

**Unite : conjunto de programas
para el tratamiento filológico
de textos en verso**

F.Marcos Marín
Universidad Autónoma de Madrid

Abstract

UNITE consists of a set of programs and instructions which help the user to obtain critical editions by comparing different versions, i.e. manuscripts, of the same work. The automatic comparison includes automatic ordering of the stanzas, graphemic neutralization, word order and clitics, and differentiation of semantic and functional words.

The package has been developed at IBM Science Centres in Madrid (Spain) and Heidelberg (Germany).

La utilización del ordenador en trabajos filológicos puede aliviar mucho la carga mecánica del investigador, facilitando su tarea. Esto es especialmente visible cuando se trata de ediciones que han de tener en cuenta distintas versiones, escritas no pocas veces con variantes dialectales, o con diferencias debidas a las distintas épocas en que se realizaron las copias. Al preparar la edición unificada de una obra de la literatura castellana del siglo XIII, el *Libro de Alexandre*, conservado en un manuscrito del siglo XIV con rasgos leoneses y otro del XV con rasgos orientales (riojanos y aragoneses), tuvimos ocasión de ir creando un conjunto de programas y dispositivos lógicos para facilitar la tarea del editor y asegurar la fiabilidad de los resultados. A partir de este conjunto se ha ido desarrollando otro más amplio y preciso, que presentamos ahora¹.

Los procedimientos en uso son excesivamente exigentes en lo que concierne a la transcripción y a la pre-edición del texto. UNITE no requiere ninguna clase de pre-edición. Para la transcripción, recomendamos que se sigan las normas del Seminario de Estudios Medievales de Madison (Wisconsin) y que se eviten todos los signos que no forman parte del teclado básico (es decir, el de una máquina de escribir con

¹ Para ello basta con entrar en contacto con el autor, bien por correo ordinario: Ap. 46348, E-28080 Madrid, España, bien por correo electrónico: MARCOS at EARN.EARNET (= BITNET), o MARCOS at EMDUAM11.EARNET (= BITNET). La fórmula de cesión dependerá del tipo de colaboración que se pueda establecer. Los programas funcionan, de momento, en ordenadores IBM 43xx y superiores, en el sistema VM/CMS, pero está previsto ofrecerlos, más adelante, en versiones que se ejecuten en IBM/PC o compatibles, escritas en TURBOPASCAL. Como procesadores de textos se usan GML (Generalized Mark-up Language) y SCRIPT/VS. Este conjunto de programas se ha realizado gracias a la ayuda imprescindible de dos instituciones. La primera de ellas ha sido la empresa IBM, que permitió al autor trabajar en sus Centros Científicos de Madrid y de Heidelberg. La estancia en Heidelberg fue posible gracias a una beca de la Alexander von Humboldt-Stiftung para 1984 - 85, ampliada para el verano del 86. El agradecimiento a las instituciones se extiende, por supuesto, a las personas V. Martínez, J. L. Becerril, R. Casajuana, L. Sopeña, G. Martínez, A. Llopis, A. Blaser, Hein Lehmann, Hubert Lehmann, H. Wrobel, J. Vaessen, el Profesor K. Baldinger, Albert Gier y Frau Cécilia Nauderer fueron sometidos a consultas, preguntas, peticiones y resolvieron toda clase de dudas, con paciencia infinita. Las becarias Celia Villar, Consuelo Rodríguez y Marina Galán merecen también nuestra gratitud. Juan de Dios Godoy, gracias a una beca de IBM España para el Centro de Investigación IAM-IBM, colaboró decisivamente en el proyecto, como informático. Los programas, macros y exec (cuando no se trata de productos públicos o internos de IBM) han sido fruto de esa colaboración, así como las modificaciones de otros programas ya existentes, y están sujetos a los términos del acuerdo firmado entre IBM y el autor.

teclado inglés) como tildes y acentos. Téngase en cuenta que es mucho más sencilla la reconversión desde signos de este teclado, visibles en cualquier sistema, que desde unos hexadecimales peculiares de uno de ellos, que pueden sufrir transformaciones impredecibles en los procesos de copia o traslado de un sistema a otro.

Observada la norma de transcripción o, en su defecto, hechos explícitos los criterios de la misma, si difieren, para que se puedan hacer las transformaciones oportunas, debe tenerse en cuenta que el conjunto de programas de UNIFICACIÓN TEXTUAL (UNITE) puede actuar sobre textos en verso que comiencen por tres líneas en blanco y que estén organizados en unidades como mínimo de un verso y como máximo de cinco, separadas por una línea en blanco de la unidad siguiente. Los programas tratan, en principio, hasta seis versiones distintas, pero pueden corregirse con facilidad para tratar el número que se indique, lo mismo sucede para las unidades o conjuntos de versos.

Las cifras de uno a cinco para las unidades de versos o de seis para las versiones indican el estándar, pero no los límites del sistema. Al principio de cada unidad debe constar su número. Este número y el texto pueden comenzar en la columna que se desee, pero esa columna debe ser siempre la misma para toda la numeración y para todos los versos. Así, si se copia un poema que puede estar compuesto por 1300 unidades de cinco versos, se aconseja que las cuatro primeras columnas se reserven para la numeración, que ésta se ordene del 0001 al 1300, que luego se deje en blanco la columna 5 y que el texto se copie a partir de la columna sexta, empezando en ella. Ahora bien, no sería problema que la numeración empezara en la columna dos, y el texto en la diez, siempre que toda la numeración ocupara las columnas 2, 3, 4, 5 y el texto de la 10 en adelante. Nótese que la numeración debe tener en cuenta el número más alto, y que deben reservarse tantas columnas como requiera este número. También es opcional numerar las unidades como 0001 o como 1, siempre que delante de ese 1 no precedido de ceros se hayan dejado las correspondientes columnas en blanco. La numeración no tiene que ser correlativa, pero el mismo número no puede repetirse.

El paquete de ficheros UNITE

Este conjunto de ficheros se puede dividir en dos grupos, según realicen la entrada de datos desde la pantalla o desde el fichero DAT_UNIT.DATOS. Tanto uno como otro utilizan los ficheros llamados FTEXi SCRIPT ($1 < i < 7$), en los que están almacenadas las versiones que se van a comparar.

Los ficheros FTEXi SCRIPT A contienen las versiones que se van a comparar en el proceso de unificación automática. El máximo de versiones que se permiten en el programa es 6, por tanto, existen seis ficheros FTEXi (FTEX1..FTEX6). Dentro de este límite se utilizarán aquellos que sean necesarios, según el número de textos que se quieran comparar. La prioridad de cada versión a la hora de la comparación depende del nombre del fichero en el que esté almacenada y va en orden descendente desde el FTEX1 hasta el FTEX6.

Reglas para el almacenamiento de las versiones

- Las tres primeras líneas de los ficheros no serán tenidas en cuenta por el programa; pueden ser utilizadas por el usuario para identificar la versión que se va a almacenar en el fichero.
- Los textos deben estar compuestos por conjuntos de versos a los que, por comodidad, llamaremos *estrofas*, cada una de las cuales puede tener un número variable de versos. Por ahora el máximo de versos está fijado en 5, el mínimo es 1.
- Entre estrofa y estrofa debe haber una línea en blanco de separación y sólo una.
- En el final del fichero puede existir como máximo una línea en blanco a partir de la última estrofa.

UNITE

teclado inglés) como tildes y acentos. Téngase en cuenta que es mucho más sencilla la reconversión desde signos de este teclado, visibles en cualquier sistema, que desde unos hexadecimales peculiares de uno de ellos, que pueden sufrir transformaciones impredecibles en los procesos de copia o traslado de un sistema a otro.

Observada la norma de transcripción o, en su defecto, hechos explícitos los criterios de la misma, si difieren, para que se puedan hacer las transformaciones oportunas, debe tenerse en cuenta que el conjunto de programas de UNIFICACIÓN TEXTUAL (UNITE) puede actuar sobre textos en verso que comiencen por tres líneas en blanco y que estén organizados en unidades como mínimo de un verso y como máximo de cinco, separadas por una línea en blanco de la unidad siguiente. Los programas tratan, en principio, hasta seis versiones distintas, pero pueden corregirse con facilidad para tratar el número que se indique, lo mismo sucede para las unidades o conjuntos de versos.

Las cifras de uno a cinco para las unidades de versos o de seis para las versiones indican el estándar, pero no los límites del sistema. Al principio de cada unidad debe constar su número. Este número y el texto pueden comenzar en la columna que se desee, pero esa columna debe ser siempre la misma para toda la numeración y para todos los versos. Así, si se copia un poema que puede estar compuesto por 1300 unidades de cinco versos, se aconseja que las cuatro primeras columnas se reserven para la numeración, que ésta se ordene del 0001 al 1300, que luego se deje en blanco la columna 5 y que el texto se copie a partir de la columna sexta, empezando en ella. Ahora bien, no sería problema que la numeración empezara en la columna dos, y el texto en la diez, siempre que toda la numeración ocupara las columnas 2, 3, 4, 5 y el texto de la 10 en adelante. Nótese que la numeración debe tener en cuenta el número más alto, y que deben reservarse tantas columnas como requiera este número. También es opcional numerar las unidades como 0001 o como 1, siempre que delante de ese 1 no precedido de ceros se hayan dejado las correspondientes columnas en blanco. La numeración no tiene que ser correlativa, pero el mismo número no puede repetirse.

El paquete de ficheros UNITE

Este conjunto de ficheros se puede dividir en dos grupos, según realicen la entrada de datos desde la pantalla o desde el fichero DAT_UNIT.DATOS. Tanto uno como otro utilizan los ficheros llamados FTEXi SCRIPT ($1 < i < 7$), en los que están almacenadas las versiones que se van a comparar.

Los ficheros FTEXi SCRIPT A contienen las versiones que se van a comparar en el proceso de unificación automática. El máximo de versiones que se permiten en el programa es 6, por tanto, existen seis ficheros FTEXi (FTEX1..FTEX6). Dentro de este límite se utilizarán aquellos que sean necesarios, según el número de textos que se quieran comparar. La prioridad de cada versión a la hora de la comparación depende del nombre del fichero en el que esté almacenada y va en orden descendente desde el FTEX1 hasta el FTEX6.

Reglas para el almacenamiento de las versiones

- Las tres primeras líneas de los ficheros no serán tenidas en cuenta por el programa; pueden ser utilizadas por el usuario para identificar la versión que se va a almacenar en el fichero.
- Los textos deben estar compuestos por conjuntos de versos a los que, por comodidad, llamaremos *estrofas*, cada una de las cuales puede tener un número variable de versos. Por ahora el máximo de versos está fijado en 5, el mínimo es 1.
- Entre estrofa y estrofa debe haber una línea en blanco de separación y sólo una.
- En el final del fichero puede existir como máximo una línea en blanco a partir de la última estrofa.

UNITE

Formato de la estrofa

- Cada estrofa debe tener un número asignado que figurará al menos en la línea correspondiente al primer verso de dicha estrofa.
- La distancia desde el principio de la línea a la numeración, la longitud máxima de dicha numeración, así como la distancia desde ésta al principio del verso propiamente dicho son optativas, pero constantes. Es muy importante que estos valores se correspondan con los aportados al programa en DAT_UNIT o en la pantalla.
- El verso propiamente dicho debe comenzar en la misma columna en todos los versos de la estrofa, es decir en aquella correspondiente a la suma de los parámetros: margen antes de la numeración, longitud máxima de ésta y distancia de dicha numeración al principio del verso.
- Las palabras pertenecientes a un verso deben estar separadas por un único espacio en blanco, ya que el fin del verso se detecta cuando se llega al fin de línea o a dos espacios blancos seguidos.
- Si falta un verso completo en una estrofa, esta línea no se debe dejar en blanco, sino rellenarse con los caracteres: .-.-. (punto y guión).
- No es necesario que las estrofas estén ordenadas según su numeración, ya que se puede utilizar para ordenarlas el módulo SORT_EST.
- El programa UNITE compara los versos en el interior de la estrofa, aunque estén en orden distinto.

Como paso previo a la ejecución de cualquiera de los dos grupos de programas, se debe utilizar el módulo SORT_EST MODULE.

Descripción del programa SORT_EST

Función

Este programa ordena de menor a mayor las estrofas almacenadas en un fichero. Si dos estrofas tienen el mismo número las deja tal y como están almacenadas.

Condiciones para su funcionamiento

El fichero cuyas estrofas se van a ordenar debe estar declarado como DESORD SCRIPT A; también debe haber declarados dos ficheros llamados FICH1 SCRIPT A y FICH2 SCRIPT A con el contenido que se quiera, éstos son utilizados como ficheros auxiliares por el programa. Una vez acabada la ejecución, los ficheros auxiliares desaparecen, mientras que el fichero DESORD SCRIPT A contiene las estrofas ordenadas.

El número de estrofa en el fichero DESORD debe estar situado en las cuatro primeras posiciones de la línea correspondiente.

Al principio del fichero no debe haber ninguna línea en blanco. Esto supone una diferencia de formato respecto a los ALEX y ALEJOs.

Al final del fichero debe haber una única línea en blanco.

Modo de empleo:

Una vez declarados los ficheros auxiliares y DESORD, es conveniente asegurarse de que existen estrofas descolocadas antes de ejecutar el programa; para ello se puede utilizar la macro AVISA, que

UNITE

muestra en la pantalla el número de estrofa descolocada. Para utilizar correctamente esta macro la línea vigente debe ser el primer verso de una estrofa (normalmente de la primera del fichero). Hay que decir que AVISA considera descolocadas aquellas estrofas que tienen el mismo número. Si detectamos estrofas descolocadas, salimos del fichero editado y ejecutamos el módulo SORT_EST con la siguiente instrucción:

SORT_EST

Cuando acaba la ejecución podemos comprobar si ésta ha sido correcta volviendo a editar el fichero DESORD y utilizando de nuevo la macro AVISA.

Ejecución con el fichero DAT_UNIT DATOS

El fichero DAT_UNIT DATOS se utiliza para que el usuario pueda controlar la ejecución del programa. En este fichero se pide una serie de valores y se indica en qué lugares se deben teclear las respuestas. Conviene recordar que DAT_UNIT sólo se puede modificar en los espacios reservados para respuestas.

Además de este fichero existen también UNITE1 PASCAL y el módulo generado a partir de él, UNITE1 MODULE. Integran también el grupo de ficheros los EXEC COMPUNII (compila en BATCH el programa UNITE1, su ejecución no es necesaria puesto que ya está generado UNITE1 MODULE) y EJECTUNII, que ejecuta en el BATCH el módulo de UNITE1.

El fichero DAT_UNIT DATOS

Se trata de un fichero de entrada de datos. En aquellos apartados en los que se pidan datos que, por respuestas anteriores, no son necesarios no hay que insertar nada. Este fichero sólo se puede modificar en los espacios reservados para las respuestas.

Introducir siempre el dato en el espacio indicado por →

Establecer si se deben ejecutar los procesos de ayuda a la selección

Ejecución de unión de palabras (s/n) →s

Ejecución de separación de palabras (s/n) →s

Ejecución de unifica varias posiciones de una palabra (s/n)? →s

Ejecución de formación de palabras a partir de letras comunes de otras(s/n)→s

Introducir los siguientes valores generales:

Deben aparecer las estrofas originales en la salida (s/n):→n

Número de versiones originales:→6

Número de estrofa para comienzo de ejecución:→1

Número de fases de unificación (1 - 5):→5

Mínimo de palabras comunes (todas vacías) entre dos versos para no tomarlos como descolocados:→3

Número de espacios desde el principio hasta la numeración de las estrofas:→0

Longitud de la numeración:→4

Distancia desde el final de la numeración hasta el principio de la estrofa:→7

UNITE

- Radio de acción del proceso union_de_palabras (número recomendado 4): → 4
- Radio de acción del proceso separacion_de_palabras (número recomendado 4): → 4
- Radio de acción de unifica_posiciones (se recomienda entre 4 y 6): → 5
- Radio de acción de letras_comunes (se recomienda 3): → 3

Introducir el mínimo de versiones de coincidencia: Se trata del mínimo de versiones en las que debe coincidir una palabra para ser seleccionada, según las versiones que se estén comparando.

- Mínimo de versiones cuando comparemos 2 textos: → 2
- Mínimo de versiones cuando comparemos 3 textos: → 2
- Mínimo de versiones cuando comparemos 4 textos: → 3
- Mínimo de versiones cuando comparemos 5 textos: → 3
- Mínimo de versiones cuando comparemos 6 textos: → 4

Introducir los códigos de las fases de unificación: Los códigos se introducirán en el orden que se desee. Las fases van numeradas del 1 al 5 y son las siguientes:

1. Fase inactiva
2. Elimina signos de puntuación
3. Cambio de mayúsculas y vocales acentuadas
4. Unificación de caracteres y grupos de caracteres hacia formas generales
5. Simplificación de letras consecutivas iguales dentro de una palabra

El número que sigue a la → es el código de la fase. El orden de aplicación depende de la voluntad del usuario. Ejemplificamos con el orden normal, creciente, de 1 a 5, pero nada impediría un orden decreciente, del 5 al 1, o mixto (3,4,2,5,1). Téngase en cuenta que la fase 1 es útil sólo cuando se aplica en primer lugar.

- Código de la primera fase aplicable: → 1
- Código de la segunda fase aplicable: → 2
- Código de la tercera fase aplicable: → 3
- Código de la cuarta fase aplicable: → 4
- Código de la quinta fase aplicable: → 5

Pasos para la ejecución:

1. Preparar los ficheros FTEXi
2. Editar el fichero DAT_UNIT DATOS y modificar si es necesario las respuestas.
3. Ejecutar el programa. La ejecución se puede hacer de dos formas: en nuestra máquina virtual o utilizando el BATCH, esta última normalmente se utilizará para procesos con textos muy largos.
 - a. UNITE1 para la primera
 - b. SUBMIT EJECUNII (el i di no sec 36 para el BATCH.
(las opciones del exec SUBMIT se pueden modificar)

UNITE

- Radio de acción del proceso union_de_palabras (número recomendado 4):→4
- Radio de acción del proceso separacion_de_palabras (número recomendado 4):→4
- Radio de acción de unifica_posiciones (se recomienda entre 4 y 6):→5
- Radio de acción de letras_comunes (se recomienda 3):→3

Introducir el mínimo de versiones de coincidencia: Se trata del mínimo de versiones en las que debe coincidir una palabra para ser seleccionada, según las versiones que se estén comparando.

- Mínimo de versiones cuando comparemos 2 textos:→2
- Mínimo de versiones cuando comparemos 3 textos:→2
- Mínimo de versiones cuando comparemos 4 textos:→3
- Mínimo de versiones cuando comparemos 5 textos:→3
- Mínimo de versiones cuando comparemos 6 textos:→4

Introducir los códigos de las fases de unificación: Los códigos se introducirán en el orden que se desee. Las fases van numeradas del 1 al 5 y son las siguientes:

1. Fase inactiva
2. Elimina signos de puntuación
3. Cambio de mayúsculas y vocales acentuadas
4. Unificación de caracteres y grupos de caracteres hacia formas generales
5. Simplificación de letras consecutivas iguales dentro de una palabra

El número que sigue a la → es el código de la fase. El orden de aplicación depende de la voluntad del usuario. Ejemplificamos con el orden normal, creciente, de 1 a 5, pero nada impediría un orden decreciente, del 5 al 1, o mixto (3,4,2,5,1). Téngase en cuenta que la fase 1 es útil sólo cuando se aplica en primer lugar.

- Código de la primera fase aplicable:→ 1
- Código de la segunda fase aplicable:→ 2
- Código de la tercera fase aplicable:→ 3
- Código de la cuarta fase aplicable: → 4
- Código de la quinta fase aplicable: → 5

Pasos para la ejecución:

1. Preparar los ficheros FTEXi
2. Editar el fichero DAT_UNIT DATOS y modificar si es necesario las respuestas.
3. Ejecutar el programa. La ejecución se puede hacer de dos formas: en nuestra máquina virtual o utilizando el BATCH, esta última normalmente se utilizará para procesos con textos muy largos.
 - a. UNITE1 para la primera
 - b. SUBMIT EJECUNII (el i di no sec 36 para el BATCH.
(las opciones del exec SUBMIT se pueden modificar)

UNITE

El resultado de esta ejecución queda almacenado en el fichero FILE SALUNIF, que aparecerá en el FILEList, o en el READER si se ha ejecutado en el BATCH.

Ejecución desde la pantalla:

En este modo de ejecución, la pantalla ocupa el lugar del fichero DAT_UNIT, es decir, el programa pide una serie de datos al usuario, para que éste teclee en cada caso la respuesta que considere oportuna. Además de esta entrada de datos, el programa va mostrando los resultados de las diferentes etapas de la ejecución.

Los ficheros que integran este grupo son : UNITE2 PASCAL, su módulo UNITE2 MODULE y el EXEC COMPUNI2, sus funciones son las mismas que las de los ficheros vistos anteriormente. Cabe destacar que al utilizar la pantalla no existe la modalidad de ejecución en el BATCH.

Pasos para la ejecución:

1. Preparar los ficheros FTEXi
2. Invocar el módulo tecleando: UNITE2

El resultado de la ejecución (al igual que en el caso anterior) queda almacenado en el fichero FILE SALUNIF.

Ejemplo del proceso de unificación.

A continuación presentamos una representación del proceso, según las sucesivas fases aplicadas para unificar la estrofa 51 del *Libro de Alexandre*, conservada en los manuscritos de París (1), de Madrid (2), en la versión impresa del *Victorial* de Gutierre Díez de Games (4) y en la versión manuscrita del mismo fragmento (5). Falta el número (3), que corresponde al fragmento de Medinaceli, en el cual no está la estrofa que unificamos. La primera línea de cada ejemplo (que lleva la numeración 51 A/B/C/D) contiene el estado que la comparación ha alcanzado al terminar cada fase, mientras que las líneas sin numerar que la siguen contienen los elementos que todavía debe ir tratando el programa o, al final, los residuos no unificables, que el editor humano tendrá que considerar para completar la tarea. No se olvide que el proceso sigue las instrucciones de DAT UNIT DATOS. En el primer intento tiene en cuenta el número de orden de las palabras en el verso. En lugar de separar el presunto clítico, a diferencia de productos anteriores, como ALCOMCLI y UNION PASCAL, lo une si va separado, con lo que se evitan falsos cortes de >-l<, >-t<, etc. en posición final. El resto del proceso es de fácil deducción, creemos.

Proceso de unificación del primer verso

Enpec'ol Aristotiles com<m>o om're bien honrrado

Comec'o don Aristotil cuemo ombr<e> bie<n> letrado

Comenzó Aristotiles, como ome bien lenguado,

comenc'o Aristotiles como onbre bien lenguado

Ejecución del primer intento de selección

```
51 A      Aristotiles como      bien
/1enpesol/                /14ome/      /1onrado/
/2comeso//2don/          /2aristotil//2cuemo//2omure//2bien//2letrado/
/45comenso/                /45lenguado/
                          /5onure/
```

Ejecución de Unifica_Posiciones

```
51 A      Aristotiles como      bien
/1enpesol/                /14ome/      /1onrado/
/2comeso//2don/          /2aristotil//2cuemo//2omure//2letrado/
/45comenso/                /45lenguado/
                          /5onure/
```

Ejecución de Letras_Comunes. Fase= 1

```
51 A come_so Aristotiles como      bien
/1enpesol/                /14ome/      /1onrado/
      /2don/          /2aristotil//2cuemo//2omure//2letrado/
                          /45lenguado/
                          /5onure/
```

Ejecución de Letras_Comunes. Fase= 2

51 A come_so Aristotiles como om__e bien
/1enpesol/ /1onrado/
/2don/ /2aristotil//2cuemo/ /2letrado/
/45lenguado/
/5onure/

Ejecución de Letras_Comunes.Fase= 3

51 A come_so Aristotiles como om__e bien le__ado
/1enpesol/ /1onrado/
/2don/ /2aristotil//2cuemo/
/5onure/

Proceso de unificación del segundo verso

fijo dixol en buena hedat sodes vos llegado
fijo a bona edat sodes embiado
é dixo: fijo, á buena edad eres llegado
e dixo fijo a buena hedad heres llegado

Ejecución del primer intento de selección

51 B llegado
/12fijo//1dixol//1en//1buena//1edat//1sodes//1uos/
/2a/ /2uona//2edat//2sodes//2enuiado/
/45e/ /45dixo//45fijo//45a//45buena//45edad//45eres/

Ejecución de Unifica_Posiciones

51 B fijo a buena llegado
/1dixol//1en/ /12edat//12sodes//1uos/
/2uona/ /2enuiado/
/45e//45dixo/ /45edad//45eres/

Ejecución de Letras_Comunes. Fase= 1

51 B fijo dixo_ e_ a buena eda_ llegado
/12sodes//1uos/
/2uona/ /2enuiado/
/45eres/

Proceso de unificación del tercer verso

pora seyer ome bueno tu lo as agujado
de seer ombre bueno tenedes lo bien guisado
de seer ome bueno: fazlo bien á guisado,
de ser hombre bueno faslo bien agujado

Ejecución del primer intento de selección

51 C de ser bueno
/1pora//1seier//14ome//1tu//12lo//1as//1aguisado/
 /2omure/ /2tenedes/ /2bien//24guisado/
 /45faslo//45bien//4a/
 /5onure/ /5aguisado/

Ejecución de Unifica_Posiciones

51 C de ser bueno bien
/1pora//1seier//14ome//1tu//12lo//1as//15aguisado/
 /2omure/ /2tenedes/ /24guisado/
 /45faslo/ /4a/
 /5onure/

Ejecución de Une_Palabras. Fase= 1

51 C de ser bueno bien aguisado
/1pora//1seier//14ome//1tu//12lo//1as/
 /2omure/ /2tenedes/
 /45faslo/
 /5onure/

Ejecución de Separa_Palabras. Fase= 1

51 C de ser bueno lo bien aguisado
/1pora//1seier//14ome//1tu/ /1as/
 /2omure/ /2tenedes/
 /45(fas)/
 /5onure/

Ejecución de Letras_Comunes. Fase= 2

51 C de ser om_e bueno _as lo bien aguisado
 /1pora//1seier/ /1tu/
 /2tenedes/
 /5onure/

Proceso de unificación del cuarto verso
 sy leuar lo qujeres com<m>o lo as enpec'ado
 si leuar lo q<ui>sierdes cuemo auedes co<m>pec'ado
 si llevarlo quisieres como has comenzado
 sy llevarlo qujsjeres como as comenc'ado

Ejecución del primer intento de selección.

51 D si
 /12leuar//12lo//1quieres//1como//1lo//1as//1enpesado/
 /2quisierdes//2cuemo//2auedes//2conpesado/
 /45lleuarlo//45quisieres//45como//45as//45comensado/

Ejecución de Unifica_Posiciones

51 D si como as
 /12leuar//12lo//1quieres//1lo/ /1enpesado/
 /2quisierdes//2cuemo//2auedes//2conpesado/
 /45lleuarlo//45quisieres//45comensado/

Ejecución de Une_Palabras. Fase= 1

51 D si leuarlo como as
 /1quieres//1lo/ /1enpesado/
 /2quisierdes//2cuemo//2auedes//2conpesado/
 /45quisieres/ /45comensado/

Ejecución de Letras_Comunes. Fase= 1

51 D si leuarlo quisier_es como as
 /1quieres//1lo/ /1enpesado/
 /2cuemo//2auedes//2conpesado/
 /45comensado/

Ejecución de Letras_Comunes. Fase= 2

51 D si leuarlo quisier_es como as co_nsado
 /1quieres//1lo/ /1enpesado/
 /2cuemo//2auedes/

ESTROFA UNIFICADA COMPLETA
(Tal como aparece en FILE SALUNIF)

51 A come_so Aristotiles como om__e bien le__ado
/1enpesol/ /1onrado/
/2don/ /2aristotil//2cuemo/
/5onure/

51 B fijo dixo_e_ a buena eda_ l llegado
/12sodes//luos/
/2uona/ /2enuiado/
/45eres/

51 C de ser om__e bueno _as lo bien aguisado
/1pora//1seier/ /1tu/
/2tenedes/
/5onure/

51 D si leuarlo quisier_es como as co__nsado
/1quieres//1lo/ /1enpesado/
/2cuemo//2auedes/

El tratamiento posterior del texto unificado

A continuación enumeramos los distintos pasos que permiten automatizar en la mayor medida posible el trabajo filológico crítico final, para que las energías del editor se concentren en lo verdaderamente importante: la elección de las formas del texto y de las variantes de las notas.

1. RENAME FILE SALUNIF * UNxxxxxx SCRIPT *

Esto nos permite procesarlo con GML y SCRIPT/VS

2. COPYfile UNxxxxxx SCRIPT * @xxxxxxx = *

El fichero UNxxxxxx se conserva como referencia para futuras búsquedas en la selección de variantes.

El fichero @xxxxxxx es el fichero de trabajo.

3. Sobre @xxxxxxx ejecutar FNMAC XEDIT

Esta macro realiza tres funciones:

- Da un número de referencia para la nota a pie de página.
- Elimina la numeración (a,b,c,d) de cada verso y la repetición de la numeración de la estrofa, que sólo se conserva en el verso primero de la misma.
- Edita la estructura convencional del GML para una nota a pie de página, con la referencia al número.

4. Intervención del editor humano, con la ayuda de XEDIT, GML, SCRIPT/VS y programas de búsqueda, tipo SCANFILE.

UNITE

La detallada descripción de las distintas etapas del trabajo pretende ofrecer a los investigadores las mayores precisiones sobre lo que se puede esperar de la metodología y del conjunto de programas. Debe tenerse en cuenta que el diseño de todos ellos permite, con suma facilidad, su adaptación a necesidades que no sean idénticas a las descritas, sino variaciones de las mismas. Tampoco es imprescindible que los textos estén escritos en castellano, ni siquiera en una lengua románica. La colaboración en este terreno permitirá, sin duda, un mayor desarrollo de nuevas técnicas filológicas.