

# ELABORACIÓN DE CONCORDANCIAS Y LÉXICOS DE AUTOR EN MICROORDENADOR (\*)

Manuel Martínez Quintana

## INTRODUCCIÓN

### a) Los procesadores de textos con lenguaje de programación

En un trabajo anterior (*Estudios Clásicos* n.º. 93, 1988) expusimos el proceso de elaboración de un «Index Verborum» en microordenador. A pesar de la complejidad que supone el hecho de tener que programar las distintas fases del proceso, la tarea es perfectamente realizable en un ordenador de pequeñas dimensiones, si se emplea un procesador de textos dotado de lenguaje de programación, como el XyWrite III Plus, uno de los más rápidos y potentes del mercado o el AppleWriter II con su WPL (Word Processing Language), de prestaciones ilimitadas. Se puede utilizar también un procesador que tenga la posibilidad de crear macros, función existente en muchos procesadores de textos actuales como el Word de Microsoft, el Multimate Advantage II, el Nota Bene, el WordStar 2000 Plus y, sobre todo, el WordPerfect, que se ha convertido en un standard del mercado actual de

---

(\*) Un resumen de este artículo fue publicado en la revista *Universidad Abierta*, núm. 10, 1989.

programas de textos. A excepción de AppleWriter II, los demás requieren alrededor de 320 Kbytes de memoria RAM (\*\*).

## b) Los contextos mecánicos

Intentamos exponer en este trabajo el procedimiento técnico para la elaboración de concordancias y léxicos de autor. Las concordancias ofrecen listas alfabéticas de todas las palabras de un texto con los contextos correspondientes; los léxicos constan de artículos estructurados en cuatro partes: lema; componente gráfico y fónico; componente morfo-sintáctico y componente semántico. El componente semántico es propiamente el cuerpo del artículo, donde hay que definir el significado de la palabra y clasificar las acepciones según el criterio que se adopte.

Los contextos extraídos por el programa que hemos confeccionado son mecánicos, es decir, a cada palabra le antecede y le sigue un número fijo de palabras «contextuales», algunas de las cuales, obviamente, en algunos casos, serán irrelevantes para la comprensión del enunciado y, por el contrario, en otros, el contexto podría ser insuficiente por falta de palabras. Esta dificultad se atenúa si tenemos en cuenta que el programa permite elegir la extensión del contexto. Eligiendo ocho o diez palabras, por ejemplo, para el contexto anterior y otras tantas para el posterior, se tiene normalmente un contexto global suficiente para la comprensión del enunciado. El límite en la elección de este número no está en el programa sino en la capacidad de almacenamiento del ordenador.

Como es un término que se repite frecuentemente, llamamos «unidad contextual» al conjunto de contexto anterior + palabra-clave + referencia + contexto posterior.

## c) El soporte material o «hardware»

El material necesario para estas tareas es una UPC con capacidad proporcional a las necesidades de memoria RAM exigida por el procesador de textos que se utilice, dos unidades de disco flexibles o, mejor, un disco duro,

---

(\*\*) Cuando en 1988 fue redactado este trabajo los lenguajes de macros no ofrecían los procedimientos de programación suficientes para llevar a cabo el proceso que aquí se describe. En la actualidad, aunque los programas que presentamos pueden ser reescritos utilizando macros, las fases del proceso y el flujo del programa permanecen idénticos.

normalmente requerido por los procesadores de textos que hemos citado; con el disco duro se consigue un mayor rendimiento del procesador de textos, pues facilita notablemente la gestión de ficheros. En el presente trabajo hemos empleado el microordenador APPLE IIe, de 128 Kbytes de memoria RAM y dos lectores de discos flexibles, el procesador de textos APPLE-WRITER II con su lenguaje de programación WPL (Word Processing Language) y una impresora IMAGEWRITER II, también de Apple Computer.

## EL REGISTRO DEL TEXTO

### a) Los «scanners» y los programas de lectura óptica

El punto de partida de este nuevo trabajo es también el registro del texto. Si entonces decíamos que la primera tarea era teclear el texto, en el momento actual podemos asegurar que ya es posible liberar el lexicógrafo de esta tarea. El empleo del ordenador combinado con un «scanner» puede hacer el trabajo a un precio razonable. En esencia, este aparato permite tomar un material que ha sido previamente escrito, dibujado o fotografiado para reutilizarlo en formato electrónico, sin necesidad de tener que reescribirlo, redibujarlo o volver a fotografiarlo.

Si se trata de lectura de caracteres, este dispositivo, conocido con las siglas OCR (Optic Characters Reader), rastrea el texto como si se tratara de una página bitonal, pero su «software» no debe únicamente procesar la imagen, sino que también ha de reconocer los caracteres de texto que la imagen representa y convertirlos a un formato ASCII o a algún formato reconocible por los procesadores de texto.

Existen numerosos programas de lectura óptica, entre cuyas especificaciones se pueden destacar las siguientes:

- Alta fiabilidad: Margen de error inferior al 1/1000.
- Velocidad: Más de 500 caracteres por minuto.
- Velocidad/Fiabilidad regulable.
- Control de espacios.
- Compatibilidad con la práctica totalidad de «scanners» y digitalizadores profesionales.
- Interpretación de cualquier tipo de textos en caracteres de imprenta sin importar su tipografía. Incluidos documentos dactilografiados.
- Juego de caracteres cargable y «editable» por el usuario.
- Reconocimiento de «kerning» (letras imbuidas).
- Resolución mínima de 300 ppp (puntos por pulgada).

## b) Condiciones del texto de entrada

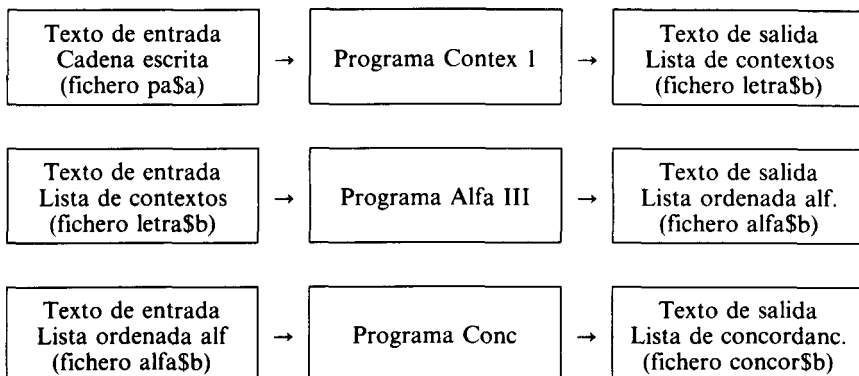
El texto de entrada se compone de bloques separados por un solo espacio. Cada bloque consta de una palabra o forma y de la referencia, separada por una marca (aquí hemos empleado el asterisco) sin solución de continuidad. Las palabras deben comenzar siempre con minúsculas, de otro modo habría problemas para la ordenación alfabética del texto ya que las mayúsculas van delante de las minúsculas en el código ASCII. Se pueden mantener los signos de puntuación. Las formas compuestas de la conjugación, perifrasias verbales, nombres propios de más de una palabra serán registrados como una sola palabra en su elemento básico o más representativo; los numerales de más de una palabra llevarán un registro en cada uno de los elementos componentes en el orden en que aparecen en el texto.

La referencia lleva en primer lugar las siglas convencionales de la obra y la indicación tradicional de libro, capítulo etc. Cada elemento de la referencia debe estar separado por una marca cualquiera que no sea la coma, si se usa Applesoft; después, esta marca puede ser convertida en coma por medio de la instrucción /Find/ (encontrar y cambiar) existente en todos los procesadores de textos. El programa que nos permite colocar automáticamente la referencia, incluida la numeración mecánica de las palabras, lo expusimos detalladamente en el trabajo aludido. El conjunto de forma y referencia formaría un bloque y ofrecería este aspecto: «auctoribus\*DmH<105>6». La palabra «auctoribus» pertenece a la obra *De motu Hispaniae* de Juan Maldonado y se encuentra en la página 105, línea 6. El programa que hemos empleado en el trabajo citado para colocar la referencia se puede adaptar sin ningún problema, cuando ésta incluye la indicación de libro, capítulo, párrafo, etc.

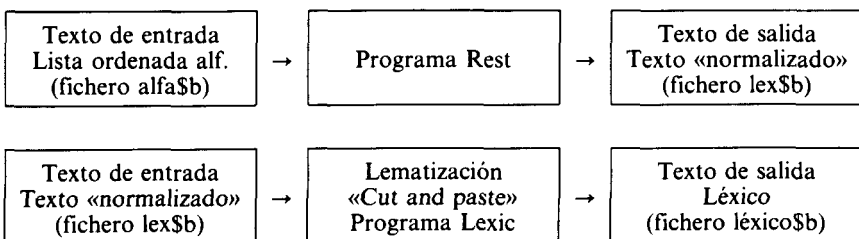
## LA EXTRACCIÓN DE CONTEXTOS MECÁNICOS (Programa Contex 1)

El programa Contex 1 tiene como finalidad extraer mecánicamente contextos de extensión variable elegida por el usuario y colocar la palabra «contextualizada» en la cabecera del contexto anterior para facilitar la ordenación alfabética. Este programa es la base para la realización tanto de concordancias como de léxicos. A grandes rasgos, el proceso tendría los siguientes pasos:

## a) Elaboración de concordancias



## b) Elaboración de léxicos



El texto de partida, para ser procesado por el programa Contex 1, tiene que constar, como hemos dicho antes, de una serie de bloques separados por un solo espacio. Aquí partimos de unos ficheros pequeños que se corresponden cada uno con una página. Por ejemplo, el fichero correspondiente a la página 162 del *De motu Hispaniae* de Juan Maldonado ofrecería este aspecto (presentamos únicamente las cinco primeras líneas):

```
praesentia*DmH<162>01 tantum*DmH<162>01 munera*DmH<162>01 trahunt,*
DmH<162>01 pollicitis*DmH<162>01 nulla*DmH<162>01 fides.*DmH<162>
01 condiciones*DmH<162>01 tamen*DmH<162>02 plerique*DmH<162>02 fe-
rebant:*DmH<162>02 prodituros*DmH<162>02 obuiam*DmH<162>02 velasco*
DmH<162>02 gratulabundos*DmH<162>02 dictoque*DmH<162>02 parituros*
DmH<162>03 ueluti*DmH<162>03 proregi.*DmH<162>03 modo*DmH<162>
03 certi*DmH<162>03 essent*DmH<162>03 uel*DmH<162>03 tria*DmH<162
>03 quae*DmH<162>03 maxime*DmH<162>03 cuperent*DmH<162>03 habi-
```

turos\*DmH < 162 > 04 a\*DmH < 162 > 04 rege,\*DmH < 162 > 04 nimirum\*DmH < 162 > 04 ut\*DmH < 162 > 04 essent\*DmH < 162 > 04 immunes\*DmH < 162 > 04 ac\*DmH < 162 > 04 liberi\*DmH < 162 > 04 a\*DmH < 162 > 04 praestandis\*DmH < 162 > 04 ugratuitis\*DmH < 162 > 05 hospitiiis\*DmH < 162 > 05 aulicae\*DmH < 162 > 05 familiae\*DmH < 162 > 05 cum\*DmH < 162 > 05 rex\*DmH < 162 > 05 diuerteret\*DmH < 162 > 05 burgos;\*DmH < 162 > 05 tum\*DmH < 162 > 05

El programa pide primero la letra inicial de las palabras cuyo contexto va a extraer, a continuación, el número de la página (o del fichero), y finalmente la extensión del contexto «ante» y del contexto «post». Una vez obtenida la lista de contextos de las palabras que empiecen por «a» de la página 162, por ejemplo, el ordenador pedirá el número de la página siguiente (este paso se puede automatizar) para obtener la lista de las palabras que comiencen por «a» de la página 163, lista que añadirá al fichero anterior y así sucesivamente; cuando se hayan terminado los ficheros, comenzará otra vez por el fichero n.º 1 para extraer las palabras que comiencen por «b» y repetirá el proceso. La lista de las palabras que comienzan por «a» de la página 162 de la obra referida con sus contextos es la siguiente:

a rege' tria quae maxime cuperent habituros @ nimirum ut essent immunes \*DmH < 162 > 04  
ac liberi rege' nimirum ut essent immunes @ a praestandis gratuitis hospitiiis \*DmH < 162 > 04  
a praestandis ut essent immunes ac liberi @ gratuitis hospitiiis aulicae familiae \*DmH < 162 > 04  
aulicae familiae liberi a praestandis gratuitis hospitiiis @ cum rex diuerteret burgos; \*DmH < 162 > 05  
ad uendum die perpetuo burgis mercatus esset @ et emendum sine ulla \*DmH < 162 > 06  
animaduersio nihil remittereturque omnium flagitiorum supplicium et @. non facile consecuturum velascum \*DmH < 162 > 08  
allaturum interim certos intra dies diploma regium @; humanitatis esse prudentia eque burg \*DmH < 162 > 11  
agebant primores tabellariorum regiae partes infirmarentur. hoc @ ac diuites ultro citro \*DmH < 162 > 13  
ac diuites partes infirmarentur. hoc agebant primores @ ultro citroque commeando nec \*DmH < 162 > 13  
anni quo quae conuenerant. Kalendis nouembris eius @ motus coeptus est' recipitur \*DmH < 162 > 22  
ac uestibus obuam primoribus phaleratis in equis @ picturatis' quo dynastae facerent \*DmH < 162 > 23

Nótese que la palabra «contextualizada» y la que le sigue han sido colocadas provisionalmente al inicio de la unidad contextual para facilitar la ordenación alfabética y que su sitio está ocupado mientras tanto por la marca /@/. Las comas han sido convertidas también provisionalmente en apóstrofes para evitar problemas en el BASIC de Applesoft, lenguaje en el que está escrito el programa de ordenación alfabética. Arbitrariamente se

han elegido cinco palabras para cada contexto, podía haber sido otro número.

A esa lista hemos llegado después de aplicar el programa principal Context 1 cuyo listado, con las explicaciones oportunas, ofrecemos a continuación:

### Programa Context 1

#### Carga de variables

psyl	(asigna 1 a la variable (y))
pin ¿Qué letra? = \$b	(asigna una letra a la variable \$b)
r pin ¿Número de la página? = \$a	(asigna un número a la variable \$a)
pin ¿Cuánto contexto ante? = \$c	(asigna un número a la variable \$c)
pin ¿Cuánto contexto post? = \$d	(asigna un número a la variable \$d)
pnđ	(no se visualiza el proceso en pantalla)
lpa\$a,d2	(carga el texto del disco 2)
f/ \$b/ \$b/	(búsqueda de palabras con inicio \$b-)
o?	(acepta la instrucción anterior con doble condición)
pgo q	(si encuentra, ejecuta esta instrucción; si no, la salta)
pgo s	(va a la etiqueta s)

#### Creación de espacios al final del fichero

q e	(colocación del cursor al final del texto)
d	(cambia la dirección de la flecha en la línea de datos)
psx 1	(asigna 1 a la variable (1))
m f// /	(crea dos espacios)
o?	(acepta con doble condición, en este caso siempre sí)
pcs/ \$d / (x) /	(compara las variables \$d y (x))
pgo n	(iguales, ejecuta esta instrucción; distintas, la salta)
psx + 1	(incremento de 1 en la variable (x))
pgo m	(va a la etiqueta m y repite el proceso)

## Creación de espacios al principio del fichero

n b	(colocación del cursor al principio del texto)
d	(cambia la dirección de la flecha)
psx l	(asigna l a la variable (x))
p f// /	(crea dos espacios)
o?	(acepta con doble condición, en este caso siempre sí)
pcs/ \$c / (x) /	(compara las variables \$c y (x))
pgo o	(iguales, ejecuta esta instrucción; distintas, la salta)
psx + l	(incremento de l en la variable (x))
pgo p	(va a la etiqueta p y repite el proceso)

Comienza la extracción de contextos; se marca la palabra-clave, el principio y el final del segmento

## Colocación de la marca central

o psx l	(asigna l a la variable (x))
b	(colocación del cursor al principio del texto)
a f/ \$b/ (x)b \$b/	(búsqueda de \$b- y colocación de la marca central (x)b)
o?	(acepta con doble condición, en este caso siempre sí)

## Colocación de la marca posterior

psz l	(asigna l a la variable (z))
b f// /	(búsqueda del primer espacio)
o?	(acepta con doble condición, en este caso siempre sí)
pcs/ \$d / (z) /	(compara las variables \$d y (z))
pgo w	(iguales, ejecuta esta instrucción; distintas, la salta)
psz + l	(incremento de l en la variable (z))
pgo b	(va a la etiqueta b y repite el proceso)
w f/ / (x)c /	(colocación de la marca posterior (x)c)
o?	(acepta con doble condición, en este caso siempre sí)
d	(cambia de dirección la flecha de la línea de datos)
f/ (x)b / (x)b /	(búsqueda de la marca central)



o?	(acepta con doble condición, en este caso siempre sí)
Ctrl-U	(carácter de control, hace avanzar un espacio al cursor)

### Colocación de la marca anterior

psz l	(asigna l a la variable (z))
c f/ / /	(búsqueda del primer espacio)
o?	(acepta con doble condición, en este caso siempre sí)
pcs/ \$c / (z) /	(compara las variables \$c y (z))
pgo g	(iguales, ejecuta esta instrucción; distintas, la salta)
psz + l	(incremento de l en la variable (z))
pgo c	(va a la etiqueta c y repite el proceso)
g f/ / (x)a /	(colocación de la marca anterior (x)a)
o?	(acepta con doble condición, en este caso siempre sí)
pcs/ (x) / l /	(compara la variable (x) con l)
pgo d	(iguales, ejecuta esta instrucción, distintos, la salta)
pgo f	(va a la etiqueta f)

### Extracción de contextos

d b	(colocación al principio del texto)
l #/(x)a/(x)c/	(carga del fichero en memoria el primer segmento entre marcas)
f!#!	(crea un «return»: # = «return»)
o?	(acepta con doble condición, en este caso siempre sí)
pgo h	(va a la etiqueta h)
f f!#!	(busca el «return»)
?	(termina la búsqueda)
d	(cambia la dirección de la flecha)
f!#!	(crea un «return»)
o?	(acepta con doble condición, en este caso siempre sí)
l#/(x)a/(x)c/	(carga los segmentos restantes)

### Borrado de las marcas del texto de partida

h f/ (x)a // o?	(borra la primera marca)
f/ (x)c // o?	(borra la marca posterior)
d	(cambio de dirección)
f/ (x)b // o?	(borra la marca central)

### Repetición del proceso

d	
f/ \$b/ \$b/ o?	(búsqueda de la próxima palabra con inicio \$b)
pgo t	(si la encuentra, ejecuta la instrucción; si no, la salta)
pgo j	(va a la etiqueta j)
t psx + 1 pgo a	(va a la etiqueta a y repite el proceso)

Sustitución del \* por el % en la palabra-clave para facilitar el borrado de la referencia en las demás palabras

j b	
psz 1	
x f/(z)b/(z)b/ o?	(búsqueda de la primera marca central)
pgo k	(si sí, la ejecuta; si no, la salta)
pgo l	(va a la etiqueta l en caso de no)
k f/*/%/ o?	(sustituye el * por %)
psz + 1 pgo x	(va a la etiqueta x y repite el proceso)

### Borrado del texto de partida

l e	(colocación al final del texto)
f!#! o?	(crea un «return»: # = «return»)
pin Cuántos ctrl-x?= \$d (\$d = caracteres del texto/1024)	
psz \$d	(asigna el número \$d a la variable (z))

z pcs/(z)/O/ pgo y	(compara la variable con O) (iguales, ejecuta esta instrucción, distintas, la salta)
x	(borra hasta el próximo «return», un máximo de 1024 carct.)
psz - 1	(decremento de la variable)
pgo z	(va a la etiqueta z y repite el proceso)

Borrado de las referencias, salvo palabra-clave, y restitución de la marca separadora de la referencia

y f!*« ! !a	(borrado de todos los segmentos comprendidos entre * y espacio)
o?	
f/%/*/a	(restitución de todas las marcas primitivas en la palabra-clave)
o?	

Subrutina para cambiar de lugar los signos de puntuación de la palabra-clave

e pgo sal	(va a la etiqueta sal: programa principal)
e e	(colocación del cursor al final y la flecha dirección izq.)
dl f/\$b*/*/	(supresión del contenido de la variable \$b)
o?	
pgo al	(si sí, la ejecuta; si no, la salta)
pgo bl	
al f/ /\$b /	(traslado del contenido de \$b al espacio anterior)
o?	
pgo dl	(repetición del proceso)
bl prt	(vuelta al programa principal)

Programa principal en el que se asignan sucesivamente los signos de puntuación a la variable \$b

sal pas. = \$b	
psr e	(va a la subrutina e)
pas. = \$b	
psr e	(va a la subrutina e)
pas; = \$b	

psr e	(va a la subrutina e)
pas" = \$b	
psr e	(va a la subrutina e)
pas! = \$b	
psr e	(va a la subrutina e)
pas? = \$b	
psr e	(va a la subrutina e)

Supresión de la palabra-clave, de su referencia y de la palabra siguiente; restitución de las dos palabras en la cabecera del contexto anterior para su ordenación alfabética y de la referencia al final del contexto posterior

Traslado de la referencia

psx l	
b	
an f/* / *	(creación de un espacio para extraer la referencia)
o?	(ejecución condicional)
pgo bn	
pgo xn	
bn f / /	(colocación del cursor en el espacio posterior de la ref.)
?	(termina la búsqueda)
d	(cambio de dirección a la izquierda)
Ctrl-W	(carácter de control que suprime la referencia)
d	(cambio de dirección a la derecha)
f!#	(busca el final del contexto posterior)
?	(termina la búsqueda)
Ctrl-W	(restituye la referencia)
psx + l	
pgo an	(etiqueta an y repetición del proceso)

Traslado de la palabra-clave y de la que le sigue

xn b	
psx l	
tn f/(x)b/@/	cambia la marca central variable por la fija @)
o?	
pgo ln	
pgo cn	

In f/ / /	(sobrepasa al primer espacio)
o?	
f/ / /	(sobrepasa a la palabra-clave)
o?	
f/ /	(sobrepasa a la siguiente)
?	(termina la búsqueda)
d	(cambio de dirección para posibilitar la supresión)
Ctrl-W	(suprime la última palabra)
Ctrl-W	(suprime la primera)
f/(x)a /	(va a la cabecera del contexto anterior)
?	
d	(cambia de dirección para posibilitar la restitución)
f/ /	(busca un espacio)
?	
Ctrl-W	(restituye la palabra-clave)
Ctrl-W	(restituye la siguiente)
psx + 1	
pgo tn	(repetición del proceso)

### Borrado de las marcas de los contextos

cn b	
psx 1	
gn f/(x)a / /	(borrado de la marca anterior)
o?	
?	
f/ (x)c//	(borrado de la marca posterior)
o?	
pgo el	
pgo fn	
el psx + 1	
pgo gn	(repetición del proceso)

### Operaciones diversas

fn psz (x)	(asigna a la (z) el valor numérico de (x))
psz - 1	(decremento de la última unidad añadida a (x))
pin Núm. de contex: (z)	(información del número de contextos extraídos para dimensionar el programa de ordenación alfabética)

b	
f/,/'/a	(eliminación provisional de las comas para evitar problemas con el Applesoft)
o?	
pgo il	
pgo il	

Grabación de los ficheros
---------------------------

il pcs/(y)/l/	(compara la variable (y) con l)
pgo ac	(iguales, ejecuta la instrucción; distintas, la salta)
pgo ol	
ac sletra\$b,d2	(graba el primer fichero en el disco 2)
pgo un	
ol sletra\$b,d2+	(graba el siguiente añadiéndolo al anterior en el disco 2)
un no	(borra la memoria)
psy + 1	
pgo r	(va a la etiqueta r y repite el proceso)
s qt	(deja el programa)

## ORDENACIÓN ALFABÉTICA

Si disponemos de unos ficheros adecuados a la potencia del microordenador, podemos proceder a la ordenación alfabética utilizando el método del intercambio, que consiste en comparar dos nombres. Si están bien ordenados, se para al siguiente, y si no, mediante una variable auxiliar, se realiza el intercambio y luego se pasa al siguiente. Para realizar la comparación, se utilizan dos bucles anidados, uno exterior que recorre desde el primer nombre hasta el penúltimo y el interior que sirve para realizar las comparaciones.

```

10 REM **** ALFA III ****
20 INPUT «¿NOMBRE DEL FICHERO?»;F$: PRINT
30 IF F$ = «» THEN 270
40 INPUT «¿NOMBRE DEL FICHERO DE DESTINO?»;M$: PRINT
50 INPUT «¿CUANTAS PALABRAS TIENE LA LISTA?»;N: PRINT
60 DIM A$(N): D$ = CHR$(4)
70 D = 2
80 PRINT D$;«OPEN»;F$;«D»;D: PRINT D$;«READ»;F$

```

```

90 FOR I = 1 TO N
100 INPUT A$(I)
110 NEXT I
120 FOR I = 1 TO N - 1
130 FOR J = 1 TO N - 1
140 IF A$(J) < = A$(J+1) THEN 180
150 X$ = A$(J)
160 A$(J) = A$(J+1)
170 A$(J+1) = X$
180 NEXT J
190 NEXT I
200 PRINT D$;«OPEN»;M$;«D»;D
210 PRINT D$;«WRITE»M$
220 FOR I = 1 TO N
230 PRINT A$(I)
240 NEXT I
250 PRINT D$;«CLOSE»M$
260 GOTO 20
270 END

```

## LAS CONCORDANCIAS (Programa Conc)

Una vez ordenada alfabéticamente la lista de los contextos, el programa *Conc* realiza funciones esencialmente de edición e impresión de tal manera que las palabras concordantes queden en el centro aisladas por dos espacios de sus contextos; la referencia aparecerá colocada en el margen derecho. En realidad se trata de imprimir en la columna izquierda la lista de los contextos anteriores y en la derecha, la lista de las palabras concordantes, la de los contextos posteriores y la de la referencia. Suponiendo una lista más amplia y uniforme, la concordancia presentaría este aspecto después de la aplicación del programa *Conc*:

etiamsi remotis in regionibus, procul	a patria, militent, quod eos nimirum	DmH 163 17
ut essent immunes ac liberi	a praestandis gratuitis hospitiiis aulicae familia	DmH 162 04
stipendium idque serum multisque modis	a quaestoribus imminutum. Tum autem cum	DmH 163 15
tria quae maxime cuperent habituros	a rege, nimirum ut essent immunes	DmH 162 04
partes infirmarentur, hoc agebant primores	ac diuites ultro citroque commendo nec	DmH 162 13

rege, nimirum ut essent immunes	<b>ac liberi a praestandis gratuitis hospitiiis</b>	DmH 162 04
obuiam primoribus phaleratis in equis	<b>ac uestibus picturatis, quo dynastae facerent</b>	DmH 162 23
habere delectus, pecunias imperare, iuuentutem	<b>ad arma cogere, toletanis obsistere, qui</b>	DmH 164 10
traxit donec discessurus in germaniam	<b>ad fasces imperii primamque purpuram capessendus</b>	DmH 165 14
maximas illis commoditates allaturum. iactarent	<b>ad haec in conciliis curiarum, quae</b>	DmH 161 17
die perpetuo burgis mercatus esset	<b>ad uendendum et emendum sine ulla</b>	DmH 162 06
occupati, partim suis a tribunis	<b>ad velasum deducti, petrus-giron, comitis uregnae</b>	DmH 165 15
ueluti specula prospectantes, secure quidem	<b>agebant, donec quidam eorum rem suam</b>	DmH 143 11
tabellariorum regiae partes infirmarentur, hoc	<b>agebant primores ac diuites ultro citro</b>	DmH 162 13
qui regis procerumque negotium dissimulante	<b>agebant ut si methina-ruisica fuisset expugnata</b>	DmH 182 18
uinculum illud maximas illis commoditates	<b>allaturum, iactarent adhaec in conciliis</b>	DmH 161 17
certos intra dies diploma regium	<b>allaturum; interim humanitatis esse prudentiaeque</b>	DmH 162 11
protinus aduolauit et, flagitantibus tum	<b>burg</b>	DmH 161 04
remittereturque omnium flagitiorum supplicium et	<b>amicis et propinquis, tum ipso denique</b>	DmH 161 08
plurimum afferre momenti solet ad	<b>animaduersio, nihil non facile consecuturum</b>	DmH 162 08
quae conuenerant. Kalendis nouembris eius	<b>animos impellendos in quamcumque partem. itaque</b>	DmH 161 10
paucis uenisset, meigarenenses, quamuis ab	<b>anni quo motus coeptus est, recipitur</b>	DmH 162 22
uocant. quod autem romano multis	<b>annis multis velascos patronos oppidi fuisse</b>	DmH 177 21
ipsius nomine dent, futuras perpetuo	<b>annis, ne dicam saeculis, imperio uos</b>	DmH 056 21
liberi a praestandis gratuitis hospitiiis	<b>annuas polliceantur, amicitiam sanciant, ingentem</b>	DmH 161 16
nequaquam libera gratuitaque foren hospitia	<b>spem</b>	DmH 162 05
	<b>aulicae familiae cum rex diuerteret burgos;</b>	DmH 162 05
	<b>aulicae familiae sed in reliqua hispania</b>	DmH 118 13

La perfección de un lenguaje tan estructurado como el WPL y la potencia de los programas Contex 1 y Conc nos han permitido llegar a este grado de sofisticación en el tratamiento de un texto normal. La lista anterior contiene una serie de unidades contextuales entresacadas de varias páginas del *De motu Hispaniae*, pero el número (26) correspondería a la media de palabras en a – de dos páginas de la citada obra. En recorrer todo el proceso de su elaboración, el ordenador ha empleado 35 segundos.



El listado del programa Conc es el siguiente:

### Programa Conc

#### Carga de variables

psy l	(asigna l a la variable (y))
n pin ¿Qué letra? = \$d	(asigna una letra a la variable \$d)
pnd	(el proceso no se visualiza en pantalla)
lalfa\$d,d2	(carga el fichero \$d del disco 2)
pin ¿Margen derecho interior? = \$a	(asigna el número de la columna elegida)
pin ¿Margen derecho exterior? = \$c	(asigna el número de la columna elegida)
pin ¿Fichero de impresión? = \$b	(para imprimir en el disco)

#### Operaciones previas

b	
f/ \$d/ \$d/	(si encuentra una palabra con \$d-, sigue el programa si no, lo deja)
o?	(ejecución condicional)
pgo i	
pgo o	
i b	
z	(«return» automático)
f/</ /a	(supresión de la marca < de la referencia)
o?	
f/>/ /a	(supresión de la marca > de la referencia)
o?	
f/./,/a	(restitución de las comas)
o?	
f// /	(crea un espacio)
o?	
f!#!# !a	(crea un espacio en la cabecera del contexto anterior # = return)
o?	

Restitución de la palabra-clave y de la siguiente a sus lugares originarios después de la ordenación alfabética

Supresión de estas palabras en la cabecera del contexto

b	
f/@ /@% /a	(se pone % detrás las @ que no llevan signo de puntuación para posibilitar un procesamiento uniforme)
o?	
a f// /	(alcanza a la primera palabra)
o?	(ejecución condicional)
pgo b	(si sí, la ejecuta; si no, la salta)
pgo c	
b f// /	(sobrepasa a la primera palabra)
o?	
Ctrl-U	(carácter de control que hace avanzar al cursor un espacio)
f/ /	(sobrepasa a la segunda)
?	(termina la búsqueda)
d	(dirección izquierda para suprimir las dos palabras)
Ctrl-W	(carácter de control para suprimir la segunda palabra)
Ctrl-W	(carácter de control para suprimir la primera palabra)

Restitución de las palabras a sus lugares originarios

d	(dirección derecha para restituir)
f/ @/#@/	(separa los dos contextos anterior y posterior para imprimir cada uno con distintos parámetros y restituir las dos palabras en la cabecera de la segunda mitad)
o?	
Ctrl-W	(restituye la primera palabra)
Ctrl-U	(avanza un espacio)
Ctrl-U	(avanza otro espacio)
Ctrl-W	(restituye la segunda palabra)
f// /	(crea un espacio)
o?	
Ctrl-J	(pasa a la línea siguiente)
pgo a	(va a la etiqueta a y repite el proceso)

## Colocación de los parámetros de impresión

c	b	
	f/%//a	(supresión de la marca normalizadora %)
	o?	
	f!#!	(crea un «return»)
	o?	
	b	
	f!# !#.md\$a#!a	(inserta en el texto el parámetro del margen derecho para el contexto anterior)
	o?	
	f! .md\$a#! .md\$a# .jd#!a	(inserta en el texto el parámetro del justificación derecha para el contexto anterior)
	o?	
	f!#@!#.md\$c#@!a	(inserta en el texto el parámetro del margen derecho para el contexto posterior)
	o?	
	f!#@#.jg#@!a	(inserta en el texto el parámetro del justificación izquierda para el contexto posterior)
	o?	
	pmg 0	(parámetro del margen izquierdo)
	ppd 8	(para imprimir en disco)
	pnp	(instrucción de imprimir)
	\$b	(nombre del fichero de impresión)

## Colocación de la referencia en las últimas columnas de la línea

no	(borra la memoria)
l\$b	(carga el fichero imprimido en el disco)
b	
f!#@ ! !a	(une los contextos)
o?	
f!###!	(crea dos «return» al principio)
o?	
pin	¿Cuántos caracteres tiene la referencia? = \$a
psz \$a	(asigna el número de caracteres a la variable (z))
psx \$c	(asigna el parámetro del margen derecho exterior)
psx - (z)	(resta el número de caracteres de la referencia)

b	Ctrl-Tp	(carácter de control para purgar el tabulador)
d	f// /	(avanza un espacio)
	o?	
	pcs/(x)/0/	(compara la variable con 0)
	pgo e	(iguales, ejecuta la instrucción; distintas, la salta)
	psx - 1	(decremento de una unidad)
	pgo d	(repetición del proceso de avance)
e	Ctrl-Tm	(cuando (x) = 0, colocará un tope del tabulador)
f	f/*//	(encuentra la referencia, * = marca separadora)
	o?	
	pgo g	
	pgo h	
g	Ctrl-I	(mueve la referencia hacia el tope del tabulador)
	pgo f	(repite el proceso)

Grabación de los ficheros
---------------------------

h	pcs/(y)/1/	(distingue la primera grabación de las siguientes)
	pgo k	(iguales, ejecuta la instrucción; distintas, la salta)
	pgo l	
k	sconcor\$d,d2	(graba el primer fichero en el disco 2)
	pgo m	
l	sconcor\$d,d2 +	(añade el siguiente)
m	psy + 1	
	no	(borra la memoria)
	pgo n	(repite el proceso)
o	qt	(deja el programa)

## EL LÉXICO (Función «cut and paste» y gestión de ficheros)

### a) Colocación del lema y de los componentes fonético-gráfico, morfo-sintáctico y etimológico

Hasta este momento tenemos la materia prima para elaborar un léxico: la lista alfabética de palabras con sus contextos. El primer paso que hay que efectuar es la lematización de las formas de la lista de partida; al tiempo

que se escribe el lema, hay que colocar a continuación el componente morfo-sintáctico y, si se quiere, también el fonético-gráfico y el etimológico. No vamos a entrar aquí en la problemática de las posibilidades (que existen) de una lematización automática. Puede ser tema de otro trabajo. Aquí realizamos la lematización de forma manual, para ello partimos de la lista alfabética de palabras con sus contextos (fichero alfa\$b) pero modificada por el programa Rest de tal manera que ofrezca el aspecto normal de un texto en prosa (para facilitar la lematización, la palabra «contextualizada» sigue enmarcada entre dos espacios):

AB, prep. abl.

etiamsi remotis in regionibus, procul **a** patria, militent, quod eos nimirum DmH 163 17  
ut essent immunes ac liberi **a** praestandis gratuitis hospitii aulicae familiae DmH 162 04  
stipendium idque serum multisque modis **a** quaestoribus imminutum. Tum autem cum DmH  
163 15  
tria quae maxime cuperent habituros **a** rege, nimirum ut essent immunes DmH 162 04

AC, conj.

partes infirmarentur. hoc agebant primores **ac** diuites ultro citroque comendo nec DmH  
162 13  
rege, nimirum ut essent immunes **ac** liberi a praestandis gratuitis hospitii DmH 162 04  
obuiam primoribus phaleratis in equis **ac** uestibus picturatis, quo dynastae facerent DmH 162  
23

AD, prep. acus.

habere delectus, pecunias imperare, iuuentutem **ad** arma cogere, toletanis obsistere, qui DmH  
164 10  
traxit donec discessurus in germaniam **ad** fasces imperii primamque purpuram cap essendum  
DmH 165 14  
maximas illis commoditates allaturum. iactarent **ad** haec in conciliis curiarum, quae DmH 161  
17  
die perpetuo burgis mercatus esset **ad** uendendum et emendum sine ulla DmH 162 0 6  
occupati, partim suis a tribunis **ad** velasum deducti, petrus-giron, comitis uregnae DmH 165  
15

ADFERO adfers, adferre, adtuli, adlatum, v. a.

uinculum illud **maximas illis commoditates allaturum**. iactarent **ad** haec in conciliis DmH 161  
17  
certos intra dies diploma regium **allaturum**; interim humanitatis esse prudentia eque burg DmH  
162 11

AGO, egi, actum, 3. v. a.

ueluti specula prospectantes, secure quidem **agebant**, donec quidam eorum rem suam DmH  
143 11

tabellariorum regiæ partes infirmarentur. hoc **agebant** primores ac diuites ultro citro DmH 162 13  
qui regis procerumque negotium dissimulanter **agebant** ut si methina-ruisica fuisset expugnata DmH 182 18

AMICVS, i, m.  
protinus aduolauit et, flagitantibus tum **amicis** et propinquis, tum ipso denique DmH 161 04.

ANIMADVERSIO, onis, f.  
remittereturque omnium flagitiorum supplicium et **animaduersio**. nihil non facile consecutum velasum DmH 162 08

ANIMVS, i, m.  
plurimum afferre momenti solet ad **animos** impellendos in quamcumque partem. itaque DmH 161 10

ANNVS, i, m.  
quæ conuenerant. kalendis nouembris eius **anni** quo motus coeptus est, recipitur DmH 162 22  
paucis uenisset, melgarenses, quamuis ab **annis** multis velascos patronos oppidi fuisse DmH 177 21  
uocant. quod autem romano multis **annis**, ne dicam saeculis, imperio uos DmH 056 21

ANNVVS, a, um, adj.  
ipsius nomine dent, futuras perpetuo **annuas** polliceantur, amicitiam sanciant, ingentem spem DmH 162 05

AVLICVS, a, um, adj.  
liberi a praestandis gratuitis hospitiiis **aulicae** familiae cum rex diuerteret burgos DmH 162 05  
nequaquam libera gratuitaque forent hospitia **aulicae** familiae sed in reliqua hispania DmH 118 13

El listado del programa Rest es el siguiente (es una adaptación del programa Conc):

## Programa Rest

### Operaciones previas

	psy l	
	pnđ	(el proceso no se visualiza en pantalla)
n	pin, ¿Qué letra? = \$d	
	lalfa\$d,d2	(mismo punto de partida que el programa Conc)
	b	
	f/ \$d/ \$d/	(si encuentra una palabra con \$d-, sigue el programa; si no lo deja)
	o?	
	pgo i	
	pgo o	
i	b	
	z	(«return» automático)
	f// /	(crea un espacio)
	o?	
	f#!# !a	(crea un «return» y un espacio al inicio de cada línea)
	o?	

Restitución de la palabra-clave y de la siguiente a sus lugares originarios después de la ordenación alfabética

Supresión de estas palabras en la cabecera del contexto

	b	
	f/@ /@% /a	(se pone % detrás las @ que no llevan signo de puntuación para posibilitar un procesamiento uniforme)
	o?	
a	f/ / /	(alcanza a la primera palabra)
	o?	(ejecución condicional)
	pgo b	(si sí, la ejecuta, si no, la salta)
	pgo c	
b	f/ / /	(sobrepasa a la primera palabra)
	o?	
	Ctrl-U	(carácter de control que hace avanzar el cursor un espacio)
	f/ /	(sobrepasa a la segunda)
	?	(termina la búsqueda)

# FICHERO DEL ARTÍCULO

<b>LEMA</b>	<b>comp. graf.- fon.</b>	<b>comp. morf.- sint.</b>	<b>comp. etimolog.</b>
-------------	--------------------------	---------------------------	------------------------

<i>marca</i>	<i>acepción</i>			<i>return</i>	
3	<u>contexto</u>	<u>ante</u>	<u>acep. 3</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret
3	<u>contexto</u>	<u>ante</u>	<u>acep. 3</u>	<u>contexto post ref.</u>	ret
2	<u>contexto</u>	<u>ante</u>	<u>acep. 2</u>	<u>contexto post ref.</u>	ret
3	<u>contexto</u>	<u>ante</u>	<u>acep. 2</u>	<u>contexto post ref.</u>	ret
4	<u>contexto</u>	<u>ante</u>	<u>acep. 4</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret
4	<u>contexto</u>	<u>ante</u>	<u>acep. 4</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 2</u>	<u>contexto post ref.</u>	ret
2	<u>contexto</u>	<u>ante</u>	<u>acep. 4</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret
4	<u>contexto</u>	<u>ante</u>	<u>acep. 4</u>	<u>contexto post ref.</u>	ret

La aplicación de la instrucción Ctrl-L #/1/ret/a dará el siguiente resultado:

1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret
1	<u>contexto</u>	<u>ante</u>	<u>acep. 1</u>	<u>contexto post ref.</u>	ret

cambiando /1/ por /2/, la instrucción Ctrl-L #/2/ret/a obtendría:

2	<u>contexto</u>	<u>ante</u>	<u>acep. 2</u>	<u>contexto post ref.</u>	ret
2	<u>contexto</u>	<u>ante</u>	<u>acep. 2</u>	<u>contexto post ref.</u>	ret

y así sucesivamente. Al final las marcas serán automáticamente convertidas en su significado por la instrucción /Find/. La tarea se hace más sencilla con un simple programa:



## Programa Lexic

### Operaciones previas

n psy 1  
pin ¿Letra del fichero? = \$d  
pnd  
llex\$d (parte del resultado del programa Rest)  
f/ \$d/ \$d/ (si encuentra un \$d-, sigue; si no, abandona)  
o?  
pgo a  
pgo o

### Clasificación de acepciones

a b  
f!#!  
o?  
psx 1  
l# <(x) < > <a (carga de la memoria los segmentos que empiecen  
por (x) y terminen en «return», > = «return»)  
pgo b  
pgo c (si no hay más segmentos, va a la etiqueta c)  
b f!#!  
o?  
psx + 1 (incremento)  
pgo a (repite el proceso)

### Borrado del texto de partida

c f!marca separadora de la copia!###! (crea tres «return» para «frenar»  
el borrado)  
o?  
e (cursor al final)  
x (tantas x como número de caracteres/1024)  
x  
etc.

## Conversión de las marcas en sus significados

b	
psx l	
d	f/(x)/«significado de la marca»/ (reescritura de cada marca)
	o?
	pgo e
	pgo h
e	psx + l
	pgo d

## Grabación de los ficheros

h	pcs/(y)/1/	(distingue la primera grabación de las siguientes)
	pgo k	(iguales, ejecuta la instrucción; distintas, la salta)
	pgo l	
k	slexico\$d,d2	(graba el primer fichero en el disco 2)
	pgo m	
l	slexico\$d,d2+	(añade el siguiente)
m	psy + l	
	no	(borra la memoria)
	pgo n	(repite el proceso)
o	qt	(deja el programa)

En resumen: Podemos decir que la elaboración de las concordancias es un proceso absolutamente automático en el que se puede partir de un texto normal, se extraen los contextos mecánicos, se clasifican alfabéticamente y se editan e imprimen destacando la palabra-clave de forma que salte a la vista la «concordancia» en distintos contextos. Para la elaboración de los léxicos de autor se puede partir, como hemos hecho nosotros, de la lista alfabética de las palabras con sus contextos. Tiene la ventaja de tener ordenadas alfabéticamente todas las unidades contextuales agrupadas bajo una misma acepción, lo cual facilitará en artículos largos la búsqueda de una determinada palabra. La confección de estos léxicos, según la hemos programado nosotros, es una tarea semiautomática ya que el ordenador se circunscribe a la clasificación material de las acepciones; no obstante, no hay que olvidar que el empleo de un procesador de textos programable, aparte de las prestaciones convencionales como procesador de textos, hace posible la confección de listas de frecuencia, la distribución de frecuencias, los extractos estadísticos de los datos gramaticales, etc., una serie de operaciones básicas para el conocimiento científico de la lengua de un autor.