

Ingeniería

Revista de la Universidad de Costa Rica

ENERO / JUNIO 1991 VOLUMEN 1 Nº 1



INGENIERIA

Revista Semestral de la Universidad de Costa Rica

Volumen I Enero-Junio 1991 Número 1

DIRECTOR

Rodolfo Herrera J.

CONSEJO EDITORIAL

Armando Castro A. (editor)
Víctor Hugo Chacón P.
Gerardo Chacón V.
Miguel Dobles U.
Ronald Jiménez Ch.
Ismael Mazón G.
Domingo Riggioni C.

CORRESPONDENCIA Y SUSCRIPCIONES

Editorial de la Universidad de Costa Rica
Apartado Postal 75
2060 Ciudad Universitaria Rodrigo Facio
San José, Costa Rica.

CANJES

Universidad de Costa Rica
Sistema de Bibliotecas, Documentación e Información
Unidad de Selección y Adquisiciones-CANJE
Ciudad Universitaria Rodrigo Facio
San José, Costa Rica

Suscripción anual:

Costa Rica: ₡500,00
Otros países: US \$20.00

Número suelto:

Costa Rica: ₡250,00
Otros países: US \$10.00





Impreso en la
Oficina de Publicaciones
de la Universidad de Costa Rica

Edición aprobada por la Comisión Editorial de la Universidad de Costa Rica
1991 EDITORIAL DE LA UNIVERSIDAD DE COSTA RICA

Todos los derechos reservados conforme a la ley

Ciudad Universitaria Rodrigo Facio

San José, Costa Rica

Edición Técnica: *Guillermo Loría M.*
Fernando Durán A.

Revisión Filológica: *María Teresa Bolaños*

Montaje Gráfico: *Adalberto Ramírez*

620.005

1-46i

Ingeniería / Universidad de Costa Rica. --

Vol. 1, no. 1 (ene./jun. 1991) -- San José, C. R. : Editorial de la Universidad de Costa Rica, 1991 -- (Oficina de Publicaciones de la Universidad de Costa Rica)

v. : il.

Semestral.

1. Ingeniería -- Publicaciones periódicas.

CCC/BUCR-250



ESTANDARES PARA SISTEMAS DE ARCHIVOS EN COMPUTADORES

José Ronald Arguello V.*

Resumen

Estandarizar el sistema de archivos, en particular la estructura de los directorios manejados en un computador, es un medio para lograr compartir, flexibilizar y hacer eficiente el uso del mismo. En este artículo se justifica la necesidad de tal estandarización, se sugiere un camino para lograrlo en un ambiente de microcomputadores bajo el sistema DOS, se dan alternativas para la nomenclatura de directorios y archivos y se sugiere una forma de definir adecuadamente los ambientes de los usuarios.

Summary

To standardize the file system, specifically the directory structures under control, into a computer is a mean to share, to make flexible and efficient the use of it. In this paper: I plead for the need to such standardization, I suggest a way to achieve it in a microcomputer environment under the DOS system, I offer alternatives for directories and files and I suggest a suitable way for defining user environments.

El presente artículo nació, como una inquietud, durante el desarrollo del proyecto **Árboles de decisión y Modelos de datos**; no como un producto directo de la investigación *per se* sino más bien, un subproducto de la misma. Cuando varios de nosotros -directamente o indirectamente- compartimos un mismo recurso como lo es un computador "personal" -el cual puede prestar valiosos frutos como instrumento compartido- nos creó la necesidad de organizar y compartir información, y también, la necesidad de idear los medios para lograrlo. Paralelamente esta necesidad fue sentida y reflejada en nuestro quehacer docente -en donde la misma falta de recursos impide la búsqueda de soluciones adecuadas.

Así, un subproducto de la investigación tiene sus repercusiones y aunque no fue un fin en sí mismo, no deja de ser tan importante como cualquier producto directo de la misma. Sin embargo, ¿No será acaso nuestra inquietud (o mi inquietud particular) por los árboles de decisión

que lleva indirectamente a comprender y entender la necesidad de las ideas reflejadas en este artículo?

Un agradecimiento a los colegas que desinteresadamente han contribuido a la elaboración de este documento con sus valiosas ideas y comentarios: Prof. David Chaves y Prof. Adolfo Di Mare.

1. INTRODUCCION

UNIX ha logrado un gran auge a nivel mundial por ser tal vez el primer sistema operativo estándar i.e., capaz de ser utilizado en cualquier equipo computacional, permitiendo entonces el aprovechamiento del aprendizaje invertido sobre el sistema operativo cuando se cambia de equipo.

El estándar de UNIX no viene únicamente en su diseño sino también en la forma de organizar su sistema de archivos y en la nomenclatura utilizada. Algo difícil de llevar a cabo tomando en cuenta que el sistema operativo puede hacer poco o nada sobre los nombres que el usuario decida dar a los archivos o directorios utilizados.

Sin embargo, el suponer -por parte de UNIX y los programas que se ejecutan bajo UNIX- que los archivos ejecutables tienen que estar en un directorio llamando /BIN o /USR/BIN permiten reforzar (sino forzar) este estándar. El usuario de

* PIBDC Programa de Investigación en Bases de Datos y del conocimiento.
ECCI-UCR Escuela de Ciencias de la Computación e Informática.
Universidad de Costa Rica.

UNIX tendrá muchos problemas si decide saltarse esto.

Cuando se introducen las microcomputadoras a una institución, poco o nada se piensa sobre estándares. Al fin y al cabo son de uso personal, y lo que yo decida hacer con mi computadora solo me afecta a mí mismo. Hay ciertos estándares que vienen *per se*: IBM compatible por lo general, DOS o cualquier versión necesaria, ciertos programas comunes (lotus, wordstar, smart, pascal, wordperfect, dbase, foxbase, clipper, etc). Hay otros estándares que es necesario reforzar (aprovechando la experiencia UNIX) y que, por ser DOS como es, no se puede tan fácilmente.

2. ESTANDARIZANDO LOS DIRECTORIOS

¿Qué ventajas tiene estandarizar una estructura de directorios?. Casi todos nosotros, usuarios diarios de una microcomputadora, tenemos tantos archivos pequeños como sea necesario (ergo decir que esto fue uno de los principales acicates para el diseño de UNIX). Esto implica necesariamente una fila interminable de discos suaves o bien un disco duro repleto de muchos pequeños archivos (pequeño puede ser considerado aquí poco menos de 100K). Cualquier usuario novato no hace nada por remediar esta situación y el número de sus archivos crecerá y empezará a perder los mismos. La solución es entonces relativamente sencilla, agrupe archivos de acuerdo a su significado (cartas, reportes, listas, etc) y haga un directorio para cada uno de estos grandes grupos. Si la situación continúa, repita el procedimiento.

Y entonces, si la solución está dicha ¿Qué objeto tiene este artículo?. Pues no, la solución no llega aquí. Si la situación continúa, usted llegará a tener tantos directorios y subdirectorios como archivos anteriormente tenía. Un problema viene por añadidura, ¿Cómo documento cada directorio para recordarme de lo que supuestamente iba a poner ahí?. Al proliferar directorios, luego no sabemos como los íbamos a utilizar y lo más común es simplemente no encontrar un archivo (¿Dónde #i\$%|/%\$ puse mi última carta?!).

Todos estos comentarios valen si usted usa su microcomputadora a nivel personal. Otra realidad mucho más compleja se presenta cuando ésta debe ser compartida por un grupo de usuarios, muchas veces no especificado. Tal es el caso de cualquier institución de enseñanza.

Cuando un grupo de usuarios potenciales se dispone a usar una microcomputadora y compartir tanto la máquina como la información en discos suaves o duros, una necesidad de comunicación nace. Si cada uno mantiene el caos descrito anteriormente, la **utilización compartida** es imposible. Así la necesidad de un estándar para el manejo de archivos es evidente y aquí la filosofía Unix puede ser útil.

Si un grupo de usuarios va a utilizar una o varias microcomputadoras, una estructuración adecuada de la jerarquía de directorios es uno de los factores principales que deben ser atacados. Note que no hago incapie si se va a utilizar información en discos suaves o discos duros. Algunas consideraciones de organización física deben ser hechas pero dejémoslas por ahora.

A través de la experiencia he logrado acumular cierta información sobre grupos significativos de archivos (directorios) que pueden ser creados. Algunos pueden diferir de lo aquí expuesto, sin embargo, la filosofía debe mantenerse. Por ejemplo, es común ya la creación de un directorio \DOS que contiene todos los programas propios de la versión del DOS que se está utilizando. De hecho ciertos programas asumen que \DOS es el lugar donde buscar por ciertos programas clave (e.g command.com).

De hecho, si varios usuarios existen, lo mejor es --en lugar de que cada uno disperse archivos por cada unidad de almacenamiento-- un directorio \USR que contendrá todos los directorios de cada uno de los usuarios. Por ejemplo, en mi caso deberá existir un subdirectorio de USR que contendrá todos mis archivos, llamémosle JRAV y el directorio es \USR\JRAV.

Cuando varios usuarios trabajan es bueno crear archivos .BAT que definan comandos rápidos o ambientes propios de cada usuario. Estos archivos pueden ser de utilización general, en cuyo caso un directorio \BAT deberá existir, o de uso particular, en cuyo caso \USR\JRAV\BAT será más adecuado. Esta filosofía es también heredado del UNIX estándar.

Para el lector interesado estoy incluyendo, al final de este artículo, la jerarquía general de directorios con el significado de cada uno. Lo más importante aquí no es crear directorios y directorios, sino justificarlos significativamente de manera que no se hagan redundancias o traslapen directorios. Esto no sería problema si alguna versión futura del DOS incluyera un comando

como el "link" de UNIX que permita tener copias fantasmas de un archivo en diferentes directorios.

La figura 1 muestra como se vería el árbol (jerarquía) con estos directorios. Solo una visión parcial del árbol es incluida, los subdirectorios marcados con * deben contener muchos más directorios.

2.1 Ventajas (+) y desventajas (-)

Auditoría, economía y especialización (+)

Es mucho más fácil controlar el uso de la máquina en esta forma. Controlar el uso del disco duro. Obtener respaldos periódicos sin respaldar archivos innecesarios. Aprovechar mejor los recursos, dado que se pueden distribuir archivos diferentes (programas) en diferentes máquinas y especializar las mismas, dado que no es posible duplicar todos los programas en todas las máquinas sin recurrir a un excesivo desperdicio de los discos.

Disk Volume: ECCI-JRAV88

3-21-89 5:16:20 pm Pag 1

C:\

```

- BAT
- BDC *
- BIN
- CAD *
- DOS
- ETC
- LEN *
- PAQ *
- TMP
- USR
  | - ADH
  | - ANA
  | - CITFG
  | - CRED
  | - DABD
  | - DASI
  | - ECCI
  | - FAG
  | - ILE
  | - JRAV
  | - MARY
  | - PIBDC
- UTL *
- WPR *
```

Actualmente la ECCI en la Universidad de Costa Rica se encuentra en la elaboración de un programa, que siguiendo los estándares y conociendo la organización previa del disco, se avoque a revisar que se mantenga en esa forma y reporte cualquier cambio en estándares u organización. Sin duda será muy útil cuando esté en uso.

Manejo (+)

Existen muchos paquetes en el mercado para la manipulación de directorios. Tal vez el más famoso es el Xtree Professional, el cual permite manipular con todas sus funciones un árbol de directorios. Se puede crear, renombrar, copiar y respaldar árboles de directorios con solo digitar unas cuantas teclas. Aún así, la elaboración de un paquete especializado con estándares es todo una tentación.

Redes (+)

Aceptación (-)

La mayor parte de nosotros estamos acostumbrados a trabajar de nuestro modo y es difícil adaptarnos. Incluso profesionales sienten cierto recelo a usarlo. Yo creo que esto es simple resistencia normal al cambio y no debiera preocuparnos. Una fase de convencimiento previa es necesaria.

Administración (-)

Mantener este estándar requiere una continua vigilancia para evitar quebrantarlo. No hay nada que fuerze este, sólo el convencimiento y el cuidado al crear nuevos directorios.

Utilización (+)

No hay duda que si todo el equipo disponible mantiene una misma estructura de directorios, el compartir archivos y programas se hace más simple y versátil. Ya no tendríamos que revisar todo el disco duro o disco pequeño de una máquina para encontrar un programa a archivo que necesitamos.

FIGURA 1. Ventajas y desventajas.

El objetivo básico para implantar una red local de cómputo es compartir y comunicar datos e información. Estándares de este tipo siguen los mismos objetivos. No hacer esto antes de implantar una red es arriesgarse a algo más que el caos descrito anteriormente.

Paquetes inadecuados (-)

Cuando se utilizan programas y paquetes que no son capaces de ser arrancados desde otro directorio, es necesario recurrir a trucos -por lo general usando la instrucción SUBST del DOS- para llevarlo a cabo. Algunas veces esto es imposible y entonces es necesario trabajar en el directorio del programa con la siguiente proliferación de archivos ahí. Esta dificultad es obviada conforme nuevas versiones más adecuadas de paquetes y programas vayan apareciendo.

3. ESTANDARIZANDO LA NOMENCLATURA DE ARCHIVOS

Tal vez es necio intentar ir más allá de estandarizar los directorios de un sistema de archivos. Sin embargo, dependiendo de las circunstancias, algunos podrían estar interesados en este aspecto.

Los problemas de nombrar archivos a gusto de cada usuario son similares a la creación de directorios: Nombres duplicados, desconocimiento del contenido de los archivos, pérdida de información, dificultades para compartir información, etc. Estos problemas son idénticos bajo cualquier sistema operativo.

Para las rutinas y programas que se están desarrollando para los sistemas de cómputo es clara la necesidad de una nomenclatura estandarizada, por razones ampliamente conocidas en medios administrativos. No es así para la generalidad de los datos almacenados.

Existen estudios e intentos de este tipo que han sido publicados, pero por alguna razón, estos no han sido tan convincentes. Tal vez, la razón más poderosa para tal falta de convencimiento es la ausencia de herramientas automatizadas que permitan mantener esto.

3.1- EXTENSIONES

Extensiones existentes

El único estándar de hecho, que ha sido utilizado bajo DOS, es el de utilizar extensiones

de tipo .COM o .EXE para los archivos ejecutables.

Las otras extensiones estándar son las introducidas por cada paquete de "software" .WK1 para LOTUS, .DBF para DBASE, etc.

En aras de no entrar en conflicto con estos paquetes, lo mejor es no tocar en un estándar de nomenclatura estas extensiones. Para estos paquetes el estándar es respetar las extensiones en estos casos.

Es conveniente entonces, conocer y divulgar las extensiones de archivos que están siendo utilizadas para evitar crear un archivo (o peor aún, un programa) que utilice extensiones ya utilizadas por otros programas.

Extensiones vacías

El no usar una extensión para denominar un archivo es sinónimo de archivo temporal y que puede ser eliminado en cualquier momento. Aparte de dar mala apariencia al directorio, elimina la posibilidad de uso de comandos que utilizan las extensiones, se pierden tres caracteres de información -vitales cuando se maneja gran cantidad de archivos similares.

Extensiones .L99

La primera letra (L) de la extensión puede utilizarse para otros tipos de significado: C - para cartas, F- para fórmulas, D - para documentos, P para partes, etc.

Los otros dos caracteres, cuando son números deberán significar, versiones, partes o secuenciación de archivos de una misma clase. Por ejemplo: Archivos que tienen un carácter histórico, pueden utilizar la extensión como V01, V02, etc para manejo de versiones.

Extensiones estándar.

Así los estándares a utilizar son:

.TXT para archivos de texto en ASCII,
 .TMP para archivos temporales de cualquier tipo,
 .PRN para archivos a imprimir,
 .OBJ para programas objeto,
 .MSJ para archivos con mensajes, o formato
 .L99 para control de versiones o partes de un archivo.

Debe respetarse extensiones ya utilizadas por lenguajes y paquetes de propósito general.

Sin embargo, legislar más sobre las extensiones es hilar demasiado fino.

3.2- NOMBRES

Los primeros 8 caracteres de un archivo son los conocidos como el nombre (propriadamente dicho) del archivo. Esta limitación del DOS implica necesariamente un estándar algo restricto. Durante mi experiencia he logrado visualizar un estructura de nombre de la siguiente forma:

AAAAXXXX

Los primeros 4 caracteres deben ser 4 letras más significativas del contenido del documento, e.g. CART para carta, para el destinatario, dueño, o autor del documento, puede ser CCSS (Caja Costarricense Seguro Social), UCOR (Universidad de Costa Rica), etc. Estos deben ser designados localmente de acuerdo a cada ambiente o bien a gusto del usuario y tal vez lo más importante significativamente.

Algunos se preguntarán por qué cuatro letras y no dos o tres. Mi experiencia muestra que muchas abreviaturas significativas pueden obtenerse adecuadamente con cuatro letras.

A manera de ejemplo lea el siguiente párrafo:

Si esto crct son esco con un poqo de cuid
cad vez que cres un nuev arch, faci pode
recd de que era el asun, loca un docu, rspl,
etc.

(Si estos caracteres son escogidos con un poquito de cuidado cada vez que creamos un nuevo archivo, fácilmente podemos recordar de que era el asunto, localizar un documento, respaldar, etc.)

Obviamente, puede parecer dificultoso y caprichoso a una primera instancia, pero se supone que nunca es la primera vez (como en el ejemplo anterior) que un usuario es enfrentado al estándar o abreviatura escogida. Siempre hay un conocimiento previo de los estándares. Si no, amigo lector, lea el texto de ejemplo anterior por segunda vez.

4.- ESTANDARIZANDO LA DEFINICION DE LOS AMBIENTES DE LOS USUARIOS.

El directorio \BAT tiene innumerables usos, uno de ellos es el de crear archivos de instruc-

ciones que contengan la definición del ambiente para cada usuario de la máquina — aquellos que tengan un subdirectorio bajo \USR. Así, si un usuario es codificado bajo el nombre JRAV i.e., existe un directorio \USR\JRAV, entonces debe existir un archivo JRAV.BAT en el directorio \BAT.

Desde este punto de vista el archivo de arranque del sistema AUTOEXEC.BAT debe contener lo mínimo indispensable para iniciar el computador, así como proveer suficiente flexibilidad para que cualquier usuario no quede amarrado con un ambiente predefinido.

Luego los archivos *.BAT en el directorio \BAT (que deben ser llamados desde el archivo de instrucciones AUTOEXEC.BAT) tendrán el complemento de instrucciones necesarias para definir un ambiente particular para cada usuario.

A manera de ejemplo, el siguiente archivo AUTOEXEC.BAT puede ser utilizado:

```
echo off
type AUTOEXEC.MSJ
\utl\nu\ask "Opcion?(1,2,3,4,5)", 12345
if errorlevel 5 goto 5
if errorlevel 4 goto 4
if errorlevel 3 goto 3
if errorlevel 2 goto 2
if errorlevel 1 goto 1
:5
\utl\pop\popdos2
\utl\prq\printq
:4
\utl\pop\popalarm
\utl\sk\sk
:3
\utl\var\vcache 32
:2
\utl\var\dosedit
\dos\keybsp
:1
set EXINIT = set eightbits native
\utl\var\lea _MSJ
path c:\bat;
prompt "Usuario?"
echo on
```

El archivo AUTOEXEC.MSJ contiene información sobre las opciones disponibles:

¿Qué desea cargar?					Utilitarios
Opciones	2	3	4	5	
1					
Nada	Tecl.Español Hist.Comando				Keybsp Dosedit Vcache
603k libres	599k libres	+ Disco Rápido	+ Alarma + Proc/Calc/Calen.	+ Colas de Impr. + Ventana Dos	Popalarm Sidekick Printq Popdos
		563k libres	440k libres	373k libres	

El archivo `._MSJ` es un archivo que puede ser "TYPE-ado" u hojeado a través de un comando que permita leer el archivo de principio a fin programa LEA en `\utl\var`. Este archivo contiene mensajes generales de utilidad para cualquier usuario. Por ejemplo puede contener esta misma información sobre el manejo de la máquina.

Después de arrancado el sistema, el mensaje

Usuario: ?

aparecerá en la pantalla indicando al usuario que debe entrar su código respectivo -por ejemplo JRAV-, lo que provocará que el archivo de instrucciones JRAV.BAT sea llamado y el ambiente de JRAV definido. Tal archivo -aunque diferente de usuario a usuario- podría seguir los siguientes lineamientos:

```
echo off
path \bat;\dos;\len\pro;\utl\nu;\bdc\3pl;
\wpr\ws;\utl\prt;(cont)
\paq\smt;\paq\lot;\len\bas;\bdc\clp;\bin;
\utl\var;\utl\xt
cd \usr\jra
prompt $p$g
cls
set comspec = \dos\command.com
set EXINIT = set eightbits native
type JRAV.MSJ
```

El archivo JRAV.MSJ reside en el directorio `\USR\JRAV` de manera que puede ser usado por cualquier usuario o programa para dejar mensajes

al usuario JRAV. Un archivo similar debe existir para cada usuario.

CONCLUSIONES

Temo que no todo lo dicho aquí es suficiente para asegurar una adecuada utilización de los microcomputadores. Ciertos aspectos pueden ser mejorados y ampliados de acuerdo a las necesidades individuales pero los principios mencionados deben prevalecer. Cualquier estándar para que sea mantenible debe ser flexible y poder ser actualizado frecuentemente. Algunos encontrarán algo engorroso el tener que digitar varias teclas antes de empezar a hacer uso efectivo de su computador, pero más engorroso sería su situación al tratar de usar un computador ajeno y no poder hacerlo del todo como estoy seguro a muchos les ha sucedido.

BIBLIOGRAFIA

1. EL IBM-PC. Jonathan Sachs. Mc Graw Hill de México. México 1985.
Contiene todos los aspectos básicos que alguien desee conocer sobre la pequeña máquina y su organización.
2. El entorno de programación UNIX. B.Kernighan. Rob Pike. Prentice Hall. 1987.
Una referencia clásica al sistema UNIX estándar.
3. COMPUTER IN FOCUS. R. H. Austing, L. Cassel Brooks/Cole Publishing Company. Monterrey, California. 1986.
Una introducción didáctica a todos los aspectos relacionados con los computadores, microcomputadores y grandes computadores en esta época. Discute tópicos elementales -desde representación de datos y programación- hasta tópicos calientes como redes, sistemas operativos e inteligencia artificial.

APENDICE

LISTA DE DIRECTORIOS CON SU SIGNIFICADO.

C:\	Directorio raíz.
-BAT	Comandos en lote (*.bat).
-BDC	Progr. Bases de Datos y Conoc.
- 3PL	Dbase 3 Plus
- DOC	DOCumentador de dbase.
- GENM	GENErador de Menús.
- QR	Quick Report de dbase.
- VWG	VieW Generator de dbase.
- CLP	CLiPper
- FOX	FOXbase
- GURU	Generador de sistemas expertos GURU.
'- DEMO	DEMOstrador de guru.
- M1	Mycin 1
- R5K	Rbase 5000 (5K)
'- VPE	VP-Expert
-BIN	Código BINario ejecutable(de UNIX-pc)
-COM	COMunicaciones
- KER	KERmit
- OMG	OMGate
'- PCW	PC-Works
-DAC	Diseño Asistido por Computadora
- ACAD2	Automatic Comp. Aided Design 2 dim.
- DIBU	Fuentes de DIBUjos.
'- DRIVES	Manejadores de periféricos.
- ACAD3	Automatic Comp. Aided Design 3 dim.
- EFLOW	Easy FLOW.
- FLOW	FLOW
- FORM	FORMtools
'- PRODES	PROfessional DESign
-DOS	archivos del DOS.
-ETC	archivos ETC de UNIX-pc.
-JUEGOS	programas de JUEGOS
-LEN	LENguajes
- BAS	BASic y versiones
- COB	COBoL
- LSP	LiSP
- MD2	MóDula dos.
- P86	Prolog 86.
- PRO	Turbo PROlog
'- TOOLS	TOOLS de turbo-prolog
- RCB	RM-CoBol
'- LIB	Rm-Cobol LIBrary.
- TCC	TurboC Compiler
- BIN	Código BINario ejecutable
- INCLUDE	Archivos para INCLUDE
'- SYS	rutinas del SYStema
'- LIB	Bibliotecas de C (LIBraries)

'-	PAS	PAScal
	- GRAPHIX	rutinas de gráficos
	- TOOLBOX	Caja de herramientas (TOOLS)
	- TUR3	TURbo pascal versión 3
	- TUR5	TURbo pascal versión 5
-	PAQ	Paquetes de programas
	- LOT	LOTus
'-	SMT	SMarT
-	PUB	Paquetes para PUBLICaciones
'-	NEWS	NEWS
-	TEMP	archivos temporales (sin ubicación)
-	TMP	archivos temporales de programas
-	UNIX	sistema operativo UNIX-pc
-	USR	directorio de usuarios
	- ANA	Usuario llamado ANA
	- JRAV	Usuario llamado JRAV
		subdirectorio CUR (de ejemplo)
	- CUR	idem
	- ECCI	idem
	- PAS	idem
	- PER	idem
	- PRO	idem
	- WK1	idem
.		
.		
-	UTL	UTiLitarios
	- CPW	CoPyWrite
	- DVW	DeskView
	- KEY	super-KEY
	- NU	Norton Utilities
	- POP	POP-dos utilities
	- PRQ	PRint-Queue
	- IMP	utilitarios de IMPresión
	- SDW	SiDeWays
	- SK	SideKick
	- VAR	utilitarios varios
	- XT	XTree Professional
-	WPR	WordProcessors
	- WS	WordStar
	- WP	WordPerfect