1.52	installer.guide/abort . . . . .	37
1.53	installer.guide/exit . . . . .	37
1.54	installer.guide/trap . . . . .	38
1.55	installer.guide/onerror . . . . .	38
1.56	installer.guide/trace . . . . .	38
1.57	installer.guide/retrace . . . . .	39
1.58	installer.guide/Órdenes de Depuración . . . . .	39
1.59	installer.guide/user . . . . .	40
1.60	installer.guide/debug . . . . .	40
1.61	installer.guide/Procedimientos Definidos por el Usuario . . . . .	40
1.62	installer.guide/Funciones . . . . .	41
1.63	installer.guide/cadena . . . . .	42
1.64	installer.guide/cat . . . . .	43
1.65	installer.guide/substr . . . . .	43
1.66	installer.guide/strlen . . . . .	43
1.67	installer.guide/transcript . . . . .	44
1.68	installer.guide/tackon . . . . .	44

---

1.69	installer.guide/fileonly	44
1.70	installer.guide/pathonly	44
1.71	installer.guide/expandpath	45
1.72	installer.guide/askdir	45
1.73	installer.guide/askfile	45
1.74	installer.guide/askstring	46
1.75	installer.guide/asknumber	46
1.76	installer.guide/askchoice	47
1.77	installer.guide/askoptions	48
1.78	installer.guide/askbool	49
1.79	installer.guide/askdisk	49
1.80	installer.guide/exists	50
1.81	installer.guide/earlier	50
1.82	installer.guide/getsize	50
1.83	installer.guide/getdevice	50
1.84	installer.guide/getdiskspace	51
1.85	installer.guide/getsum	51
1.86	installer.guide/getversion	51
1.87	installer.guide/getenv	52
1.88	installer.guide/getassign	52
1.89	installer.guide/iconinfo	53
1.90	installer.guide/database	54
1.91	installer.guide/select	55
1.92	installer.guide/patmatch	55
1.93	installer.guide/symbolval	55
1.94	installer.guide/Comparación	56
1.95	installer.guide/Matemáticas básicas	56
1.96	installer.guide/Funciones lógicas	56
1.97	installer.guide/Funciones lógicas orientadas a bits	57
1.98	installer.guide/Conmutación de bits	57
1.99	installer.guide/Comprobación de bits	57
1.100	installer.guide/Sumario de Parámetros	57
1.101	installer.guide/Variables Predefinidas	61
1.102	installer.guide/Referencia rápida del lenguaje de Installer	62
1.103	installer.guide/Introducción (resumen)	63
1.104	installer.guide/Introducción rápida al lenguaje	63
1.105	installer.guide/Variables predefinidas (resumen)	65
1.106	installer.guide/Variables de cadenas de ayuda por omisión	66
1.107	installer.guide/Órdenes (resumen)	66
1.108	installer.guide/Funciones (resumen)	70

# Chapter 1

## installer

### 1.1 installer.guide

Installer

\*\*\*\*\*

Este documento contiene toda la información necesaria para aprender a utilizar la herramienta Installer de Amiga.

Derechos de Copyright

Historia

Presentación

Guía de estilo ¡Lea esto!

Tutorial del lenguaje de scripts

Referencia del lenguaje de Installer

Referencia rápida del lenguaje de Installer

NOTA: En versiones de AmigaGuide inferiores a la V39 (< Workbench 3.0), es posible que observe algunas marcas en pantalla que parecen fuera de lugar. Se trata de comandos de formateo V39, que hacen que el texto tenga un mejor aspecto en versiones de AmigaGuide >=V39, pero que en ningún caso reducen la cantidad de información mostrada en este documento.

### 1.2 installer.guide/Derechos de Copyright

Derechos de Copyright y Sentimientos

\*\*\*\*\*

La herramienta Installer y su documentación son ©1995-2001 de Amiga, Inc.

Basadas en el trabajo previo de Loren Wilton, la herramienta Installer V43 y su documentación son obra de Heinz Wrobel. Este trabajo está dedicado a Joan Thuesen.

Las actualizaciones presentes en Installer V44 fueron hechas

---

por Jochen Becher y el equipo de HAAGE&Partner Computer GmbH.

## 1.3 installer.guide/Historia

### Historia

\*\*\*\*\*

Tradicionalmente, la instalación en disco duro de aplicaciones servidas en disquetes ha resultado una tarea muy poco consistente y a menudo muy frustrante para la mayoría de los usuarios. Esto ha estado motivado por muchos factores, algunos de los cuales son:

- a. Muchos productos no vienen acompañados de una utilidad o script para instalar la aplicación en el disco duro.
- b. Muchos productos asumen por parte del usuario un gran nivel de familiaridad con las nociones básicas del Amiga y las aplicaciones, incluyendo asignaciones, nombres de unidades (diferenciándolos de los nombres de volumen), etc.
- c. Los scripts o utilidades de instalación incluidos con ciertos productos varían enormemente en su capacidad para enfrentarse a diferentes entornos y sistemas.

En 1991, Commodore se propuso poner remedio a esta situación, desarrollando una herramienta estándar que los desarrolladores pudiesen incluir con sus productos, y que ofreciese al usuario una forma estándar de instalar aplicaciones. Las características de Installer se basaron en una serie de premisas:

- a. Los requerimientos de una determinada instalación pueden variar mucho: algunas necesitan asignaciones, otras precisan la creación de nuevos cajones, algunas instalan cosas en los cajones del sistema -como puede ser 'fonts'-, un producto puede ser simplemente una actualización y la instalación podría necesitar saber qué versión hay actualmente instalada (si es que la hay), etc.
  - b. Diferentes usuarios se desenvuelven con distinta facilidad y comodidad al intentar instalar software, e Installer debería servir a todos. Muchos scripts de instalación asumen un elevado grado de conocimientos, lo cual puede resultar muy intimidatorio para un novato.
  - c. La herramienta Installer debe poder ser muy flexible internamente, al tiempo que debe presentar al usuario un interfaz gráfico consistente y agradable, que sólo le muestre información o le plantee las preguntas estrictamente necesarias. Installer debería tener en cuenta la resolución, el número de colores y el tipo de letra preferidos del usuario.
  - d. Escribir scripts para instalar una aplicación precisará cierto esfuerzo, pero desde luego no más que escribir un script AmigaDOS
-

equivalente, mientras que el procedimiento de instalación resultante será más amistoso, flexible y de mucho mejor aspecto que este último.

Amiga Technologies mejoró Installer para permitir desarrollar procedimientos de instalación aún mejores y más versátiles, al tiempo que se mantiene la orientación generalista. Durante el desarrollo de Installer V43 se añadieron muchas novedades.

Amiga, Inc. mejora Installer para permitir implantar efectos visuales y pantallas multimedia bajo el control de la herramienta. Además, se ha implantado un mecanismo de vuelta atrás.

## 1.4 installer.guide/Presentación

### Presentación

\*\*\*\*\*

Installer es un programa dirigido por scripts, que presenta al usuario un entorno de instalación consistente. El usuario nunca llega a ver el script; simplemente se le presentan una serie de sencillas preguntas del tipo sí/no, y se le puede consultar para que indique lugares específicos de su sistema donde instalar cosas.

Para acomodar distintos tipos de usuarios, éstos pueden elegir entre ejecutar la herramienta en modo novato, medio o experto. Los scripts pueden incluir texto de ayuda para documentar las opciones que el usuario puede tomar. En cada paso, el usuario dispone asimismo de la posibilidad de interrumpir la instalación.

Invocación estándar  
Operaciones iniciales  
Ventanas de inicio  
Operaciones de instalación

## 1.5 installer.guide/Invocación estándar

### Invocación estándar

=====

Normalmente, Installer se lanza desde un icono de proyecto en el Workbench que tiene el mismo nombre que el script que se ha de interpretar, y que cuenta con Installer como herramienta por omisión. Se dispone de una serie de tipos de herramienta con los que puede modificarse el funcionamiento de Installer:

'SCRIPT'



Senda que apunta a un fichero script a utilizar con Installer.

**'APPNAME'**

Nombre de la aplicación que se va a instalar (aparece en la ventana de inicio). Este tipo de herramienta ES fundamental.

**'MINUSER'**

El nivel mínimo posible de uso de la instalación para un script. Su valor puede ser NOVICE ('Novato', todas las decisiones serán tomadas por Installer), AVERAGE ('Medio', el usuario sólo tomará las decisiones importantes) o EXPERT ('Experto', el usuario deberá confirmar todas las operaciones). El valor por omisión es NOVICE.

**'DEFUSER'**

Indica el botón que debería estar seleccionado inicialmente eligiendo el nivel de uso. Los valores posibles son los mismos que en el caso de MINUSER, siendo precisamente el valor por omisión el indicado con MINUSER (que será NOVICE si no se ha dado un valor distinto a MINUSER).

**'NOPRINT'**

Si su valor es FALSE, la opción de enviar a la impresora una copia de las operaciones llevadas a cabo durante la instalación estará difuminada y no podrá seleccionarse.

**'PRETEND'**

Si su valor es FALSE, no podrá emplearse el modo 'simulado' del funcionamiento del script.

**'LANGUAGE'**

Se utiliza para definir la variable '@language'. El valor por omisión para '@language' es el nombre del idioma elegido en el editor Locale, o "english" en caso de no haber ninguno disponible. El uso de esta variable queda a cargo del script de instalación, y debería utilizarse para adaptar el idioma de los mensajes presentados por el script. Tenga en cuenta que en Installer V42 y superiores, el idioma seleccionado en el S.O. por medio del correspondiente editor de preferencias será empleado de forma automática en caso de que no se indique un idioma en particular. Probablemente, este tipo de herramienta sólo debería ser utilizado para forzar el uso de un determinado idioma durante el proceso de pruebas de un script, o cuando se espera que el script sea ejecutado en un sistema con el S.O. 2.04. En este último caso, se recomienda emplear el comando 'getversion' para conocer la versión del S.O.

**'LOGFILE'**

El nombre del fichero de registro que debería utilizar Installer. Deberá indicarse una senda completa. El valor por omisión es 'registro\_de\_instalación' en el caso de Installer V42+ y un sistema localizado al español.

**'LOG'**

En modo NOVICE, se crea por omisión un fichero de registro que se guarda en disco. Si el argumento de este tipo de herramienta es FALSE, la creación del fichero de registro en modo NOVICE quedará desactivada.

---

A pesar de que Installer puede invocarse desde Shell, no es el método de invocación recomendado. La invocación desde Shell está soportada principalmente para propósitos de depuración de scripts. La plantilla del comando es:

```
'SCRIPT,APPNAME,MINUSER,DEFUSER,LOGFILE,LANGUAGE,NOPRETEND/S,NOLOG/S,
NOPRINT/S'
```

## 1.6 installer.guide/Operaciones iniciales

Operaciones iniciales

=====

La primera cosa que hace Installer es compilar el script de instalación en un formato interno que puede ser fácilmente interpretado. En caso de haber en el script algún error de sintaxis, quedará atrapado durante esta fase.

## 1.7 installer.guide/Ventanas de inicio

Ventanas de inicio

=====

A continuación, Installer pregunta al usuario por el Modo de instalación en el que correr, que puede ser Novato, Medio o Experto. Si el usuario elige Novato, no se le harán más preguntas (aunque se le puede pedir que haga algunas cosas, como insertar un disquete). En el resto de niveles de usuario, aparecerá una segunda ventana preguntando al usuario si desea hacer una instalación "real" o bien "simulada", y si desea una transcripción por escrito del proceso de instalación, y en caso informativo si la quiere en un fichero o por impresora. En modo Experto, el usuario debería ser consultado con cada opción de configuración, como es el caso de la instalación de cajones, mientras que en el nivel Medio sólo deberían plantearse las cuestiones más importantes.

## 1.8 installer.guide/Operaciones de instalación

Operaciones de instalación

=====

A continuación Installer interpreta su versión interna del guión. Cualquier comando que invoque a un interfaz de usuario hará que Installer genere de forma algorítmica una ventana, incluyendo siempre botones para permitir ofrecer ayuda sensible al contexto, así como la posibilidad de interrumpir la instalación.

---

## 1.9 installer.guide/Guía de estilo

### Guía de Estilo

\*\*\*\*\*

Crear un script de instalación para una aplicación típica es bastante fácil con Installer V43. Pero, para conseguir un proceso de instalación realmente brillante, por favor, sírvase adherirse a las directrices que se exponen a continuación:

Installer V43 ofrece una serie de novedades respecto a versiones anteriores, como mejores valores de retorno para ciertas órdenes, o la presentación de las opciones con un tipo de letra de anchura proporcional, a título opcional. La versión mínima necesaria de Installer se indica en cada descripción. Por favor, examine el contenido de la variable '@installer-version' antes de utilizar estas nuevas características. Esta variable vale 0 por omisión en el caso de viejas versiones de Installer como la 1.24, de forma que su comprobación es fácil y segura. Por favor, no haga una comprobación basada en la igualdad; compruebe la presencia de una versión mínima, tal como lo haría en el caso de las librerías.

A partir de la versión V42 existen una serie de directrices perfectamente razonables a la hora de nombrar cosas. Por favor, respételas a menos que quiera que la actualización de Installer resulte problemática para el usuario. No utilice el signo '@' en el nombre de sus variables; este signo marca las variables de sistema de Installer. Por favor, utilice un prefijo en el nombre de sus variables, como '#', y nombre sus procedimientos cuidadosamente empleando asimismo un prefijo, como 'P\_'. Los nombres que no cuenten con un prefijo podrían dar serios problemas con futuras versiones de Installer.

Siempre, siempre, siempre incluya textos de ayuda de fácil comprensión en su script de instalación.

No instale partes de su aplicación en cajones estándar del S.O. como son 'DEVS:', 'SYS:' o 'FONTS:' si no es absolutamente necesario. Mucha gente intenta mantener lo más separados posible su S.O. y sus aplicaciones, para de esta forma poder disfrutar de unas fáciles actualizaciones. Nunca olvide que 'PROGDIR:' podría ser lo que realmente necesita.

Si no puede mantener juntos los ficheros necesarios, al menos deje abierta la posibilidad de cambiar el destino a un lugar distinto de, por ejemplo, 'LIBS:'.

Siempre, siempre, siempre incluya en su script una opción de des-instalación. Ésta es otra razón para evitar desperdigar ficheros por todas partes.

Si está instalando o des-instalando librerías, procure no limitarse a sobreescribir o a borrar una librería ya existente, cuando otro programa podría necesitarla. En caso de duda, PREGUNTE AL USUARIO, claro está siempre y cuando éste \*no\* haya elegido el nivel Novato.

No tenga al usuario esperando a menos que sea necesario. Si su aplicación cabe en un disco, haga *\*al principio\** todas las preguntas necesarias, y a continuación instale los ficheros.

Si tiene que preguntar al usuario por el directorio donde instalar una aplicación, deje bien claro si su script va a crear el directorio en sí para la aplicación, o si el usuario tendrá que crear e indicar el directorio por su cuenta. Ésta ha sido siempre una fuente de gran confusión en muchos scripts de instalación.

Evite hacer preguntas en el nivel Novato; si no le queda más remedio que hacerlas, no soporte este nivel. Cualquier instalación efectuada en este nivel debería ser capaz de transcurrir con éxito de principio a fin sin la intervención del usuario.

Establezca una diferencia entre los niveles Medio y Experto. En el nivel Experto, el usuario debería poder intervenir prácticamente en todas las decisiones. En el nivel Medio, sólo deberían plantearse las cuestiones más importantes. No deje que los niveles Medio y Experto den el mismo resultado práctico. Dé al usuario experto algo con lo que entretenerse, y evite confundir al usuario medio.

A partir de su versión V44, Installer permite volver atrás en la instalación. En caso de utilizar este mecanismo, el botón "Interrumpir instalación" será sustituido en la mayoría de los casos por un botón "Volver atrás". Debería definir un parámetro '(back)' para todas las órdenes y funciones que soporten la intervención del usuario y que deban presentar este botón; si no lo hace, el botón "Volver atrás" quedará desactivado, lo cual podría resultar frustrante para el usuario.

## 1.10 installer.guide/Tutorial del lenguaje de scripts

### Tutorial del Lenguaje de Scripts

\*\*\*\*\*

El lenguaje de scripts de Installer está basado en LISP. No es difícil de aprender, pero requiere un montón de paréntesis. Es fácil hacer que un script de Installer resulte perfectamente legible a simple vista.

- Elementos básicos
- Caracteres de escape
- Símbolos (Variables)
- Tipos de símbolos
- Órdenes
- Tipos de datos
- Características especiales
- Miscelánea

## 1.11 installer.guide/Elementos básicos

Elementos básicos  
=====

Tan sólo hay unos cuantos elementos básicos en el lenguaje de Installer. Aquí va una lista de estos elementos básicos, junto con algunos ejemplos de cada uno.

enteros decimales

`'5', '32769', '-3'`

enteros hexadecimales

`'$a000', '$FB'`

enteros binarios

`'%0010010', '%11'`

cadenas

`"Hola", 'Hola'`

símbolos

`'#x', '#varbucle', 'P_funcvar'`

comentarios

`'; esto es un comentario ( )'`

`'( )'`

para la definición de órdenes

espacio (o cualquier espacio en blanco)

delimita símbolos

## 1.12 installer.guide/Caracteres de escape

Caracteres de escape  
=====

Los caracteres de escape se soportan de la misma forma que en el lenguaje C++:

`'\n'`

nueva línea

`'\r'`

retorno del carro

`'\t'`

tabulador

`'\h'`

tabulador horizontal (V42.6)

---

```

'\v'
    tabulador vertical (V42.6)

'\b'
    retroceso (V42.6)

'\f'
    salto de página (V42.6)

'\"'
    comillas "

'\''
    comilla simple '

'\'\'
    barra invertida

'\ooo'
    número octal 'ooo'. Puede utilizar '\\0' para obtener un
    carácter NUL. (V42.6)

'\xXX'
    número hexadecimal 'XX'. (V42.6)

```

NOTA: Dependiendo de la versión de AmigaGuide o de MultiView que esté utilizando, es posible que observe aquí arriba un carácter de barra invertida adicional donde sólo debería haber una. Aquí debería ver sólo una barra: '\\'. Y aquí debería ver dos: '\\\'.

## 1.13 installer.guide/Símbolos (Variables)

Símbolos (Variables)  
=====

Un símbolo es cualquier secuencia de caracteres rodeada de espacios que no es una cadena citada, un entero o un carácter de control. Esto significa que los símbolos pueden contener signos de puntuación y otros caracteres especiales. A continuación se citan algunos símbolos válidos:

```

'x'

'total'

'esto-es-un-símbolo'

'**nombre**'

'@#__#@'

```

Al nombrar variables, no debería utilizar '@' como prefijo para sus nombres; sí en cambio se recomienda emplear un prefijo como '#' para evitar colisiones en futuras versiones de Installer.

## 1.14 installer.guide/Tipos de símbolos

Tipos de símbolos  
=====

Existen tres tipos de símbolos:

- a. símbolos definidos por el usuario. Se crean empleando la orden 'set'.
- b. nombres de órdenes o funciones incorporadas de serie. Incluyen cosas como '+' y '\*', así como nombres textuales como pueden ser 'delete' o 'rename'.
- c. símbolos especiales. Se trata de variables creadas por Installer antes de que el script se ejecute realmente, y se utilizan para informar al script de ciertas cosas acerca del entorno. Estos símbolos siempre empiezan por un signo '@'. Un ejemplo sería '@default-dest', que le informa del directorio por omisión que ha sido seleccionado por Installer para instalar el producto.

Para ver las convenciones empleadas con los nombres, consulte la Guía de estilo.

## 1.15 installer.guide/Órdenes

Órdenes  
=====

El formato de una orden es:

```
(operador <operando1> <operando2> ...)
```

Una orden para asignar el valor '5' a la variable '#x' sería:

```
(set #x 5)
```

Puede leer esto como "pon '#x' a 5". Recuerde que la variable '#x' no tiene por qué ser declarada previamente; esta orden la crea automáticamente.

Tenga en cuenta que no existe diferencia entre órdenes y operadores: la orden 'set' y el operador aritmético '+' se usan exactamente de la misma forma.

Combinación de órdenes: Una orden puede ser utilizada como operando de otra orden, tal como ilustra este ejemplo:

---

```
(set #var (+ 3 5))
```

En este caso, la orden `'(+ 3 5)'` se evalúa primero, y el resultado es 8. Puede considerar esto como que la parte `'(+ 3 5)'` ha sido sustituida por un 8. Así que ahora nos queda:

```
(set #var 8)
```

que presenta el mismo formato que el primer ejemplo citado.

Dese cuenta de que la parte `'(+ 3 5)'` ha producido un valor: 8. A ésto se le llama el resultado de la orden. Muchas órdenes devuelven resultados, incluyendo algunas que podrían sorprenderle, como pueden ser `'set'` e `'if'`.

## 1.16 installer.guide/Tipos de datos

Tipos de datos

=====

En Installer, todos los tipos de datos son dinámicos, lo que equivale a decir que el tipo de una variable viene determinado por la información que contiene. Así, si asigna la cadena `"Hola, mundo"` a la variable `'#x'`, `'#x'` será una variable de tipo CADENA. Posteriormente puede asignar un entero a `'#x'`, en cuyo caso la variable pasará a ser de tipo ENTERO. Al utilizar variables dentro de las expresiones, el intérprete intentará hacer la conversión al tipo apropiado, si es posible.

Formas especiales: Existen dos excepciones en la forma de una orden. El primer tipo se utiliza para la sustitución de cadenas: si el primer elemento indicado entre paréntesis es una cadena de texto en lugar del nombre de una orden, el resultado de la cláusula será otra cadena, creada a base de tomar la cadena original y realizar sobre ella una operación de formateo del tipo `'sprintf'` al estilo C, empleando el resto de argumentos de la orden como parámetros en la operación de formateo.

Así, la orden:

```
("Me llamo %s y tengo %ld años" "María" 5)
```

Se convierte en:

```
"Me llamo María y tengo 5 años"
```

NOTA: Puesto que la operación de formateo emplea la rutina `'RawDoFmt()'` de la ROM, los valores decimales deberán expresarse siempre con `'"%ld"'` en lugar de con `'"%d"'` (el intérprete siempre pasa las cantidades numéricas como palabras largas de 32 bits). Tenga en cuenta que una variable que contenga una cadena puede ser utilizada en lugar de la cadena en sí.

El segundo tipo de excepción se produce si los elementos indicados



entre paréntesis son en sí mismos órdenes entre paréntesis. En este caso, el intérprete asume que todos los elementos son órdenes que debe ejecutar de forma secuencial.

Por ejemplo, la orden siguiente define el valor de tres variables diferentes: `#var1`, `#var2`, y `#var3`.

```
((set #var1 5) (set #var2 6) (set #var3 7))
```

Esta característica permite al lenguaje tener una estructura de bloque, donde una orden `'if'` puede tener varias órdenes en su cláusula `'then'` o `'else'`. Tenga en cuenta que el resultado de esta orden será el resultado de la última orden que compone la secuencia.

Órdenes complejas: Aquí va un ejemplo de cómo las órdenes en el lenguaje de scripts pueden combinarse para formar expresiones complejas. Comenzaremos con una orden `'if'`. El formato básico de una orden `'if'` es:

```
(if <condición> <orden-then> [<orden-else>])
```

La condición debería ser una orden que devuelva un valor. Tanto la parte `'then'` como la parte opcional `'else'` deberían ser órdenes. Recuerde que si las órdenes `'then'` o `'else'` producen un resultado, la orden `'if'` también tendrá este resultado.

Nuestro primer ejemplo es un poco extraño: utilizamos una orden `'if'` para simular un operador `'NOT'` booleano. Recuerde que dispone de formas más sencillas de hacer esto en el lenguaje de scripts.

```
(set #bandera 0) ; pone una bandera en FALSE

(set #bandera (if #bandera 0 1)) ; un NOT booleano
```

Básicamente, la orden `'if'` comprueba la variable `'bandera'`. Si `'bandera'` no vale cero, produce el valor `'0'`; de lo contrario, el resultado es `'1'`. En cualquier caso, `'bandera'` se define en función del resultado de la orden `'if'`.

Ahora, añadamos algunas órdenes reales a nuestra orden `'if'`.

```
(if #bandera ; prueba condicional
  (message "'bandera' no valía cero\n") ; cláusula "then".
  (message "'bandera' valía cero\n") ; cláusula "else".
) ; if ; cerramos paréntesis
```

Fíjese en el estilo de la justificación. Esto permite una lectura más fácil del programa. La justificación preferida es a cuatro espacios.

Ahora, añadamos una condición real. `'='` comprueba la igualdad entre dos elementos.

```
(if (= #a 2) ; prueba condicional
  (message "a vale 2\n") ; cláusula "then"
  (message "a no vale 2\n") ; cláusula "else"
) ; if ; cerramos paréntesis
```

Por último, y sólo para hacer esto más interesante, vamos a hacer de la cláusula 'else' una orden compuesta. Fíjese en el uso extendido de paréntesis para generar exactamente las dos cláusulas necesarias. ¡Installer no comprobará las cosas si Ud. mete aquí la pata!

```
(if (= #a 2)                                ; prueba condicional
  (                                           ; cláusula "then"
    (message "a vale 2\n")
  )
  (                                           ; orden compuesta "else"
    (message "a no valía 2\n")
    (set #a 2)
    (message "¡pero ahora sí!\n")
  )
) ; if                                       ; fin de la orden compuesta
                                           ; fin de if
```

## 1.17 installer.guide/Características especiales

### Características especiales

=====

Cuando Installer comienza a funcionar, intenta determinar el mejor lugar para instalar la aplicación. Cualquier volumen llamado 'WORK:' tiene preferencia, dado que ésta es la forma estándar en que el Amiga viene configurado de fábrica por Amiga Technologies. A partir de la versión V42, eso sí, también se tendrá en cuenta como destino por omisión la partición escribible con más espacio libre. Esto cubre los casos más típicos.

Existen dos atajos de teclado. Siempre que haya activo un botón 'Ayuda', pulsar la tecla <Help> también hará aparecer la ventana de ayuda. Además, siempre que haya activo un botón 'Interrumpir', pulsar <Esc> hará aparecer el requester de interrupción. También, siempre que Installer esté ocupado, pulsar <Esc> forzará la aparición del requester de interrupción; en la barra de título se muestra un texto al efecto.

Si una aplicación requiere el establecimiento de asignaciones u otras operaciones durante el arranque del sistema, Installer añadirá esto a un fichero llamado 'S/User-Startup' en el volumen de arranque. Si este fichero no existe, se intentará con 'S:User-Startup'. Installer añadirá entonces las líneas

```
if exists S:user-startup
execute S:user-startup
endif
```

al fichero 'Startup-Sequence' del usuario. Installer intentará determinar el verdadero volumen de arranque del sistema al buscar los ficheros 'Startup-Sequence' y 'User-Startup', y es capaz de manejar cualquier script AmigaDOS ejecutado desde 'Startup-Sequence' con hasta 10 niveles de anidamiento.

Installer puede crear una asignación a una unidad, volumen o

asignación lógica. Esto resulta útil cuando se quiere actualizar una aplicación que se suministra en un volumen llamado 'MiAplicación' y se da la casualidad de que la versión instalada se encuentra en un directorio con la asignación lógica 'MiAplicación:'.

Installer siempre copia ficheros en el modo 'CLONE' del AmigaDOS, lo que significa que todas las banderas de protección, comentarios y fechas de última modificación quedan preservados. Al copiar ficheros, Installer presenta una lectura del tipo "indicador de progreso" para mostrar el progreso del proceso de copia.

Installer puede encontrar el número de versión de cualquier fichero ejecutable que tenga o bien una RomTag con una cadena de IDentificación (como es el caso de las librerías y los drivers) o bien una cadena de versión conforme al estándar dado en la Guía de Estilo para Interfaz de Usuario Amiga. Installer también soporta la suma de comprobación de ficheros (también conocida como "checksum").

## 1.18 installer.guide/Miscelánea

Miscelánea

=====

Para realizar una serie de operaciones con todo el contenido de un directorio que responda a un determinado patrón, puede utilizar el operador 'foreach'. Para llevar a cabo una serie de operaciones sobre un grupo específico de ficheros, las siguientes órdenes de Installer pueden utilizarse a modo de plantilla:

```
(set #n 0)
(while (set estefichero (select #n "fich1" "fich2" "fich3" ""))
(
  (set #n (+ #n 1))
  (... sus operaciones con este fichero ...)
)
) ; fin de while
```

Recuerde que una cadena vacía es considerada como valor FALSE por cualquier operador condicional.

Para lanzar un comando Shell externo que normalmente requiere la participación del usuario, redirija la entrada desde un fichero con las respuestas necesarias. Por ejemplo, para formatear un disco, podría combinarse la orden mencionada a continuación con un fichero que contenga sólo un carácter de salto de línea.

```
(run "format <fichero_sl drive DF0: name Vacío")
```

## 1.19 installer.guide/Referencia del Lenguaje de Installer

## Referencia del Lenguaje de Installer

\*\*\*\*\*

## NOTAS MUY IMPORTANTES

Órdenes

Órdenes de Control

Órdenes de Depuración

Procedimientos definidos por el Usuario

Funciones

Sumario de Parámetros

Variables Predefinidas

## 1.20 installer.guide/NOTAS MUY IMPORTANTES

## Notas Muy Importantes

=====

- a. 'diskimage' es un nombre reservado (V42). No intente usar este nombre.  
¡Por favor, siga leyendo!
  - b. A partir de la V42, existen una serie de normas razonables a seguir a la hora de nombrar cosas. Por favor, adhiérase a ellas si no quiere que la actualización de Installer por parte del usuario resulte un quebradero de cabeza. No utilice el signo '@' al nombrar sus variables. Este signo marca variables de sistema de Installer. Por favor, utilice un prefijo en los nombres de sus variables, como '#', y nombre sus procedimientos cuidadosamente empleando un prefijo como 'P\_'. Los nombres que carezcan de un prefijo podrían ser una fuente de problemas en futuras versiones de Installer.
  - c. Recuerde que Installer no soporta variables locales.
  - d. Cuando el script finaliza -ya sea por llegar al final, o por alcanzar una orden 'exit'-, se muestra un mensaje informando del lugar donde se ha instalado la aplicación y dónde se ha guardado el fichero de registro, si lo hay. Recuerde que debe indicar en '@default-dest' el lugar donde ha instalado realmente la aplicación.
  - e. Un carácter de salto de línea (\n, '0x0a') producirá una ruptura de línea cuando Installer realice el corte de palabras. Un carácter de espacio duro (ALT-espacio, '0xa0') impedirá una ruptura de línea en ese caso. Además, las partes entrecomilladas serán consideradas como una sola palabra por lo que respecta al corte de palabras. Por ejemplo, si se utiliza el texto de ayuda siguiente:  
  
"El disco \"FrameZapper 2.0\" es necesario para la instalación."  
  
el texto "FrameZapper 2.0" no sufrirá un corte de palabra antes del "2".
  - f. El tamaño máximo de una cadena dentro de un script es de 512 bytes.
-

El tamaño máximo de cualquier variable de cadena es de 10000 bytes. Si necesita crear por ejemplo un texto de ayuda muy largo, divídalo en bloques de 512 bytes, y entonces utilice la función de concatenación automática de cadenas de Installer para crear la gran cadena final. Tampoco pase por alto la división en varias líneas de sus cadenas dentro de los scripts, para hacer que éstos resulten más legibles. Si alguna vez se encuentra con que Installer informa de un error de rebosamiento de la pila, compruebe si ha sido causado por la concatenación de demasiadas cadenas de pequeño tamaño, y júntelas en bloques más grandes.

- g. Los patrones con comodines para nombres de ficheros y directorios dentro de un script no deben exceder los 64 caracteres de largo.

## 1.21 installer.guide/Órdenes

### Órdenes

=====

#### Gestión de variables

```
'set'  
'symbolset'    (V42.9)
```

#### Ayuda con los ficheros

```
'mkdir'  
'copyfiles'  
'copylib'  
'startup'  
'tooltype'  
'textfile'
```

#### Ejecución de partes externas a la instalación

```
'execute'  
'run'  
'rexx'
```

#### Más ayuda con los ficheros

```
'makeassign'  
'rename'  
'delete'  
'protect'
```

#### Informar al usuario

---

```
`complete'
`message'
`working'
`welcome'
```

Multimedia

```
`effect'      (V44)
`showmedia'   (V44)
`setmedia'    (V44)
`closemedia'  (V44)
```

## 1.22 installer.guide/set

La orden `'set'`

-----

```
(set <variable> <valor> [<variable2> <valor2> ...])
```

Da a la `'<variable>'` el valor indicado. Si `'<variable>'` no existe todavía, se creará en ese momento. La orden `'set'` devuelve el valor de la última asignación.

NOTA: Todas las variables carecen de tipo definido, y cualquier variable puede ser utilizada dondequiera que pudiera emplearse una cadena. Todas las variables son globales. Para las convenciones con los nombres, por favor, consulte la Guía de Estilo.

La orden `'set'` puede utilizarse para convertir una cadena en un valor entero:

```
(set <valor-entero> (+ <variable-de-cadena>))
```

Para hacer la operación inversa, utilice `'cat'`.

## 1.23 installer.guide/symbolset

La orden `'symbolset'` (V42.9)

-----

```
(symbolset <símbolo> <valor> [<símbolo2> <valor2> ...])
```

Da a la variable nombrada por el contenido de la variable de cadena o expresión `'<símbolo>'` el valor indicado. Si la variable a la que `'<símbolo>'` hace referencia no existe todavía, se creará en ese momento. La orden `'symbolset'` devuelve el valor de la última asignación.

En combinación con la función `'symbolval'`, esta orden está pensada para permitir cierto tipo de manejo dinámico de los nombres de variable, por ejemplo para crear variables con un índice arbitrario en el nombre.

NOTA: Debería leer la descripción de `'set'`, puesto que el resto de la semántica es la misma.

## 1.24 installer.guide/makedir

La orden `'makedir'`

-----

(makedir <nombre> <parámetros>)

Crea un nuevo directorio. A partir de Installer V42.9, esta orden intentará crear la senda completa, con todos los directorios intermedios que se indiquen. Parámetros:

`'prompt'`

dice al usuario lo que va a ocurrir

`'help'`

texto del mensaje de ayuda

`'infos'`

crea un icono para el directorio

`'confirm'`

si está presente esta opción, se consultará al usuario; de lo contrario, se creará el directorio sin más dilación.

`'safe'`

crea el directorio, aunque el script corra en modo PRETEND

## 1.25 installer.guide/copyfiles

La orden `'copyfiles'`

-----

(copyfiles <parámetros>)

Copia uno o más ficheros del disco de instalación al directorio destino. Cada fichero se mostrará con un botón de marca al lado de su nombre para poder indicar si el fichero se va a copiar o no. Recuerde que un fichero protegido contra escritura también se considera como "protegido contra borrado". Parámetros:

`'prompt'`  
`'help'`  
dicen al usuario lo que va a ocurrir

`'source'`  
nombre del directorio o fichero origen

`'dest'`  
nombre del directorio destino, que se creará automáticamente en caso de que no exista todavía. Recuerde que tanto origen como destino pueden ser una senda relativa.

`'newname'`  
si se va a copiar solamente un fichero, y el fichero va a cambiar de nombre, éste será el nuevo nombre

`'choices'`  
una lista de los ficheros/directorios a copiar (opcional)

`'all'`  
todos los ficheros/directorios del directorio origen deberán copiarse

`'pattern'`  
indica los ficheros/directorios a copiar del directorio origen, por medio de un patrón. El patrón no deberá exceder los 64 caracteres de largo. Recuerde que sólo puede utilizar a un tiempo uno de los parámetros `'choices'`, `'all'` o `'pattern'`.

`'files'`  
copiar solamente ficheros. Por omisión, Installer también copia los subdirectorios.

`'infos'`  
interruptor que permite copiar los iconos asociados con otros ficheros/directorios

`'noposition'`  
reinicializar ("descongelar") la posición de cada icono copiado

`'fonts'`  
interruptor para no mostrar los ficheros `'.font'`, pero aún así copiar cualquiera que coincida en un directorio que se va a copiar

`'nogauge'`  
no mostrar el indicador de progreso

`'(optional <opción> <opción> ...)'`  
dicta lo que será considerado como un fallo en la copia. Las primeras tres opciones son mutuamente exclusivas, es decir, sólo puede utilizar una de ellas a un tiempo.

`'fail'`  
Installer interrumpe la instalación si no puede copiar algo (valor por omisión)

---



``nofail'`

Installer seguirá adelante aunque no haya podido copiar algo

``okndelete'`

interrumpe la instalación si no puede copiar, a menos que el motivo sea "protección contra borrado"

Las siguientes dos opciones pueden utilizarse en combinación con cualquier otra opción del grupo `'optional'`.

``force'`

desproteger el objeto destino

``askuser'`

preguntar al usuario si desea que el fichero sea desprotegido (pero no en modo NOVICE). Al emplear `'askuser'`, el valor por omisión para el modo NOVICE es una respuesta negativa; por lo tanto, tal vez le interese utilizar `'force'` para hacer que la respuesta por omisión en modo NOVICE sea afirmativa.

``(delopts <opción> <opción> ...)'`

elimina las opciones definidas por `'optional'`

``confirm'`

si está presente esta opción, se pedirá al usuario que indique qué ficheros desea copiar; de lo contrario, los ficheros se copiarán sin más.

``safe'`

copiar ficheros incluso en modo PRETEND

``compression'`

si está activada esta bandera, todos los ficheros origen deberán haber sido comprimidos con el algoritmo LZW `'compress'` de Un\*x, versión 3.0 o superior. Los ficheros serán descomprimidos de forma automática durante el proceso de copia al destino. Tenga en cuenta que la compresión con más de 13 bits requiere una cantidad notable de memoria cuando se descomprime un fichero. El número máximo soportado de bits es de 16. ¡Esto necesitaría casi 500KB de memoria al descomprimir! Haga una sabia elección con el número máximo de bits al comprimir un fichero, de lo contrario la descompresión podría fallar por falta de memoria.

AVISO: La implantación de la compresión sigue estando todavía en un estado experimental y en fase alfa. Puede jugar con ella y enviar informes de fallos. ¡NO UTILICE ESTO EN SU SCRIPT FINAL!

## 1.26 installer.guide/copylib

La orden ``copylib'`

-----

(copylib <parámetros>)

Copia un fichero empleando comprobación de versión, es decir, sólo sobrescribe el fichero destino si el nuevo fichero cuenta con un número de versión/revisión superior. La orden 'copylib' creará el directorio destino siempre que sólo falte un nivel. Por ejemplo, al copiar a un inexistente directorio 'DEVS:midi' se crearía el directorio 'midi', pero si intenta copiar a 'DEVS:midi/extras' cuando ni 'midi' ni 'extras' existen, se produciría un fallo. Recuerde que una librería protegida contra escritura es considerada también como "protegida contra borrado". Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'confirm'

si está presente esta opción, se pedirá al usuario que confirme la operación de copia; de lo contrario, los ficheros se copiarán sin más. Recuerde que en el modo EXPERT, el usuario podrá sobrescribir un fichero más reciente con otro más antiguo.

'safe'

copiar ficheros incluso en modo PRETEND

'source'

nombre del fichero origen

'dest'

nombre del directorio destino, que se creará automáticamente si no existe todavía. Recuerde que tanto origen como destino pueden ser una senda relativa.

'newname'

si el fichero va a ser cambiado de nombre, éste será el nuevo nombre

'infos'

interruptor para copiar también el icono

'noposition'

reinicializar ("descongelar") la posición de todo icono copiado

'nogauge'

no mostrar el indicador de progreso

'(optional <opción> <opción> ...)'

dicta lo que será considerado como un fallo en la copia. Las primeras tres opciones son mutuamente exclusivas, es decir, sólo puede utilizar una de ellas a un tiempo.

'fail'

Installer interrumpirá la instalación si no puede copiar algo (valor por omisión)

'nofail'

Installer seguirá adelante aunque no pueda copiar algo

'oknodelete'

interrumpe la instalación si no puede copiar, a menos que el

---

motivo sea "protección contra borrado"

Las siguientes dos opciones pueden utilizarse en combinación con cualquier otra opción del grupo 'optional'.

'force'

desproteger el objeto destino

'askuser'

preguntar al usuario si desea que el fichero sea desprotegido (pero no en modo NOVICE). Al emplear 'askuser', el valor por omisión para el modo NOVICE es una respuesta negativa; por lo tanto, tal vez le interese utilizar 'force' para hacer que la respuesta por omisión en modo NOVICE sea afirmativa.

'(delopts <opción> <opción> ...)'

elimina las opciones definidas por 'optional'

## 1.27 installer.guide/startup

La orden 'startup'

-----

(startup <aplicación> <parámetros>)

Este comando edita el fichero 'S:User-Startup', que es ejecutado por el fichero 'Startup-Sequence' del usuario (Installer modificará el fichero 'Startup-Sequence' si es necesario, aunque de una forma amistosa). El parámetro 'command' se utiliza para declarar líneas de comando AmigaDOS que serán ejecutadas. Las líneas de comando se agrupan por aplicación, empleando el argumento indicado en 'aplicación'. Si ya existe información en 'S:User-Startup' para esa aplicación, las nuevas líneas de comando sustituirán completamente a las antiguas.

NOTA: Los parámetros 'prompt' y 'help' de la orden 'startup' se utilizan solamente en la ventana de confirmación a la hora de editar el fichero 'User-Startup'. Esto sólo ocurre en los modos AVERAGE y EXPERT, o en modo NOVICE en caso de que se incluya el parámetro 'confirm'.

Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'confirm'

si está presente esta opción, se pedirá al usuario que confirme la operación; de lo contrario, la modificación tendrá efecto inmediatamente

'command'

---

se utiliza para declarar una línea de comando AmigaDOS a ejecutar en el momento del arranque del sistema

Tenga en cuenta que Installer procurará encontrar el fichero User-Startup en el directorio de scripts del volumen de arranque. Sólo en caso de que no se encuentre allí se recurrirá a la asignación 'S:'.

## 1.28 installer.guide/tooltype

La orden 'tooltype'

-----

(tooltype <parámetros>)

Modifica los tipos de herramienta -y más cosas- de un icono. Normalmente, los nuevos valores habrán sido indicados con anterioridad por medio de varias órdenes en el script de instalación, es decir, el usuario no tiene realmente que teclear los nuevos valores. Por ejemplo, podría usar un 'askchoice' para preguntar al usuario por la resolución de pantalla que desea y entonces definir un tipo de herramienta de acuerdo a la opción indicada. El trabajo de 'tooltype' consiste simplemente en pedir una confirmación antes de hacer la escritura en sí.

Tenga en cuenta que el parámetro 'settooltype' puede ir acompañado de uno o dos argumentos, con el formato general '(settooltype "tipo de herramienta")' o '(settooltype "tipo de herramienta" "valor")'. El primer caso se utiliza para borrar un tipo de herramienta de un icono. El segundo dará un nuevo argumento a un tipo de herramienta nuevo o ya existente. El nombre del tipo de herramienta se pasará siempre a letras mayúsculas. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'dest'

el nombre del icono a modificar. No hay necesidad de indicar una extensión '.info'

'settooltype'

cadena con el nombre del tipo de herramienta y su valor

'setdefaulttool'

nombre de la herramienta por omisión para un proyecto

'setstack'

define el tamaño de la pila

'noposition'

reinicializar la posición del icono a NOICONPOSITION (sin posición definida). Tenga cuidado y no manipule arbitrariamente la posición de los iconos del usuario al hacer una actualización.

---

``setposition (V42.12)'`  
dos valores enteros que indican la nueva posición del icono en las coordenadas X e Y. No tome esto a la ligera. Está pensado para mantener las posiciones de los iconos durante las actualizaciones con ayuda de la nueva función ``iconinfo'`. Cambiar arbitrariamente las posiciones de los iconos sólo le deparará usuarios enfadados, al emplear diferentes configuraciones y tipos de letra en su Workbench.

``swapcolors'`  
¡OBSOLETO! ¡NO UTILICE ESTO!

``confirm'`  
si está presente esta opción, se pedirá confirmación al usuario; de lo contrario, la modificación tendrá lugar sin más dilación

``safe'`  
hace los cambios incluso en modo PRETEND

## 1.29 installer.guide/textfile

La orden ``textfile'`

-----

(textfile <parámetros>)

Crea un fichero de texto a partir de otros ficheros de texto o cadenas de texto suministradas. Esto puede utilizarse para crear ficheros de configuración, programas ARexx o scripts AmigaDOS. Parámetros:

``prompt'`  
``help'`  
dicen al usuario lo que va a ocurrir

``dest'`  
el nombre del fichero de texto que se va a crear

``append'`  
una cadena que será añadida al nuevo fichero de texto

``include'`  
un fichero de texto que se va a añadir al nuevo fichero. Si el fichero indicado no existe, será tratado como un fichero vacío e ignorado completamente.

``confirm'`  
si está presente esta opción, se pedirá confirmación al usuario; de lo contrario, la escritura será efectuada sin más dilación

``safe'`  
crea el fichero incluso en modo PRETEND

### 1.30 installer.guide/execute

La orden 'execute'

-----

(execute <argumento> ...)

Ejecuta un script AmigaDOS con los argumentos dados. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'confirm'

si está presente esta opción, se pedirá confirmación al usuario;  
de lo contrario, la ejecución será efectuada sin más dilación

'safe'

ejecutar el script incluso en modo PRETEND

Esta orden devuelve directamente el resultado primario del script,  
y a partir de Installer V42, el resultado secundario en la variable  
'@ioerr'.

### 1.31 installer.guide/run

La orden 'run'

-----

(run <argumento> ...)

Ejecuta un programa compilado con los argumentos dados. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'confirm'

si está presente esta opción, se pedirá confirmación al usuario;  
de lo contrario, la ejecución será efectuada sin más dilación

'safe'

ejecutar el guión incluso en modo PRETEND

Esta orden devuelve directamente el resultado primario del programa,  
y a partir de Installer V42, el resultado secundario en la variable  
'@ioerr'.

---

### 1.32 installer.guide/rexx

La orden 'rexx'  
-----

(rexx <argumento> ...)

Ejecuta un script ARexx con los argumentos dados. Si el servidor ARexx no ha sido lanzado todavía, se generará un error. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'confirm'

si está presente esta opción, se pedirá confirmación al usuario;  
de lo contrario, el script ARexx se ejecutará sin más dilación

'safe'

ejecutar el script incluso en modo PRETEND

Esta orden devuelve directamente el resultado primario del script, y a partir de Installer V42, el resultado secundario en la variable '@ioerr'.

### 1.33 installer.guide/makeassign

La orden 'makeassign'  
-----

(makeassign <asignación> [<senda>] (parámetros))

Crea la '<asignación>' a la '<senda>'. Si no se indica '<senda>', la asignación será eliminada. Parámetros:

safe

(des)hacer la asignación incluso en modo PRETEND

NOTA: '<asignación>' debe indicarse sin ningún signo de dos puntos; es decir, por ejemplo '"ENV"' y no '"ENV:"'.

### 1.34 installer.guide/rename

La orden 'rename'  
-----

(rename <nombre antiguo> <nuevo nombre> <parámetros>)

---

Renombra un fichero o directorio. Si se indica el parámetro 'disk', entonces el comando cambiará la etiqueta del disco al que hace referencia '<nombre antiguo>'. Al cambiar la etiqueta de un disco, incluya el signo de dos puntos sólo al indicar el nombre antiguo. Devuelve 1 si la operación ha finalizado con éxito, o 0 en caso de que haya fallado. Parámetros:

```
'prompt'
'help'
    dicen al usuario lo que va a ocurrir

'confirm'
    si está presente esta opción, se pedirá confirmación al usuario;
    de lo contrario, la operación de cambio de nombre será efectuada
    sin más dilación

'disk'
    interruptor para producir el cambio de etiqueta de un disco

'safe'
    cambiar el nombre incluso en modo PRETEND
```

### 1.35 installer.guide/delete

La orden 'delete'

-----

```
(delete <fichero> <parámetros>)
```

Borra un fichero. Tenga en cuenta que un fichero protegido contra escritura se considera también "protegido contra borrado". A partir de Installer 42.9, puede indicar un patrón AmigaDOS y la opción 'all'. Parámetros:

```
'prompt'
'help'
    dicen al usuario lo que va a ocurrir

'confirm'
    si está presente esta opción, se pedirá confirmación al usuario;
    de lo contrario, el borrado tendrá lugar sin más dilación

'(optional <opción> <opción> ...)'
    decide si se forzará el borrado. Opciones:

    '"force"'
        desproteger el destino

    '"askuser"'
        preguntar al usuario si desea desproteger el fichero (pero
        no en modo NOVICE)

    En el caso de '"askuser"', el valor por omisión en el modo NOVICE
    es una respuesta negativa; por lo tanto, tal vez le interese usar
    '"force"' para hacer que la respuesta por omisión en el modo NOVICE
```



sea afirmativa.

```
'(delopts <opción> <opción> ...)'
    elimina las opciones definidas por 'optional'

'safe'
    borrar incluso en modo PRETEND

'infos'
    borrar también los iconos correspondientes. No utilice esta
    opción junto con 'all'.

'all'
    comprobar también todos los subdirectorios que coincidan (V42.9)
```

NOTA: Por favor, utilice con precaución las funciones de alusión por patrón y de comprobación de subdirectorios. El usuario no podrá interrumpir el comando mientras éste está ejecutándose. Podría ser una mala idea borrar una gran cantidad de información en un sólo paso.

## 1.36 installer.guide/protect

La orden 'protect'

-----

```
(protect <fichero> [<cadena de banderas a cambiar>] [<máscara
                    decimal>] <parámetros>)
```

Devuelve el estado de protección de un fichero (en caso de que no se indique un segundo argumento), o bien lo define. Existen dos métodos para la definición del estado: cadena (por ejemplo, '"+rgr -wgr +e -dgd"') o numérico (por ejemplo, 5). El método de la cadena permite cambiar cualquiera de las banderas de forma individual, mientras que el método numérico escribe todas las banderas a la vez (posiblemente cambiando banderas involuntariamente). Los bits en la representación binaria de la máscara decimal se corresponden con las banderas, de la forma siguiente:

```
8 7 6 5 4 3 2 1  <- número de bit

h s p a r w e d  <- bandera de protección correspondiente

^ ^ ^ ^ ^ ^ ^ ^
| | | | | | | |
| | | | | | | +- \
| | | | | | | +--- | 0 = bandera activada
| | | | | | | +----- | 1 = bandera desactivada
| | | | | | | +----- /
| | | |
| | | |
| | | +----- \
| | +----- | 0 = bandera desactivada
| +----- | 1 = bandera activada
```

```
+----- /
```

Recuerde que el significado de los bits en el valor numérico sigue la convención del DOS en la que un 1 en los cuatro bits más altos (banderas "hspa") indica que la bandera se activa, mientras que un 1 en los bits más bajos (banderas "rwed") indica que la bandera se desactiva. Puede utilizar los modificadores '"g"' y '"o"' para definir las banderas de grupo y otras. Estos modificadores son válidos para todo lo que siga inmediatamente a las banderas '"rwed"'.

Por favor, evite usar la bandera '"h"'. Su significado está por definir y podría ser reutilizada en el futuro por Amiga Technologies.

Al definir banderas, 'protect' devuelve 1 si la operación ha tenido éxito, o 0 si no es así. La obtención de las banderas devuelve el valor numérico del estado de protección (vea más arriba la forma de interpretarlo), o bien -1 en caso de fallo. Parámetros:

'safe'

cambiar la protección incluso en modo PRETEND

## 1.37 installer.guide/complete

La orden 'complete'

```
-----
```

```
(complete <número>)
```

Esta orden se utiliza para informar al usuario de cuán adelantado se encuentra el proceso de instalación. El número (que debe estar comprendido entre 0 y 100) se imprimirá en la barra de título de la ventana de Installer junto con un signo '%'.

## 1.38 installer.guide/message

La orden 'message'

```
-----
```

```
(message <cadena> <cadena> ... <parámetros>)
```

Esta orden muestra al usuario un mensaje en una ventana, junto con los botones 'Continuar', 'Interrumpir' y opcionalmente 'Ayuda'. Tenga en cuenta que estos mensajes no se muestran en el nivel de usuario 0 (NOVICE) a menos que se emplee 'all'. Parámetros:

'all'

Si indica este parámetro opcional, el mensaje se mostrará para todos los niveles de usuario (V42.4). Use este parámetro de forma inteligente. No confunda con mensajes a un usuario novato.

### 1.39 installer.guide/working

La orden 'working'

-----

```
(working <cadena> <cadena> ...)
```

Las cadenas se concatenarán para formar un mensaje que se mostrará bajo una línea estándar que reza "Instalando...". Resulta útil en caso de que esté haciendo una larga operación distinta de la copia de ficheros (que cuenta con su propio indicador de progreso).

### 1.40 installer.guide/welcome

La orden 'welcome'

-----

```
(welcome <cadena> <cadena> ...)
```

Durante la compilación, Installer busca la presencia de esta orden en el script de instalación. Si no está presente (como es el caso de los scripts más antiguos), se presentará al usuario la ventana 'Bienvenido a la instalación de <APLICACIÓN>' tan pronto como la compilación haya finalizado. Si está presente esta orden, Installer no hará aparecer la ventana 'Bienvenido...' hasta llegar a la orden 'welcome'. Esto permite la ejecución de código antes de que aparezcan las primeras ventanas. Tenga en cuenta que el estado de las variables '@user-level' y '@pretend' estará basado en los valores por omisión iniciales, incluyendo cualquier modificación hecha desde los tipos de herramienta. Los argumentos de cadena preceden al texto de ayuda estándar para cualquiera de las dos ventanas iniciales que aparezca primero.

### 1.41 installer.guide/effect

La orden 'effect' (V44)

-----

```
(effect <posición> <efecto> <color 1> <color 2>)
```

La orden 'effect' es especial porque se interpreta antes que todos los demás comandos del script, independientemente del lugar donde se encuentre en él. Sus argumentos deben ser valores literales (constantes de cadena y de entero), y no pueden ser evaluados.

Al utilizar esta orden, Installer abre una nueva pantalla que es un clon de la pantalla pública por omisión (la del Workbench).

'<posición>' define la posición de la ventana de Installer, y deberá ser una de las siguientes cadenas:

`'upper_left'`

La ventana se coloca en la esquina superior izquierda de la pantalla.

`'upper_center'`

La ventana se coloca centrada horizontalmente en la parte superior de la pantalla.

`'upper_right'`

La ventana se coloca en la esquina superior derecha de la pantalla.

`'center_left'`

La ventana se coloca centrada verticalmente en el lado izquierdo de la pantalla.

`'center'`

La ventana se coloca en el centro de la pantalla.

`'center_right'`

La ventana se coloca centrada verticalmente en el lado derecho de la pantalla.

`'lower_left'`

La ventana se coloca en la esquina inferior izquierda de la pantalla.

`'lower_center'`

La ventana se coloca centrada horizontalmente en la parte inferior de la pantalla.

`'lower_right'`

La ventana se coloca en la esquina inferior derecha de la pantalla.

`'<efecto>'` define el efecto gráfico que se utilizará para generar el fondo. Puede ser una de las siguientes cadenas:

`'horizontal'`

Se dibuja un degradado de colores con líneas horizontales que van del `'<color 1>'` al `'<color 2>'`.

`'radial'`

Se dibuja un degradado de colores a base de círculos concéntricos que van del `'<color 1>'` al `'<color 2>'` y que parten de un punto situado en la esquina superior derecha de la pantalla.

Si el efecto definido no puede generarse correctamente debido al escaso número de colores de la pantalla, se empleará un efecto que utilice menos colores. El efecto radial (sólo posible en pantallas a color real) es sustituido por un degradado horizontal. Si el efecto horizontal no puede mostrarse, será sustituido por un degradado animado.

`'<color 1>'` es un valor entero e indica el valor RGB de 24 bits empleado para el primer color. Lo típico es emplear un número en base hexadecimal.

`'<color 2>'` es un valor entero e indica el valor RGB de 24 bits empleado para el segundo color. Todos los efectos crean una gama

---

cromática que va del primer color al segundo.

## 1.42 installer.guide/showmedia

La orden 'showmedia' (V44)

-----

(showmedia <nombre> <fichero> <posición> <tamaño> <marco> ...)

'showmedia' abre un fichero por medio de datatypes (requiere el AmigaOS 3.0 o superior) y lo presenta al usuario. Exceptuando el caso de ficheros de música y sonido, se abre una ventana con una posición y un tamaño determinados para mostrar el contenido del fichero. Si el fichero no puede abrirse, se devuelve un 0; de lo contrario, si el fichero llega a abrirse, se devuelve un 1.

'<nombre>' es un valor de cadena. El valor se utiliza como nombre de una variable que queda asociada al fichero. Esta variable tiene que utilizarse en el resto de órdenes de este tipo.

'<fichero>' es el nombre del fichero a mostrar.

'<posición>' es una de las siguientes cadenas:

'upper\_left'

La ventana se coloca en la esquina superior izquierda de la pantalla.

'upper\_center'

La ventana se coloca centrada horizontalmente en la parte superior de la pantalla.

'upper\_right'

La ventana se coloca en la esquina superior derecha de la pantalla.

'center\_left'

La ventana se coloca centrada verticalmente en el lado izquierdo de la pantalla.

'center'

La ventana se coloca en el centro de la pantalla.

'center\_right'

La ventana se coloca centrada verticalmente en el lado derecho de la pantalla.

'lower\_left'

La ventana se coloca en la esquina inferior izquierda de la pantalla.

'lower\_center'

La ventana se coloca centrada horizontalmente en la parte inferior de la pantalla.

---

`'lower_right'`

La ventana se coloca en la esquina inferior derecha de la pantalla.

`'<tamaño>'` es una de las siguientes cadenas:

`'none'`

La ventana toma su tamaño del fichero que se está abriendo (por ejemplo, el tamaño de la imagen).

`'small'`

La ventana es de pequeñas dimensiones.

`'small_medium'`

La ventana tendrá una anchura pequeña y una altura media.

`'small_large'`

La ventana tendrá una anchura pequeña y una gran altura.

`'medium_small'`

La ventana tendrá una anchura media y una pequeña altura.

`'medium'`

La ventana tendrá unas dimensiones medias.

`'medium_large'`

La ventana tendrá una anchura media y una gran altura.

`'large_small'`

La ventana tendrá una gran anchura y una pequeña altura.

`'large_medium'`

La ventana tendrá una gran anchura y una altura media.

`'large'`

La ventana tendrá grandes dimensiones.

Si `'<marco>'` es 1, la ventana dispondrá de un marco con flechas y deslizadores proporcionales, así como de un botón de cambio de tamaño (preferido para ficheros AmigaGuide y de texto). Si `'<marco>'` es 0, la ventana no tendrá marco alguno (valor preferido para imágenes y animaciones).

A continuación del valor del marco, pueden indicarse más cadenas, que definirán ciertos atributos del fichero abierto:

`'wordwrap'`

Se mostrará un texto empleando el corte de palabras.

`'panel'`

Si el datatype cuenta con un panel de control, se utilizará (ficheros AmigaGuide y de animación).

`'play'`

El fichero comenzará a reproducirse inmediatamente (animaciones).

---

``repeat'`

El fichero se repetirá indefinidamente una vez llegue a su final (esto parece no estar soportado por ningún datatype).

## 1.43 installer.guide/setmedia

La orden ``setmedia'` (V44)

-----

(setmedia <nombre> <operación> [parámetro])

Por medio de la orden ``setmedia'` puede realizarse una operación con un fichero abierto mediante datatypes.

`'<nombre>'` es la variable que se ha definido con la orden ``showmedia'` para aludir a este fichero. `'<operación>'` es una de las siguientes cadenas:

``pause'`

Hace una pausa en la reproducción

``play'`

Inicia la reproducción

``contents'`

Mostrar el contenido

``index'`

Mostrar el índice

``retrace'`

Volver al nodo visitado anteriormente

``browser_prev'`

Ir al nodo anterior

``browser_next'`

Ir al nodo siguiente

``command'`

Enviar un comando al datatype. El `'<parámetro>'` indica el comando del que se trata.

``rewind'`

Ir al comienzo

``fastforward'`

Avance rápido

``stop'`

Detiene la reproducción

``locate'`

Localiza una posición. El `'<parámetro>'` es un valor entero (número

de fotograma, por ejemplo).

## 1.44 installer.guide/closemedia

La orden 'closemedia' (V44)

-----  
(closemedia <nombre>)

Cierra un fichero abierto por medio de datatypes y lo elimina de la memoria.

## 1.45 installer.guide/Órdenes de Control

Órdenes de Control

=====

NOTA: Pueden utilizarse cadenas como resultado de una expresión a prueba. Una cadena vacía se considera un valor FALSE, mientras que todas las demás se consideran como TRUE.

Cambio condicional, control de flujo

'if'  
'select'

Construcción de bucles

'while'  
'until'  
'foreach'

Flujo lineal de órdenes

'Secuencia de órdenes'

Finalización de un script

'abort'  
'exit'

Manejo de errores

'trap'  
'onerror'

---



Mecanismo de vuelta atrás

```
`trace'      (V44)
`retrace'    (V44)
```

## 1.46 installer.guide/if

La orden ``if'`

-----

```
(if <expresión> <orden TRUE> <orden FALSE>)
```

Funciona como una orden estándar `'if'-'then'`.

## 1.47 installer.guide/select

La orden ``select'`

-----

```
(select <n> <elemento1> <elemento2> ...)
```

Sólo se evaluará el elemento seleccionado. De esta manera, puede utilizarse `'select'` como una construcción selectiva.

## 1.48 installer.guide/while

La orden ``while'`

-----

```
(while <expresión> <orden> ... )
```

Funciona como una orden `"do-while"` estándar.

## 1.49 installer.guide/until

La orden ``until'`

-----

```
(until <expresión> <orden> ... )
```

Funciona como una orden `"do-until"` estándar.

---

## 1.50 installer.guide/foreach

La orden 'foreach'

-----

```
(foreach <directorio> <patrón> <orden>)
```

Por cada fichero o directorio aludido por el '<patrón>' y que se encuentre dentro del '<directorio>' indicado, se ejecutará la '<orden>'. Las variables especiales '@each-name' y '@each-type' contendrán el nombre del fichero y el tipo de objeto DOS, respectivamente (por "objeto DOS" queremos decir el mismo valor que se encontraría con 'fib\_DirEntryType' en caso de que se examinase el objeto con 'Examine()'). Valores negativos de tipo de objeto DOS indican ficheros; valores positivos representan directorios. Los patrones especificados en un script no pueden exceder los 64 caracteres de largo.

## 1.51 installer.guide/secuencia

Secuencia de órdenes

-----

```
((...) (...) (...))
```

Ejecuta una secuencia de órdenes. Las órdenes entre paréntesis serán ejecutadas consecutivamente una tras otra. Esto no es necesario en el nivel más alto. Si la secuencia de órdenes es un argumento para una orden o función, el valor de la secuencia será el valor de la última orden ejecutada en la secuencia.

## 1.52 installer.guide/abort

La orden 'abort'

-----

```
(abort <mensaje> <mensaje> ...)
```

Fuerza la salida del proceso de instalación mostrando los mensajes indicados, y a continuación procesa las órdenes 'onerror' (si las hay).

## 1.53 installer.guide/exit

La orden 'exit'

-----

```
(exit <cadena> <cadena> ... (quiet))
```

Motiva la finalización normal de un guión. Si se indican cadenas, se mostrarán. A continuación se presenta el mensaje "instalación concluida". Las órdenes 'onerror' no se ejecutan. Si se indica el interruptor '(quiet)', se prescindirá de la ventana informativa final.

## 1.54 installer.guide/trap

La orden 'trap'

-----

```
(trap <banderas> <órdenes>)
```

Se utiliza para atrapar errores. Funciona de forma muy similar a la orden 'longjmp' de C, es decir, cuando se produce un error, el control pasa a la orden situada inmediatamente a continuación de 'trap'. '<banderas>' determina qué errores se atraparán. La orden 'trap' en sí devuelve el tipo de error, o bien 0 si no se ha producido error alguno. Los actuales valores de tipo de error son:

```
1 == interrupción causada por el usuario
2 == memoria agotada
3 == error en el script
4 == error DOS (véase '@ioerr')
5 == información de parámetros inapropiada
```

## 1.55 installer.guide/onerror

La orden 'onerror'

-----

```
(onerror <órdenes>)
```

Cuando se produce un error fatal que no ha sido atrapado, pueden invocarse una serie de órdenes para hacer limpieza después del script. Estas órdenes se indican previamente empleando la construcción 'onerror'. Recuerde que 'onerror' puede utilizarse varias veces para permitir una terminación sensible al contexto.

## 1.56 installer.guide/trace

La orden 'trace' (V44)

-----

(trace)

La orden 'trace' fija una posición de vuelta atrás. Un "repaso" forzará la continuación a partir de esa posición en el guión. Puede fijar tantas posiciones de vuelta atrás como desee.

Ejemplo:

```
(trace) (message "fijada una posición") (trace) (message "fijada una
segunda posición") (retrace)
```

Este ejemplo correrá hasta la orden (retrace) y continuará de nuevo a partir de la primera orden (trace), es decir, se salta una posición y se utiliza la anterior. Esto es un bucle sin fin que presentará al usuario un número infinito de veces ambas páginas de mensajes.

## 1.57 installer.guide/retrace

La orden 'retrace' (V44)

-----

(retrace)

Invoca el mecanismo de vueltra atrás. Se salta la última posición de vuelta atrás y busca la inmediatamente anterior.

Las posiciones de vuelta atrás se pierden si la ejecución del script se sale fuera de los márgenes.

Ejemplo:

```
(trace) (message "a continuación, una orden if") (if (= 1 1)
((trace) (message "then"))      (message "else") ) (trace) (retrace)
```

El script salta atrás hasta la primera orden (trace) y no a la parte "then" de la orden 'if', con lo cual verá los mensajes "a continuación..." y "then" una y otra vez, hasta que interrumpa el script.

## 1.58 installer.guide/Órdenes de Depuración

Órdenes de Depuración

=====

'user'

---

```
`debug'
```

## 1.59 installer.guide/user

La orden `user`

-----

```
(user <nivel de usuario>)
```

Se utiliza para cambiar el nivel de usuario de la instalación actual. Esta orden debe utilizarse sólo durante la depuración de scripts. Elimine este tipo de órdenes de su script antes de la distribución final de su producto. Devuelve el nivel de usuario actual.

## 1.60 installer.guide/debug

La orden `debug`

-----

```
(debug <algo> <algo> ...)
```

Si Installer se ejecuta desde una ventana Shell, `debug` imprimirá los valores de los parámetros con un espacio entre cada parámetro. Por ejemplo, las órdenes

```
(set #variable 2)
(debug "El valor de 'variable' es" #variable)
```

mostrarán "El valor de 'variable' es 2". Si el parámetro es una variable no inicializada, `debug` mostrará '<NIL>' como su valor.

## 1.61 installer.guide/Procedimientos Definidos por el Usuario

Procedimientos Definidos por el Usuario

=====

Installer soporta procedimientos definidos por el usuario (subrutinas). Esta funcionalidad, sin embargo, es de momento muy primitiva. No existen las variables locales. A partir de Installer V42.7, eso sí, se dispone de una forma cómoda de pasar argumentos, que se describe a continuación. Para definir un nuevo procedimiento, utilice el comando 'procedure':

```
(procedure <procedimiento> [<argumentos>] <órdenes>)
```

Posteriormente puede invocar el procedimiento de esta forma:

```
(<procedimiento>)
```

Recuerde que '`<procedimiento>`' no es una cadena, sino simplemente un nombre simbólico. Debería utilizar un prefijo como '`P_`' para nombrar sus procedimientos; de lo contrario, podría tener problemas con futuras versiones de Installer.

El valor de retorno de un procedimiento es el valor de la última orden ejecutada en la secuencia de órdenes que componen el procedimiento.

A partir de Installer V42.7 hay una forma más fácil de pasar argumentos. '`<argumentos>`' es una lista de nombres de variables. Si invoca el procedimiento con argumentos, las variables tomarán como valor esos argumentos. Si al final faltan argumentos, las variables correspondientes mantendrán su valor anterior. Un ejemplo:

```
(procedure P_SUMAMUL arg1 arg2 arg3
  (* (+ arg1 arg2) arg3)
)

(message (P_SUMAMUL 1 2 3))      ; muestra 9
(message (P_SUMAMUL 4 5))      ; muestra 27, ya que arg3 sigue igual
```

## 1.62 installer.guide/Funciones

Funciones  
=====

Manejo de cadenas

```
Sustitución de cadenas
'cat'
'substr'
'strlen'

'transcript'
```

Manejo de nombres de ficheros y sendas

```
'tackon'
'fileonly'
'pathonly'
'expandpath'
```

Consultas al usuario

```
'askdir'
'askfile'
'askstring'
'asknumber'
'askchoice'
'askoptions'
```

---

```
`askbool`  
`askdisk`  
  
`exists`  
`earlier`
```

Obtención de información del DOS

```
`getsize`  
`getdevice`  
`getdiskspace`  
`getsum`  
`getversion`  
`getenv`  
`getassign`
```

Información sobre iconos

```
`iconinfo`      (V42.12)
```

Obtención de información sobre configuración

```
`database`
```

Recuperación selectiva de valores

```
`select`  
`patmatch`  
`symbolval`     (V42.9)
```

Todo tipo de operaciones matemáticas

Comparación de valores  
101 operaciones matemáticas  
Funciones lógicas

Funciones lógicas orientadas a bits  
Funciones de conmutación de bits  
Comprobación de bits

## 1.63 installer.guide/cadena

La función de sustitución de cadenas

```
(<cadena> <argumentos> ...)
```

Siempre que una cadena de texto sea el primer elemento de un grupo indicado entre paréntesis, los argumentos serán sustituidos en la

cadena empleando la función del Exec `'RawDoFmt()'`.

NOTA: Esta función no realiza comprobación alguna del tipo de argumento.

## 1.64 installer.guide/cat

La función `'cat'`

-----

```
(cat <cadena> <cadena> ...)
```

Concatena las cadenas y devuelve la cadena resultante.

Para convertir un entero en una cadena, utilice la función `'cat'`. Todos los argumentos enteros dados a `'cat'` son convertidos en cadenas durante la concatenación. Utilice `'set'` para convertir una cadena en un entero. Quizás desee echarle una ojeada también a `'symbolset'`.

## 1.65 installer.guide/substr

La función `'substr'`

-----

```
(substr <cadena> <inicio> [<cuenta>])
```

Devuelve una subcadena de `'<cadena>'`, comenzando por el carácter situado en la posición `'<inicio>'` (que sería 0 en el caso del primer carácter) y tomando el número de caracteres indicado por `'<cuenta>'`. Si se omite `'<cuenta>'`, se devolverá todo el resto de la cadena, hasta el final.

## 1.66 installer.guide/strlen

La función `'strlen'`

-----

```
(strlen <cadena>)
```

Devuelve la longitud de la cadena indicada.

---



## 1.67 installer.guide/transcript

La función 'transcript'

-----

```
(transcript <cadena> <cadena> ...)
```

Concatena las cadenas, añade un salto de línea y a continuación escribe la cadena resultante en el fichero de registro (si lo hay).

## 1.68 installer.guide/tackon

La función 'tackon'

-----

```
(tackon <senda> <fichero>)
```

Concatena el nombre del fichero con la senda, devolviendo la cadena resultante. La función responde correctamente a cualquier barra '/' que anteceda o a cualquier signo ':' que se encuentre en el parámetro '<fichero>'.

## 1.69 installer.guide/fileonly

La función 'fileonly'

-----

```
(fileonly <senda>)
```

Devuelve solamente la parte de una senda que indica el nombre del fichero.

## 1.70 installer.guide/pathonly

La función 'pathonly'

-----

```
(pathonly <senda>)
```

Devuelve solamente la parte de una senda que corresponde a la senda en sí, sin ningún nombre de fichero.

---

## 1.71 installer.guide/expandpath

La función 'expandpath'

-----

(expandpath <senda>)

Devuelve la senda completa, partiendo de una senda abreviada.  
Por ejemplo, podría expandir 'SYS:c' a 'System2.x:c'.

## 1.72 installer.guide/askdir

La función 'askdir'

-----

(askdir <parámetros>)

Pregunta al usuario por el nombre de un directorio, con un requester en forma de lista deslizante. El usuario puede o bien crear un nuevo directorio, o elegir uno ya existente. Si el usuario cancela el requester, la rutina producirá la interrupción del script.

NOTA: Siempre es mejor asegurarse de antemano de que el volumen que le interesa está montado, empleando la función 'askdisk'.

Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'default'

nombre por omisión del directorio a elegir. Recuerde que puede ser el nombre de una senda relativa.

'newpath'

permite indicar sendas inexistentes como directorio por omisión

'disk'

muestra primero la lista de unidades

'assigns'

indica que las asignaciones lógicas también deben satisfacer al requester

## 1.73 installer.guide/askfile

La función `'askfile'`

-----

`(askfile <parámetros>)`

Pregunta al usuario por el nombre de un fichero, presentando un requester en forma de lista deslizable. La senda por omisión puede hacer referencia a un fichero o a un directorio. En el primer caso, la caja con el nombre de fichero aparecerá ya cubierta. Parámetros:

`'prompt'`

`'help'`

dicen al usuario lo que va a ocurrir

`'newpath'`

permite indicar sendas inexistentes como directorio por omisión

`'disk'`

muestra primero la lista de unidades

`'default'`

nombre por omisión del fichero a elegir. Recuerde que puede ser el nombre de una senda relativa.

## 1.74 installer.guide/askstring

La función `'askstring'`

-----

`(askstring <parámetros>)`

Solicita al usuario que introduzca una cadena de texto. Parámetros:

`'prompt'`

`'help'`

dicen al usuario lo que va a ocurrir

`'default'`

la cadena de texto por omisión

## 1.75 installer.guide/asknumber

La función `'asknumber'`

-----

`(asknumber <parámetros>)`

Solicita al usuario que introduzca un valor entero, indicando el margen de valores aceptados inmediatamente debajo de la caja de

enteros en caso de que se utilice el parámetro 'range', e impidiendo al usuario seguir adelante mientras no introduzca un número válido. Si no se indica margen alguno, el margen de valores permitidos consistirá en todos los números no negativos. Sin embargo, si se indica un valor por omisión que es negativo, y no se determina margen alguno, el margen de valores permitidos se ampliará hacia abajo para llegar hasta el valor dado por omisión. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'range'

margen de números entendidos como válidos; para un funcionamiento correcto, el primero deberá ser inferior o igual al segundo.

'default'

valor por omisión

## 1.76 installer.guide/askchoice

La función 'askchoice'

-----

(askchoice <parámetros>)

Solicita al usuario que elija una de entre N opciones, empleando botones radiales. Como resultado se devuelve una máscara de bits, en la que el primer bit indica el estado de la primera opción, etc. Al tratarse de una máscara de bits, puede especificar un máximo de 32 opciones posibles. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'choices'

una lista de cadenas con opciones, tales como '"Manzanas"', '"Cerezas"', etc.

'default'

el número de la opción seleccionada inicialmente (por omisión, 0)

NOTA: A partir de Installer V42.6, puede hacer cosas mágicas con 'choices':

1. Si utiliza una cadena vacía para describir las opciones, la opción será invisible para el usuario, es decir, no se mostrará en pantalla. Utilizando variables, puede disponer un número programable de opciones, manteniendo la numeración de bits.
2. Las antiguas versiones de Installer no soportaban bien los tipos de letra de anchura proporcional, y alguna gente se basaba en la

presentación no proporcional de la ventana para la disposición de opciones en forma de tabla. Por tanto, Installer continuará presentando las opciones con un tipo de letra de anchura no proporcional, a menos que inicie una de las opciones con la secuencia especial de escape "<ESC>[2p". Esta secuencia de escape permite la presentación proporcional. Es una sabia decisión indicar esta secuencia sólo con la primera opción de la lista. No olvide esto. (V42)

## 1.77 installer.guide/askoptions

La función 'askoptions'

-----

(askoptions <parámetros>)

Solicita al usuario que elija cualquier número de opciones entre N posibles, empleando botones en forma de marca de revisado. Como resultado se devuelve una máscara de bits, en la que el primer bit indica el estado de la primera opción, etc. Al tratarse de una máscara de bits, puede especificar un máximo de 32 opciones posibles. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'choices'

una lista de cadenas con opciones, tales como '"Manzanas"', '"Cerezas"', etc.

'default'

una máscara de bits indicando los botones que aparecen seleccionados inicialmente (por omisión, -1)

NOTA: A partir de Installer V42.6, puede hacer cosas mágicas con 'choices':

1. Si utiliza una cadena vacía para describir las opciones, la opción será invisible para el usuario, es decir, no se mostrará en pantalla. Utilizando variables, puede disponer un número programable de opciones, manteniendo la numeración de bits.
  2. Las antiguas versiones de Installer no soportaban bien los tipos de letra de anchura proporcional, y alguna gente se basaba en la presentación no proporcional de la ventana para la disposición de opciones en forma de tabla. Por tanto, Installer continuará presentando las opciones con un tipo de letra de anchura no proporcional, a menos que inicie una de las opciones con la secuencia especial de escape "<ESC>[2p". Esta secuencia de escape permite la presentación proporcional. Es una sabia decisión indicar esta secuencia sólo con la primera opción
-

de la lista. No olvide esto. (V42)

## 1.78 installer.guide/askbool

La función 'askbool'

-----

(askbool <parámetros>)

Solicita al usuario que elija "Sí" o "No". Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'default'

0 = no, 1 = sí

'choices'

cambia los textos de las respuestas. Los valores por omisión son '"Sí"' y '"No"'. Así, para cambiar el texto por '"Continuar"' y '"Cancelar"', usaría '(choices "Continuar" "Cancelar")'.

## 1.79 installer.guide/askdisk

La función 'askdisk'

-----

(askdisk <parámetros>)

Solicita al usuario que inserte un disco, de una forma amistosa. Por ejemplo, la ventana puede describir el disco de acuerdo con su etiqueta, como "Disco de Programa". Esta función no finalizará hasta que se inserte el disco correcto, o el usuario interrumpa el script. Parámetros:

'prompt'

'help'

dicen al usuario lo que va a ocurrir

'dest'

el nombre de volumen del disco a insertar

'newname'

un nombre que se deba asignar al disco para una futura referencia. Esta asignación se realiza incluso en modo PRETEND, puesto que es considerada como "segura".

'disk'

interrupción para hacer que se muestre al principio una lista de unidades

``assigns'`

hace que las asignaciones lógicas también satisfagan al requester

Nota: El nombre de volumen debe indicarse sin signo de dos puntos; es decir, por ejemplo "ENV" y no "ENV:".

## 1.80 installer.guide/exists

La función ``exists'`

-----

`(exists <fichero> (noreq))`

Devuelve 0 si el `<fichero>` no existe, 1 si se trata de un fichero, o 2 si en realidad se trata de un directorio. Si se indica `'noreq'`, no se mostrará requester alguno en caso de que la senda indicada no se encuentre en ninguno de los volúmenes montados; en este caso, el resultado será 0.

## 1.81 installer.guide/earlier

La función ``earlier'`

-----

`(earlier <fichero-1> <fichero-2>)`

Devuelve TRUE si `<fichero-1>` es más antiguo que `<fichero-2>`.

## 1.82 installer.guide/getsize

La función ``getsize'`

-----

`(getsize <fichero>)`

Devuelve el tamaño de un fichero, en bytes.

## 1.83 installer.guide/getdevice

La función 'getdevice'

-----

```
(getdevice <senda>)
```

Devuelve el nombre del volumen en el que se encuentra la <senda>. Por ejemplo, 'c:mount' como senda podría devolver 'WB\_2.x'.

## 1.84 installer.guide/getdiskspace

La función 'getdiskspace'

-----

```
(getdiskspace <senda>)
```

Devuelve el espacio libre disponible en el disco al que hace referencia la <senda>, en bytes. Devuelve -1 si la senda es incorrecta o si no puede obtenerse la información necesaria del sistema de ficheros, a pesar de que el nombre de la senda pudiese ser válido.

## 1.85 installer.guide/getsum

La función 'getsum'

-----

```
(getsum <fichero>)
```

Devuelve la suma de comprobación de un fichero, para comparar versiones. Puede utilizar esta función para obtener la suma de comprobación al preparar el script y para comprobarla mientras el script está corriendo.

## 1.86 installer.guide/getversion

La función 'getversion'

-----

```
(getversion <fichero> (resident))
```

Si el fichero nombrado tiene una RomTag con una cadena de identificación o una cadena de versión al estilo del S.O. 2.x, esta función devolverá el número de versión. Si no se indica el nombre de un fichero, se devolverá en su lugar la versión del S.O. Tenga en cuenta que esta función NO asume que los ficheros cuyo nombre termina en '.library' o '.device' residen en un lugar

---



en particular; deberá especificar la senda completa. Si se usa 'resident', la función intentará devolver el número de versión de la librería o driver situado en la memoria. Por ejemplo:

```
(getversion "intuition.library" (resident))
```

devolvería la versión/revisión de Intuición. Recuerde que usar el parámetro 'resident' hace que se compruebe primero la lista de librerías y a continuación la de drivers.

El número de versión se devuelve en forma de valor de 32 bits, donde la palabra de 16 bits de orden más alto es la versión y la palabra de orden más bajo es la revisión. Aquí van algunas órdenes de ejemplo para interpretar el número de versión:

```
(set vernum (getversion "c:iconx"))
(set ver (/ vernum 65536))
(set rev (- vernum (* ver 65536) ) )

(message
  ("Tiene la versión %ld.%ld" ver rev)
)
```

Si el fichero en cuestión no puede encontrarse, se devolverá una versión 0.

## 1.87 installer.guide/getenv

La función 'getenv'

-----

```
(getenv <nombre>)
```

Devuelve el contenido de la variable de entorno (ENV:) indicada.

## 1.88 installer.guide/getassign

La función 'getassign'

-----

```
(getassign <nombre> <opciones>)
```

Devuelve la senda correspondiente al objeto al que hace referencia '<nombre>'. Por omisión sólo se examinan las asignaciones lógicas, pero esto puede modificarse utilizando una cadena de opciones donde los caracteres tienen las siguientes correspondencias:

```
'v'
sólo examinar los volúmenes
```

`'a'`  
sólo examinar las asignaciones lógicas

`'d'`  
sólo examinar las unidades

Por tanto, `'a'` equivaldría a no usar opción alguna. Devuelve una cadena vacía en caso de producirse un error.

Notas:

`'<nombre>'` no debe llevar ningún signo de dos puntos; es decir, `'"ENV"'` y no `'"ENV:"'`. En lugar del `<nombre>` puede utilizarse una variable previamente definida como tal.

Si se utiliza el nombre de una unidad y la búsqueda se limita a nombres de unidad, `'getassign'` devolverá el nombre de unidad o de volumen en caso de que exista la unidad; de lo contrario, devolverá una cadena vacía. Un ejemplo de uso sería `'(getassign "df1" "d")'`.

`'getassign'` no puede manejar asignaciones no vinculantes, debido a una limitación del AmigaDOS. En este caso, devolverá una cadena vacía a partir de Installer V42. Lo mismo ocurrirá con las asignaciones de espera (creadas por ASSIGN DEFER) que por alguna razón no puedan resolverse con un simple acceso.

## 1.89 installer.guide/iconinfo

La función `'iconinfo'` (V42.12)

-----

`(iconinfo <parámetros>)`

Obtiene información sobre los tipos de herramienta y otros datos de un icono. Excepto por el resultado, esta función difiere de otras. Los argumentos para la mayoría de los parámetros no son valores, sino nombres de símbolos que se definirán de acuerdo con esos valores por medio de esta función. ¡Tenga cuidado!

Parámetros:

`'prompt'`  
`'help'`  
dicen al usuario lo que va a ocurrir

`'dest'`  
el nombre del icono a examinar. No hay necesidad de indicar una extensión `' .info'`.

`'gettooltype'`  
cadena con el símbolo que indica el nombre y el resultado del tipo de herramienta

```
`getdefaulttool'
    símbolo para el nombre de la herramienta por omisión de un
    proyecto

`getstack'
    símbolo para el tamaño de pila actual del icono

`getposition'
    Dos símbolos para la posición guardada del icono en las coordenadas
    X e Y. No tome esto a la ligera. Está pensado para mantener las
    posiciones de los iconos durante las actualizaciones con ayuda de
    la orden `tooltype'. Cambiar arbitrariamente las posiciones de
    los iconos sólo le deparará usuarios enfadados, al emplear diferentes
    configuraciones y tipos de letra en su Workbench. Si el icono no
    tiene ninguna posición definida, se devolverá -1 para el respectivo
    valor posicional. Esta información puede pasarse a `tooltype'.
```

## 1.90 installer.guide/database

La función `database'

-----

```
(database <característica> [<valor>])
```

Devuelve información sobre el modelo de Amiga en el que está corriendo Installer. El argumento '*<característica>*' debe ser una cadena. Esta función siempre devuelve un resultado en forma de cadena, aunque el resultado tenga un aspecto numérico. Si no se reconoce la característica solicitada, la función devolverá `"unknown"`. Las características comprendidas actualmente son:

```
`"vblank"'
    `"50"', `"60"'
```

```
`"cpu"'
    `"68000"', `"68010"', `"68020"', `"68030"', `"68040"', `"68060"'
```

```
`"fpu"'
    `"NOFPU"', `"68881"', `"68882"', `"FPU40"'
```

```
`"graphics-mem"'
    devuelve una cadena que representa la cantidad de memoria gráfica
    libre, en bytes
```

```
`"total-mem"'
    devuelve una cadena que representa la cantidad total de memoria
    libre, en bytes
```

```
`"chiprev"'
    `AA', `ECS', `AGNUS'.
```

Si utiliza la cadena opcional '*<valor>*', se devolverá un resultado booleano en lugar de uno de cadena, indicando si se produce una coincidencia entre el argumento de cadena y el '*<valor>*' dado.

Esto es una comprobación exacta, ofrecida en forma de operador. Para hacer una comparación con varias posibles CPUs, debería hacer algo como esto:

```
(not (patmatch "68000|68010" (database "cpu")))
```

Intente pensar siempre de forma defensiva al utilizar la función `'database'`. Diferentes configuraciones podrían devolver resultados inesperados. Prepare su script también para ellos.

## 1.91 installer.guide/select

La función `'select'`

-----

```
(select <n> <elemento1> <elemento2> ...)
```

Devuelve el valor del elemento número N.

## 1.92 installer.guide/patmatch

La función `'patmatch'`

-----

```
(patmatch <patrón> <cadena>)
```

Determina si una cadena es aludida por un patrón del AmigaDOS. Devuelve TRUE o FALSE.

## 1.93 installer.guide/symbolval

La función `'symbolval'` (V42.9)

-----

```
(symbolval <símbolo>)
```

Evalúa la expresión de cadena `'<símbolo>'` y recupera el valor del símbolo nombrado para ofrecerlo a través del resultado. En combinación con la orden `'symbolset'`, esta función puede emplearse para disponer de cierto tipo de manejo dinámico en los nombres de las variables.

NOTA: ¡Por favor, lea también la descripción de `'set'`!

## 1.94 installer.guide/Comparación

Las funciones de comparación

-----

```
(= <expresión-1> <expresión-2>)
(> <expresión-1> <expresión-2>)
(>= <expresión-1> <expresión-2>)
(< <expresión-1> <expresión-2>)
(<= <expresión-1> <expresión-2>)
(<> <expresión-1> <expresión-2>)
```

Se trata de las expresiones relacionales estándar.

## 1.95 installer.guide/Matemáticas básicas

Las funciones matemáticas básicas

-----

```
(+ <expresión> ...)
```

Devuelve la suma de todos los argumentos.

```
(- <expresión-1> <expresión-2>)
```

Devuelve la diferencia entre el primer argumento y el segundo.

```
(* <expresión> ...)
```

Devuelve el producto de todos los argumentos.

```
(/ <expresión-1> <expresión-2>)
```

Devuelve el cociente de dividir el primer argumento por el segundo.

## 1.96 installer.guide/Funciones lógicas

Las funciones lógicas

-----

```
(AND <expresión-1> <expresión-2>)
(OR <expresión-1> <expresión-2>)
(XOR <expresión-1> <expresión-2>)
(NOT <expresión>)
```

Funciones lógicas estándar.

---

## 1.97 installer.guide/Funciones lógicas orientadas a bits

Las funciones lógicas orientadas a bits

-----

```
(BITAND <expresión-1> <expresión-2>)
(BITOR  <expresión-1> <expresión-2>)
(BITXOR <expresión-1> <expresión-2>)
(BITNOT <expresión>)
```

Versiones orientadas a bits de las funciones lógicas estándar.

## 1.98 installer.guide/Conmutación de bits

Las funciones de conmutación de bits

-----

```
(shiftright <número> <cantidad a saltar>)
(shiftleft <número> <cantidad a saltar>)
```

Estas funciones realizan un salto orientado a bits equivalente a la cantidad especificada. Los ceros se pasan y se justifican en el lado contrario.

## 1.99 installer.guide/Comprobación de bits

La función 'IN'

-----

```
(IN <expresión> <número de bit-1> ...)
```

Devuelve 0 si ninguno de los números de bit (comenzando por 0 para el bit menos significativo) está activado en el resultado de la expresión; de lo contrario, devuelve una máscara de los bits que estaban activados.

## 1.100 installer.guide/Sumario de Parámetros

Sumario de Parámetros

=====

'(all)'

En la orden 'copyfiles', hace que se copien todos los ficheros.

'(append <cadena>)'

Dentro de una orden 'textfile', añadirá la cadena al fichero de

---

texto.

``(assigns)'`

Una opción empleada con la función ``askdisk'` para indicar que las asignaciones lógicas también serán examinadas por la petición de ``askdisk'`.

``(choices <cadena-1> <cadena-2> ...)'`

Se utiliza para mostrar una serie de marcas de revisado o de botones radiales, dependiendo de la función utilizada. Este parámetro es utilizado por las funciones ``askchoice'` y ``askoptions'` para indicar de qué opciones dispondrá el usuario. También puede utilizarse con la orden ``copyfiles'` para indicar que sólo pueden copiarse unos ficheros determinados. En su ausencia, se empleará otro criterio para determinar qué ficheros se copian.

NOTA: A partir de Installer V42.6, puede hacer cosas mágicas con `'choices'`:

1. Si utiliza una cadena vacía para describir las opciones, la opción será invisible para el usuario, es decir, no se mostrará en pantalla. Utilizando variables, puede disponer un número programable de opciones, manteniendo la numeración de bits.
2. Las antiguas versiones de Installer no soportaban bien los tipos de letra de anchura proporcional, y alguna gente se basaba en la presentación no proporcional de la ventana para la disposición de opciones en forma de tabla. Por tanto, Installer continuará presentando las opciones con un tipo de letra de anchura no proporcional, a menos que inicie una de las opciones con la secuencia especial de escape "`<ESC>[2p`". Esta secuencia de escape permite la presentación proporcional.

``(command <texto> ...)'`

Especifica de forma textual un comando a insertar en el fichero `'S:User-Startup'`. Los argumentos de cadena se concatenarán.

``(confirm <nivel de usuario>)'`

En el caso de ciertas órdenes, el usuario sólo será informado de la operación y le será permitido cancelarla en caso de que esté presente la opción `'confirm'`. El nivel de usuario puede ser `'EXPERT'` o `'AVERAGE'`. El valor por omisión es `EXPERT`.

``(default <valor>)'`

Indica el valor por omisión de una función ``askchoice'`, ``askstring'` o ``asknumber'`.

``(delopts <opción> <opción> ...)'`

Indica a las órdenes ``copyfiles'`, ``copylib'` y ``delete'` que las opciones listadas deberán eliminarse de la lista interna global de opciones para esa orden. La opción global por omisión es `"fail"`.

``(dest <fichero>)'`

Indica el fichero o directorio a modificar como parte de la orden.

``(disk)'`

Cuando se utiliza con la orden `'rename'`, sirve para indicar que en realidad va a hacerse un cambio en la etiqueta de un disco. Cuando se utiliza con las funciones `'askdir'` o `'askfile'`, indica que deberá presentarse inicialmente una lista de unidades, en lugar de una lista de ficheros.

``(fonts)'`

Indica a la orden `'copyfiles'` que copie también los ficheros `'.font'` que encuentre.

``(help <cadena-1> <cadena-2> ...)'`

Se utiliza para indicar el texto de ayuda para cada operación.

``(infos)'`

Indica a la orden `'copyfiles'` que copie también los ficheros `'.info'` que encuentre. Si el directorio destino no existe aún, se creará un icono por omisión para el directorio generado por Installer. También es válido para la orden `'delete'`.

``(include <fichero>)'`

Dentro de una orden `'textfile'`, añadirá el contenido del fichero indicado al fichero de texto. Si el fichero no existe, esta operación será ignorada.

``(newname <nombre>)'`

Se utiliza con `'copyfiles'` para indicar que un fichero cambiará de nombre una vez que haya alcanzado su destino. También se usa con `'askdisk'` para asignar un nuevo nombre al disco insertado, y con `'copylib'` para renombrar la librería una vez que se encuentre en su destino.

``(newpath)'`

Se utiliza con `'askdir'` y `'askfile'` para permitir indicar como directorio por omisión una senda inexistente.

``(nogauge)'`

Cuando se utiliza con las órdenes `'copyfiles'` y `'copylib'`, desactiva el indicador de progreso.

``(noposition)'`

Se utiliza para modificar el emplazamiento de un icono y así "descongelarlo" en lo que respecta al Workbench (`'NO_ICON_POSITION'`).

``(pattern <cadena>)'`

Se utiliza con la orden `'copyfiles'` para indicar un patrón con comodines.

``(prompt <cadena-1> <cadena-2> ...)'`

Se utiliza para dar el "título" de la ventana que explica al usuario qué se va a hacer en este paso de la instalación.

``(range <mínimo> <máximo>)'`

Indica el margen de valores aceptables en una función `'asknumber'`.

``(safe)'`

Ordena a Installer que efectúe una operación que normalmente no

---



haría en modo PRETEND.

``(setdefaulttool <valor>)``

Se utiliza para modificar la herramienta por omisión de un icono.

``(setstack <valor>)``

Se utiliza para modificar el tamaño de pila definido en un icono.

``(settooltype <tipo de herramienta> <valor>)``

Se utiliza para modificar un tipo de herramienta. Si el tipo de herramienta no existe aún, será creado en ese momento. Si el parámetro '`<valor>`' se omite, lo que se hará es borrar del icono el tipo de herramienta citado. Puede añadirse un tipo de herramienta sin argumento alguno, de la manera siguiente:

```
(settooltype <tipo de herramienta> "")
```

Recuerde que ``(tooltype <tipo de herramienta>)`` sin más borrará el tipo de herramienta indicado.

``(source <fichero>)``

Indica el fichero o directorio a leer como parte de esta orden.

``(swapcolors)``

Este parámetro está obsoleto y no debería usarse más. Sigue estando aquí por compatibilidad hacia abajo, careciendo de cualquier otro cometido. ¡NO LO USE!

``(optional <opción> <opción> ...)``

Indica a las órdenes `'copyfiles'` y `'copylib'` que un fallo en la copia no es un error fatal. También se utiliza con `'delete'` para indicar si deberá forzarse el borrado.

``(resident)``

Se utiliza con `'getversion'` para indicar que deberán examinarse las listas de librerías y drivers en memoria.

``(back <código>)``

Se utiliza para definir el código para el mecanismo de vuelta atrás. Si el usuario pulsa el botón "Volver atrás", se ejecutará el código. Lo típico es que este código contenga una orden `(retrace)` que apunta a la última orden.

Ejemplo:

```
(set #siono (askbool (prompt "¿Desea instalar la documentación?")
                    (help @askbool-help)
                    (choices "Sí" "No")
                    (back (retrace))
                    )
)
```

Debe usar este parámetro para activar el botón "Volver atrás".

## 1.101 installer.guide/Variables Predefinidas

### Variables Predefinidas

=====

Existen una serie de variables predefinidas a disposición del script de instalación. Estas variables pueden ser modificadas en vivo, pero su tipo no puede cambiarse (por ejemplo, para convertir una cadena en un valor numérico) a menos que la cadena careciese de valor alguno anteriormente.

`'@abort-button'`

Texto que sustituye al del botón 'Interrumpir instalación'.

`'@app-name'`

El valor de 'APPNAME' indicado en el arranque.

`'@icon'`

El nombre de la senda completa del icono empleado para lanzar Installer -sin incluir sufijo '.info' alguno- en la invocación desde el Workbench. ¡Este no es necesariamente el icono del script! A partir de Installer V42.12, esta variable contendrá la senda completa del script en el caso de una invocación desde Shell.

`'@execute-dir'`

Si esta variable contiene una senda válida, Installer utilizará esta senda como directorio actual siempre que se utilice una orden 'run' o 'execute'.

`'@default-dest'`

El lugar sugerido por Installer para instalar una aplicación. Si ha instalado la aplicación en alguna otra parte (como resultado de haber consultado al usuario), debería modificar este valor; esto permitirá que la orden "final" funcione correctamente. Tenga en cuenta que crear un directorio y colocar la aplicación en él es considerado como instalar la aplicación en otra parte. Dé a esta variable el valor '""' si realmente no hay un lugar definido en el que la "aplicación" haya sido instalada. El fichero de registro se copiará al directorio indicado por '@default-dest' a menos que se le haya dado el valor '""'.

`'@language'`

Se utiliza para definir la variable '@language'. El valor por omisión para '@language' es el nombre de la localización del sistema, o bien "english". El uso de esta variable queda a cargo del script de instalación.

`'@pretend'`

El estado de la bandera "Pretend" (1 en modo PRETEND).

`'@user-level'`

El nivel de usuario en el que corre el script: 0 para NOVICE, 1 para AVERAGE, 2 para EXPERT.

``@installer-version'`

Un entero que representa la versión actual de Installer.  
El valor coincide con el que ``getversion'` devolvería en el caso de Installer. Esta variable está disponible a partir de Installer V42.

``@error-msg'`

El texto que se mostraría en caso de producirse un error fatal, pero que no aparecería en el caso de emplearse una orden ``trap'`.

``@special-msg'`

Si en un script desea mostrar su propio texto para cualquier error fatal que se produzca en varios puntos del script, esta variable debería contener ese texto. El texto original de error se añadirá a continuación de éste, entre paréntesis. Defina esta variable como `'"'` para eliminar el uso del mensaje especial.

``@ioerr'`

El valor del último error DOS. Puede utilizarse junto con la orden ``trap'` para disponer de más información sobre un error. A partir de Installer V42, esta variable también es soportada por las órdenes ``run'`, ``execute'` y ``rexx'`.

``@each-name'`

``@each-type'`

Se utilizan en un bucle ``foreach'`.

``@askoptions-help'`

``@askchoice-help'`

``@asknumber-help'`

``@askstring-help'`

``@askdisk-help'`

``@askfile-help'`

``@askdir-help'`

``@copylib-help'`

``@copyfiles-help'`

``@makedir-help'`

``@startup-help'`

Texto de ayuda por omisión para varias funciones. Estos textos pueden añadirse a la explicación ofrecida por el sistema para una operación en particular, o bien usarse completamente como reemplazo.

## 1.102 installer.guide/Referencia rápida del lenguaje de Installer

Referencia Rápida del Lenguaje de Installer

\*\*\*\*\*

Introducción

Introducción rápida al lenguaje

Variables predefinidas

Variables de cadenas de ayuda por omisión

Órdenes

## Funciones

### 1.103 installer.guide/Introducción (resumen)

#### Introducción

=====

Intenta instalar en 'Work:' por omisión, si existe, o bien (V42) en la partición escribible con mayor espacio libre.

La tecla <Help> hace aparecer ayuda sensible al contexto.  
La tecla <Esc> hace aparecer el requester de interrupción.

Puede añadir asignaciones al fichero 'S:User-Startup', y añade líneas a 'S:Startup-Sequence' si es necesario para asegurarse de que 'S:User-Startup' se ejecute durante el arranque. Utiliza el volumen de arranque para buscar 'S:'.

Puede comprobar la versión de ficheros y librerías.

La instalación puede correr en modo "real" (para llevarla a cabo realmente) o "simulado" (instalación ficticia).

### 1.104 installer.guide/Introducción rápida al lenguaje

#### Introducción rápida al lenguaje

=====

El lenguaje es similar a LISP (montones de paréntesis ()) (-:).

Las variables carecen de tipo definido (al estilo ARexx), es decir, cadenas y números son tratados indistintamente.

Las cadenas se delimitan con ''' o ''.

En las cadenas pueden utilizarse ciertas secuencias embebidas:

'\n'  
salto de línea (nueva línea)

'\r'  
carácter de retorno del carro

'\t'  
tabulador

'\h'  
tabulador horizontal (V42.6)

'\v'

```
    tabulador vertical (V42.6)

'\b'
    retroceso (V42.6)

'\f'
    salto de página (V42.6)

'\"'
    comillas "

'\''
    comilla '

'\'\'
    barra invertida

'\ooo'
    número octal 'ooo' (V42.6)

'\xXX'
    número hexadecimal 'XX' (V42.6)
```

Las órdenes se indican entre paréntesis ( ). El formato general es:  
'(operador <operando1> <operando2> ...)'

Por ejemplo, para asignar el valor '5' a la variable 'x', usaría  
'(set x 5)'.

Para obtener la suma de dos números, utilice '(+ 5 9)'

Tenga en cuenta que no hay diferencia alguna entre operadores y órdenes; la orden 'set' y el operador aritmético '+' se emplean exactamente de la misma forma.

Combinación de órdenes: Una orden puede utilizarse como operando de otra orden. Por ejemplo:

```
(set x (+ 3 5))
```

En este caso, la orden '(+ 3 5)' se evalúa primero, y el resultado es 8. Puede imaginar esto como que la parte '(+ 3 5)' ha sido sustituida por un 8, lo que nos deja:

```
(set x 8)
```

Tenga en cuenta que la parte '(+ 3 5)' ha producido realmente un valor: '8'. A esto se le llama el "resultado" de la orden. Muchas órdenes devuelven resultados, incluso algunas que podrían sorprenderle (como 'set' e 'if').

Los comentarios van precedidos de un signo ';'.

Los números hexadecimales van precedidos de un signo '\$' (por ejemplo, '\$23').

Los números binarios van precedidos de un signo '%' (por ejemplo,

```
'%0101').
```

Muchas órdenes devuelven un valor que puede utilizarse en asignaciones, pruebas, etc.

La información puede formatearse empleando una cadena literal con emplazadores de argumentos, por ejemplo:

```
("Mido %ld metro y %ld centímetros." 1 80)
;Produce una cadena donde los %ld's se sustituyen con 1 y 80.
;Recuerde que los valores decimales deben indicarse en formato
;de palabras largas.
```

## 1.105 installer.guide/Variables predefinidas (resumen)

Variables predefinidas

=====

```
`@default-dest'
    directorio donde el script colocará por omisión las cosas

`@each-name'
`@each-type'
    se utilizan con `foreach'.

`@error-msg'
    mensaje que se mostraría si un error no es atrapado. Véase `trap'.

`@execute-dir'
    Installer pasará a este directorio antes de realizar una orden
    `run', o `execute'.

`@icon'
    nombre de la senda del script de instalación invocado por arranque
    desde el Workbench, o bien (V42.12) del script de instalación abierto
    por una invocación desde el Shell.

`@installer-version'
    versión de Installer, en el formato de `version'.

`@language'
    idioma indicado en los tipos de herramienta o en el Shell (nombre
    de la localización empleada si está disponible, o bien `"english").

`@pretend'
    estado de la bandera `pretend' (instalación ficticia): 0-Real, 1-Ficticia.

`@special-msg'
    mensaje de error fatal especial.

`@user-level'
    0-Novice, 1-Average, 2-Expert.
```

---

## 1.106 installer.guide/Variables de cadenas de ayuda por omisión

Variables de Cadenas de Ayuda por Omisión

=====

Puede dar un valor a estas variables que de esta forma sustituya a los textos de ayuda empleados por omisión por parte de los comandos correspondientes.

@askoptions-help

@askchoice-help

@asknumber-help

@askstring-help

@askdisk-help

@askfile-help

@askdir-help

@copylib-help

@copyfiles-help

@mkdir-help

@startup-help

## 1.107 installer.guide/Órdenes (resumen)

Órdenes

=====

Muchos comandos tienen parámetros estándar (algunos opcionales):

'(all)'

hace que se tengan en cuenta todos los ficheros

'(append <cadena>)'

añade la cadena al fichero de texto (para 'textfile')

'(choices <cadenal> <cadena2> ...)'

opciones para los botones de revisado o radiales

'(command <cadenal> <cadena2>...)'

comandos a añadir a 'S\User-Startup'

'(confirm <nivel de usuario>)'

confirmación

``(default <valor>)``  
valor, opción, cadena, etc. por omisión

``(dest <fichero>)``  
salida al <fichero>

``(help <cadena1> <cadena2> ...)``  
define el texto de información para el mensaje de ayuda actual

``(include <fichero>)``  
insertar este fichero como parte de una orden ``textfile``

``(infos)``  
copiar/borrar también los ficheros ``.info``

``(newname <nombre>)``  
indica el nuevo nombre del fichero o disco

``(noposition)``  
"descongela" el icono

``(pattern <cadena>)``  
se utiliza con ``files`` para emplear patrones

``(prompt <cadena1> <cadena2> ...)``  
texto a mostrar al usuario

``(range <mínimo> <máximo>)``  
margen para la entrada numérica (véase ``asknumber``)

``(safe)``  
obliga a Installer a realizar la operación incluso en modo ``Pretend``

``(settooltype <tipo de herramienta> <valor>)``  
define un tipo de herramienta de un icono

``(setdefaulttool <valor>)``  
define la herramienta por omisión de un icono

``(setstack <valor>)``  
definir el valor de pila de un icono

``(source <fichero>)``  
leer del <fichero>

``(swapcolors)``  
intercambiar los primeros dos planos de la imagen de un icono si la versión del S.O. es inferior a la V36

``(back <código>)``  
Activar el botón "Volver atrás" y asignarle el código correspondiente (V44)

Nota: Los parámetros especiales se muestran abajo entre corchetes `< >`, mientras que los parámetros estándar se indican con el formato ``(parámetro...)``, donde ``parámetro`` es ``help``, ``prompt``, ``safe``, etc.

---



Véase más arriba más información sobre los parámetros estándar.

```
'(welcome <cadena> <cadena> ...)'
    invoca la pantalla de "bienvenida"
```

```
'(abort <cadena1> <cadena2> ...)'
    abandona la instalación
```

```
'(complete <número>)'
    muestra en la barra de título un valor porcentual que hace referencia al
    progreso de la instalación
```

```
'(copyfiles (prompt..) (help..) (source..) (dest..) (newname..) (choices..)'
    '(all) (pattern..) (files) (infos) (confirm..) (safe) (optional
    <opción> <opción> ...) (delopts <opción> <opción> ...) (nogauge)
    (back..))'
```

copia ficheros (y subdirectorios, por omisión). La opción 'files'  
dice NO subdirectorios

```
'(copylib (prompt..) (help..) (source..) (dest..) (newname..) (infos) (confirm)'
    '(safe) (optional <opción> <opción> ...) (delopts <opción> <opción> ...)
    (nogauge) (back..))'
```

instala una librería en caso de que se trate de una versión más  
reciente

```
'(delete file (help..) (prompt..) (confirm..) (infos) (optional <opción>
    <opción> ...) (all) (delopts <opción> <opción> ...) (safe) (back..))'
```

borra un fichero

```
'(execute <arg> (help..) (prompt..) (confirm) (safe) (back..))'
    ejecuta un script
```

```
'(exit <cadena> <cadena> ... (quiet))'
    finaliza la instalación después de mostrar las cadenas (en caso de que
    se especifiquen)
```

```
'(foreach <directorio> <patrón> <órdenes>)'
    realizar cierta operación con los elementos del directorio
```

```
'(if expresión ordenessisecumple ordenessinosecumple)'
    condicional
```

```
'(makeassign <asignación> <senda> (safe)) ; nota: la <asignación> no
    necesita ':'
```

crear/anular una asignación

```
'(mkdir <nombre> (prompt..) (help..) (infos) (confirm..) (safe) (back..))'
    crear un directorio
```

```
'(message <cadena1> <cadena2>... (all) (back..))'
    mostrar un mensaje con botones Continuar, Interrumpir
```

```
'(onerror (<órdenes>))'
```

```
trampa general de errores

'(protect <fichero> [<cadena de banderas a cambiar>] [<máscara decimal>] < ←
  parámetros>)'
  obtener/definir banderas de protección de un fichero

'(rename <antiguo> <nuevo> (help..) (prompt..) (confirm..) (safe) (back..))'
  renombrar ficheros

'(retrace)'
  invocar el mecanismo de vuelta atrás (V44)

'(rexx <argumento> (help..) (prompt..) (confirm..) (safe))'
  ejecutar script ARexx

'(run <argumento> (help..) (prompt..) (confirm..) (safe))'
  ejecutar programa

'(set <variable> <expresión>)'
  asignar una variable a una variable

'(startup (prompt..) (command..))'
  añadir un comando a los scripts de arranque (Startup-sequence,
  User-startup)

'(symbolset <símbolo> <expresión>)'
  asignar un valor a una variable a la que hace referencia el resultado
  de cadena de <símbolo> (V42.9)

'(textfile (prompt..) (help..) (dest..) (append) (include..) (confirm..)
  (safe) (back..))'

  crear un fichero de texto a partir de otros ficheros y cadenas de texto

'(tooltype (prompt..) (help..) (dest..) (settooltype..) (setstack..))'
  '(setdefaulttool..) (noposition) (confirm..) (safe))'

  modificar la información de un icono

'(trace)'
  fijar posición de vuelta atrás (V44)

'(trap <banderas> <órdenes>)'
  atrapa errores. banderas: 1-interrupción, 2-memoria agotada, 3-error,
  4-error DOS, 5-argumentos erróneos

'(until <expresión> <órdenes>)'
  estructura condicional do-until (probar fin del bucle)

'(welcome <cadena> <cadena> ...)'
  permite el inicio de la instalación en sí

'(while <expresión> <órdenes>)'
  estructura condicional do-while (probar inicio del bucle)

'(working)'
  indicar al usuario que Installer está ocupado haciendo cosas
```

---

## 1.108 installer.guide/Funciones (resumen)

### Funciones

=====

```
`(= <expresión1> <expresión2>)`  
    prueba de igualdad (devuelve 0 ó 1)  
  
`(> <expresión1> <expresión2>)`  
    prueba "mayor que" (devuelve 0 ó 1)  
  
`(>= <expresión1> <expresión2>)`  
    prueba "mayor o igual que" (devuelve 0 ó 1)  
  
`(< <expresión1> <expresión2>)`  
    prueba "menor que" (devuelve 0 ó 1)  
  
`(<= <expresión1> <expresión2>)`  
    prueba "menor o igual que" (devuelve 0 ó 1)  
  
`(+ <expresión1> <expresión2> ...)`  
    devuelve la suma de las expresiones  
  
`(- <expresión1> <expresión2>)`  
    devuelve la diferencia entre '<expresión1>' y '<expresión2>'  
  
`(* <expresión1> <expresión2> ...)`  
    devuelve el producto de las expresiones  
  
`(/ <expresión1> <expresión2>)`  
    devuelve el cociente de dividir '<expresión1>' por '<expresión2>'  
  
`(AND <expresión1> <expresión2>)`  
    devuelve el resultado lógico 'AND' de '<expresión1>' y '<expresión2>'  
  
`(OR <expresión1> <expresión2>)`  
    devuelve el resultado lógico 'OR' de '<expresión1>' y '<expresión2>'  
  
`(XOR <expresión1> <expresión2>)`  
    devuelve el resultado lógico 'XOR' de '<expresión1>' y '<expresión2>'  
  
`(NOT <expresión>)`  
    devuelve el resultado lógico 'NOT' de '<expresión>'  
  
`(BITAND <expresión1> <expresión2>)`  
    devuelve el resultado lógico orientado a bits 'AND' de '<expresión1>'  
    y '<expresión2>'  
  
`(BITOR <expresión1> <expresión2>)`  
    devuelve el resultado lógico orientado a bits 'OR' de '<expresión1>'  
    y '<expresión2>'  
  
`(BITXOR <expresión1> <expresión2>)`
```

---

devuelve el resultado lógico orientado a bits 'XOR' de '<expresión1>' y '<expresión2>'

'(BITNOT <expresión>)'  
devuelve el resultado lógico orientado a bits 'NOT' de '<expresión>'

'(shiftright <número> <cantidad a saltar>)'  
salto lógico a izquierda

'(shiftright <número> <cantidad a saltar>)'  
salto lógico a derecha

'(IN <expresión> <número de bit> <número de bit>...)'  
devuelve el resultado de '<expresión>' 'AND' bits

'(<cadena de formateo> <argumento1> <argumento2> ...)'  
clon de printf

'(askdir (prompt..) (help..) (default..) (newpath) (disk) (back..))'  
solicita el nombre de un directorio

'(askfile (prompt..) (help..) (default..) (newpath) (disk) (back..))'  
solicita el nombre de un fichero

'(askstring (prompt..) (help..) (default..) (back..))'  
solicita una cadena

'(asknumber (prompt..) (help..) (range..) (default..) (back..))'  
solicita un número

'(askchoice (prompt..) (choices..) (default..) (back..))'  
ofrece elegir 1 opción entre varias posibles

'(askoptions (prompt (help..) (choices..) default..) (back..))'  
ofrece elegir n opciones

'(askbool (prompt..) (help..) (default..) (choices..) (back..))'  
0=no, 1=sí

'(askdisk (prompt..) (help..) (dest..) (newname..) (assigns) (back..))'  
solicita la inserción de un disco

'(cat <cadenal> <cadena2>...)'  
devuelve una concatenación de cadenas

'(exists <fichero> (noreq))'  
0 si no existe, 1 si es un fichero, 2 si es un directorio

'(expandpath <senda>)'  
expande una senda abreviada a su equivalente en forma de senda completa

'(earlier <fichero1> <fichero2>)'  
cierto si fichero1 es más antiguo que fichero2

'(fileonly <senda>)'  
devuelve la parte de fichero de una senda (véase 'pathonly')

---

```
`(getassign <nombre> <opciones>)`  
    devuelve el valor del nombre lógico (no utilice `:`)  
    `<opciones>`: 'v'-volúmenes, 'a'-asignaciones, 'd'-unidades  
  
`(getdevice <senda>)`  
    devuelve el nombre de la unidad donde se encuentra la <senda>  
  
`(getdiskspace <senda>)`  
    devuelve el espacio disponible  
  
`(getenv <nombre>)`  
    devuelve el valor de la variable de entorno  
  
`(getsize <fichero>)`  
    devuelve el tamaño  
  
`(getsum <fichero>)`  
    devuelve la suma de comprobación del fichero para hacer una comparación  
  
`(getversion <fichero> (resident))`  
    devuelve el número de versión/revisión de un fichero, librería, etc.  
    como un número de 32 bits  
  
`(iconinfo <parámetros>)`  
    devuelve información sobre un icono (V42.12)  
  
`(pathonly <senda>)`  
    devuelve la parte de directorio de una senda (véase `fileonly`)  
  
`(patmatch <patrón> <cadena>)`  
    ¿ Alude el <patrón> a la <cadena> ? TRUE : FALSE  
  
`(select <n> <elemento1> <elemento2> ...)`  
    devuelve el elemento número n  
  
`(strlen <cadena>)`  
    longitud de la cadena  
  
`(substr <cadena> <inicio> [<cuenta>])`  
    devuelve una subcadena de la <cadena>  
  
`(symbolval <símbolo>)`  
    devuelve el valor del símbolo al que hace referencia la expresión  
    de cadena `<símbolo>` (V42.9)  
  
`(transcript <cadena1> <cadena2>)`  
    inserta las cadenas concatenadas en el fichero de registro  
  
`(tackon <senda> <fichero>)`  
    devuelve senda y fichero convenientemente concatenados
```

---