SISTEMATIZACIÓN DEL PROCESO DE DEPURACIÓN DE LOS DATOS EN ESTUDIOS CON SEGUIMIENTOS

ALBERT BONILLO MARTÍN

Doctorado de Psicopatología Infantojuvenil

Proyecto de Investigación dirigido por

Dr. JOSÉ MARÍA DOMÉNECH I MASSONS

Dra. ROSER GRANERO PÉREZ

Departament de Psicobiologia i Metodologia de les Ciències de la Salut
Universitat Autònoma de Barcelona

Bellaterra, Septiembre 2003

ÍNDICE DE CONTENIDOS

PRESENTACIÓN

El desarrollo de la informática en los últimos años ha permitido utilizar los ordenadores de forma rutinaria en todos los ámbitos científicos. Así, y centrándonos en el ámbito de la Psicológia, ha permitido crear programas que facilitan el proceso de medición de conductas, la administración y corrección de pruebas, la planificación de la recogida de la información, el almacenamiento de la misma, e incluso, la emisión de un diagnóstico (Westmeyer y Hageböck, 1992).

En el ámbito científico, el desarrollo de la informática ha permitido aumentar la cantidad de información a procesar y maximizar la productividad, facilitando tareas que antes eran mucho más tediosas o prácticamente imposibles de realizar. Los programas actuales facilitan un fácil procesamiento de los datos y la gestión de los mismos por usuarios que no necesitan conocimientos profundos en materias como la teoría de las bases de datos o el análisis estadístico.

Sin embargo, la aplicación de la informática para maximizar la calidad de los datos no ha ido pareja a su uso para facilitar la cantidad de información procesada. Así, aunque se han creado gran cantidad de programas de análisis de datos y de *interfaces* que pretenden facilitar su uso, no se han desarrollado programas que incidan en los procesos que permiten aumentar la calidad de la información que se maneja. Sin duda, este es un asunto pendiente de afrontar.

En cuanto a las técnicas disponibles para garantizar la calidad de la información, la mayoría de métodos implementados se centran en la fase de recogida de datos pero no aportan ninguna estrategia para la información que ya ha sido grabada. Este trabajo aporta soluciones a esta situación concreta.

El núcleo fundamental de esta tesis se centra en describir las tipologías de error que pueden contener los datos ya grabados. También se aportan los algoritmos necesarios para automatizar la detección y corrección de estas incidencias.

Este trabajo se estructura en seis capítulos, los tres primeros de carácter teórico y los dos últimos aplicados. El capítulo primero revisa de forma sucinta los conceptos básicos de “medida” en psicología y “error”, así como los controles que se deben implementar para minimizar las incidencias en el proceso de los datos. Entre los controles a implementar se distingue entre aquellos que son aplicables a la captura diferida y los que lo son a la captura directa.

El segundo capítulo se centra en la definición operativa del error en el proceso de la gestión de los datos, exponiendo la necesidad de evaluar la calidad de la información previamente a su análisis estadístico. Se introduce el concepto de depuración que se define como un proceso a realizar necesariamente tras haber capturado los datos para detectar y corregir los errores que contienen.

En el tercer capítulo se revisan múltiples controles propuestos para distintas tipologías de variables. Se muestran técnicas de detección de errores por registros duplicados, en variables cuantitativas, en variables categóricas, etc. Se presentan

técnicas novedosas, como el uso de “Data Mining” en la detección de errores, y se introducen procedimientos que, pese a ser habituales en otras disciplinas como la economía o la auditoría, no lo son en absoluto en las Ciencias de la Salud.

En el cuarto capítulo, de carácter aplicado, se expone el proceso de depuración propuesto en esta tesis y se especifican controles y chequeos para todas las tipologías de variables descritas en los apartados teóricos, se detalla el tipo de comprobación que debe efectuarse y el algoritmo en pseudocódigo que permite su implantación. Asimismo, para cada tipo de variable se ha programado una macro en lenguaje SPSS que permite automatizar el control.

En el quinto capítulo se detalla la aplicación de la metodología de depuración propuesta a datos reales: la Encuesta Sociodemográfica, el Conjunto de Datos Mínimo de Alta Hospitalaria y la Historia Clínica Electrónica. Estos ejemplos se caracterizan por bases de datos extensas y de estructura compleja. En esta parte del trabajo se expone de qué modo se ha realizado la depuración y se valora el comportamiento mostrado por este proceso. El procedimiento propuesto ha demostrado ser flexible y fácilmente adaptable a estos estudios. Las conclusiones derivadas de estas experiencias también han servido para mejorar nuestro análisis sobre el proceso de depuración.

Finalmente, en el último capítulo se exponen las principales conclusiones y se discuten las implicaciones teóricas y prácticas de este trabajo.

1
ERROR EN EL PROCESO DE MEDICIÓN EN PSICOLOGÍA

1.1 INTRODUCCIÓN

El modelo general de investigación científica se articula en tres niveles: teórico-conceptual, técnico-metodológico y analítico-estadístico (Arnau, 1996). Este modelo se desarrolla mediante un proceso que acontece de forma integradora, iterativa y cíclica, y constituye el vehículo común de adquisición y actualización de conocimientos para las denominadas ciencias empíricas. Su referente más próximo se encuentra en el método hipotético-deductivo (o "teórico-formal", como se ha denominado de forma clásica), y una de sus características más distintivas radica en el principio de verificación empírica que facilita la descripción, la clasificación y la explicación de los fenómenos y de sus relaciones.

Dentro del modelo de investigación científica, el nivel teórico-conceptual constituye un estadio general (punto de partida y finalización del proceso) que facilita la elaboración del marco teórico y la definición de las hipótesis objeto de contrastación empírica. Autores como Bayés (1984) indican que el planteamiento de esta etapa responde al reconocimiento de lagunas en las distintas áreas (tanto teóricas como aplicadas) que integran las disciplinas, así como también a la obtención de resultados contradictorios o a la acumulación de evidencias aparentemente desconectadas de los conocimientos disponibles. Las hipótesis que se especifican constituyen tentativas de solución ante estas carencias, y requieren de una formulación que guíe el desarrollo de un adecuado plan de investigación que permita su verificación empírica. Por otro lado, como punto final del proceso investigador, el nivel teórico implica la discusión, generalización y comunicación de los resultados que se obtienen en la fase estadístico-analítica.

En el nivel técnico-metodológico se concretan las formas en que se procederá a la verificación de las hipótesis. Esta etapa se caracteriza por la realización de dos tareas fundamentales: el diseño del plan de investigación y la elaboración de la estrategia que permitirá recoger los datos necesarios para dar respuesta a las preguntas que se formulan.

Finalmente, en el nivel analítico-estadístico se reúnen los datos, se elaboran y se obtienen conclusiones. El objetivo principal consiste en la valoración del grado de credibilidad que merecen las hipótesis formuladas, y para ello se procede a su contrastación con los datos empíricos. Precisamente, los resultados obtenidos de este análisis permiten "retornar" al nivel conceptual, con objeto de aceptar la hipótesis teórica (de forma provisional, hasta que se encuentren suficientes evidencias para desestimarla) o reexpresarla. De este modo, se reinicia un nuevo ciclo.

La falta de rigor en cualquiera de los tres niveles que conforman el modelo general de investigación científica conlleva necesariamente la falta de confianza en los resultados que se obtienen y en las conclusiones que se formulan. El estudio de las deficiencias que afectan los niveles teórico y metodológico ha recabado un gran interés

en las últimas décadas, aunque no constituyen el objetivo de este trabajo. Por otro lado, los estudios realizados para la identificación de las potenciales fuentes de error en el nivel analítico han centrado su atención en el análisis estadístico, ya que se considera la fase más emblemática del proceso. Actualmente, sin embargo, se evidencia que las posibles deficiencias producidas durante el análisis de los datos son más fáciles de detectar y de solucionar que las que se producen en otras etapas. Asimismo, una de las consecuencias de este acrecentado interés ha consistido en la falta de atención a los múltiples errores que se pueden cometer durante la obtención, grabación y el manejo de los datos.

Este trabajo encuentra su justificación en el estudio de uno de los procesos básicos que garantizan que la información que se analiza está exenta de inconsistencias: la depuración. Su formalización teórica conlleva el planteamiento y discusión de dos aspectos cruciales en el contexto de la investigación: la gestión y la calidad de datos.

1.2 EL PROBLEMA DE LA MEDIDA EN PSICOLOGÍA

El estudio de la gestión y la depuración de datos entronca con conceptos fundamentales en el ámbito de la Psicología, *la medición* y *el error de medida*. En los siguientes subapartados se revisan de forma breve ambos aspectos.

1.2.1 La medición

La medición constituye una operación inherente al propio proceso científico, y según numerosos investigadores se encuentra en la base del uso de los métodos cuantitativos en Psicología. Hoy resulta obvio que la investigación científica en nuestra disciplina versa sobre fenómenos no cuantificables de forma precisa y estable en el tiempo. El requisito indispensable para el estudio empírico es que el fenómeno acontezca de forma relativamente regular (Arnau, 1996).

La medición se puede definir como la asignación de fenómenos psicológicos a números en base a unas reglas preestablecidas, que permitirán cuantificar su magnitud y facilitar su comparación con otras medidas análogas. Para profundizar en las controversias que ha generado a lo largo de la historia la posibilidad de medir de forma efectiva los fenómenos psicológicos se recomienda revisar el trabajo de Jáñez (1989).

Entre las principales razones por las cuales es imprescindible la medición para que pueda hablarse de ciencia destacan dos: a) la operativización de la medida posibilita a los investigadores que trabajan en entornos semejantes la comparación entre fenómenos y el intercambio de información, y b) la medición de las conductas permite operar matemáticamente sobre los valores obtenidos y realizar análisis a partir de ellos.

Es evidente que sólo a partir de la medición y del registro puede surgir la posibilidad del error en el fenómeno objeto de estudio. Diversos autores han alertado sobre esta posibilidad. Por ejemplo, Huber (1981) especulaba que era de esperar que al menos entre un 2% y un 5% de las observaciones contuvieran errores.

1.2.2 El error

Podríamos definir el error como la inexactitud entre el valor de una medida y el valor real del objeto medido. Esta inexactitud puede darse por diversos motivos y ser vista desde distintos ángulos. Este apartado pretende distinguir, de forma sucinta, entre el concepto de error que utilizaremos a lo largo de esta tesis y el concepto de error que se maneja en otros ámbitos de la metodología en psicología.

Desde la psicometría se suele diferenciar entre error sistemático y error aleatorio (Losada y López-Feal, 2003). El error sistemático se asocia al instrumento o método utilizado en la medición. Se manifiesta en el mismo sentido y puede atribuirse a factores tales como la interpretación de los resultados de una prueba o los criterios de puntuación.

El error aleatorio se asocia a variables de difícil, o imposible, control por parte del investigador, y su existencia se deduce de las "pequeñas" discrepancias en la medición que aportan instrumentos de medida análogos en el mismo momento, e incluso del mismo instrumento administrado en dos momentos distintos. Entre los motivos que inciden en la magnitud de este error podríamos nombrar variables relacionadas con el investigador o con imperfecciones del instrumento, entre otros.

La estadística, por su parte, utiliza conceptos muy semejantes. Así, las observaciones se constituirían de una parte sistemática y otra aleatoria. La primera de ellas se referiría a la parte de la medición que se corresponde con el objeto real al que representan, mientras que la parte aleatoria representaría la incertidumbre sobre la realidad objeto de estudio.

Un objeto de investigación ha sido el estudio de la reducción del error. Dependiendo del enfoque, se han desarrollado estrategias para minimizar su impacto sobre las medidas que serían analizadas.

En esta tesis se aborda el concepto de error como la discrepancia, producida durante la fase de gestión de los datos entre el valor medido y el registrado magnéticamente. Prescindiremos, por tanto, de errores en el sentido de divergencia respecto a la realidad, causados por los instrumentos de medida y/o la situación del sujeto, así como de conceptos análogos.

Lamentablemente, los estudios realizados para minimizar los errores producidos durante la gestión de los datos se han desarrollado de forma tardía. En concreto, la definición de tipologías de error y el establecimiento de rutinas para chequear y asegurar la calidad de la información a analizar es un ámbito que necesita ampliar su desarrollo. La calidad de los datos es un tema casi inexistente en los textos de psicometría y de estadística, que suelen ocuparse de técnicas de descripción y de modelado de los datos ya grabados y obvian todo lo ocurrido anteriormente.

1.3 CONTROLES DEL ERROR

La Figura 1-1 presenta cada una de las fases de la gestión de los datos. En sombreado se incluyen las diversas estrategias propuestas para reducir el error. Para garantizar la calidad final resulta imprescindible definir con precisión todas las fases del proceso de datos de un estudio, procurando seleccionar e implementar adecuadamente las mejores estrategias (Moritz et al., 1995).

Atendiendo a la primera fase de la Figura 1-1, es frecuente que el diseño de la estructura de la base de datos de un estudio se realice mediante el uso de un sistema gestor de bases de datos (SGBD). Si es de tipo relacional, como es habitual, este diseño conlleva un proceso de normalización (Codd, 1985a, 1985b), consistente en la aplicación iterativa de un conjunto de reglas conocidas como formas normales. El objetivo que se persigue es: 1) evitar redundancias, duplicidades y valores nulos innecesarios; 2) facilitar los cambios que en un futuro se tengan que realizar en la estructura de las tablas; y 3) reducir el impacto de los cambios de estructura de la base en las aplicaciones que acceden a los datos (Jennings, 1997).

El proceso de normalización mejora de varios modos la calidad de la gestión de datos de un estudio. Por ejemplo, la aplicación de la primera forma normal garantiza que los registros están unívocamente identificados, evitando tanto registros duplicados como carentes de identificador, y que no existen grupos de repetición (campos de ocurrencia múltiple). Mediante la segunda y tercera formas normales se elimina la información redundante que persiste tras la aplicación de la primera forma normal asegurando que sea independiente entre sí. Finalmente, la cuarta y quinta formas normales garantizan la actualización de los datos al mantener la integridad referencial y la recuperación de la información (Date, 1999).

Una vez definida la estructura de la base de datos se procede a la recogida y captura de los datos. La Figura 1-1 muestra que la grabación puede realizarse de modo diferido (“off line”) o directo (“on line”). En el primer modo los datos se encuentran contenidos en otros soportes, habitualmente en papel, y la tarea del operador consiste en transferirlos a soporte magnético. En la captura directa la grabación se realiza de forma simultánea al registro del dato. En líneas generales, las estrategias utilizadas para mejorar la calidad de los datos durante su grabación deben garantizar que la información esté exenta de inconsistencias, que contenga el menor número de valores desconocidos (*missing*) y que se ajuste a los momentos temporales de registro establecidos en el diseño (especialmente en el caso de estudios longitudinales).

1.3.1 Captura diferida (“off line”)

Aunque en la captura diferida es posible implementar controles de calidad automáticos y no automáticos, en este apartado nos centraremos en los segundos por ser exclusivos de este tipo de grabación. Entre los principales controles de calidad no automáticos destacan (Granero, 1999): la comparación visual, la doble entrada, la verificación aleatoria y las técnicas de lectura óptica. El objetivo común de estos procedimientos es garantizar una correspondencia perfecta entre la información contenida en las fuentes originales y los datos grabados.

Más adelante se detallan controles automáticos que también tienen cabida en la captura diferida. El objetivo de programar estos controles (de forma simultánea a técnicas como la doble entrada o la verificación aleatoria) es mejorar la consistencia de

la información que está siendo grabada en la base de datos. En este caso, los valores inconsistentes no serán grabados y estos campos no coincidirán con la fuente original.

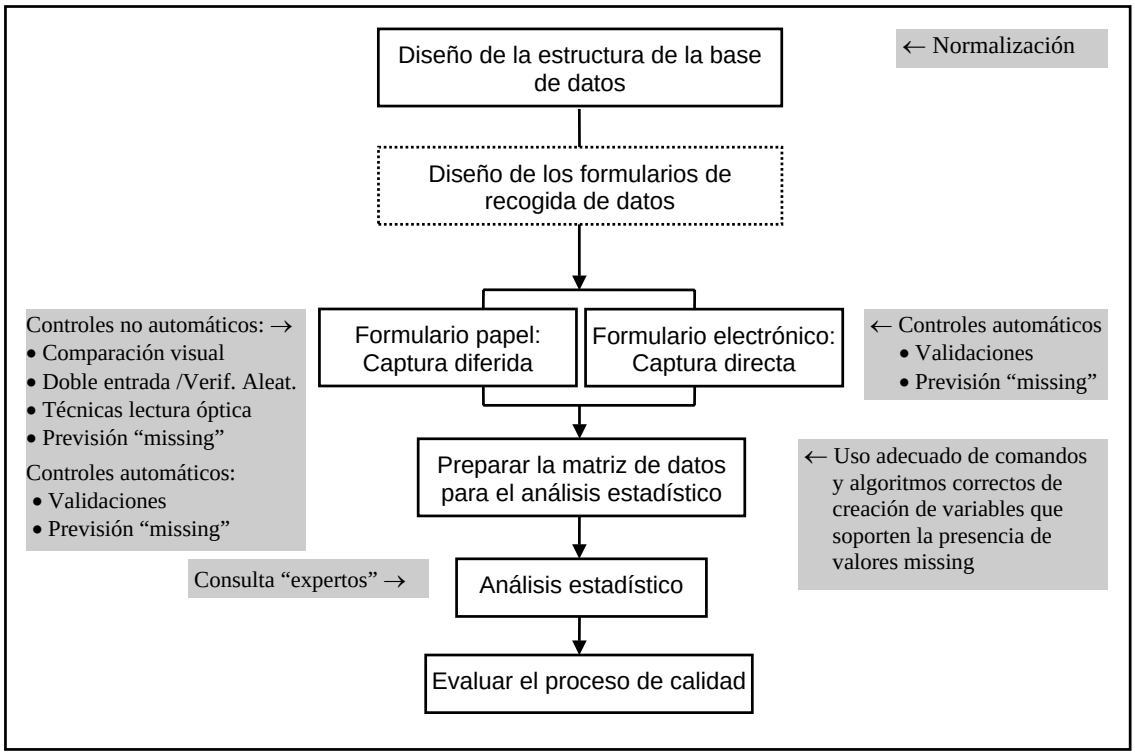


Figura 1-1: Fases y controles básicos de calidad en la gestión de datos (modificada de Granero, 1999).

Es también importante destacar que cuando se efectúa una captura diferida, el uso combinado de controles automáticos y no automáticos no equivale a la captura directa con protecciones. Registrar la información de forma directa, programando todos los controles posibles, posee una sustancial ventaja: la proximidad física al dato en el momento de la captura permite que éste sea contrastado (y si es preciso recuperado) ante cualquier incidencia que pueda surgir.

1.3.1.1 Comparación visual

La comparación visual consiste en que una o más personas cotejen visualmente los datos grabados con los registrados en las fuentes originales. La mayor ventaja relacionada con esta técnica radica en que permite verificar la calidad de las dos fuentes: la que contiene los datos originales y la de los datos grabados. Sin embargo, sólo se considera una técnica útil cuando no es posible la adopción de otras más fiables, ya que constituye un procedimiento muy elemental y costoso. En otras palabras, la comparación visual queda relegada a aquellos estudios en los que se manejan pocos datos y cuando se dispone de la garantía de que el sujeto que la realiza está altamente motivado porque asume la responsabilidad de otras fases de la investigación y también de la presencia de errores (Blumenstein, 1993).

1.3.1.2 Doble entrada

La doble entrada (DE) es una técnica de control no automática consistente en grabar los datos dos veces, cotejar ambas grabaciones y revisar los campos discrepantes. Es una técnica basada en la recaptura de datos.

En cuanto a su realización, la DE se puede llevar a cabo por un único operador (encargado de entrar la misma serie en dos ocasiones diferentes) o por operadores distintos (cada uno de los cuales graba la misma serie en una única ocasión). Esta última modalidad constituye el procedimiento de elección ya que si un dato presenta problemas de legibilidad resulta imprescindible disponer de interpretaciones diferentes e independientes (Blumenstein, 1993; Cobos, 1995; Gassman et al., 1995).

Tras la doble captura se activa el proceso de comparación automática, que puede efectuarse de forma directa por los mismos operadores encargados de introducir los datos o de forma diferida por un tercer sujeto encargado de revisar y resolver las incongruencias aparecidas. Sea cual sea la modalidad que se adopte, resulta imprescindible que toda modificación realizada sobre los datos se realice atendiendo a un protocolo específicamente diseñado con este fin. El máximo nivel de garantía se obtiene cuando no sólo se graban los valores corregidos sino también los valores originales, la/s fecha/s de cada modificación, las razones que han conducido a la corrección y las personas que las realizan y/o autorizan (Gassman et al., 1995).

Pese a que es una técnica que ha demostrado suficientemente su eficacia, ya que reduce la proporción de errores hasta en un 30% (Gibson, Harvey, Everett y Parmar, 1994; Neaton, Duchene, Svendsen y Wentworth, 1990; Wolf, 1993) en ocasiones también se ha considerado poco eficiente porque el coste que supone la doble grabación la relega a estudios con recursos suficientes (Reynolds-Haertle y McBride, 1992). Por otro lado, se ha argumentado que en gran medida el éxito de la DE radica en la disponibilidad de operadores con experiencia, entrenados en el uso de los formularios utilizados en el estudio y concienciados de la importancia que supone grabar los datos sin ningún error (Gibson et al., 1994; Stellman, 1989). Finalmente, se ha indicado que la DE no garantiza totalmente la calidad del registro si los datos no son sometidos también a chequeos para detectar las inconsistencias que podrían contener las fuentes documentales originales (Cobos, 1995).

1.3.1.3 Verificación aleatoria

La falta de eficiencia de la doble entrada ha provocado la búsqueda de alternativas que reduzcan la cantidad de datos que deben ser recapturados. Por ejemplo, se ha propuesto implementar procedimientos de verificación selectiva que obliguen a grabar dos veces los datos considerados más importantes y sólo una vez el resto de la información (Mullooly, 1990; Neaton et al., 1990).

La verificación aleatoria es una estrategia de recaptura que consiste en reintroducir un porcentaje determinado de los valores escogidos al azar y solicitados al operador con suficiente retardo como para evitar el efecto del recuerdo. Esta técnica requiere necesariamente que sea el mismo operador quien reintroduzca los datos, y tiene la ventaja de proporcionar un “feed-back” inmediato de los errores que se van cometiendo. Los trabajos realizados hasta el momento indican que la verificación aleatoria ofrece prestaciones semejantes a la doble entrada reduciendo sensiblemente los costes de la grabación (Doménech, Losilla y Portell, 1998; Granero y Doménech, 2001;

Granero, Doménech y Bonillo, 2001). Por otro lado, este sistema también facilita una estimación del porcentaje de errores contenidos en los datos y un índice de la aptitud y/o eficacia de los operadores.

1.3.1.4 Lectura óptica

Existen tecnologías que simplifican la grabación de los datos mejorando notablemente su calidad (Barton et al., 1991), como por ejemplo las técnicas de lectura óptica que escanean las fuentes documentales originales (Cardiff Software Inc., 1998a, 1998b).

Los formularios de lectura óptica son especialmente adecuados cuando la mayoría de variables registradas son de alternativa múltiple. Cuando los formularios incluyen respuestas manuscritas, por ejemplo respuestas numéricas, el escaneo requiere un estudio previo de fiabilidad del reconocimiento óptico de estos caracteres que se realizará a partir de una muestra aleatoria de formularios. Si la fiabilidad del reconocimiento es aceptable la grabación requerirá una doble captura con distintos operadores que efectúen interpretaciones independientes para resolver las alternativas que el software presenta ante los caracteres con problemas de identificación.

La lectura óptica permite evitar los errores introducidos por el operador y disminuye el coste de la grabación. En contrapartida, esta técnica de captura posee sus propias deficiencias derivadas de: a) la falta de formación del personal que diseña los formularios; b) el uso de software poco fiable; c) formularios rellenados con marcas deficientes; d) dificultad en el reconocimiento de caracteres; y e) la falta de protocolos fiables para solventar los problemas de legibilidad e interpretabilidad durante la captura.

Cuando se presenta un problema de legibilidad o de interpretabilidad, los programas de lectura óptica requieren del operador la decisión final sobre el valor que será grabado. Este *modus operandi* es asimilable a la situación que se plantea cuando un grabador de datos decide qué valor introducir ante un carácter "dudoso". Lamentablemente, los programas de lectura óptica no disponen de opciones de recaptura para los datos introducidos manualmente. Por consiguiente, la secuencia óptima para efectuar una captura óptica consiste en realizar el análogo a una doble entrada: a) el escaneo se realiza dos veces con dos operadores diferentes; b) a continuación se comparan las dos tablas de datos; y c) se toman decisiones sobre las discrepancias entre ambas grabaciones.

La lectura óptica adolece del mismo problema que la doble entrada y la verificación aleatoria: si los datos no son sometidos a chequeos que detecten inconsistencias en las fuentes originales, los errores que éstas contengan serán grabados y propagados a las fases posteriores, afectando el análisis estadístico y las conclusiones que se formulan. Y es que estas técnicas de control no automáticas detectan las discrepancias entre los valores contenidos en las fuentes originales y los valores grabados, pero no son sensibles a las inconsistencias del dato obtenido directamente del informante.

1.3.1.5 Previsión de valores desconocidos ("missing")

La existencia de valores desconocidos es una constante en cualquier investigación. Independientemente de la rigurosidad con la que se haya planificado la recogida de datos, al final de un estudio es frecuente no disponer de toda la información necesaria debido a diferentes causas (Fowler, 1993): 1) por falta de cooperación de los informantes (habitualmente como consecuencia de una actitud recelosa que provoca un rechazo a contestar las preguntas que se formulan); 2) por mero desconocimiento u olvido de los contenidos que se preguntan; 3) por las propias interferencias que provocan los problemas de los individuos (que en ocasiones pueden ocasionar falta de inteligibilidad o comprensión por parte de los respondientes); 4) por las dificultades del propio investigador a la hora de conducir la recogida de datos (por ejemplo por la falta de familiaridad o entrenamiento con los formularios que se utilizan).

Existe una actitud general de olvido ante la presencia de datos desconocidos. Se considera que la proporción de valores desconocidos constituye un indicador más de la calidad del proceso de recogida de los datos y de la información que se maneja, se analiza y se interpreta. Sin embargo, en los trabajos publicados raramente se explica cómo se ha abordado el tema de los datos desconocidos ni cuál es su magnitud. En la mayoría de estudios no se discuten ninguno de los aspectos concernientes al plan de recogida de los datos, y las soluciones propuestas se centran en las fases de preparación de la matriz de datos y en el análisis estadístico (Vach y Blettner, 1991).

Actualmente, los trabajos aparecidos en ámbitos más teóricos conceden gran importancia al uso de estrategias secuenciales que minimicen las consecuencias de la falta de respuesta (Taylor y Amir, 1994). Estas propuestas encuentran su aplicación desde la fase de recogida de datos. En el caso de la captura diferida, el control relacionado con la falta de información comporta diseñar los formularios de manera que incluyan la opción de valor desconocido para permitir posteriormente tratar esta "falta de información". Los argumentos utilizados para defender esta postura son básicamente dos: 1) la disponibilidad de un código de "no respuesta" permite distinguir la falta real de respuesta por parte del sujeto de los "descuidos" del registrador (esto es, del uso inadecuado de los instrumentos de medida); 2) cuando en un formulario que no contempla la posibilidad de dejar campos vacíos no se codifican algunos ítems se incide sobre las propiedades psicométricas del instrumento, sobre todo cuando se crean puntuaciones totales, ya que en estos casos rara vez existen indicaciones sobre cómo agregar los campos sin un valor válido (Granero y Doménech, 1997).

1.3.2 Captura directa ("on line")

Si el diseño del estudio lo permite, la captura directa es el procedimiento de elección, ya que se logran efectos análogos a la doble entrada y a la doble lectura óptica pero de forma mucho más eficiente. En este apartado, se entenderá como captura directa el registro de las respuestas de un sujeto a una prueba autoadministrada o a una entrevista. La captura directa aventaja a la doble entrada porque sólo es necesario introducir una vez la serie y es preferible a la lectura óptica porque su proximidad al dato permite resolver problemas acaecidos durante su medición. Además, la captura directa elimina el *prerregistro* de los datos y, por lo tanto, impide la falta de correspondencia entre el formato de papel (que deja de existir) y el registro informático.

En la actualidad existe una fructífera línea de trabajo relacionada con la construcción de instrumentos de medida informatizados: la denominada *Human Computer Interaction*. En este contexto se realizan numerosos estudios para obtener

conocimiento sobre qué elementos de los formularios, de las *interfaces* y de los modos de interacción influyen y mejoran en aspectos perceptuales, atencionales y motivacionales de los respondientes y de los operadores. El objetivo no es otro que el de facilitar el procesamiento de la información y obtener de los sujetos las mejores respuestas, disminuyendo cuanto sea posible los errores de medida (Carroll, 1993; González, 1993; P.E. Johnson, 1992; Martin y Fuerst, 1987; Pocius, 1991; Sánchez, 1992; Schneiderman, 1992; Wallace y Anderson, 1993).

Sin embargo, y a pesar de sus ventajas, la captura directa plantea un problema común a la doble entrada: no asegura la consistencia de la información, y por lo tanto no garantiza por sí misma la calidad del registro. Para conseguir este objetivo se deben programar un conjunto de controles automáticos que incorporen todas las restricciones posibles y así asegurar que los datos son consistentes.

Atendiendo al momento en que son evaluadas, es posible establecer tres tipos de restricciones automáticas entre los datos (Date, 1999): prevalidaciones, invalidaciones y postvalidaciones. Estos controles pueden ser definidos de forma bastante sencilla por gran parte de los SGBD, y garantizan que cada variable cumpla las condiciones de consistencia que afectan a (Gassman et al., 1995): a) la propia variable; b) otras variables contenidas en el mismo formulario; y c) las condiciones de consistencia de otros cuestionarios relacionados (requisito indispensable cuando se programan seguimientos).

1.3.2.1 Pre-validaciones, in-validaciones y post-validaciones

Una pre-validación es una condición lógica asociada a un campo que se evalúa previamente a introducir ningún valor en él y que determina si se permite su edición o si se le asigna un valor automáticamente que puede ser “no aplicable”, desconocido o “deducible” (conceptos que se expondrán en el apartado 4.2). Por ejemplo, una pre-validación no debe permitir que las preguntas referentes a un diagnóstico de anorexia se formulen a un informante sabiendo que éste no ha perdido peso (en este caso, estos campos se deben completar de forma automática con valor “no aplicable”).

Una in-validación es una protección que sólo permite grabar datos cuyo formato corresponda con el definido para el correspondiente campo. Esta protección está implementada en todos los programas que contienen el concepto de “formato del campo”. Por ejemplo, una in-validación no permite que se introduzca un valor cadena en un campo definido como fecha, y tampoco un número con decimales si el formato del campo se ha definido para valores numéricos enteros.

Una post-validación es una protección que determina el rango de valores admisible en un campo y las condiciones que éstos deben cumplir respecto a otros campos. Por ejemplo, una post-validación no debe permitir grabar como valor de CI, medido en escala Weschler, un dato negativo o superior a 150.

1.3.2.2 Previsión de valores desconocidos

En el caso de la captura directa, la previsión de valores desconocidos implica un diseño adecuado de los instrumentos que limite la posibilidad de que el operador que

entra los datos deje campos vacíos cuando éstos cumplen las condiciones de edición (Granero, 1999). Sin embargo, cuando se realiza esta previsión es importante permitir que el operador encargado de recoger la información pueda dejar campos *pendientes de edición* con el objetivo de poder registrar y/o revisar los datos con posterioridad. Por ejemplo, si en el momento de efectuar una entrevista personal el entrevistado no recuerda cuánto pesa, este campo se puede dejar pendiente para ser completado en otro momento (durante otra entrevista, llamándole por teléfono, etc.).

1.3.3 Preparación de la matriz de datos

En ocasiones, las variables que se analizan estadísticamente no son las respuestas directas dadas por los sujetos, sino indicadores que se han creado combinando la información que ha sido recogida a través de tests, entrevistas, cuestionarios, etc. Puesto que los algoritmos que permiten la creación de estos indicadores raramente están disponibles en lenguaje formal, cada investigador asume la responsabilidad de su definición. Y es en este punto, precisamente, donde aparecen las dos dificultades principales que se relacionan con la falta de calidad durante la fase de preparación de la matriz de datos: 1) la falta de conocimiento sobre la forma en que funcionan los comandos de creación y transformación de variables del "software" utilizado; y 2) la definición de algoritmos que no soporten la presencia de valores desconocidos.

Atendiendo al razonamiento anterior, es importante destacar que los diferentes sistemas de proceso estadístico de datos tienen un comportamiento muy distinto frente a los datos desconocidos. Por ejemplo, algunos son extraordinariamente sensibles y dejan el indicador en blanco cuando alguna de las variables que se incluyen en su definición está vacía; otros simplemente ignoran esta falta de valor. Precisamente, este comportamiento tan diversificado hace que el usuario deba conocer extensamente las instrucciones de creación y transformación de variables.

Por otro lado, preveer durante la fase de captura la posibilidad de valores desconocidos contribuye a disminuir su presencia en la base de datos del estudio, pero no los elimina. Una de las primeras dificultades que se plantea ante la falta de respuesta tiene que ver con la decisión de si es pertinente o no el uso de procedimientos de estimación para imputar estos valores. Los partidarios de utilizar estas técnicas argumentan que la no imputación de los valores desconocidos afecta notablemente la calidad de la investigación, tanto a nivel de validez interna como externa, ya que implica la eliminación de una parte de la muestra y los riesgos que se derivan (Huberty y Julian, 1995; Orme y Reis, 1991; Vach y Blettner, 1991; Whitehead, 1994): 1) el número de sujetos que entran a formar parte del análisis estadístico puede no ser constante (especialmente cuando se realizan ajustes de modelos con procedimientos de entrada o exclusión secuencial), provocando variaciones importantes en los niveles de significación e inestabilidad en los resultados; 2) la distribución de los valores desconocidos acostumbra a no ser aleatoria, factor que atenta gravemente sobre el grado de representatividad de la muestra (especialmente porque los sujetos que se eliminan suelen poseer características particulares que se relacionan tanto con los predictores como con las respuestas); y 3) la eliminación indiscriminada de sujetos sin valor en alguna variable reduce la potencia de los análisis estadísticos, la precisión de las estimaciones y la posibilidad de generalización de los resultados.

Hay técnicas de estimación de valores desconocidos simples y complejas. Entre los procedimientos simples destacan la imputación directa de la media, mediana o

moda. Entre los complejos destaca el modelado estadístico (Espeland et al., 1992; W.D. Johnson, George, Shahane y Fuchs, 1992; Wei y Tanner, 1991) y en especial las técnicas de imputación múltiples (Efron, 1994; Rubin, 1987, 1996; Rubin y Schenker, 1991). Es importante destacar que la imputación de valores desconocidos no siempre constituye la mejor solución, y en la actualidad están apareciendo trabajos que alertan sobre los peligros que se relacionan con cada procedimiento de estimación y con su posible falta de validez (Crawford, Tennstedt y McKinlay, 1995; Greenland y Finkley, 1995; Kochhar, 1991; Schafer, 1997). Por otro lado, las técnicas de estimación de valores desconocidos raramente están incorporadas en los programas que se utilizan para el manejo y análisis de los datos. En síntesis, aunque se trata de un área en pleno desarrollo, en la actualidad se considera tan relevante el análisis de las causas de la falta de respuesta como el desarrollo de técnicas de imputación directa. Para ampliar en el estudio de los datos desconocidos remitimos a la lectura del texto de Little y Rubin (1987), donde además se incluye una útil clasificación de los mecanismos que pueden provocar la falta de respuesta y diversas propuestas para su tratamiento.

Finalmente, otra de las dificultades que se relaciona con la preparación de la matriz de datos consiste en que el software no permite diferenciar de forma automática entre códigos de valor “no aplicable” y valor desconocido (véase apartado 4.2). A pesar de los diferentes significados conceptuales, en la práctica el valor no aplicable y el desconocido se suelen asimilar a un mismo código: “vacío”. La no diferenciación de estas tipologías es también causa frecuente de error durante el análisis estadístico (Sonquist y Dunkelberg, 1977).

1.3.4 Análisis estadístico

Durante esta fase también se pueden dar situaciones que atentan contra la calidad: 1) el desconocimiento sobre cuáles son las mejores estrategias para efectuar el análisis estadístico; 2) el uso de programas o instrucciones que generan resultados erróneos; o 3) la disponibilidad de sistemas con una interfaz diseñada de forma que induzca a errores en su manejo. Puesto que estas dos últimas dificultades raramente son conocidas por una gran mayoría de los usuarios, las normas de publicación de la mayoría de revistas científicas exigen indicar el programa utilizado y la versión.

De todos modos, como ya hemos apuntado, los errores en esta etapa son de solución relativamente simple, ya que en muchas ocasiones únicamente implican la elección de estrategias o programas de análisis distintos. Generalmente, la simple consulta con expertos basta para subsanar las potenciales deficiencias.

Por otro lado, durante la fase de análisis estadístico también se puede efectuar una nueva inspección para detectar valores anómalos en los datos (esto es, sospechosos de ser erróneos). Así, mediante análisis exploratorio, examinando residuales en modelos multivariantes, detectando sujetos influyentes en modelos de regresión y mediante técnicas de *data mining* es posible detectar patrones de respuestas que comporten sospechas de la veracidad del registro.

2
ERROR EN EL PROCESO DE GESTIÓN DE DATOS

2.1 EL PROCESO DE GESTIÓN DE DATOS

La gestión de datos ("data management") se entiende como una tarea compleja que abarca un conjunto de fases secuenciales (Rondel, Varley y Weeb, 1999): 1) definir la estructura de la base de datos; 2) definir e implementar todas las protecciones posibles entre los datos para detectar y eliminar el máximo número de incongruencias durante la recogida de la información; 3) garantizar que los datos que han sido registrados no contienen inconsistencias; 4) preparar la matriz de datos para el análisis estadístico, creando las variables necesarias para contrastar las hipótesis empíricas; 5) emplear las técnicas de análisis estadístico adecuadas; 6) evaluar la calidad de las estrategias utilizadas durante todo el proceso de recogida y manejo de la información; y 7) consensuar procedimientos de control para garantizar la calidad de la gestión de datos que sean aplicables en futuras investigaciones, valorando en cada caso los costes de su implementación.

De lo expuesto en el párrafo anterior se deduce que garantizar la calidad resulta un objetivo inherente al propio proceso de gestión de los datos. Sin embargo, aunque se tienen referentes de estudios en esta línea desde la década de 1980 (Ember, 1986), la calidad del proceso de gestión de datos no ha recabado plena atención hasta hace apenas unos años (Cobos, 1995; Cody, 1999; Doménech, Losilla y Portell, 1998; Gassman, Owen, Kuntz, Martín y Amoroso, 1995; Granero, 1999; Granero, Doménech y Bonillo, 2001; Magnusson y Bergman, 1990).

La falta de calidad durante el proceso de gestión de los datos afecta gravemente a la validez interna y externa de un estudio, pudiendo hacerlo incongruente y difícilmente replicable, ya que todo error de gestión (por pequeño que éste sea) se propaga y se magnifica en fases posteriores (Abelson, 1998; Lewis-Beck, 1995). Este reconocimiento ha facilitado el desarrollo de estrategias para mejorar globalmente la calidad del proceso (Freedland y Carney, 1992; Groves, 1989; Moritz et al., 1995; G.R. Norman y Streiner, 1996; Redman, 1992; Rowley, 1989; Saris, 1991; Yeoh y Davies, 1993), y ha permitido comprobar que el mero uso de sistemas informáticos no asegura el correcto funcionamiento de todas las operaciones que conlleva el manejo y el análisis de la información (Butcher, 1994). El síndrome GIGO, iniciales de "Garbage In–Garbage Out" (Cobos, 1995), ejemplifica esta situación, y evidencia que cuando los datos analizados no se corresponden con la realidad que supuestamente representan, de nada sirven las herramientas estadísticas más potentes, los programas informáticos de última generación o el personal más competente: si los datos son "basura" las conclusiones que facilitan también lo serán.

El hecho que tradicionalmente la calidad en la gestión de los datos no haya sido un motivo de preocupación para la ciencia no implica que en la actualidad no deba

serlo. La ciencia no es ajena al concepto de excelencia que tan de moda ha estado en los últimos tiempos. En este sentido, consideramos que el coste de adoptar medidas para proteger la calidad de los datos es muy inferior al coste que representa poner en duda la validez de una investigación; además, este coste se reduce de forma paralela y proporcional al desarrollo de nuevos programas informáticos que faciliten dichos controles. Y es que, como afirma Ember (1986), *"los resultados obtenidos en pequeños estudios con datos de calidad pueden resultar mucho más interesantes y concluyentes que los obtenidos en grandes estudios con datos sin control de calidad o cuestionables en lo que se refiere a este aspecto"*.

Existen algunos ámbitos de investigación donde la preocupación por la calidad de los datos y la adopción de controles no es reciente y se ha venido usando de forma sistemática: los ensayos clínicos (ICH, 1994, 1996, 1998) y la investigación y desarrollo militar (DISA, 2001) son los principales exponentes. Estos ámbitos gozan de las condiciones necesarias para desarrollar y adoptar costosos controles que garanticen la calidad de la gestión de los datos y para someter a auditoría este proceso: a) realizar una actividad con suficiente repercusión social como para que la toma de precauciones resulte imprescindible, b) disponer de una estricta legislación, y c) disponer de los recursos suficientes para aplicar estos controles.

En el ámbito de la medicina, la normativa vigente en nuestro país en relación al proceso de gestión de datos de ensayos clínicos se recoge en el Real Decreto 561/1993 (B.O.E., 1993). Autores como Espinosa, Zamora y Feliu (1996) o García (1993) presentan una revisión crítica de la misma. Las conclusiones que se derivan de estos trabajos, así como también de las estrategias que se proponen para mejorar la calidad de la investigación en conjunto, resultan generalizables a la gran mayoría de los estudios que se realizan y publican en nuestra disciplina. Muchas de estas directrices podrían implementarse en los trabajos de investigación empírica que se realizan en Psicología, pero otras, sin embargo, son de aplicación más difícil dada la escasez de recursos económicos para investigar en nuestra disciplina.

Por otro lado, Mora (1999) discute el uso y abuso de la metodología estadística en el contexto de la investigación clínica, analiza las principales causas que originan los errores más habituales durante el proceso de investigación científica (especialmente en aspectos referidos a diseño y análisis de datos), valora el importante rol que juegan las guías y listas de comprobación estadística de cara a la mejora de la calidad del proceso de búsqueda de conocimientos y presenta una exhaustiva casuística de las mismas.

En síntesis, el estudio del proceso de gestión de datos ha originado una fructífera línea de investigación entorno al concepto de calidad de los datos ("data quality"), y los estudios aparecidos en la literatura denotan lo complejo de su delimitación y tratamiento. Por ejemplo, en algunos trabajos se proponen criterios para clasificar las fuentes de error que pueden atentar contra la calidad de los datos (Groves, 1989); en otros se muestran estrategias para minimizar el impacto de estas potenciales fuentes de error (Saris, 1991). Por otro lado, autores como Redman (1992) se centran en el estudio de las primeras etapas de la investigación, y destacan que las principales deficiencias acontecen como consecuencia de los errores debidos a la no observación o a una observación deficiente. Groves (1989), sin embargo, se centra en fases posteriores del proceso de gestión y alerta sobre las deficiencias que pueden acontecer durante la captura y la elaboración de la información.

En líneas generales, y tal como se verá a lo largo de este trabajo, hablar de *datos de calidad* implica que éstos representen con exactitud la información, que no tengan valores desconocidos, que sean consistentes entre si y que sean actuales (DISA, 2001).

A continuación se describen algunas de las estrategias que pueden contribuir a minimizar los errores de gestión y garantizar la calidad de los datos. Nuestro interés se centrará especialmente en la fase de captura, ya que constituye el paso previo e inmediato a la depuración y determina en gran medida cuándo y cómo realizar ésta.

2.1.1 Evaluar el proceso de calidad: verificación

La última etapa de la gestión de los datos debe ser siempre una evaluación de la calidad de la misma mediante su verificación (Connet y Lee, 1990).

La verificación comporta una serie de procedimientos, normalmente mediante la introducción de un banco de datos ficticios de test, que permiten comprobar el correcto funcionamiento de los controles implementados. También es usual obtener valoraciones sobre la funcionalidad de los operadores mediante sistemas de recaptura (Blumenstein, 1993; Gassman et al., 1995; Gibson et al., 1994; Wolf, 1993).

2.1.2 Sobre el uso de *interfaces* adecuadas

Antes de finalizar esta revisión sobre técnicas para mejorar la calidad de los datos es imprescindible recordar la enorme importancia de emplear interfaces adecuadas a lo largo del proceso de gestión de la información. Aunque parezca una obviedad, toda interfaz que facilite al usuario su aprendizaje y uso mejorará la calidad de los datos, y por consiguiente de los resultados que se publican.

Al plantearnos el modo en que las interfaces pueden contribuir a mejorar la calidad de la gestión de datos volvemos a encontrarnos con la *Human Computer Interaction*, ya que desde esta línea de investigación se trata el diseño, el análisis y la valoración de sistemas informáticos interactivos. Su principal objetivo es la identificación de los elementos que influyen en las formas en que se producen las interacciones “hombre-ordenador”, así como también el desarrollo de métodos y guías que ayuden a la construcción de sistemas informáticos. Desde la década de 1970 ha aparecido una considerable literatura en esta línea que aborda conceptos tales como “usabilidad” o “amigabilidad” (Batra y Srinivasan, 1992; Booth, 1991; Eason, 1991; González, 1993; Hartson y Hix, 1989; Lowgren y Lauren, 1993; Rath y Brown, 1995; Senay, 1992; Werner, 1996). A pesar de todo, los problemas de interacción hombre-ordenador no están en absoluto resueltos, tal vez porque las soluciones que se necesitan requieren de un tratamiento altamente interdisciplinar con aportaciones de áreas tan distintas como la Psicología cognitiva, la Psicología experimental, la ergonomía, la ingeniería y la propia tecnología informática (Dix, Finlay, Abowd y Beale, 1993; M.A. Norman y Thomas, 1991). Sea como sea, los estudios empíricos disponibles hasta el momento concluyen que cuando se desarrollan diseños de interfaces centrados tanto en el usuario como en las tareas que se deben llevar a cabo, la actitud de los usuarios finales es más favorable, se reduce el tiempo requerido de entrenamiento y se ofrecen menos oportunidades para entrar errores en los datos (Bannert y Kunkel, 1991).

2.2 CONTROLES GENERALES DE DETECCIÓN DE ERRORES

De los controles expuestos hasta ahora para garantizar la calidad de los datos podríamos distinguir entre los que previenen y los que detectan el error. En la Figura

2-1 se muestran los controles básicos que se dedican a la detección: en negrita y en la parte superior de los procesos vemos el objetivo de los controles y en la parte inferior las técnicas que nos permiten alcanzar el objetivo propuesto. Estos controles se articulan en tres etapas ordenadas cronológicamente.

En primer lugar, las técnicas de recaptura detectan posibles no correspondencias entre el dato a grabar en soporte magnético y el registrado en papel.

En segundo lugar, las comprobaciones automáticas (pre-validaciones, in-validaciones y post-validaciones) y el diseño adecuado de la base de datos detectan valores desconocidos y aseguran la consistencia básica de los valores. Se asegura que el dato es consistente respecto a su campo y respecto a otros valores.

Por último, las técnicas estadísticas pueden detectar falta de consistencia en el conjunto de los datos. Mediante análisis estadísticos se pueden detectar, por ejemplo, patrones de respuesta aberrantes y valores dentro del dominio pero extremos a su grupo de pertenencia, que pueden ser sospechosos de contener errores dignos de revisión.

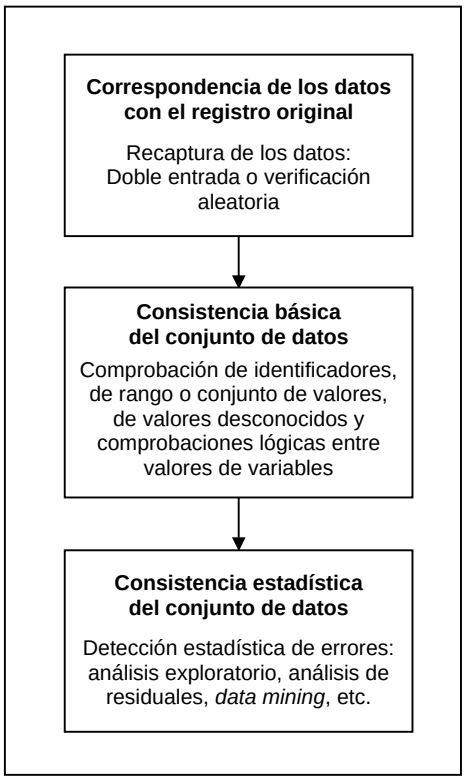


Figura 2-1: Principales controles para la detección de errores.

2.3 DEPURACIÓN

La mayor parte de las técnicas presentadas se implementan con anterioridad a la captura y tienen efecto durante la misma. Aunque se ha prestado escasísima atención a las estrategias a adoptar cuando la captura ya ha sido realizada, algunos autores han hecho hincapié en la necesidad de contar con herramientas que permitan chequear la consistencia de los datos capturados (Fowler, 1993). Acorde con estas propuestas, algunos paquetes estadísticos han incorporado en sus últimas versiones instrucciones dedicadas a este fin.

Así, Stata (2001) ha incorporado en su versión 7 la instrucción ICD9, que actúa a modo de postvalidación. Con esta instrucción se chequea que una variable sólo contiene códigos válidos CIE-9 (siglas de Clasificación Internacional de Enfermedades, 9ª revisión. Modificación Clínica; Ministerio de Sanidad y Consumo, 1989).

El paquete SAS (1999) incorpora desde su versión 7 una serie de instrucciones, llamadas genéricamente *Integrity Constraints*, que especifican reglas a cumplir tanto por los valores existentes antes de especificarlas como por los datos agregados después. Mediante estas instrucciones podemos:

- Mantener la integridad referencial de una tabla respecto a otra.
- Exigir que los registros estén unívocamente identificados.
- Impedir que un campo pueda quedar vacío.
- Limitar el dominio de una variable a un rango, a una lista de valores o a los valores de otras variables.

Es probable que otros paquetes estadísticos se sumen a esta tendencia e incorporen instrucciones específicas para verificar los datos tras su grabación. Ahora bien, no debemos olvidar la limitación intrínseca de estos chequeos: sólo pueden detectar inconsistencias entre los datos.

En la Figura 2-2 se ubica la gestión de datos en el proceso general de investigación científica. La parte sombreada corresponde a la fase concreta de captura y depuración de los datos, y muestra en qué situaciones es imprescindible efectuar un proceso de depuración. El examen de esta figura ilustra que sólo si los datos han sido introducidos con protecciones exhaustivas podemos garantizar que se hallan libres de inconsistencias y se puede proceder a la definición de la matriz de datos y al análisis estadístico.

Realizar un proceso de depuración resulta imprescindible cuando:

- Los datos se han grabado con un software que no tiene implementado el sistema de protecciones.
- Los datos se han grabado con un conjunto de protecciones incompleto. Esta situación puede darse como consecuencia de: a) no definir *a priori* todas las condiciones que deben darse para asegurar la consistencia de la información; b) haber definido o programado mal alguna protección.
- Se añaden datos procedentes de otros registros, tanto si están informatizados como si no. En este caso, aunque los datos incorporados tengan garantizada su validez interna porque fueron sometidos a comprobaciones exhaustivas, es necesario comprobar la coherencia de los datos importados respecto a los existentes en la base de datos del estudio.

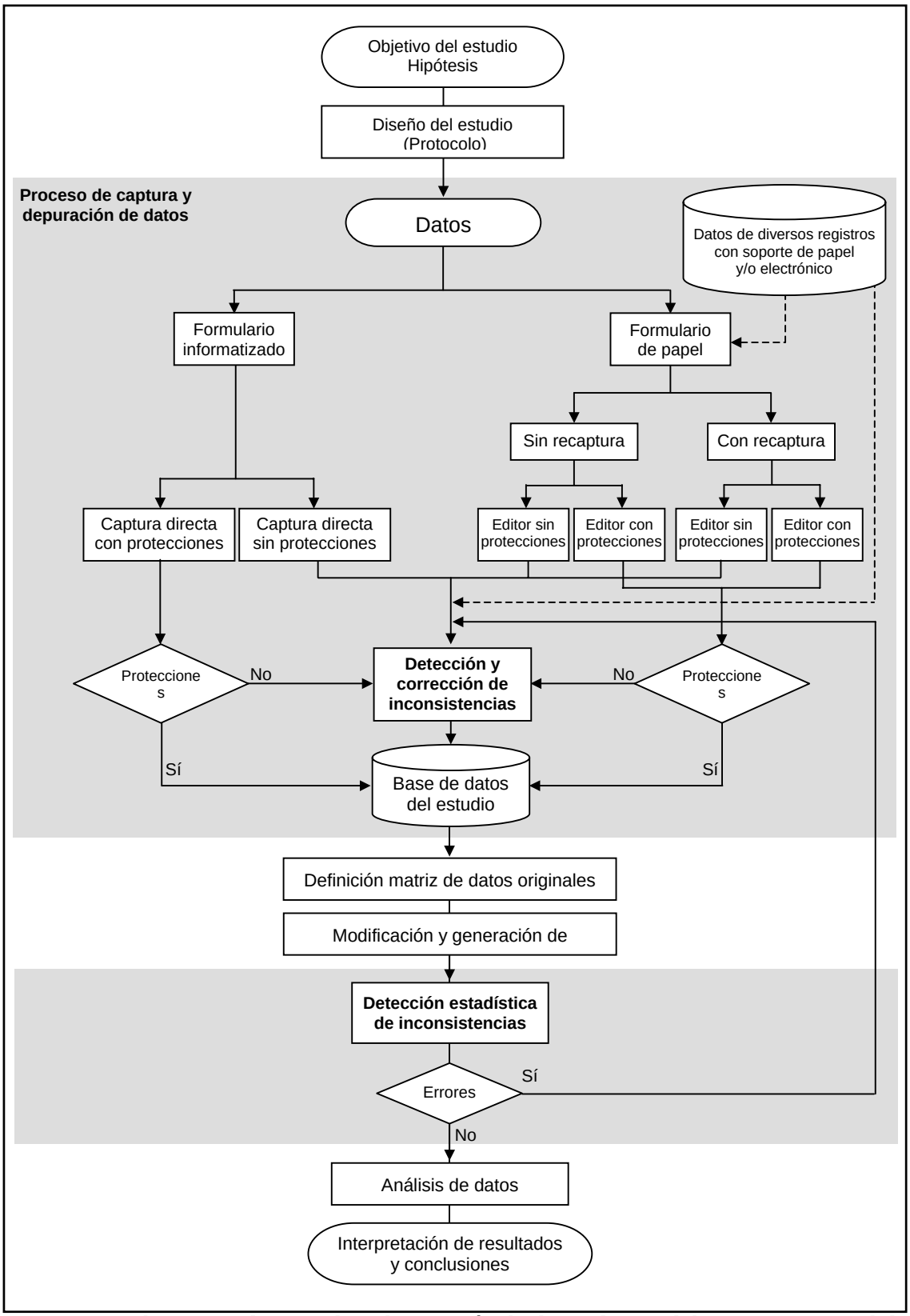


Figura 2-2: El proceso de entrada y depuración de datos dentro del contexto general de investigación científica (modificada de Granero, Doménech y Bonillo, 2001).

Es importante subrayar que cuando se utilizan estrategias de recaptura como la DE se elimina un tipo de error de diferente naturaleza al que se detecta tras efectuar el análisis de consistencia y detección de valores desconocidos que comporta la depuración propuesta en este trabajo.

Si se elimina la etapa de DE, el proceso de depuración detectará la totalidad de datos que se han olvidado grabar, pero sólo se identificarán una pequeña parte de los valores grabados erróneamente: aquellos que producen inconsistencias, por ejemplo errores de formato, valores fuera de rango o incompatibles con otros valores. Si los datos grabados son consistentes pero no coinciden con los registrados en el documento original, este error nunca será detectado por el proceso de depuración.

En síntesis, la depuración garantiza que el conjunto de datos que se someten a la etapa de definición y generación de variables contienen el menor número posible de valores desconocidos y no incluyen inconsistencias.

3
CONTROLES PROBABILÍSTICOS: ALGORITMOS PROPUESTOS

3.1 PRESENTACIÓN

Entre los distintos controles que han sido propuestos para detectar errores en los datos se distingue entre los controles lógicos y los probabilísticos o estadísticos. Los controles lógicos serían aquellos que detectan errores en los datos a partir de condiciones que no pueden darse. Los probabilísticos, por su parte, no garantizan que lo detectado sea un error, pero sí que hay una alta probabilidad, controlada o sabida por el usuario, de que lo sea. El sistema de depuración de este trabajo, que será expuesto en el capítulo 4, son en su gran mayoría lógicos. El motivo de ello es que es difícilmente justificable realizar propuestas a nivel probabilístico cuando la literatura no ha solucionado completamente todos los controles lógicos a realizar. Sin embargo, se hace imprescindible realizar una revisión sobre las técnicas probabilísticas de uso más habitual, ya que hay muchas situaciones en que los controles lógicos no pueden solucionar los problemas que el usuario tiene en sus datos. Cabe decir que muchos de los controles y técnicas que se expondrán en este capítulo se hallan en plena fase de desarrollo y optimización.

3.2 CONTROLES DE LA DEPURACIÓN

3.2.1 Identificadores

En los siguientes subapartados se revisan de forma sucinta los principales controles propuestos en la literatura para detectar casos que comparten identificadores pese a que no coincidan exactamente. Las técnicas que serán expuestas se utilizan para detectar duplicados y para fusionar registros de tablas distintas.

3.2.1.1 Detección de duplicados no exactos

3.2.1.1.1. Detección de duplicados a partir de identificadores semejantes: algoritmo de Smith-Waterman

El método habitualmente utilizado para detectar duplicados exactos consiste en ordenar la tabla de datos y chequear si los identificadores de los registros son idénticos (Bitton y DeWitt, 1983). Este método puede flexibilizarse para detectar registros que sin ser duplicados idénticos son suficientemente semejantes como para sospechar que corresponden a la misma entidad del mundo real (Melgratti y Yankelevich, 2000).

La detección de registros duplicados no exactos se basa en la presunción de que los identificadores del registro pueden contener errores tipográficos y/o pueden haber sido introducidos de manera no estandarizada. Los errores tipográficos aparecen en gran medida cuando se utilizan como identificadoras variables tales como el nombre del sujeto o su dirección. Estas variables de tipo cadena son susceptibles de ser abreviadas o

de padecer errores (o simplemente discrepancias), debido al uso de diferentes idiomas o a homonimias (Monge, 2000a y 2000b).

La Tabla 3-1 ilustra con un ejemplo esta situación. Los dos primeros registros corresponden al mismo sujeto, pero su nombre no coincide porque en el primer caso ha sido introducido el nombre antes del apellido, mientras que en el segundo registro ha sucedido lo contrario. Los registros tercero y cuarto también corresponden al mismo sujeto, pero en este caso no hay coincidencia exacta porqué en el registro tercero se ha registrado el nombre en castellano mientras que en el registro cuarto se ha hecho en catalán. Los registros quinto y sexto no son coincidentes, puesto que representan sujetos distintos.

El ejemplo anterior muestra la necesidad de contar con algoritmos que proporcionen medidas de las similitudes entre los identificadores de los registros. Clásicamente, se ha utilizado el algoritmo de Smith-Waterman (Smith y Waterman, 1981). Este algoritmo proporciona una medida del "coste" que tendría convertir una cadena de caracteres en otra, lo que equivale a una medida de su similitud. Por este motivo, el algoritmo también ha recibido el nombre de "algoritmo de alineamiento" o algoritmo de la "distancia editada" (Monge, 1998, 2000a y 2000b; Monge y Elkan, 2001; Galhardas et al., 2000a y 2000b; Hernández y Stolfo, 1998).

Los creadores de este algoritmo lo concibieron para hallar relaciones evolutivas entre diferentes proteínas y en series de ADN y ARN. Así, dos cadenas distintas pero muy semejantes de ADN hacen sospechar que las dos especies tienen un ancestro común (Practical Bioinformatics, 2000). El problema mayor que plantea esta situación consiste en que la medida de semejanza debe ser "relativamente" tolerante a los *gaps* (fragmento de aminoácidos no común a las dos cadenas originales insertado en cualquier lugar de la doble hélice del ADN). El algoritmo debe detectar que aunque un fragmento sea distinto en ambas cadenas, el resto sí es igual o muy semejante. Sin entrar en demasiados detalles, el algoritmo tiene tres parámetros: *m*, *s* y *c*. El parámetro *m* representa a los caracteres alfanuméricos que son coincidentes entre ambas cadenas, mientras que *s* y *c* representan la penalización por iniciar y por continuar un *gap*, respectivamente.

Tabla 3-1: Ejemplo de pares de registros y su estatus

Nombre	Dirección
Juan Manuel Camarena	*Vacío*
Camarena, JM	*Vacío*
Juan Manuel Camarera	Pl. Mayor 160
Joan Manel Acamarena	Mayor, Plaza 160 Esc. B
J.M. Camarera	Apartado de Correos 254
Paz Pau	Apartado de Correos 214

Una situación muy parecida a la que se da en estos estudios filogenéticos se produce cuando se observan semejanzas entre variables tipo cadena que identifican a los registros, como los nombres o las direcciones. En muchas ocasiones, las diferencias que se dan entre las cadenas identificadoras de varios registros que corresponden a un mismo sujeto son del tipo *gap*. Véase, por ejemplo, los registros tercero y cuarto (Tabla 3-1), que no son coincidentes pero sí son muy semejantes. Así, los nombres de "Juan

Manuel Camarera" y "Joan Manel Acamarena" son distintos por la diferencia de idioma y por un error tipográfico en el apellido. Un algoritmo que "alineara" las cadenas del apellido proporcionaría un valor de coincidencia bajo, ya que el *gap* se encuentra en la primera posición del apellido; a partir de este carácter la coincidencia exacta entre ambas cadenas es nula. Sin embargo, es en esta situación donde el algoritmo de Smith-Waterman muestra su potencia porque contempla lo que se ha llamado "no alineamiento". Es decir, detecta que exceptuando el primer carácter, el resto del apellido es el mismo.

Una de las mejoras implantadas en el algoritmo de Smith-Waterman es su uso transitivo (Gallardas et al. 2000a; Monge, 1997). En la propuesta original, las cadenas identificadoras de todos los casos deben ser comparados a pares, lo que hace crecer de forma exponencial el tiempo de procesamiento y hace inaplicable el algoritmo ante grandes volúmenes de datos. Para remediar esto se ha propuesto aceptar la transitividad del algoritmo, es decir, si $i_1=i_2$ y $i_2=i_3$ se acepta que $i_1=i_3$, lo que reduce mucho el número de comparaciones a realizar y hace asumible el coste computacional.

Una buena exposición del algoritmo de Smith-Waterman y de sus características se encuentra en Monge (1997 y 2000). Además, en Practical Bioinformatics (2000) se presenta un claro ejemplo que ayuda a implementar este algoritmo de manera sencilla.

3.2.1.1.2. Alternativas al algoritmo de Smith-Waterman

Se han formulado diferentes alternativas al algoritmo de Smith-Waterman para superar algunos de sus problemas conocidos, como es el coste computacional o su rigidez ante la detección de la transposición de cadenas. Entre ellas destacan los algoritmos *K-way Sorting Method* (Feekin y Chen, 2000) y *Basic Sorted-Neighborhood Method* (Hernández y Stolfo, 1995).

El *K-way Sorting Method* se basa en ordenar los datos repetidamente a través de las variables cadena candidatas a identificar los casos y comparar éstas entre registros. Si el porcentaje de coincidencias exactas supera un valor umbral predeterminado se considera que ambos registros representan la misma entidad del mundo real. Las ordenaciones se realizan repetidamente para evitar que la ordenación decidida inicialmente, y que puede ser arbitraria, incida en la decisión a tomar.

Implementar este procedimiento en el ejemplo mostrado en la Tabla 3-1 sería sencillo. Suponga que aceptamos un valor umbral de coincidencias exactas del 50%. Dicho de otro modo, se considerarían duplicados aquellos casos cuyos nombres o direcciones coincidieran exactamente. Veamos un poco más detalladamente como se realizaría esta comparación.

En primer lugar, se deberían ordenar los datos a través de la variable *Nombre* y realizar la comparación entre registros. Observe que tan sólo coincidirían los registros primero y tercero. En un segundo paso, se reordenarían los casos a través de la variable *Dirección* y se compararían de nuevo los registros. En este paso no se produciría ninguna coincidencia. Finalizado el proceso se debe tomar la decisión de si los pares de casos son o no duplicados. En el ejemplo mostrado sólo los registros primero y tercero serían considerados como tales puesto que alcanzan el valor umbral del 50% de coincidencia. Tal y como ocurre con el algoritmo de *Smith-Waterman* se acepta su uso transitivo, con lo que se reduce el número de comparaciones y, consecuentemente, el coste computacional.

3.2.1.1.3. Uso de la detección de duplicados no exactos en Psicología

El uso de las técnicas de detección de duplicados no exactos en Psicología es similar al realizado en otras disciplinas. Como se ha indicado, se trata de técnicas complejas que requieren un alto coste computacional, y, en muchas ocasiones, que el usuario realice la programación *ad hoc*, ya que no hay software en el mercado que maneje de manera genérica estos problemas. Puesto que tiene un alto coste, sólo es interesante implementar un sistema tan sofisticado si se trabaja con muchos casos. En estos casos, en los que se registra un gran número de informaciones por parte de distintos informantes y/o que pueden proceder de centros distintos, es interesante comprobar que no se han introducido accidentalmente duplicados.

Un argumento adicional favorable al uso de este tipo de algoritmos es el hecho de que posibilitan el aumento de la calidad de los datos al poder "fusionar" dos registros, que suponemos repetidos, en uno solo que contenga el máximo de información extraíble de los dos. La detección de un duplicado no tiene por qué significar la eliminación inmediata de uno de ellos, en especial cuando no hay criterios externos para suponer que uno es más válido que el otro. Lo correcto, en este caso, es obtener un registro final tan completo como sea posible, tomando los valores válidos de un registro si en las variables del otro están vacías, o guardando aquellos campos más largos (como en el caso de las direcciones) pues se puede suponer que son más completos.

3.2.1.2 Problemas de integración: "Merge/Purge Problem"

El problema de la integración de información aparece cuando en una tabla se pretenden añadir una serie de variables contenidas en otra u otras tablas que hasta ese momento no han estado relacionadas. El problema que surge en este momento es similar al visto en el caso de los duplicados: es posible que los valores usados como clave relacional no coincidan de manera exacta, aunque representen al mismo sujeto.

Se hace necesario, de nuevo, disponer de algoritmos que relacionen los registros de ambas tablas y, en caso de que representen al mismo sujeto, los unan, y, en caso de que no sea así, los desechen (Hernández y Stolfo, 1995). Este problema denominado genéricamente "merge/purge problem" aparece descrito en otros trabajos como *record linkage* (Fellegi, 1969) y *semantic integration problem* (Wang, 1989).

Hernández y Stolfo (1995) propusieron el algoritmo *Basic Sorted-Neighborhood Method*. Éste es un algoritmo muy similar al expuesto en el apartado anterior (*K-way Sort Method;* Feekin y Chen, 2000), aunque su formulación fue previa. De nuevo, se basa en extraer subcadenas "importantes" de las variables identificadoras, o candidatas a serlo, tanto de la tabla principal como de la tabla o tablas que se desean fusionar con la principal. De estas subcadenas importantes se eliminarían previamente las porciones que no aportan información porque son muy comunes. En el caso de las direcciones se deberían eliminar elementos como las palabras "calle", "plaza", "avenida", así como todas las abreviaciones de éstas que conociéramos. En el caso de los nombres, deberíamos primero intentar homogeneizar los datos cambiando las abreviaciones por sus nombres, como "Paco" por "Francisco" o "Pepe" por "José" o, simplemente, formando la subcadena importante sólo con las iniciales.

Tras ordenar ambas tablas por esta nueva variable se comprobaría su coincidencia exacta y en caso afirmativo se realizaría la fusión. El algoritmo se iteraría variando la composición de esta nueva cadena (tomando más o menos caracteres de las cadenas de

los nombres y de las direcciones, por ejemplo, o variando el orden de concatenación de la cadena del nombre para así recoger aquellos casos en los que el orden de captura de nombre y apellidos está transpuesto). En definitiva, la fusión se realizaría únicamente cuando un porcentaje considerado como "suficiente" de iteraciones aconsejaran hacerlo.

De este ejemplo se deduce una de las propiedades que ha sido argumentada como aspecto negativo de este algoritmo: debe ser implementado de manera muy específica. En otras palabras, es fundamental conocer profundamente el tipo de variables que identifican los registros y tener experiencia en la tipología de errores que pueden hallarse en ellas. Sólo así se pueden crear de manera precisa las subcadenas que serán utilizadas para comprobar si el registro es un duplicado o no.

Además de este procedimiento, se ha propuesto el uso del algoritmo de Smith-Waterman (Monge y Elkan, 2001), también visto en el apartado anterior. El modo de operar de éste es muy semejante al ya visto en el caso de los duplicados. Se comparan las variables cadenas identificadoras de los registros de las tablas a fusionar con las cadenas identificadoras de la tabla principal y sólo en el caso de que el valor proporcionado por el estadístico sea superior a un determinado umbral se realiza la fusión de variables.

3.2.1.2.1. Uso de la fusión de tablas con identificadores no exactos en Psicología

El alto coste computacional y la posibilidad de integrar en un solo registro a casos que no representan la misma entidad de la realidad, hacen desaconsejable utilizar algoritmos automáticos para fusionar tablas en investigaciones que manejen un volumen pequeño de datos. Ahora bien, existen varias situaciones en la que sería interesante recurrir a estas técnicas.

En registros de tipo epidemiológico puede ser interesante fusionar datos que provienen de distintas bases de datos y que no tienen por qué tener un identificador unívoco. Suponga un registro que monitoriza una población a la que, en un momento dado, surge la posibilidad de añadir datos procedentes de otras investigaciones. Si ambas investigaciones no han tenido la precaución de utilizar un identificador de tipo universal y unívoco (el DNI, por ejemplo), los registros sólo pueden ser eficientemente fusionados utilizando algoritmos del tipo expuesto. Además, hay ámbitos de investigación en los que, incluso, se carece de este tipo de identificador o es poco frecuente, como puede ser el caso de la pediatría. En estos ámbitos es habitual que cada estudio cree un identificador *ad hoc* para la investigación, dificultando luego su fusión con otros registros.

Otra situación semejante a la anterior es aquella en que diferentes operadores capturan datos de manera simultánea, careciendo, de nuevo, de un identificador universal y unívoco. Suponga que un hospital registra datos de sus pacientes desde distintas áreas sin un identificador como el descrito o, siendo éste susceptible de tener errores y no ser válido como campo de fusión. En esta situación, podría ser eficiente utilizar las variables cadena del nombre y la dirección por ejemplo, para maximizar la fusión.

En general, toda captura de datos realizada de manera distribuida (desde distintos terminales con diferentes operadores y/o en diferentes momentos temporales) es susceptible de necesitar de algoritmos de fusión de registros, a menos que la captura se

haya realizado accediendo directamente a la base de datos central, lo que es poco habitual.

3.2.2 Datos transversales. Variables cuantitativas: La Ley de Benford

3.2.2.1 Definición

Newcomb (1881), en su artículo pionero demuestra que "la ley de probabilidad de la ocurrencia de los dígitos es tal que las mantisas[1] de sus logaritmos son equiprobables".

Benford (1937) define la Ley de los Números Anómalos como aquella que produce que un conjunto de datos tomados de fuentes distintas ajusten a una distribución logarítmica.

Hill (1996a, 1996b, 1996c, 1997 y 1999) define La Ley de los Dígitos Importantes a partir de la observación constatada en muchas series de datos cuantitativos, arguyendo que la distribución de los primeros dígitos no es uniforme (como intuitivamente se podría pensar) sino que sigue una distribución logarítmica.

Nigrini (2000) define la Ley de Benford como aquella que predice la frecuencia esperada de aparición de los dígitos en series de números.

Aunque las aportaciones y leyes anteriores sean aparentemente tan dispares, todas son ciertas y todas versan sobre el mismo fenómeno, La Ley de Benford.

A modo de definición general, esta Ley, también llamada Ley de los Dígitos Importantes o Ley del Primer Dígito, se presenta como una distribución de tipo logarítmica utilizada para calcular las distribuciones esperadas de aparición de los dígitos en función de su importancia respecto al número que forman. Un dígito es tanto más importante en un número cuanto más a la izquierda del mismo se halla, prescindiendo de los 0 iniciales. Por ejemplo, los números 314, 3.14 y 0.00314 tienen 3 dígitos importantes, que son, en este orden, el 3, el 1 y el 4.

La Ley de Benford, ya en el trabajo de Newcomb, proporciona la frecuencia de aparición esperada del primer y del segundo dígito. Posteriormente, el propio Benford (1938) desarrolló las distribuciones del tercer dígito, y las distribuciones conjuntas de los dos primeros dígitos y de los tres primeros dígitos.

Un clásico y sencillo ejemplo será de utilidad para entender qué es la Ley de Benford y el por qué muchas variables registradas se ajustan a esta distribución (Nigrini, 2000, pág 9; y Hill, 1999).

Imaginar una población de 10000 habitantes. Suponer también que esta población crece un 10% al año. El valor "1" será el primer dígito del censo de esta población hasta que alcance los 20000 habitantes, cosa que tardará bastante tiempo en ocurrir visto su crecimiento. Una vez la población tenga 20000 habitantes, y manteniendo constante la tasa de crecimiento, el tiempo que tarde en tener 30000 habitantes será menor del que tardó en alcanzar los 20000, pero este período será a su vez mayor que el que tardará en lograr 40000 habitantes. Esta sucesión se reinicializará cuando la población alcance los 100000 habitantes, volviéndose a estancar en el valor "1" como primer dígito. Este

[1] La mantisa es la parte decimal de un logaritmo.

ejemplo muestra que el registro de habitantes de las poblaciones se ajusta a una distribución de Benford.

3.2.2.2 Antecedentes históricos

El primero en hacer notar que los primeros dígitos de los números no se distribuyen de manera equiprobable fue el astrónomo y matemático Simon Newcomb (1881). Éste, que ha pasado a la historia por trabajos tan distintos como sus teorías sobre los orígenes de los asteroides o por afirmar con rotundidad que ningún aeroplano podría volar (Tenn, 1987), observó que las primeras páginas de los libros de tablas de logaritmos estaban sistemáticamente más desgastadas que las últimas. Cabe tener presente que antes de la aparición de las calculadoras las tablas de logaritmos eran utilizadas para hacer productos, cocientes y raíces, facilitando así la tediosa operación manual (Nigrini, 2000). El mayor desgaste de las primeras páginas del libro de logaritmos consultado por Newcomb implicaba, necesariamente, que los números con mantisa 1 estaban más presentes en la realidad que los que tenían mantisa 2, y éstos más que los que tenían mantisa 3, etc. Aunque el trabajo de Newcomb no formula justificación alguna que explique este fenómeno ni se sustenta en formulaciones matemáticas sí que afirma que "la ley de probabilidad de la ocurrencia de números es tal que las mantisas de sus logaritmos son equiprobables" (traducción de Perera y Ayllón, 1999, pág. 339).

La observación de Newcomb tuvo una nula repercusión (Hill, 1998), probablemente por la dificultad de difundir los trabajos en esa época y porque Newcomb no menciona ninguna aplicación práctica de su descubrimiento, fruto de una ocasional curiosidad. Frank Benford, físico de General Electric, publica *The Law of anomalous numbers* 57 años más tarde (1938), y lo hace desconociendo el trabajo de Newcomb, aunque curiosamente a partir del mismo fenómeno: la observación del desgaste desigual de las páginas de los libros de logaritmos. A diferencia del trabajo de Newcomb, Benford no sólo formula la Ley que finalmente tomará su nombre, sino que se dedica a recoger una cantidad ingente (20229 observaciones) y variopinta de datos, que van desde el área y longitud de los ríos, hasta los pesos atómicos de los elementos de la tabla periódica, pasando por estadísticas de la liga americana de béisbol o cualquier número aparecido en las páginas de Reader's Digest (Benford, 1938; Hill, 1998; Nigrini 2000). Benford, probablemente con mucho tiempo libre debido a que durante la Depresión de los 30 trabajaba a media jornada en General Electric (Nigrini, 2000), demuestra que en todas estas series los dígitos que aparecen en los primeros lugares lo hacen con una probabilidad desigual, siendo más probable la aparición de dígitos pequeños que de dígitos grandes. En su trabajo, además, formula la probabilidad de aparición de los dígitos en primer lugar, en segundo y en tercero, así como las probabilidades conjuntas de los dos y de los tres primeros dígitos.

La repercusión del trabajo de Benford no es ni mucho menos inmediata y los artículos sobre la Ley de los Números Anómalos, como había pretendido denominarla su autor, se suceden con cuentagotas (Nigrini, 2000). Durante los años 40 los trabajos son muy escasos y en su mayoría críticos (Goudsmit y Furry, 1944; Furry y Hurwitz, 1945). En 1961, Pinkham publica una imprescindible aportación al trabajo de Benford: si hay una Ley que gobierna la distribución de los dígitos, ésta debe ser necesariamente escala invariante. Intuitivamente parece lógico pensar que la distribución de los dígitos en las series de datos utilizadas por Benford debería mantenerse aunque se cambiaran

las unidades de medida (por ejemplo, medir los ríos en metros y no en millas). Pinkham demuestra que en las series utilizadas por Benford la multiplicación (o división) de los datos por una constante diferente de 0 sigue manteniendo inalterable la distribución original de dígitos. Sin embargo, es Raimi (1969a, 1969b) quien demuestra finalmente la independencia de la unidad de medida de la Ley de Benford. Raimi aporta no sólo el fundamento matemático de esta independencia de la escala, sino que además, en un intento divulgativo, añade explicaciones intuitivas de la invariabilidad.

En los años 70 se inician aproximaciones pragmáticas al uso de la Ley de Benford. Así, Hamming (1970) cree que se puede utilizar para detectar errores de redondeo en el funcionamiento de las computadoras de la época, y corregir así su diseño. Varian (1972) señala que si los datos originales de cualquier serie siguen una Ley de Benford, las predicciones que cualquier modelo estadístico haga sobre el mismo fenómeno también deberían seguir la Ley. En caso contrario, el modelo debería ser revisado.

En los años 80, aparecen publicaciones que serán importantes para el uso de la Ley de Benford. Por un lado, Carslaw (1988) publica un artículo que resultará fundamental: sugiere que siguiendo la misma lógica de detección de errores de redondeo en cálculos con ordenadores, podrían detectarse "redondeos" malintencionados en las cuentas empresariales. Así, lleva a cabo un estudio en diversas empresas neozelandesas que demuestra que la frecuencia de aparición del 0 como segundo dígito es mayor de la esperada, mientras que la del 9 es menor. Llega entonces a la conclusión que los mandos de estas empresas han redondeado sistemáticamente al alza los ingresos para así obtener mayores incentivos. Por otra parte, Hill (1988) lleva a cabo un experimento con sus alumnos en el que demuestra que cuando alguien es interpelado a inventar "números al azar", éstos no cumplen las propiedades de la Ley de Benford. Así, como segundo dígito, los alumnos de Hill utilizaban en su mayoría dígitos próximos al 5, como el 3 y el 7 (con una frecuencia del 55% y del 61%, respectivamente).

En la década de los 90 se producen avances significativos en la investigación y en el uso de la Ley de Benford. Hill (1996) demuestra algo ya sugerido por Boyle (1994): la Ley de Benford o distribución logarítmica de los primeros dígitos, "es la distribución *de todas las distribuciones*" (cursiva en el original de Nigrini, 2000). Esto es, que si tomamos una serie de distribuciones seleccionadas al azar de manera insesgada, y de estas distribuciones extraemos valores, los primeros dígitos del *conjunto* de valores convergen a una distribución logarítmica. En cuanto a su uso, primero Christian y Gupta (1993) pero especialmente Nigrini (1994, 1996 y 2000) utilizan la Ley de Benford para detectar fraudes empresariales en auditorías. Nigrini observa los últimos dígitos que aparecen como ingresos y como gastos en las cuentas anuales presentadas por algunas grandes empresas norteamericanas. Mientras los gastos se ajustan de manera perfecta a la Ley de Benford, en el apartado de ingresos se observa una aparición mayor de dígitos "pequeños" de la esperada por la Ley de Benford, sugiriendo que los datos de este apartado habían sido redondeados a la baja. Los trabajos de Nigrini y su aplicación en el Ministerio de Hacienda Holandés (Nigrini, 2000) hacen que la Ley de Benford salte a la prensa y aparecen numerosos artículos en revistas económicas como Expansión (Conthe, 2001) y en periódicos de gran difusión como *The New York Times* (Browne, 1998), o *Business Week* (Bernstein, 1998) o *The Wall Street Journal* (Berton, 1995). Incluso revistas generalistas, como el *USA Today* (Maney, 2000) se

hacen eco de esta Ley y de sus aplicaciones en la detección de fraude empresarial y de falseamiento de datos en general.

En la actualidad, Nigrini imparte cursos de sus técnicas a empresas auditoras, y comercializa el software DATAS *(Digital Analysis Tests and Statistics)*, especializado en la aplicación de la Ley de Benford a datos contables.

3.2.2.3 Características y derivaciones matemáticas

El logaritmo decimal (logaritmo en base 10) de un número *x* se define como el valor *y* del exponente al que debe elevarse la base 10 para obtener *x*:

$$\log_{10} x = y \quad \Leftrightarrow \quad x = 10^{y}$$

La parte entera del valor de un logaritmo se denomina *característica* y la parte decimal positiva se denomina *mantisa*. La *característica* es un número entero que representa la "magnitud" de *x* (por ejemplo, cientos, miles, etc.) mientras que la *mantisa* representa las "cifras significativas" de *x*.

Vamos a explicar con un ejemplo esta importante propiedad de la *mantisa*. Puesto que cualquier número decimal *x* puede ser expresado como producto de un número real que pertenezca al intervalo [1 ÷ 10) por una potencia de 10, por las propiedades del logaritmo se verifica que:

$$x = r \times 10^{c} \quad \Leftrightarrow \quad \log_{10} x = c + \log_{10} r$$

Suponga, por ejemplo, los números 314, 31.4, 3.14 y 0.314, todos ellos formados por las cifras significativas "314". La Tabla 3-2, que ilustra la descomposición anterior, permite comprobar como la *mantisa* del logaritmo decimal de cualquier número formado por las cifras significativas "314" es constante y vale 0.4969:

Tabla 3-2: Ejemplo de cálculo de la característica y la mantisa de un número real

x	=	$r \times 10^{c}$	$\log_{10} x$	=	$c + \log_{10} r$
314	=	3.14×10^{2}	$\log_{10} 314$	=	$2 + 0.4969...$
31.4	=	3.14×10^{1}	$\log_{10} 31.4$	=	$1 + 0.4969...$
3.14	=	3.14×10^{0}	$\log_{10} 3.14$	=	$0 + 0.4969...$
0.314	=	3.14×10^{-1}	$\log_{10} 0.314$	=	$-1 + 0.4969...$
					↑ Característica ↑ Mantisa

La ley de probabilidad de ocurrencia de los dígitos, enunciada por Newcomb (1881), indica que las mantisas de sus logaritmos son equiprobables.

La Ley de Benford a modo general puede ser formulada de la siguiente manera (Hill, 1995, 1996, 1997; Nigrini, 2000):

Ecuación 3-1

$$\text{Prob}(D_1=d_1, \ldots, D_k=d_k) = \log_{10}\left[1+\left(\sum_{i=1}^{k} d_i \times 10^{k-i}\right)^{-1}\right]$$

siendo *k* valores enteros positivos, todo $d_1 \in \{1, 2, ..., 9\}$, y todo $d_k \in \{0, 1, 2, ..., 9\}$ si $i > 1$

Para hallar probabilidad de aparición del primer dígito a partir de la Ecuación 3-1 no hay más que sustituir i=1, y se obtiene:

Ecuación 3-2 $$P(d_1) = \log_{10}\left(1+\frac{1}{d}\right) \qquad d_1 \in \{1, 2, ..., 9\}$$

siendo *d* el dígito para el cual se desea calcular su probabilidad de aparición. Para el segundo dígito obtendríamos:

Ecuación 3-3 $$P(d_2) = \sum_{k=1}^{9} \log_{10}\left(1+\frac{1}{d_1 \times d_2}\right) \qquad d_2 \in \{0, 1, 2, ..., 9\}$$

En la Tabla 3-3 podemos ver las frecuencias de aparición esperadas para el primer y segundo dígito, aplicando la Ecuación 3-2 y la Ecuación 3-3, respectivamente.

Tabla 3-3: Frecuencias esperadas en función del orden

D	0	1	2	3	4	5	6	7	8	9
$P(d_1)$		0.3010	0.1761	0.1249	0.0969	0.0792	0.0669	0.0580	0.0512	0.0458
$P(d_2)$	0.1197	0.1139	0.1088	0.1043	0.1003	0.0967	0.0934	0.0904	0.0876	0.0850

En la tabla anterior se observa que, por ejemplo, la frecuencia de aparición esperada para el dígito 3 en la primera posición es de un 12.49% mientras que la probabilidad de que aparezca en segundo lugar es de 10.43%.

Como ya se ha mostrado con anterioridad, la probabilidad de aparición de los dígitos distintos del primero es dependiente de los anteriores. Así, la probabilidad de los dos primeros dígitos viene dada por:

Ecuación 3-4 $$P(d_1d_2) = \log\left(1+\frac{1}{d_1 d_2}\right) \qquad d_1d_2 \in \{10, 11, ..., 99\}$$

y para los tres primeros dígitos:

Ecuación 3-5 $$P(d_1d_2d_3) = \log\left(1+\frac{1}{d_1 d_2 d_3}\right) \qquad d_1d_2d_3 \in \{100, 101, ..., 999\}$$

Aplicando las fórmulas anteriores, la probabilidad de que los dos primeros dígitos sean un 33 es P(33) = log (1+1/33) = 0.01295 y de que los tres primeros sean 333 es P(333) = log (1+1/333) = 0.0013.

3.2.2.4 Supuestos de una distribución de tipo Benford

Como señalan Nigrini (2000, pág. 24) y Hill (1996, 1998, 1999) el requisito fundamental que debe cumplir una variable cuantitativa para ajustarse a una distribución de Benford es que siga una secuencia geométrica. Aún así, y para mejorar el ajuste a la distribución, toda variable que mida un fenómeno cuyo crecimiento se ajuste a una sucesión de tipo geométrico no debe tener ni un mínimo ni un máximo

teórico. La razón es que los dígitos que componen estos mínimos y máximos aparecen con una frecuencia mucho mayor de la esperada por la distribución. Supongamos que en un experimento psicológico que recoge tiempos de reacción se limita el tiempo de respuesta (supongamos que a 120 segundos) y en caso de que el sujeto tarde más tiempo en responder, se le asigna este valor como máximo teórico. Para ver si estos datos ajustan de manera adecuada a una distribución de tipo Benford deberíamos eliminar las respuestas correspondientes al tiempo máximo de respuesta, ya que sino observaríamos frecuencias de aparición del dígito 1 (en primer lugar) y del 2 (en segundo lugar) superiores a lo esperado.

De igual modo, si se examina el ajuste de una variable que recoge gastos empresariales y el mínimo valor que se registra, obviando el resto por negligibles, es de 55€, se observaría que estos dígitos aparecen en mayor medida de lo esperado.

Otro requisito importante de la Ley de Benford es la cantidad de dígitos importantes que tiene la variable cuyo ajuste vamos a examinar. Así, diversos trabajos han mostrado que, idealmente, deben haberse registrado 4 o más dígitos (Nigrini, 2000, pág, 25). Aún así, los trabajos de Benford y otros posteriores muestran que con tres dígitos importantes se pueden obtener excelentes ajustes. Estrictamente hablando, cualquier rango de valores sería susceptible de ser ajustado a una distribución de Benford pero las diferencias entre las proporciones esperadas entre dígitos son mayores, y por tanto más fáciles de detectar, cuantos más dígitos tiene registrados la variable examinada.

3.2.2.5 Base invariante

La idea de que una variable debe seguir la distribución de Benford independientemente de su escala de medida aparece ya sugerida en el trabajo pionero de Benford (1938) y se da como establecida en trabajos posteriores (Flehinger, 1966 y Cohen, 1976). Intuitivamente, si se toman como ejemplo las mediciones originales de Benford, parece lógico pensar que si la longitud de los ríos norteamericanos ajusta de manera satisfactoria a la distribución esperada, esto debería ocurrir de igual manera si las mediciones se realizan en metros o en yardas.

No es el propósito de este trabajo profundizar de manera excesiva en la fundamentación puramente formal de la Ley de Benford; para obtener una excelente exposición formal de la invariabilidad de la Ley de Benford basta acudir a los numerosos trabajos de Hill (1995, 1996, y 1997). Sin embargo, y a modo intuitivo, es sencillo entender por qué la Ley de Benford es escala invariante. Si tomamos la ley original que parte de la equiprobabilidad de las mantisas de los logaritmos, es sencillo ver que transformar una variable original mediante una multiplicación o división es equivalente a cambiar la base de su logaritmo. Si se acepta que un cambio de escala representa siempre una multiplicación o división de las medidas originales y se asume que la equiprobabilidad de las mantisas no depende de la base del logaritmo, es sencillo aceptar la independencia de la escala de medida.

3.2.2.6 Distribución de dígitos dependiente

Como muestra la Ecuación 3-1, la distribución del dígito segundo y sucesivos es dependiente de los anteriores. Podemos ver en la Tabla 3-3 que la probabilidad de que el segundo dígito sea un 2 es 0.1088, pero de que el segundo dígito sea 2 dado que el

primero sea 1 y desarrollando la Ecuación 3-4 es $\log_{10}(1+1/12) = 0.0348$. Sería incorrecto multiplicar las probabilidades $Pd_1(1) = 0.301 \times Pd_2(2) = 0.1088$ como si éstas fueran independientes.

Por ello, para analizar la bondad de ajuste de una variable a la distribución teórica de Benford partiendo de los dos primeros dígitos, no se deben multiplicar las probabilidades del primer y segundo dígito sino utilizar la Ecuación 3-4.

Como resulta intuitivo y además se muestra en la Tabla 3-3 la frecuencia de aparición tiende a uniformarse al aumentar el número de dígitos considerados.

3.2.2.7 Pseudo-teorema central del límite de la Ley de Benford

Ya en el propio trabajo de Benford se observó que aunque algunas de las variables utilizadas para formular su Ley no seguían la distribución esperada, sí lo hacía la unión de todas ellas (Perera y Burguillo, 1999). Como afirma Hill (1996) si se escoge al azar una muestra de diversas distribuciones, y de éstas se toman muestras al azar, la muestra resultante converge a la Ley de Benford. Esta evidencia ha hecho que se hable de Ley de Benford como la distribución de distribuciones, pues emula en su comportamiento al teorema central de límite.

3.2.2.8 Pruebas de bondad de ajuste de una distribución observada a la Ley de Benford

Para examinar la bondad de ajuste de los datos observados a la distribución teórica de Benford se han adaptado los tests clásicos Z, χ^2, la prueba de Kolmogorov-Smirnoff y la desviación media absoluta (Nigrini, 2000, pág. 73).

En el caso del estadístico Z, la prueba de bondad de ajuste recoge las desviaciones de los datos observados vs. la Ley de Benford para cada uno de los dígitos. De este modo, permite comprobar si la frecuencia de aparición del dígito 1, 2, ..., 9, en una posición dada, es mayor de la esperada. La fórmula para calcular este estadístico Z proporcionada por Nigrini es la habitual:

Ecuación 3-6

$$Z = \frac{|p_o - p_e| - \frac{1}{2n}}{\sqrt{p_e \times \frac{1-p_e}{n}}}$$

donde p_e denota la proporción esperada, p_o la observada y n el número de observaciones. El término 1/2n es una corrección de continuidad que únicamente se utiliza cuando su valor es menor que el del primer término del numerador. Las proporciones esperadas y observadas deben obtenerse a partir de la prueba de ajuste que se desee realizar, sea ésta la del primer dígito, la del segundo o las de los dos o tres primeros.

En la actualidad, y fruto de estudios de simulación, se considera que la corrección de continuidad es excesivamente conservadora y la mayoría de trabajos desaconsejan que se aplique para muestras grandes [np_e y $n(1-p_e) \geq 5$] (Haviland, 1990). Por ello, es preferible no realizar esta corrección de continuidad y aplicar la siguiente ecuación en lugar de la Ecuación 3-6:

Ecuación 3-7

$$Z = \frac{\left|p_o - p_e\right|}{\sqrt{p_e \times \frac{1 - p_e}{n}}}$$

En el caso del estadístico χ^2, Nigrini propone su uso para calcular la bondad de ajuste de todos los dígitos respecto a lo esperado por la Ley de Benford. Así, las desviaciones de todos los dígitos se sumarían y el valor de este sumatorio es comparado con el valor crítico de $\chi^2_{(0.05,gl)}$. La fórmula para calcular este estadístico es también la habitual:

Ecuación 3-8

$$\chi^2 = \sum_{i=0}^{k} \frac{(O - E)^2}{E}$$

siendo *k* los dígitos examinados, *i* la posición del mismo, y O y E representan, respectivamente, las frecuencias observadas y esperadas de aparición de los dígitos.

Aunque Nigrini propone utilizar una u otra prueba de conformidad en función de si se examina la conformidad respecto a la distribución teórica de un dígito o del conjunto de ellos, es conocido que el estadístico Z es un caso particular del χ^2 para 1 grado de libertad.

Independientemente de que se utilice una u otra prueba de bondad de ajuste, sí es recomendable realizar la prueba de conformidad dígito a dígito y del conjunto. La primera permitiría detectar sobreapariciones de un dígito concreto mientras que la segunda localizaría patrones desajustados. Obsérvese que en absoluto son pruebas excluyentes sino complementarias; hallar desajuste en los datos respecto los valores esperados nos haría sospechar del conjunto del registro, se haya utilizado un test u otro.

Además de estos dos test, Nigrini ha propuesto también el uso del test de Kolmogorov-Smirnoff. Éste no es más que una prueba de bondad de ajuste de la densidad de probabilidad teórica de aparición de dígitos menores de uno dado respecto a lo observado. Esta prueba escoge la discrepancia mayor entre lo observado y lo esperado y a continuación la compara con los valores críticos dados por la distribución de Kolmogorov-Smirnoff.

Tabla 3-4: Valores de corte de la Desviación Media Absoluta para valorar el ajuste a la distribución de Benford

Valores de MAD	Interpretación
0.0000 – 0.0006	Ajuste perfecto
0.0006 – 0.0012	Ajuste aceptable
0.0012 – 0.0018	Ajuste marginal
mayor de 0.0018	Desajuste

Para finalizar, y como solución al conocido problema de que los test anteriores están fuertemente determinados por el tamaño de la muestra, Nigrini propone usar el test de la desviación media absoluta, que no está afectado por la *n*. Esta prueba promedia los valores absolutos de las discrepancias entre las proporciones esperadas por la distribución y las observadas. El valor obtenido puede interpretarse como la proporción máxima de desviación. El propio Nigrini sugiere usar los valores de corte dados en la Tabla 3-4 para interpretar el grado de desajuste.

3.2.2.9 Aplicaciones a la depuración de datos

Nigrini (1994, 1996 y 2000) ha aplicado la Ley de Benford a la auditoría empresarial, detectando con ello fraudes en numerosas ocasiones. La detección de un fraude contable no es más que sospechar que una o varias variables cuantitativas que registran gastos o ingresos han sido manipuladas y sus valores no se corresponden con los datos reales. Esta idea de detectar manipulaciones en series de datos puede ser fácilmente trasladada a la investigación en Psicología y, en general, a cualquier ámbito de las Ciencias de la Salud.

En el caso de la psicología, una aplicación inmediata consiste en detectar si las variables cuantitativas registradas en investigaciones ajustan a la Ley de Benford. En el caso de que no sea así, y siempre y cuando teóricamente debieran hacerlo puesto que siguen una progresión geométrica y tienen suficientes dígitos registrados, sería legítimo dudar de la validez de estas variables. Ante la aparición de la sospecha, los responsables de auditar investigaciones podrían solicitar información adicional que permita garantizar que la falta de ajuste es casual y no malintencionada o debida a errores técnicos. Un ejemplo de este tipo de variable seria una puntuación no tipificada de un test, siempre y cuando su rango de variación fuera lo suficientemente amplio.

Una cuestión fundamental antes de realizar las pruebas de bondad de ajuste es escoger con detenimiento el sujeto objeto de estudio. Éste puede ser un operador, la institución responsable de la captura de los datos, o un subconjunto de sujetos con alguna característica en común. La elección de un sujeto de estudio u otro depende de que pensemos que éste es el responsable de haber introducido, bien sea por error bien por mala fe, datos que no se corresponden con la realidad.

Si, por ejemplo, se dispone de una matriz de datos con múltiples variables introducidas cada una de ellas por distintos operadores, podríamos examinar estas variables por separado. Obtendríamos entonces una prueba de la validez del operador. Si, por el contrario, un mismo operador al que pretendemos auditar ha introducido varias variables podríamos ganar potencia en la prueba de bondad de ajuste añadiendo estas variables y examinando de manera conjunta su ajuste. Si los operadores no

introducen variables sino casos podríamos examinar el ajuste del registro y no del campo.

En definitiva, es imprescindible clarificar al inicio quien es el sujeto auditado, examinar si los datos pueden ser agregados y realizar las pruebas de ajuste tomando el conjunto de datos de los cuales el sujeto auditado es responsable.

3.2.3 Comprobación a través de los ratios entre variables cuantitativas

La recogida de gran cantidad de datos cuantitativos es especialmente frecuente en el ámbito de los estudios económicos. Desde mediados de los 80, se han propuesto técnicas destinadas a detectar errores en este tipo de variables que pudieran producir grandes sesgos en los resultados (Bureau, Michaud y Sistla, 1986; Hidiroglou y Bertholot, 1986).

La mayor parte de los métodos de detección de error propuestos se basan en calcular, para todos los casos disponibles, la razón entre dos variables cuantitativas correlacionadas. A partir de estas razones se calculan medidas de posición, tales como la mediana y los cuartiles, y se comprueban aquellos valores cuyas razones se hallan fuera de unos límites calculados para esta distribución. La utilización de esta metodología tiene algunas ventajas evidentes, ya que: 1) no obliga al usuario a especificar a priori los límites de comprobación; y 2) se utilizan los datos de la propia muestra para la comprobación. Esto último es especialmente importante, ya que las razones entre dos variables pueden ser muy distintas en función de la composición de la muestra que se ha utilizado para su cálculo; prefijar los límites produciría un gran número de falsos positivos/negativos.

Expondremos brevemente el método propuesto por Hidiroglou y Bertholot (1986), ya que la mayoría de metodologías posteriores no son más que mejoras sucesivas de esta técnica original. Para una revisión más exhaustiva se recomienda acudir al trabajo de Villan y Bravo (1990). Consideremos x_j y x_i, teniendo en cuenta que x_i puede ser la misma variable x medida en otro momento temporal o bien una variable distinta que correlacione con ésta. Así, debemos calcular,

para todos los casos $r_i = x_i/x_j$

y para toda la muestra $max_{ij}= max(x_i/x_j)$

A partir de la desigualdad de Chebyshev podemos calcular el porcentaje de casos que deberían estar fuera de este intervalo:

Ecuación 3-9 $$\bar{r} - k s_r \div \bar{r} + k s_r$$

siendo k una constante que proponen asignar a valores cercanos a 40, y s_r la desviación estándar. Sin embargo, Hidiroglou y Bertholot (1986) consideran estos límites demasiado sensibles a la presencia de *outliers*, que se produce al trabajar con una medida afectada por ellos como es la media aritmética. Para superar este problema, proponen utilizar el siguiente método de cálculo del intervalo:

Ecuación 3-10 $$r_m - k r_{q1} \div r_m + k r_{q3}$$

siendo r_m la mediana de los ratios r_i, r_{q1} y r_{q3} el primer y tercer cuartil, respectivamente (Villan y Bravo, 1990).

Los propios Hidiroglou y Bertholot reconocen dos problemas para este procedimiento. Por un lado, adolece de una alta dependencia de la simetría de la

distribución. Por otro, los límites dependen del tamaño de la muestra, siendo los límites mayores para muestras grandes que para muestras pequeñas.

Para superar el problema de la simetría proponen transformar la variable r de la siguiente forma:

Ecuación 3-11 $$s_i = \frac{1 - r_m}{r_i} \div \frac{r_m}{r_i - 1}$$

Para superar el problema del tamaño de la muestra, los autores proponen un cambio de escala que pondere las unidades mayores. La transformación se realizaría del siguiente modo:

Ecuación 3-12 $$e_i = s_i \times \max_{ij}^{u}$$

donde u es una constante entre 0 y 1.

Tras estas transformaciones, los límites se obtendrían mediante la siguiente ecuación:

Ecuación 3-13 Límite inferior: $e_m - d2$

Límite superior: $d3 - e_m$

siendo e_m la mediana de los valores e_i calculados en la *Ecuación 3-12* y calculándose d2 y d3 del siguiente modo:

Ecuación 3-14 $$d2 = k \times \max(e_m - e_{q1}, A \times e_m)$$

$$d3 = k \times \max(e_{q3} - e_m, A \times e_m)$$

En estas transformaciones hay tres constantes que se deben estimar y que constituyen los parámetros del sistema. Todaro (1999) realiza un estudio donde propone asignar a estas constantes los valores u=0.4 y k=41, manteniendo A= 0.05, como hacen Hidiroglou y Bertholot (1986) en su artículo original.

Esta metodología, y pequeñas variantes que suelen cambiar los valores de los parámetros u, k y A, son ampliamente utilizadas en estudios que recogen datos económicos (Bienias et al., 1997; Sigman, 2001; Thompson, 1999;; van der Pol et al. 1997; Winkler, 1999). Así, los datos recogidos por el *Bureau of the Census Economic Programs* (organismo asimilable a nuestros institutos de estudios demográficos y económicos) son sometidos a controles semejantes a los expuestos (Thompson y Sigman, 1998). Entre las información que recoge este organismo destacan, por su importancia y repercusión en los ámbitos financieros, el *Annual Survey of Manufactures*.

También el US Census Bureau (asimilable al Instituto Nacional de Estadística de nuestro país) utiliza esta metodología de detección de errores en los datos que maneja (Sigman, 2001). Entre éstos destacan los censos poblacionales norteamericanos, así como índices de productos manufacturados, de construcción y de servicios comerciales, entre otros.

Finalmente, otro organismo que también utiliza derivaciones de la metodología de Hidiroglou y Bertholot es el Nacional Agricultural Statistics Service (1999), que recoge datos sobre la producción agrícola Norteamérica.

3.2.4 Depuración de datos en las pruebas de ejecución

Según Martínez Arias (1996, pág. 32) y Cronbach (1970) los tests pueden ser clasificados, entre otros muchos criterios, por el planteamiento del problema. Bajo este criterio, los tests se pueden etiquetar como de ejecución máxima o como de ejecución típica. En los tests de ejecución máxima, el respondiente ha de resolver los ítems poniendo a prueba sus conocimientos o aptitudes. Los ítems tienen, por tanto, respuesta o respuestas correctas, y los sujetos pueden ordenarse en función de su rendimiento. En los tests de ejecución típica, el respondiente contesta en función de sus hábitos de comportamiento o creencias. Obviamente, los ítems de estos instrumentos carecen de respuesta correcta.

En ambos tipos de pruebas se ha desarrollado investigación para detectar patrones de respuesta de cuya fiabilidad se duda. En los dos subapartados siguientes se revisan algunos de estos aspectos brevemente.

3.2.4.1 Pruebas de ejecución máxima

Las pruebas de ejecución máxima pretenden medir una determinada habilidad o conocimiento del sujeto. Se componen de una serie de ítems que suelen servir para calcular una puntuación total que, además de ser válida y fiable, permita ordenar a los respondientes. La investigación en esta área se ha dedicado al estudio de patrones, también llamados vectores, de respuestas que no son adecuados, o cuando menos normativos, respecto a la puntuación total alcanzada por el sujeto (Levine y Drasgow, 1983).

Los motivos por los cuales se generan estos patrones anómalos, llamados Patrones Aberrantes de Respuesta (PAR), pueden ser múltiples. Se pueden deber a que el sujeto no está familiarizado con el formato de respuesta del instrumento (Wright y Stone; 1979). Así, un patrón puede ser anómalo porque una persona, aún teniendo profundos conocimientos sobre una materia, puede no estar en absoluto habituada a responder a ítems de elección múltiple.

Además de la poca familiarización con las pruebas, otro motivo contemplado en la literatura que genera patrones aberrantes de respuesta es el adoptar determinadas estrategias para contestar el instrumento o el tener un “estilo” propio inadecuado (Ellis y van den Wollenberg, 1993; Holland, 1990). Entre estas estrategias o estilos inadecuados, y a modo de ejemplo, los autores señalan el “comportamiento durmiente”, y el “falseamiento”. Como “comportamiento durmiente” se entiende aquel sujeto que contesta de manera correcta los ítems más difíciles, pero falla los más fáciles, debido que le cuesta aclimatarse a la prueba. El “falseamiento” se puede producir porque, por ejemplo, un sujeto copia de otro parcialmente sus respuestas, acertando ítems difíciles que no le correspondería acertar, dada su puntuación total.

En este trabajo no se pretende exponer las tipologías de PAR que la literatura describe ni investigar las causas que los producen. Lo que sí es destacable es que la detección de un PAR puede ser útil para revisar en profundidad el caso, en la medida de lo posible, y decidir si los datos son fiables o no.

El índice más utilizado para detectar PAR es el *Modified Caution Index* (en adelante, MCI). Propuesto por Harnisch y Linn (1981), es una variación del conocido como *Caution Index*, propuesto a su vez por Sato (1975). Su fórmula general es:

Ecuación 3-15

$$G_i = \frac{\sum_{g=1}^{r} w_g - \sum_{g=1}^{k} x_g w_g}{\sum_{g=1}^{k} w_g - \sum_{g=k-r+1}^{k} w_g}$$

donde w_g es la dificultad teórica de los ítems y $x_g w_g$ representa la dificultad del patrón totalmente erróneo. El MCI fluctúa entre 0 y 1, y cuanto más cercanos son sus valores al 1 más aberrante es el patrón analizado. Se considera como valor umbral a partir del cual debe revisarse el patrón el 0.3 .

Uno de los problemas más habituales de los algoritmos de detección de PAR es que es difícil encontrar programas informáticos que los implementen.

3.2.4.2 Pruebas de ejecución típica

La investigación desarrollada en las pruebas de ejecución típica ha sido mucho menor que la que se ha realizado para las pruebas de ejecución máxima. El motivo de esto es que el hecho de no contar con una respuesta correcta hace que sea mucho más compleja la detección de los PAR.

El principal índice utilizado en la detección de PAR en pruebas de ejecución típica ha sido el Z_L *Scalability Index* (Drasgow, Levine y McLaughlin, 1985). Representa la verosimilitud estandarizada del patrón de ítems de un individuo dados unos parámetros estimados de TRI (Teoría de Respuesta al Ítem). Su fórmula viene dada por:

Ecuación 3-16

$$Z_{L|\Theta} = \frac{\sum V_{L|\Theta} - \sum E_{L|\Theta}}{\sqrt{\sum V_{L|\Theta}}}$$

donde $V_{L|\Theta}$ es la variancia y $E_{L|\Theta}$ es el valor esperado de $L|\Theta$.

Este índice se interpreta del siguiente modo: valores negativos de Z_L indican poco ajuste del patrón al modelo, mientras que valores altos indican lo contrario.

El índice Z_L ha sido aplicado a datos reales obteniendo resultados desiguales. Aplicado a escalas de personalidad, han mostrado que su capacidad de detección de PAR es, cuando menos, discutible (Reise y Flannery, 1996). Por poner sólo algunos ejemplos, Tellegen y Atkinson (1974) detectaron sólo un 46% de los patrones aberrantes generados por un grupo de sujetos que respondieron al azar el *Tellengen's Absorption Scale*.

3.2.5 Depuración de datos a través de técnicas de *Data Mining*

El término *data mining* se considera una etapa dentro de un proceso mayor llamado extracción de conocimiento en bases de datos (*Knowledge Discovery in Databases* o KDD). Una definición ortodoxa de *data mining* la proporcionan Piatetsky-

Shapiro y Frawley (1991), al indicar que es *"Un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos"*. En definitiva, podríamos decir que el *data mining* es una metodología, que recoge técnicas de la estadística tradicional, la inteligencia artificial y los procedimientos gráficos avanzados. Su objetivo es extraer información de los datos. El auge de estas técnicas en los últimos años se explica por el salto cualitativo en la calidad del Hardware y del Software que permite tratar cantidades masivas de datos de manera eficiente (Molina, 2000).

El estudio de la calidad de los datos no ha permanecido indiferente ante el auge de las técnicas de *data mining*. Puesto que esta metodología no renuncia a las técnicas estadísticas más clásicas, algunas de las propuestas en el seno del *data mining* son un tanto simples, ya que se limitan a detectar y verificar el valor de casos que se hallan fuera del intervalo de confianza de una regresión lineal o casos identificados como extremos o alejados en un diagrama de caja *(boxplot).*

Otra aplicación más interesante de las técnicas de *data mining* es el uso de redes neuronales para detectar valores erróneos, aunque hay pocos trabajos que informen sobre resultados destacables.

Finalmente, dentro del nombre genérico de *data mining* se han agrupado muchas de las técnicas descritas en este trabajo, como la detección y eliminación de duplicados, o la fusión de registros que comparten identificador.

Para finalizar este apartado, señalar que las aportaciones del *data mining* en el estudio de la calidad de los datos es un tema aún pendiente. Para una revisión más exhaustiva se recomienda el trabajo de Dasu y Johnson (2003).

4
PROCEDIMIENTO DE DEPURACIÓN DE UNA TABLA

4.1 INTRODUCCIÓN

Tradicionalmente la depuración de los datos ha constituido una práctica muy poco frecuente. La mayoría de los estudios se han venido realizando presuponiendo la calidad de la información que se maneja y analiza. Sin embargo, en la actualidad el concepto de depuración ha ganado un notable protagonismo debido al auge de técnicas como el *data mining*, que facilitan la gestión de grandes volúmenes de datos pero como contrapartida requieren que éstos estén libres de errores. Esta situación ha originado un creciente interés por todos aquellos procesos que se diseñan e implementan con el objetivo de garantizar la calidad de la información.

Hasta el presente la depuración de los datos se ha venido realizando en contextos donde se maneja información de dudosa calidad. Puesto que no se dispone de procedimientos estandarizados ni de un software adecuado que los automatice, en la práctica la depuración de los datos implica repetir las mismas secuencias (o similares) para todas las variables de cada estudio. Esta actuación denota una total falta de eficiencia, ya que es vulnerable de verse afectada por un gran número de errores al realizar las mismas operaciones repetidas veces.

En este capítulo se efectúa un riguroso análisis del proceso de depuración identificando cuáles son los aspectos sistemáticos generales a todo estudio, cuáles los aspectos sistemáticos particulares para conjuntos de estudios y cuáles los aspectos particulares aplicables a una base de datos concreta. A continuación se propone un procedimiento estandarizado que automatiza el conjunto de comprobaciones sistemáticas generales que también contempla la incorporación de las comprobaciones de tipo no sistemático.

A nivel algorítmico se proporcionan definiciones expresadas en lenguaje formal a través de macros SPSS (SPSS Inc., 1999; Bonillo, Doménech y Granero, 2000) que permiten realizar de forma automática la depuración exhaustiva de los datos contenidos en la tabla de un estudio. La lógica de estos algoritmos es fácilmente transportable a otros lenguajes (por ejemplo SAS ó SQL).

4.2 ANÁLISIS DE LA CONSISTENCIA BÁSICA DE LOS DATOS: TIPOS DE COMPROBACIONES

La mayoría de trabajos publicados adolece de poca operativización en cuanto a los controles a realizar (Melgratti y Yankelevich, 2000). Los artículos publicados suelen centrarse en un tipo de control concreto y ni siquiera mencionan el resto (Bobrowski, Marré y Yankelevich, 2001; Maletic, 1999). Son escasos los trabajos que sí realizan un estudio sobre cuáles son las comprobaciones de tipo sistemático que deben efectuarse sobre los datos grabados (Clarke, 1993; Cody, 1999).

Los chequeos que propone Cody consisten en detectar: a) errores de formato si los datos están grabados en ASCII; b) valores fuera de rango; c) fechas en orden no secuencial; d) inconsistencias entre variables; e) valores desconocidos; f) identificadores duplicados; g) vulneraciones a la integridad referencial; y h) número inadecuado de réplicas por caso.

Nuestra propuesta añade a los estudios de Clarke (1993) y Cody (1999) la identificación de las distintas tipologías de valor válido, no válido, desconocido y "no aplicable", además de detectar inconsistencias entre valores registrados a lo largo de seguimientos. Las comprobaciones consisten en detectar cualquier incidencia que se deriva de una inconsistencia debida a errores de formato, datos fuera de rango o no contenidos en una lista de valores válidos, valores incongruentes en variables dentro de saltos, valores incoherentes entre seguimientos e incompatibilidades lógicas entre variables. Además de estos controles, se detectan los datos desconocidos que podrían ser recuperados. A lo largo del capítulo se detallan estas comprobaciones y se presentan ejemplos para cada tipo de incidencia.

La comprobación de datos desconocidos requiere una consideración especial. Es común asociar el valor desconocido con la falta de respuesta e identificarlo con un vacío en la matriz de datos. Sin embargo, las situaciones generadoras de datos desconocidos son muy diversas y, además, el vacío no siempre indica olvido o falta de respuesta.

La Figura 4-1 presenta tres tipologías básicas de falta de datos para una variable contenida en un salto. En el ejemplo 1 el sujeto se niega a contestar con qué frecuencia consume drogas, y por consiguiente se le asigna código 9 para marcar que se trata de un dato desconocido y no recuperable.

En el siguiente ejemplo el entrevistador ha registrado que el sujeto consume drogas pero ha olvidado preguntar por la frecuencia. En este caso el campo aparece *vacío* y corresponde a un valor desconocido que podría ser recuperado.

En el tercer ejemplo el sujeto no consume drogas y por lo tanto no es pertinente preguntar la frecuencia. En este caso el campo aparece *vacío* y corresponde a un valor "no aplicable".

4.2.1 Comprobaciones sistemáticas generales

Dentro del conjunto de posibles comprobaciones sistemáticas sobre los datos se debe distinguir entre comprobaciones sistemáticas generales y sistemáticas particulares. Las primeras son generales a toda variable; las segundas son particulares porque son aplicables sólo en algunos ámbitos de investigación, aunque su frecuencia de uso las hace interesantes de sistematizar.

Estos controles generales aplicables a todas las variables del estudio están formados por comprobación de identificadores, comprobaciones de rango y de valores desconocidos.

En los protocolos estructurados en forma de árbol lógico (con variables de salto o filtro), cuando una rama no deba editarse se comprobará que dichas variables están vacías o contienen el valor deducible. Por ejemplo, si FUMA es una variable de filtro que afecta a las variables "consumo de tabaco" (TAB) y "tipo de tabaco" (TIPTAB), para los no fumadores se debe comprobar que TAB vale 0 y TIPTAB está vacía. En cuanto a los valores permitidos dentro de estructuras de salto, la Figura 4-3, que se mostrará en el apartado 4.7.4, detalla la problemática de los valores válidos en función del valor de salto.

Si los datos están grabados en ASCII se debe comprobar que los valores contenidos en campos numéricos y fecha son consistentes con su formato.

La implementación en lenguaje formal de todas estas comprobaciones sistemáticas generales se realizará según la escala de medida de la variable sea cuantitativa o categórica (nominal u ordinal) o según el campo contenga una fecha o un identificador.

4.2.1.1 Variables cuantitativas

Se comprobará si sus valores están dentro del intervalo que determina el conjunto de valores válidos y si tienen la precisión especificada. Los límites del intervalo pueden ser límites de aviso. Si la variable se ha registrado como cadena se comprobará que sólo contiene caracteres numéricos.

Por ejemplo, podremos comprobar si la talla de un sujeto está comprendida entre 0.95 y 2 metros, y si el valor registrado no supera las 2 cifras decimales. En este caso una talla de 2.03 metros producirá un mensaje de aviso, aunque dicho valor podría ser correcto.

4.2.1.2 Variables categóricas

Se comprobará si sus valores pertenecen a un determinado conjunto de valores válidos. Si los códigos son valores numéricos y la variable se ha registrado como cadena se comprobará que sólo contiene caracteres numéricos.

La implementación informática de esta comprobación dependerá del número de elementos del conjunto y de su codificación; para conjuntos grandes de códigos no secuenciales (por ejemplo códigos CIE) será conveniente utilizar una tabla de claves.

Ejemplos de estos tipos de controles sería comprobar que el "Sexo" se haya codificado únicamente con valores F o M, que el "Nivel socioeconómico" sea un número natural entre 1 y 5 (SES en la escala de Hollingshead) o que el código diagnóstico de una patología exista en la clasificación CIE-9.

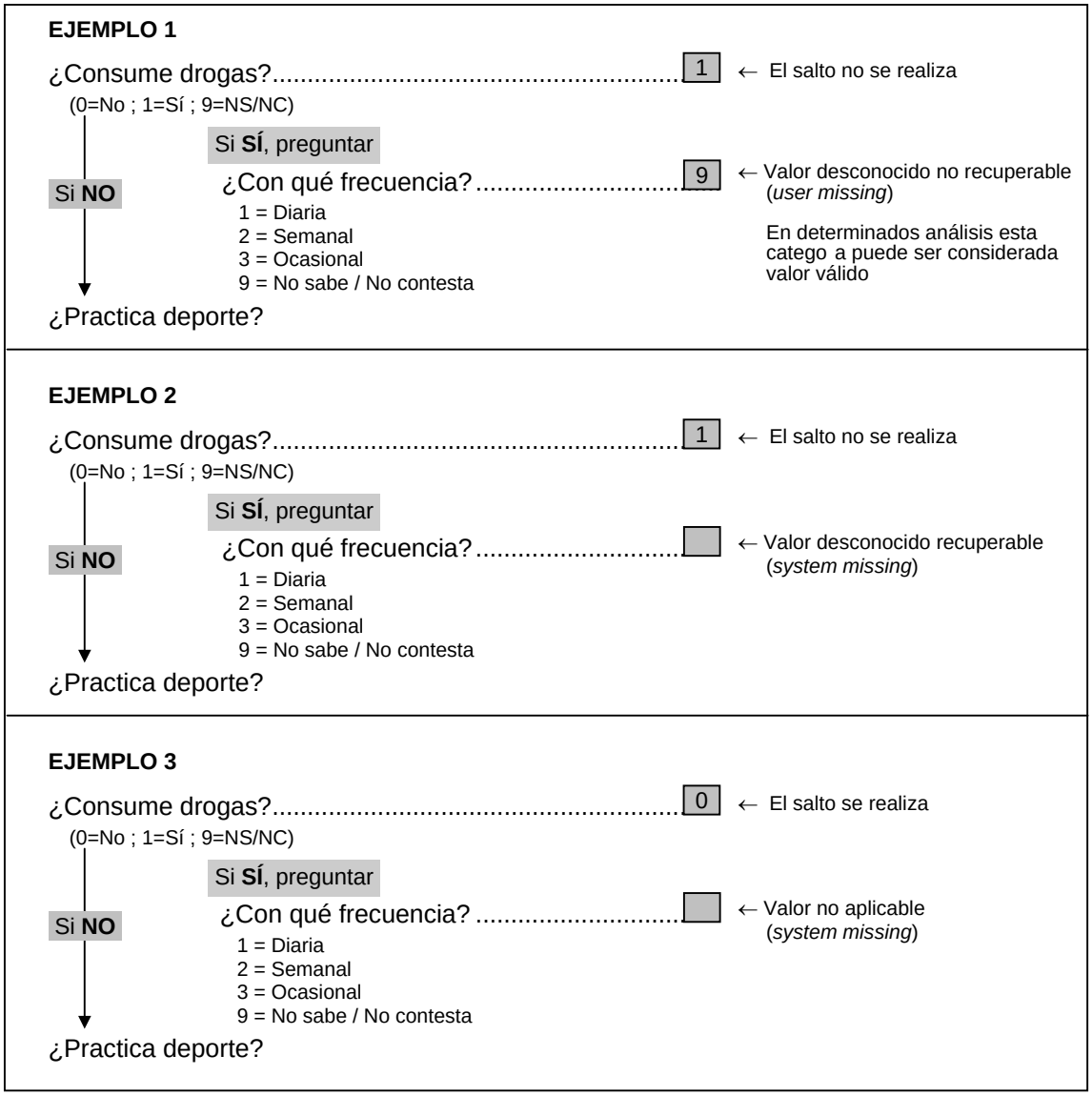


Figura 4-1: Tipologías básicas de valores desconocidos.

4.2.1.3 Fechas

Existen dos modos de comprobar el rango de los campos fecha. En algunos casos conviene comprobar si están dentro de un determinado intervalo de fechas; por ejemplo, que la fecha de respuesta a un cuestionario está comprendida entre el 1-4-1993 y el 30-12-1993.

En otros casos conviene comprobar si el intervalo de tiempo transcurrido respecto a una fecha de referencia pertenece a un determinado rango. Por ejemplo, si la edad de los sujetos cuando contestan un cuestionario está comprendida entre 6 y 45 años; en este caso la fecha de nacimiento se comprueba respecto a la fecha de respuesta (referencia).

4.2.1.4 Depuración de variables cuantitativas, categóricas y fechas ante la presencia de seguimientos

Además de las comprobaciones anteriores, si las variables han sido objeto de seguimiento, es decir, han sido registradas repetidamente a lo largo del tiempo, es imprescindible comprobar ciertos supuestos.

En primer lugar, existen valores que no deberían variar a lo largo de los seguimientos. Este tipo de variables suelen ser de tipo censal y sus valores o son inmutables o son tan estables a lo largo del tiempo que se considera que no se ha dado el lapso suficiente entre seguimientos para que se produzcan cambios. En el caso de variables cuantitativas podríamos dar como ejemplos la altura en sujetos adultos o la puntuación obtenida en una escala psicológica realizada en el pasado. En el caso de variables categóricas podríamos tener ejemplos semejantes en el sexo de los sujetos o en la provincia de nacimiento. En el caso de fechas, podríamos hallar tipologías semejantes en la fecha de nacimiento o la fecha de ingreso en un servicio de salud.

Sin embargo, hay otras variables que, obviamente, no sólo pueden variar entre seguimientos sino que lo habitual es que lo hagan, haciendo coherente la monitorización del registro. Ejemplos de ello los tendríamos en el resultado de una prueba psicológica realizado en varios momentos a lo largo del tiempo o en el valor de la presión arterial sistólica en pacientes hipertensos.

Otra comprobación susceptible de realizarse sería verificar que la variación entre valores registrados de una variable cuantitativa no excede de un porcentaje, o de un valor máximo, predeterminado. Aunque es obvio que una variable debe variar a lo largo del tiempo pueden revisarse valores que hayan tenido una variación que se considere "excesiva" respecto a su medición anterior. Como "excesiva" en este contexto debemos entender no aquella que perjudique al sujeto o que resulte clínicamente alarmante, sino aquella que resulte sospechosa de ser un error y no un valor correcto. En cuanto a la presión arterial sistólica podría considerarse que una variación entre seguimientos superior a un 30-35%, o a 40-50mmHg, podría constituir un error. Otro ejemplo semejante sería aquel en que un sujeto que responde a pruebas de algún constructo que se considere relativamente estable, por ejemplo la inteligencia, obtiene a lo largo del tiempo valores que fluctúan un 20-30% o variaciones superiores a 15 en una escala de tipo Weschler (que en este contexto designa una desviación estándar).

La última de las comprobaciones susceptible de realizarse ante variables registradas a lo largo del tiempo es aquella que produce el cese del seguimiento. Ante la presencia de un valor dado, que puede ser el "exitus" del sujeto o una variable indicadora del abandono del estudio, no deberían haberse registrado seguimientos posteriores. Si se han producido, o bien no era correcta la codificación del "exitus" o bien el seguimiento no se ha realizado.

4.2.1.5 Identificadores

El identificador de un caso puede estar formado por una o más variables. Los identificadores no pueden estar duplicados ni vacíos. Tras la corrección de duplicados, cada una de las variables que forman el identificador debe ser sometida a los controles sistemáticos que le son propios según su escala de medida.

Si el archivo de datos a depurar está relacionado con otros se deberá comprobar también la integridad referencial entre sus identificadores.

Si el archivo debe contener un determinado número de réplicas de cada sujeto (por ejemplo, un número fijo de seguimientos) se deberá comprobar que el identificador completo (incluido el identificador de réplica) no presenta duplicados y también que el número de veces que se repite la parte común del identificador corresponde al valor fijado. La Figura 4-4, en el apartado 4.8, mostrará con un sencillo ejemplo estas comprobaciones.

4.2.2 Comprobaciones sistemáticas particulares

Existe un conjunto de comprobaciones particulares para ciertas variables, pero comunes para diversos estudios, que pueden ser sistematizadas y acumuladas con objeto de aplicarlas repetidamente.

Un ejemplo sería la comprobación de las variables talla y peso en función de la edad, incluyendo la restricción que la masa corporal para cualquier edad está dentro de un determinado rango de valores. Este control se podría aplicar a todos los estudios con sujetos en fase de crecimiento que registren estas tres variables.

4.2.3 Comprobaciones no sistemáticas

Las comprobaciones no sistemáticas son controles particulares de un determinado estudio que difícilmente son generalizables a otras investigaciones. Se trata, en general, de incompatibilidades de tipo lógico entre conjuntos de variables del estudio. Por ejemplo, comprobar que los sujetos con edad menor de 11 años no fuman de forma habitual.

4.3 PRESENTACIÓN DE UN ARCHIVO DE COMPROBACIÓN

Para formalizar la propuesta de depuración presentada en este trabajo se ha diseñado el archivo de test TestNoDep.DAT, cuyos datos se reproducen en la Figura 4-2. Este archivo está formado por un conjunto de 20 registros que representan seguimientos realizados a 12 sujetos ficticios que incorporan la totalidad de incidencias de tipo general descritas. El objeto de este archivo de test es verificar el buen funcionamiento del procedimiento propuesto y de las macros que lo automatizan. Este archivo va acompañado de la descripción de sus variables junto con los controles a aplicar, y de dos tablas, CENSAL.SAV (para verificar la integridad referencial del identificador) y CIE9.SAV (para verificar la existencia de códigos CIE9).

Es poco habitual verificar de forma sistemática el comportamiento de las rutinas que se utilizan (Marcus y Robins, 1998). Sin embargo, el archivo de test propuesto cumple esta función. También puede servir como criterio de referencia para que otros investigadores comprueben la validez de sus algoritmos y/o estrategias de depuración.

El Listado 4-1 muestra el contenido de la tabla principal “CENSAL.SAV”, que incluye las variables H (código de hospital) y CASO (número de sujeto) que identifican cada sujeto, junto al nombre, apellido y código postal.

El Listado 4-2 presenta una parte del archivo diccionario “CIE9.SAV”, que contiene las variables CIE (código CIE9) y DES (descripción) de todas las patologías.

En la Figura 4-2 se presentan los datos del archivo de comprobación y los atributos de las variables: formato, códigos de los valores desconocidos (si los tiene), etiqueta descriptiva de la variable y el rango o lista de valores válidos de cada una de ellas. En la tabla superior de esta figura hemos señalado en negrita los datos erróneos que deben ser detectados.

Los sujetos hipotéticos de este archivo proceden de dos hospitales, padecen patologías del aparato circulatorio y los datos han sido recogidos durante los tres últimos trimestres de 1993. Se han registrado dos variables identificadoras: el hospital de procedencia (H) y el número identificador de su historial clínico (CASO); este último puede coincidir con el de un paciente del otro hospital.

Listado 4-1: Contenido de la tabla principal "CENSAL.SAV".

```
GET FILE='CENSAL.SAV'.
LIST.

H CASO NOMBRE APE1     CPOSTAL

A   12 José   Calvo    08080
A   14 Juan   López    08400
A   19 Pedro  Roca     08140
A   21 Mario  Rios     08016
A   51 Jorge  Lli      08002
A   94 Luz    Villa    08011
A  133 Paz    Pino     08400
B   10 Mar    Vega     08002
B   16 Eva    Ortiz    08210
B   17 Ester  Casal    08013
B   82 Jaime  Coma     08614
B   94 Luís   Casas    08015
B  103 Sara   Cid      08015
C   10 Laura  Luna     08400
C   12 José   Font     08080
C   94 Lucas  Pez      08140
```

Listado 4-2: Contenido de la tabla de claves "CIE9.SAV".

```
GET FILE='CIE9.SAV'.
LIST.

CIE    DES
...    ...
012.8  Otras tbc respiratorias especificadas
012.80 Otras tbc respiratorias especificadas-neom
012.81 Otras tbc respiratorias especificadas-no examen
...    ...
975.8  Envenenam-otros farmacos respiratorios y farm respirat neom
997.3  Complicaciones quirurgicas-respiratorias
...    ...
```

Para cada paciente se han recogido, en formato papel y al inicio del estudio, la fecha de respuesta (FR), la fecha de nacimiento (FN), el sexo (SEXO), la talla (TALLA) y los factores práctica de deporte (DPT) y descanso de forma regular (DCS) cuyas respuestas se han registrado con una misma escala ordinal. Se ha preguntado si el sujeto fuma (pregunta de filtro) y en caso afirmativo se registra el número de cigarrillos fumados al día (TAB) y el tipo de tabaco (TIPTAB).

Archivo TestNoDep.DAT

	H	CASO	FR	FN	SEXO	TALLA	DPT	DCS	FUMA	TAB	TIPTAB	CIE	PAD	PAS	EXITUS
1	A	.	11.07.1993	12.04.1965	M	1.69	1	1	1	20	NE	398.90	104	162	0
2	A	21	30.07.1993	12.04.1965		1.69	1	1	1	20	NE	398.90	90	168	**1**
3	A	21	17.09.1993	12.04.1965	M	1.69	1	1	1	20	NE	398.90	**40**	174	0
4	A	94	08.06.1993	15.07.1966	F	1.79	2	3	0	**15**			94	182	0
5	a	12	01.11.1993	07.11.1966		1.70	**0**		9	**0**		432	102	154	0
6	A	133	21.06.1993	25.05.1949	F		3	1	7			415.11	86	104	0
7	A	133	14.08.1993	25.05.1949	M		3	1				415.11	84	**184**	0
8	A	133	14.12.1993		F		3	1				415.11	**20**	104	0
9	A	14	**13.17.1993**	30.01.1954	M	**2.75**	2	2			RU		96	178	0
10	A	14	13.07.1993	30.01.1954	M	1.75	2	2			RU		96	178	0
11	A	**5.1**	21.05.1993	11.06.1968	m	1.73			1	15	nR	435.9		138	0
12	B	16	14.12.1993	**16.10.**	F	1.77	1	3	**O**	0		423.9	94	150	0
13	B	17	**07.09.1994**	**01.11.1987**	F	**.981**		2	1	**0**		411.0	88	132	0
14	B	82	**20.11.1962**	**13.05.1993**	M		2	1	0	0		398.90	100	156	0
15	B	82	01.08.1993	20.11.1962	M		2	1	0	0		398.90	90	150	0
16	B	82	05.11.1993	20.11.1962	M		2	1	0	0		422.90	95	130	0
17		94	22.04.1993	20.10.1961	**V**	1.78	2	2	1	**-1**	RU	429.9	86	162	0
18	**C**	10	05.11.1993	03.04.1952	M	1.66	3	3	1	10	N	**030.3**	94	138	0
19	B	103	29.11.1993	05.11.1972		1.58	3	3	0	0			88	138	0
20	B	103	29.11.1993	05.11.1972	F	1.58	3	3	0	0			88	138	0

↑

Variable	Formato[1]	Valor desconocido	Descripción de la variable	Rango (o lista) de valores válidos
h*	Cadena	blanco	Código de hospital	A, B } Identificador:
caso*	Num. (0)		Número historial	001 a 150 } Tabla CENSAL.SAV
fr	Fecha (E)		Fecha de respuesta	FR–FN: 6 a 45 años
fn	Fecha (E)		Fecha de nacimiento	1-4-1993 a 30-12-1993
sexo	Cadena	blanco	Sexo	M=Masculino; F=Femenino
talla	Num. (2)		Talla (m)	0.95 a 2.00 metros
dpt	Num. (0)		¿Practica deporte?	1,2,3
dcs	Num. (0)		¿Descansa regularmente?	1,2,3
fuma**	Num. (0)	9	¿Es fumador?	0=No ; 1=Sí ; 9=NS/NC***
tab	Num. (0)		Consumo tabaco (c/d)	1 a 80
tiptab	Cadena	blanco	Tipo de tabaco	NE=Negro; RU=Rubio; NR=Ambos
cie	Cadena	blanco	Código CIE	Diccionario CIE9.SAV Códigos principales: 390.0 a 459.9
pad	Num. (0)		Presión arterial diastólica	20 a 300 mmHg (30% de var.)
pas	Num. (0)		Presión arterial sistólica	50 a 400 mmHg (30% de var.)
exitus	Num. (0)		Muerte del paciente	0,1

[1] Tipos de formato: Num. (n): Número con *n* decimales. Fecha (E): Fecha europea.
(*) Variables que forman el identificador de registro.
(**) Pregunta de salto (filtro o cribado).
(***) Este código sirve para distinguir el olvido de la respuesta del caso en que el sujeto no desea contestar.

Figura 4-2: Contenido del archivo de prueba TestNoDep.DAT.

Además de estas variables, registradas únicamente en el momento inicial del estudio, se registró cada vez que el paciente acudió a consulta, también en formato papel, el código CIE9 correspondiente al diagnóstico principal (CIE9), las presiones arteriales diastólicas (PAD) y sistólicas (PAS) y una variable indicadora (EXITUS) que registraba el fallecimiento del paciente si éste se había producido. En adelante,

identificaremos este concepto de segunda o posterior asistencia a consulta como seguimiento.

Puede observarse como algunos pacientes han asistido varias veces a consulta, por ejemplo el caso 133 proveniente del hospital A que ocupa los registros 6, 7 y 8, mientras que otro pacientes sólo han acudido una vez, por ejemplo el caso 94 proveniente del hospital A.

Tras el registro inicial en formato papel los datos fueron transferidos a formato magnético y agregados en un solo archivo tal y como se muestra en la Figura 4-2.

4.4 PROPUESTA DE UN PROCEDIMIENTO DE DEPURACIÓN

El procedimiento sistemático de depuración que proponemos está centrado en cada una de las variables a depurar V_i del estudio: se aplica el conjunto de comprobaciones descritas a todos los $casos_j$ para cada variable V_i y el resultado de este chequeo se codifica en una variable auxiliar numérica $@V_i$.

La Tabla 4-1 presenta los códigos resultantes del chequeo. Observe que los códigos negativos indican valores correctos y los positivos incidencias, siendo los valores menores de la decena asignados a incidencias entre variables del mismo seguimiento y los iguales o superiores a diez reservados para inconsistencias entre seguimientos, siempre del mismo caso. Los valores iguales o superiores a cincuenta se reservan para las incidencias no sistemáticas, concepto visto en el apartado 4.2.3.

En cuanto a los valores correctos, observe que la variable auxiliar $@V_i$ tomará el valor –4 si V_i contiene un valor válido, –3 si V_i contiene un valor correcto que es deducible, –2 si V_i tiene valor "no aplicable" y –1 si V_i tiene un valor desconocido no recuperable.

En cuanto a los valores que codifican incidencias entre variables del mismo seguimiento, observe que la variable $@V_i$ tomará el valor 0 cuando el valor de V_i sea desconocido recuperable (*missing*), valor 1 cuando V_i está contenida dentro de un salto y tiene un valor incongruente con dicho salto, valor 2 cuando V_i tiene un error de formato, valor 3 cuando tiene un valor que no pertenece al rango o conjunto de códigos válidos, valor 4 cuando una diferencia de fechas no pertenece al rango y valor 5 cuando V_i tiene formato numérico con más decimales de los especificados.

La variable $@V_i$ tomará el valor 10 cuando V_i, debiendo permanecer constante, varíe a lo largo de los seguimientos, valor 11 cuando el rango de variación absoluto o relativo de V_i exceda del fijado por el usuario y valor 12 cuando un registro no debiera aparecer porque el valor de V_i indica que debería haber sido dado de baja en el estudio.

Finalmente, $@V_i$ tomará valores a partir de 50 (50, 51 ...) cuando se incumplan condiciones particulares entre V_i y otras variables del mismo seguimiento. A lo largo del trabajo se presentan distintas situaciones que ilustran errores que darán lugar a cada uno de estos códigos.

El sistema de codificación propuesto es amplio para poder identificar las diferentes tipologías de valores válidos y no válidos. Por ejemplo, la distinción entre valores desconocidos recuperables, no recuperables y no aplicables, permitiría disponer de una matriz final de datos depurados en la cual una parte de los *system missing* pudieran ser reconvertidos a códigos *user missing* (que identificaran los no aplicables y los recuperables) para poder ser utilizados en los análisis estadísticos.

Tabla 4-1: Correspondencia entre el valor original de la variable y el código de la variable auxiliar[1].

Variable original V	Variable auxiliar @V	Ejemplos
Valor válido	–4	talla = 1.75 → @talla = –4
Valor deducible	–3	tab = 0 si fuma = 0 → @tab = –3
Valor no aplicable	–2	tiptab = *vacío* si fuma = 0 → @tiptab= –2
Valor desconocido (no recuperable)	–1	fuma = 9 (NS/NC) → @fuma= –1
Valor desconocido (recuperable)	0	talla = *vacío* → @talla = 0
Valor inconsistente con el salto	1	tab ≠ 0 si fuma = 0 → @tab = 1 tab = *vacío* si fuma = 7 → @tab = 1
Error de formato	2	fuma = O → @fuma = 2
Valor fuera de rango	3	talla = 2.75 → @talla = 3
Diferencia de fechas fuera de rango	4	fn=13.05.1993 y fr=20.11.1962→ @fn = 4
Error de precisión	5	talla = 0.981 → @talla = 5
Error en una constante[2]	10	$talla_j$ = 1.69 y $talla_{j+1}$ ≠ 1.69 → @talla = 10
Exceso de variación	11	pad_j = 130 y pad_{j+1} =60 → @pad = 11
Baja de un caso	12	$exitus_j$ = 1 y $caso_{j+1}$ ≠ *vacío* → @exitus=12
Errores por inconsistencias entre variables	50, 51 ...	fuma = 1 y edad < 11 → @fuma = 50

[1] La primera parte de la tabla incluye los valores correctos, la segunda las incidencias entre variables del mismo seguimiento y la tercera entre variables del mismo caso pero de distintos seguimientos. Finalmente, contiene los códigos reservados para incidencias no sistemáticas. [2] El subíndice j indica el valor de un caso en una variable i, siendo j+1 el valor del mismo caso en un seguimiento posterior.

Una vez creada la variable auxiliar $@V_i$ se seleccionan los casos con incidencias ($@V_i \geq 0$) y se listan los identificadores, las variables V_i y $@V_i$, y el resto de variables implicadas en los chequeos realizados.

Finalizado este proceso, que se repite para cada una de las variables V_i a depurar, se realiza un informe de incidencias. Se lista, para cada caso que presenta alguna incongruencia, el conjunto de incidencias detectadas en él. Este informe de incidencias por caso es el documento de trabajo para iniciar la búsqueda de los valores correctos en las fuentes originales y efectuar las correcciones pertinentes.

El segundo ciclo del chequeo comienza escribiendo para cada modificación una instrucción que cambie el valor erróneo por el correcto, si se conoce, o por *vacío* si se desconoce.

Cuando no es posible recurrir a las fuentes originales, los errores detectados se deben pasar de forma sistemática y automática a valor desconocido. En este caso, el proceso de chequeo excluirá listar los valores desconocidos porque no dispondremos de información para recuperarlos.

Seguidamente se activa de nuevo el proceso de chequeo para el conjunto de variables V_i, ya que los errores detectados podrían enmascarar nuevos errores o se podría haber introducido algún nuevo error con los cambios realizados.

Este proceso iterativo se repite hasta que las únicas incidencias detectadas sean valores desconocidos no recuperables. Llegado a este punto se generan los siguientes cuatro documentos necesarios para una auditoría de calidad:

1. Listado en lenguaje de ordenador del conjunto de chequeos realizado.
2. Informe de las incidencias detectadas en los datos.
3. Listado de las instrucciones que contienen cada uno de los cambios realizados en los datos originales o del algoritmo que ha asignado los errores a valor desconocido.
4. Estadísticas por sujetos y variables de los valores desconocidos presentes en la matriz de datos depurada.

4.4.1 Requisitos del procedimiento

El procedimiento que vamos a presentar tiene como requisitos previos: a) utilizar nombres de variables de un máximo de 7 caracteres y que el primero no sea el carácter @; b) transformar las letras de todas las cadenas a mayúscula, siempre que mayúsculas y minúsculas representen el mismo valor; y c) definir una variable con el orden secuencial que ocupan los sujetos.

La limitación de nombre de variable a 7 caracteres está motivada porque la variable auxiliar debe estar formada por el carácter '@' y el nombre de la variable original, y muchos programas (entre ellos SPSS) tienen limitado los nombres de los campos a 8 caracteres.

La transformación de las letras a mayúsculas permite simplificar el conjunto de valores válidos en los campos cadena que contienen letras.

La definición de una variable con el orden secuencial de los sujetos permite localizar fácilmente en el archivo de datos los registros con errores en el identificador.

Listado 4-3: Lectura de los datos del archivo de test grabado en formato DBF.

```
*Lectura del archivo de datos de prueba con estructura DBF.
GET TRANSLATE FILE='C:\...\Escritorio\TestNoDep.DBF' /TYPE=DBF.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab cie.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.
```

4.5 LECTURA DE LOS DATOS

Si los datos están en una tabla de base de datos se transfieren automáticamente a SPSS con las instrucciones GET DATA o GET TRANSLATE. Si los datos están en un archivo de texto en formato ASCII la lectura se realiza con el procedimiento DATA LIST.

El Listado 4-3 presenta la lectura del archivo de test grabado en formato DBF y las instrucciones para transformar el conjunto de variables cadena a mayúsculas y para crear la variable con el orden secuencial de los sujetos.

La parte superior del Listado 4-4 presenta la lectura del archivo de test grabado en formato ASCII. La parte inferior muestra el listado estándar de incidencias que presenta la instrucción DATA LIST cada vez que lee un dato incompatible con el formato.

Este listado automático de errores es de lectura muy compleja y en la práctica sólo resulta útil cuando hay muy pocos errores de formato. Por este motivo incluiremos en nuestra propuesta de algoritmos de depuración el chequeo de estas incidencias. Para ello realizaremos una doble lectura de todos los campos no cadena por ser susceptibles de contener este tipo de error. La primera lectura se efectúa con su formato correcto (numérico o fecha) y la segunda con formato cadena para poder leer los caracteres grabados en ASCII (Cody, 1999). El error de formato se detectará cuando la variable leída con formato cadena contiene algún carácter diferente de blanco y la variable leída con su formato adecuado toma valor *system missing*.

El Listado 4-5 presenta la instrucción DATA LIST para realizar la doble lectura del archivo de test. Observe que las variables cadena de doble lectura toman el nombre de la variable a depurar precedido del carácter "•". Para evitar que la instrucción DATA LIST genere el listado automático de incidencias se ha desactivado esta opción con la instrucción SET ERRORS = NONE.

Listado 4-4: Lectura de los datos del archivo de test grabado en formato ASCII y listado automático de los errores.

```
*Lectura de datos ASCII.
DATA LIST FILE=' C:\...\Escritorio\TestNoDep.DAT'
/ h 2(A) caso 4-6(F) fr 8-17(EDATE) fn 19-28(EDATE) sexo 30(A) talla 32-35(F,2)
  dpt 37(F) dcs 39(F) fuma 41(F) tab 43-44(F) tiptab 46-47(A) cie 49-54(A)
  pad 56-58(F) pas 60-62(F) exitus 64 (F).
EXECUTE.
```

```
>Warning # 1151
>A field to be read under the EDATE format is invalid.  The field must
>contain day, month, and year separated by spaces, dashes, slashes, decimal
>points, or commas.  Note that American style dates (month/day/year) can be
>read under the ADATE format.

>Command line:   8  Current case: 9  Current splitfile group: 1
>Field contents: '13-17-1993'
>Record number: 9  Starting column: 8  Record length: 8192

>Warning # 1102
>An invalid numeric field has been found.  The result has been set to the
>system-missing value.

>Command line:   8  Current case: 12  Current splitfile group: 1
>Field contents: 'O'
>Record number: 12  Starting column: 41  Record length: 8192

>Warning # 1151
>A field to be read under the EDATE format is invalid.  The field must
>contain day, month, and year separated by spaces, dashes, slashes, decimal
>points, or commas.  Note that American style dates (month/day/year) can be
>read under the ADATE format.

>Command line:   8   Current case: 12  Current splitfile group: 1
>Field contents: '16-10-  '
>Record number: 12  Starting column: 19  Record length: 8192
```

Error detectado: falta especificar el año

El examen del DATA LIST indica que en la columna 19 empieza la variable FN

El error se encuentra en el 12avo caso de la ventana de datos

Listado 4-5: Lectura de datos propuesta para el archivo de test grabado en formato ASCII.

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='C:\...\Escritorio\TestNoDep.DAT'
  /h 2(A)
   caso   4-6(F)     ·caso  4-6(A)
   fr    8-17(EDATE) ·fr    8-17(A)
   fn   19-28(EDATE) ·fn    19-28(A)
   sexo   30(A)
   talla  32-35(F,2) ·talla 32-35(A)
   dpt    37(F)      ·dpt   37(A)
   dcs    39(F)      ·dcs   39(A)
   fuma   41(F)      ·fuma  41(A)
   tab    43-44(F)   ·tab   43-44(A)
   tiptab 46-47(A)
   cie    49-54(A)
   pad    56-58(F)   ·pad    56-58(A)
   pas    60-62(F)   ·pas    60-62(A)
   exitus 64   (F)   ·exitus 64   (A).
SET ERRORS=NONE.  /*Desactiva el listado de errores de formato del DATA LIST.
EXECUTE.
SET ERRORS=ON.    /*Activa de nuevo el listado de mensajes de error.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab cie.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.
```

4.6 COMPROBACIONES NO SISTEMÁTICAS

Este tipo de comprobaciones se efectúa para valorar la consistencia de una variable V_i respecto a otras y son específicas de cada estudio. Generalmente se implementan en forma de condiciones lógicas. El procedimiento propuesto consiste en asignar a la variable auxiliar $@V_i$ un código a partir de 50 cuando se incumpla dicha condición.

Por ejemplo, en el archivo de test se debe detectar a aquellos sujetos que afirman que fuman y todavía no han cumplido los 11 años. La parte superior del Listado 4-6 presenta las instrucciones que implementan dicha condición. El procedimiento comporta asignar el error detectado a una de las variables implicadas en la condición lógica, generalmente la primera contenida en el archivo de datos aunque esta elección es arbitraria (en este ejemplo lo haremos sobre @FUMA). El código que registra el error es 50 porque es la primera comprobación no sistemática que se efectúa sobre la variable FUMA. Si se programaran otras condiciones no sistemáticas sobre esta misma variable se utilizarían códigos de error 51 y sucesivos.

Si se realiza una comprobación no sistemática sobre otra variable, por ejemplo TALLA, el incumplimiento de la primera condición lógica quedaría recogido con el código 50 (@TALLA=50). Esta duplicidad no puede inducir a error porque el significado del código 50 es específico de cada variable.

Listado 4-6: Detección de errores no sistemáticos de la variable FUMA.

```
* Condición @FUMA=50: Los fumadores deben tener más de 10 años.
IF (fuma=1 AND CTIME.DAYS(fr-fn)<(365.25*11)) @fuma=50.
FORMATS @fuma (F2).
TEMPORARY.                          /*Selección y listado de errores.
SELECT IF (@fuma >= 50).
LIST @casenum h caso @fuma ·fuma fn fr.
```

```
@CASENUM H CASO @FUMA ·FUMA         FN         FR
      13 B   17    50  1    01.11.1987 07.09.1994

Number of cases read:  1
```

En la parte inferior del Listado 4-6 se lista el error detectado: el caso 17 fuma y aún no ha cumplido los 11 años (@FUMA=50). Observe que se ha listado los identificadores (@CASENUM, H y CASO), las variables @FUMA y •FUMA, y las fechas de nacimiento y de respuesta (FN y FR) implicadas en la condición lógica.

4.7 COMPROBACIONES SISTEMÁTICAS

La implementación práctica del procedimiento de depuración se efectúa en función de la escala de medida de las variables y de si en el estudio se han producido seguimientos. A continuación se presentan ejemplos que ilustran la depuración sistemática de variables cuantitativas, categóricas y campos fecha. Al final del apartado se aborda la problemática que afecta la depuración de variables contenidas dentro de un salto.

4.7.1 Depuración de variables cuantitativas

Para depurar una variable cuantitativa es necesario verificar que sus valores: a) no son desconocidos; b) no contienen errores de formato; c) están dentro del rango válido; d) no se han registrado con más decimales de los posibles; e) si están dentro de un salto son congruentes con él; f) si han sido recogidos durante varios seguimientos que sus valores son consistente entre ellos, y g) no tienen incompatibilidades lógicas con otras variables. El Algoritmo 4-1 muestra en pseudo-código el proceso para depurar este tipo de variables.

La macro !DR implementa en sintaxis SPSS el proceso de depuración contenido en el Algoritmo 4-1. En la Documentación 4-1 se muestra cómo efectuar la llamada de esta macro para depurar variables cuantitativas. El parámetro obligatorio V recoge la lista de variables a depurar. Es imprescindible que cuando en una llamada se designe una lista de variables éstas compartan exactamente las mismas restricciones, ya que se realizarán los chequeos especificados en la llamada para todas ellas. Con el parámetro LV se especifica el rango de valores válidos de la variables; estos rangos se pueden definir en sintaxis SPSS, separando los valores con comas o blancos, o bien utilizando las palabras clave LOW, HI y THRU. El parámetro opcional ND permite especificar el número máximo de decimales para variables continuas.

El parámetro obligatorio C recoge la lista de variables identificadoras del registro. El parámetro CN, también obligatorio, recoge la variable que contiene la posición secuencial de los sujetos en el archivo de datos, y por defecto toma valor @CASENUM. El parámetro obligatorio L permite obtener un listado de las incidencias

detectadas para cada variable evaluada, pudiéndose listar errores y valores desconocidos (valor por defecto), sólo errores o no listar ninguna incidencia, dejando las transformaciones realizadas sobre los datos pendientes de ejecutar. Esta última posibilidad permite reducir enormemente el tiempo de procesamiento y es indicada cuando se trabaja con volúmenes enormes de datos. Los parámetros MV y MVr recogen, respectivamente, los códigos que identifican los valores desconocidos no recuperables (por ejemplo el código de no sabe-no contesta) y recuperables. Los parámetros VS, XS, y VD son optativos y se utilizan para depurar variables contenidas dentro de saltos. VS recoge la variable de filtro. XS permite indicar los valores para los que se ejecuta el salto. VD especifica el valor deducible que debe tomar la variable cuando el salto se ha efectuado. MVS y MVSr recogen, respectivamente, las listas de valores desconocidos no recuperables y recuperables de la variable de salto.

Si la variable ha sido recogida a lo largo de seguimientos puede ser pertinente utilizar los parámetros VAR, OUT y XOut. El parámetro VAR recoge, en formato numérico, la variación máxima permisible entre seguimientos. Este parámetro puede expresarse en valor absoluto o bien en porcentaje. Los parámetros OUT y XOut permiten especificar, respectivamente, la variable y la lista de valores de la misma, indicadores de que el caso no tendría que haberse registrado en seguimientos posteriores porque debería haber causado baja en el estudio.

Algoritmo 4-1: Depuración de variables cuantitativas.

Asignación de los parámetros:

1. Asignar a V la lista de variables a depurar.
2. Asignar a LV el rango válido de V.
3. Asignar a MV, si procede, la lista de valores desconocidos no recuperables de V.
4. Asignar a MVr, si procede, la lista de valores desconocidos recuperables de V.
5. Asignar a VAR, si procede, el valor absoluto de V, o porcentaje, máximo de variación entre dos seguimientos consecutivos. (Por defecto VAR=0).
6. Asignar a OUT, si procede, la variable que indica la baja del caso en siguientes seguimientos.
7. Asignar a XOut, si procede, el valor de OUT que implica la baja del caso en siguientes seguimientos.
8. Si se han leído los datos en ASCII y se desea detectar los errores de formato, Asignar FORMAT = 1.
9. Si hay variables implicadas en las condiciones lógicas previas, Asignar la lista de variables implicadas a LVL.
10. Asignar a ND, si procede, el número de decimales de V
11. Si V está dentro de un salto, Asignar a VS la variable que determina el salto.
    - 11.1. Asignar a XS el valor de VS que determina saltar. (Por defecto XS=0).
    - 11.2. Asignar a MVS si procede, la lista de valores desconocidos no recuperables de VS. (Por defecto MVS=9).
    - 11.3. Asignar a MVSr si procede, la lista de valores desconocidos recuperables de VS.
    - 11.4. Asignar a VD el valor de V si se salta. (Por defecto VD=vacío).
12. Asignar a C las variables que forman el identificador.
13. Asignar a CN la variable que contiene el número de secuencia de los casos. (Por defecto CN= @casenum).
14. Asignar a L el valor 0 si se desea omitir el listado de incidencias, 1 si se desea listar errores y valores desconocidos ó 2, si se desea listar errores pero no valores desconocidos. (Por defecto L=1).

Proceso (para cada caso):

Sea V_i el elemento i de la lista V y $@V_i$ la correspondiente variable auxiliar numérica.

15. Si $@V_i >= 50$ y $LVL \neq \varnothing$ (error en condiciones lógicas previas), Ir al paso 31.
16. $@V_i$ = SYSMIS.
17. Si V_i = MV, $@V_i = -1$.
18. Si $V_i = \varnothing$ ó V_i = MVr, $@V_i = 0$.
19. Si $V_i = \varnothing$ y $\bullet V_i \neq$ blanco, $@V_i = 2$.
20. Si $V_i \in$ LV, $@V_i = -4$.
21. Si $V_i \notin$ LV, $@V_i = 3$.
22. Si (Módulo 1 de $V_i \times 10^{ND}$) $\neq 0$, $@V_i = 5$. (V_i tiene más decimales que ND)
23. Si VAR $\neq 0$ y VAR $\neq$ vacío y $(V_i - V_{j+1}) >$ VAR, $@V_i = 11$.
24. Si VAR= 0 y $(V_j \neq V_{j+1}) \neq 0$, $@V_i = 10$.
25. Si OUT$\neq$vacío y VAR_{OUT} = XOut y Registro $_{j+1} \neq$ vacío, $@V_i = 12$.
26. Si VD$\neq$vacío y VS = XS y V_i = VD, $@V_i = -3$.
27. Si VD=vacío y VS = XS y V_i = VD, $@V_i = -2$.
28. Si VS = XS y $V_i \neq$ VD, $@V_i = 1$.
29. Si VS $\neq$ XS y VS $\neq$ XnS y VS $\neq$ MV, $@V_i = 1$.
30. Si VS = MV y $V_i \neq$ vacía, $@V_i = 1$.
31. Si L=1 y $@V_i >= 0$:
    - 31.1. Si LVL=@NULL, LVL = $\varnothing$.
    - 31.2. Si FORMAT = 1, Listar: CN, C, VS, $@V_i$, $\bullet V_i$, LVL.
    - 31.3. Si FORMAT = 0, Listar: CN, C, VS, $@V_i$, V_i, LVL.

Documentación 4-1: Parámetros de la macro !DR.

```
DEPURACIÓN DE DATOS: RANGO DE VARIABLES NUMERICAS Y CADENA
Creación 30.11.1998 Última revisión 07.07.2003
(c) A.Bonillo & JM.Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
!DR     V= Lista de variables a verificar
    [/LV]= Lista de valores válidos (en formato SPSS)
    [/ND]= Número máximo de cifras decimales de los valores (opcional)
    [/MV]= Lista de valores user missing (no recuperables) de las variables V
   [/MVr]= Lista de valores user missing (recuperables) de las variables V
[/FORMAT]= Verificación de formato (1=Sí, 0=No)      (por defecto, 0)
   [/LVL]= Lista de otras variables a listar
               La variable @V NO se inicializa a SYSMIS cuando se especifica
                nombres de variables o la palabra clave @NULL
               Si se omite el parámetro la variable @V se inicializa a SYSMIS
       /C= Lista de variables identificadoras del sujeto
  [/INDX]= Variable de índice
      /CN= Variable identificadora de la secuencia de los sujetos
                                                    (por defecto, @casenum)

      /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)
                                                    (por defecto,  1)
--------------------------------------------------------------------------------
Parámetros para variables dentro de un salto:
    [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)
    [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)
                                                    (por defecto,  0)
    [/VD]= Valor deducible si VS indica saltar      (en formato SPSS)
                                                    (por defecto, vacío)
   [/MVS]= Lista de valores user missing (no recuperables) de la variable VS
                                                    (por defecto,  9)
  [/MVSr]= Lista de valores user missing (recuperables) de la variable VS
--------------------------------------------------------------------------------
Parámetros para variables implicadas en seguimientos:
   [/VAR]= Valor, o porcentaje, máximo de variación entre seguimientos
                                                    (en formato SPSS)
   [/OUT]= Variable indicadora de la baja del registro en futuros seguimientos
  [/XOUT]= Valor de OUT que indica la baja del registro en futuros seguimientos
--------------------------------------------------------------------------------

Ejemplos de llamada:
      !DR V=tab /LV=1 thru 80 /ND=0 /VS=fuma /VD=0 /FORMAT=1 /MVS=9.
      !DR V=tiptab /LV="NE","RU","NR" /MVr=" " /VS=fuma /C=h caso.
      !DR V=sexo /LV="M","F" /MVr=" " /C=h caso.
      !DR V=dpt dcs /LV=1,2,3 /FORMAT=1 /C=h caso.
      !DR V=pad /LV=20 THRU 300 /ND=0 /VAR=30% /FORMAT=1 /C=h caso.
      !DR V=nombre  /MVr=" " /C=h caso.
```

4.7.1.1 Ejemplo de depuración de una variable cuantitativa

En el Listado 4-7 podemos ver la detección de incidencias en la variable TALLA utilizando la macro !DR. En la llamada a la macro se especifica que los valores grabados deben estar comprendidos entre 0.95 y 2 metros (LV=0.95 thru 2), y que no deben tener más de dos decimales (ND=2). Además se indica que se efectúe el chequeo de formato (FORMAT=1). Puesto que esta variable no debería variar entre seguimientos el parámetro VAR ha sido asignado a 0. Los parámetros C y INDX, obligatorios, recogen las variables que forman el identificador y el identificador de réplica, en el ejemplo la fecha de respuesta.

Listado 4-7: Detección de incidencias en la variable cuantitativa TALLA.

```
!DR V=talla /LV=0.95 THRU 2 /VAR=0 /ND=2 /VAR=0 /FORMAT=1 /C=h caso
   /INDX=fr.
```

```
@CASENUM H CASO         FR @TALLA                          ·TALLA
      6  A  133 21.06.1993  0: Missing Rec
      7  A  133 14.08.1993  0: Missing Rec
      8  A  133 14.12.1993  0: Missing Rec
      9  A   14           .  3: Fuera Rango                  2.75
     10  A   14 13.07.1993 10: Err Cnstant (Val. ant:2.75) 1.75
     13  B   17 07.09.1994  5: Err Num Dec                   .981
     14  B   82 20.11.1962  0: Missing Rec
     15  B   82 01.08.1993  0: Missing Rec
     16  B   82 05.11.1993  0: Missing Rec

Number of cases read:  9
```

En el listado de incidencias se comprueba que han sido detectadas 9 inconsistencias. Los casos 133 y 82 tienen talla desconocida, por lo cual la variable auxiliar @TALLA toma valor 0. El caso 14 tiene, en el primer seguimiento, la talla fuera de rango (2.75 m) por lo que la variable auxiliar toma valor 3; en el segundo seguimiento, la talla toma valor dentro del rango válido de valores pero distinto del del primer seguimiento, por lo que la variable auxiliar toma valor 10. Finalmente, el caso 17 ha sido registrado con precisión incorrecta, ya que tiene 3 decimales; la variable auxiliar toma para este registro el valor 5.

4.7.1.2 Ejemplo de depuración de variables cuantitativas implicadas en seguimientos

En el Listado 4-8 podemos ver la detección de incidencias en las variables PAD y PAS utilizando la macro !DR. En la llamada a la macro para depurar la variable PAD se especifica que los valores grabados deben estar comprendidos entre 20 y 300 mmHg (LV=20 thru 300), que no deben tener decimales (ND=0) y que se efectúe el chequeo de formato (FORMAT=1). A estos parámetros, ya vistos en el ejemplo anterior, se debe añadir el parámetro VAR, especificando que es tolerable una variación del 30% entre cada seguimiento y el posterior.

En la llamada a la macro para depurar la variable PAS se especifica que los valores grabados deben estar comprendidos entre 50 y 400 mmHg (LV=50 thru 400) y, al igual que ocurría con la llamada de la macro PAD, que es tolerable una variación del 30%.

En el listado de incidencias de la variable PAD se comprueba que han sido detectadas 3 inconsistencias. Los casos 21 y 133 han sido detectados porque su valor, aunque dentro de los valores válidos, excede del rango de variación del 30% respecto al seguimiento anterior. El caso 51 es listado porque su valor es desconocido susceptible de ser recuperado.

En el listado de incidencias de la variable PAS se comprueban dos inconsistencias: el caso 133 excede del rango de variación del 30% respecto al seguimiento anterior en dos ocasiones.

Listado 4-8: Detección de incidencias en la variable cuantitativa PAD.

```
!DR V=pad /LV=20 THRU 300 /ND=0 /VAR=30% /FORMAT=1 /C=h caso  /INDX=fr.
```

```
@CASENUM H CASO         FR @PAD                              ·PAD
      3  A   21 17.09.1993 11: Excso Varcn (Seg. ant: 90)  40
      8  A  133 14.12.1993 11: Excso Varcn (Seg. ant: 84)  20
     11  A    5 21.05.1993  0: Missing Rec

Number of cases read:  3    Number of cases listed:  3
```

```
!DR V=pas /LV=50 THRU 400 /ND=0 /VAR=30% /FORMAT=1 /C=h caso  /INDX=fr.
```

```
@CASENUM H CASO         FR @PAS                              ·PAS
      7  A  133 14.08.1993 11: Excso Varcn (Seg. ant:104) 184
      8  A  133 14.12.1993 11: Excso Varcn (Seg. ant:184) 104
Number of cases read:  2    Number of cases listed:  2
```

4.7.2 Depuración de variables categóricas

Para depurar una variable categórica es necesario verificar que sus valores: a) no son desconocidos; b) no contienen errores de formato si se codifica con valores numéricos; c) pertenecen a un conjunto de valores válidos; d) si están dentro de un salto son congruentes con él; f) si han sido recogidos durante varios seguimientos que sus valores son consistente entre ellos, y g) no tienen incompatibilidades lógicas con otras variables. En relación al punto c, existen dos alternativas para la depuración: comprobar que los valores de la variable están incluidos en una lista o bien en una “tabla de claves” (diccionario).

El Algoritmo 4-2 muestra en pseudo-código el proceso para depurar variables categóricas cuando se dispone de una lista de valores válidos. Este algoritmo es general y permite detectar inconsistencias tanto si la variable se ha registrado con valores numéricos como con valores cadena.

La macro !DR también permite depurar gran parte de variables categóricas. En esta situación particular el parámetro ND no tiene sentido porque las variables categóricas no se miden con decimales. Asimismo, la comprobación de formato sólo tiene interés cuando la variable ha sido grabada con valores numéricos.

4.7.2.1 Depuración de una variable categórica grabada en formato cadena

En el Listado 4-9 podemos ver la detección de incidencias en la variable SEXO utilizando la macro !DR. En la llamada a la macro se ha especificado la lista de valores válidos a través del parámetro LV; puesto que la variable SEXO tiene formato cadena los valores válidos se especifican entrecomillados. El parámetro MVr=“ ” indica que el valor *user missing* recuperable para esta variable es el blanco. Puesto que se trata de una variable que no debe variar entre seguimientos se ha especificado el parámetro VAR, asignando su valor a 0. Aunque el valor por defecto de este parámetro ya es 0, y podría no haberse especificado, se pretende mostrar el modo de depurar una variable cadena cuyos valores deben ser constantes entre seguimientos.

Listado 4-9: Detección de incidencias en la variable categórica cadena SEXO.

```
!DR V=sexo /LV='M','F' /VAR=0 /MVr=' ' /C=h caso  /INDX=fr.

@CASENUM H CASO         FR @SEXO                              SEXO
       2  A   21 30.07.1993 10: Err Cnstant (Val. ant:M  )
       5  A   12 01.11.1993  0: Missing Rec
       7  A  133 14.08.1993 10: Err Cnstant (Val. ant:F  ) M
       8  A  133 14.12.1993 10: Err Cnstant (Val. ant:M  ) F
      17  B   94 22.04.1993  3: Fuera Rango                  V

Number of cases read:  5    Number of cases listed:  5
```

Listado 4-10: Depuración de las variables categóricas con formato numérico DPT y DCS.

```
!DR V=dpt dcs /LV=1,2,3 /VAR=0 /FORMAT=1 /C=h caso  /INDX=fr.

@CASENUM H CASO         FR @DPT                               ·DPT
       5  A   12 01.11.1993  3: Fuera Rango                  0
      11  A   51 21.05.1993  0: Missing Rec
      13  B   17 07.09.1994  0: Missing Rec
Number of cases read:  3    Number of cases listed:  3

@CASENUM H CASO         FR @DCS                               ·DCS
       5  A   12 01.11.1993  0: Missing Rec
      11  A   51 21.05.1993  0: Missing Rec
Number of cases read:  2    Number of cases listed:  2
```

En el listado de incidencias vemos que para el caso 12 el SEXO es desconocido y que el caso 94 tiene un valor no válido (V). Los casos 21 y 133 tienen ambos un error de variación. En el primero de los dos casos el valor del SEXO del registro 2 es desconocido, mientras que en el registro anterior tomaba el valor M. El caso 133 toma en los registros 7 y 9 el valor F mientras que en el registro 7 toma M. Ambas inconsistencias son listadas.

4.7.2.2 Depuración de una lista de variables categóricas grabadas en formato numérico

En el ejemplo presentado, las variables que preguntan sobre hábitos de salud (práctica deportiva y descanso regular) son categóricas, están grabadas numéricamente y comparten las mismas restricciones. Su depuración se realizará a partir de la llamada a la macro !DR que aparece en el Listado 4-10. El parámetro V incluye el nombre de las variables a chequear. El parámetro LV contiene la lista de códigos válidos común para ambas variables. Puesto que esta variable se únicamente al inicio del estudio el parámetro VAR toma el valor 0. Esta palabra clave especifica que los valores de DPT y DCS no pueden variar entre seguimientos. Finalmente, con FORMAT=1 se indica que se efectúe la comprobación de formato. Como se ha mostrado en llamadas anteriores, los parámetros C e INDX recogen, respectivamente, las variables identificadoras y el identificador de réplica.

En la parte inferior del Listado 4-10 vemos las incidencias detectadas. En primer lugar, el caso 12 tiene un código no válido en la variable DPT (el valor igual a 0) y valor desconocido en DCS. El caso 51 no tiene grabadas las variables DPT y DCS. Finalmente, al caso 17 le falta valor en DPT.

Algoritmo 4-2: Depuración de variables categóricas a través de una lista de valores.

Asignación de los parámetros:

1. Asignar a V la lista de variables a depurar.
2. Asignar a LV la lista de valores válidos de V.
3. Asignar a MV, si procede, la lista de valores desconocido no recuperables de V.
4. Asignar a MVr, si procede, la lista de valores desconocido recuperables de V.
5. Asignar a VAR, si procede, el valor 0 (si el valor debe ser constante a lo largo de los seguimientos) ó HI, si se permite la variación a lo largo de los seguimientos.
   (Por defecto VAR=0).
6. Asignar a OUT, si procede, la variable que indica la baja del caso en siguientes seguimientos.
7. Asignar a XOut, si procede, el valor de OUT que implica la baja del caso en siguientes seguimientos.
8. Si se han leído los datos en ASCII y se desea detectar los errores de formato,
   Asignar FORMAT = 1.
9. Si hay variables implicadas en las condiciones lógicas previas,
   Asignar la lista de variables implicadas a LVL.
10. Si V está dentro de un salto, Asignar a VS la variable que determina el salto.
    10.1. Asignar a XS el valor de VS que determina saltar. (Por defecto XS=0).
    10.2. Asignar a MVS si procede, la lista de valores desconocido no recuperables de VS.
    (Por defecto MVS=9).
    10.3. Asignar a MVSr si procede, la lista de valores desconocido recuperables de VS.
    10.4. Asignar a VD el valor de V si se salta. (Por defecto VD=vacío).
11. Asignar a C las variables que forman el identificador.
12. Asignar a CN la variable que contiene el número de secuencia de los casos.
    (Por defecto CN= @casenum).
13. Asignar a L el valor 0 si se desea omitir el listado de incidencias. (Por defecto L=1).
14. Asignar a L el valor 0 si se desea omitir el listado de incidencias, 1 si se desea listar errores y valores desconocidos ó 2, si se desea listar errores pero no valores desconocidos. (Por defecto L=1).

Proceso (para cada caso):

Sea V_i el elemento i de la lista V y $@V_i$ la correspondiente variable auxiliar numérica.

15. Si $@V_i >= 50$ y $LVL \neq \varnothing$ (error en condiciones lógicas previas), Ir al paso 29.
16. $@V_i$ = SYSMIS.
17. Si V_i = MV, $@V_i = -1$.
18. Si $V_i = \varnothing$ ó V_i = MVr, $@V_i = 0$.
19. Si $V_i = \varnothing$ y $\bullet V_i \neq$ blanco, $@V_i = 2$.
20. Si $V_i \in$ LV, $@V_i = -4$.
21. Si $V_i \notin$ LV, $@V_i = 3$.
22. Si VAR ≠ 0 y VAR ≠ vacío y $(V_j - V_{j+1}) > VAR$, $@V_i = 11$.
23. Si OUT≠vacío y VAR_{OUT} = XOut y Registro $_{j+1} \neq$ vacío, $@V_i = 12$.
24. Si VD≠vacío y VS = XS y V_i = VD, $@V_i = -3$.
25. Si VD=vacío y VS = XS y V_i = VD, $@V_i = -2$.
26. Si VS = XS y $V_i \neq$ VD, $@V_i = 1$.
27. Si VS ≠ XS y VS ≠ XnS y VS ≠ MV, $@V_i = 1$.
28. Si VS = MVS y $V_i \neq$ blanco, $@V_i = 1$.
29. Si L=1 y $@V_i >= 0$:
    29.1. Si LVL=@NULL, LVL = ∅.
    29.2. Si FORMAT = 1, Listar: CN, C, VS, $@V_i$, $\bullet V_i$, LVL.
    29.3. Si FORMAT = 0, Listar: CN, C, VS, $@V_i$, V_i, LVL.

Listado 4-11: Depuración de la variable categórica FUMA implicada en una condición lógica.

```
*Condición @FUMA=50: Los fumadores deben tener más de 10 años.
COMPUTE @fuma=$SYSMIS.
IF (fuma=1 AND CTIME.DAYS(fr-fn)<(365.25*11)) @fuma=50.

*Comprobación sistemática de la variable FUMA.
!DR V=fuma /LV=0,1 /MV=9 /VAR=0 /LVL=fn fr  /FORMAT=1 /C=h caso  /INDX=fr.
```

```
@CASENUM H CASO         FR @FUMA                              ·FUMA         FN         FR

      6  A  133 21.06.1993  3: Fuera Rango                   7     25.05.1949 21.06.1993
      7  A  133 14.08.1993 10: Err Cnstant (Val. ant:  7)          25.05.1949 14.08.1993
      8  A  133 14.12.1993  0: Missing Rec                              . 14.12.1993
      9  A   14          .  0: Missing Rec                         30.01.1954          .
     10  A   14 13.07.1993  0: Missing Rec                         30.01.1954 13.07.1993
     12  B   16 14.12.1993  2: Err Formato                    O             . 14.12.1993
     13  B   17 07.09.1994 50: Err Cond                       1     01.11.1987 07.09.1994

Number of cases read:  7    Number of cases listed:  7
```

4.7.2.3 Depuración de una variable categórica implicada en una condición lógica

La variable categórica FUMA tiene la particularidad de estar implicada en una condición lógica: se desea detectar todos aquellos fumadores con edades inferiores a los 11 años. La lógica de este tipo de comprobación consiste en efectuar primero una post-validación que recoja esta restricción y a continuación los chequeos sistemáticos con la macro !DR.

El Listado 4-11 recoge la secuencia que permite detectar las incidencias en la variable FUMA. La condición de post-validación se comprueba asignando un código 50 a los sujetos que no la cumplen (dicen ser fumadores y tienen edades inferiores a 11 años).

Cuando hay una o más condiciones lógicas, éstas se sitúan antes del chequeo sistemático de la variable FUMA para que el listado que efectúa la macro !DR incorpore tanto los errores sistemáticos como los detectados por las condiciones lógicas. En este caso la macro no debe inicializar la variable @FUMA a SYSMIS. La macro no inicializa la variable @ si encuentra el parámetro LVL que contiene la lista de variables que pudieran estar implicadas en las condiciones lógicas (en este caso FN y FR) y que aparecerán en el listado de incidencias. Si no hay otras variables implicadas o no se desea listar ninguna variable adicional se asignará @NULL a este parámetro.

Puesto que la macro no inicializa la variable @FUMA, conviene hacerlo previamente a las condiciones lógicas con la instrucción:

```
COMPUTE @FUMA = $SYSMIS.
```

En la llamada a la macro !DR, el parámetro LV indica la lista de valores válidos, el parámetro MV recoge el código de valor *user missing* no recuperable, el parámetro LVL contiene las variables FN y FR implicadas en la condición lógica, y el parámetro FORMAT=1 indica que se efectúe el chequeo de formato. El parámetro VAR se asigna a 0 ya que es una variable que se registra sólo al inicio del estudio.

Algoritmo 4-3: Depuración de variables categóricas a través de una tabla de claves.

Asignación de los parámetros:

1. Asignar a V la lista de variables a depurar.
2. Asignar a TABLE el nombre (y ruta) de la tabla de claves.
3. Asignar a IDT el nombre del identificador de TABLE (Por defecto, V).
4. Asignar a MV, si procede, la lista de valores desconocidos no recuperables de V.
5. Asignar a MVr, si procede, la lista de valores desconocidos recuperables de V.
6. Asignar a VAR, si procede, el valor 0 (si el valor debe ser constante a lo largo de los seguimientos) ó HI, si se permite la variación a lo largo de los seguimientos. (Por defecto VAR=0).
7. Asignar a OUT, si procede, la variable que indica la baja del caso en siguientes seguimientos.
8. Asignar a XOut, si procede, el valor de OUT que implica la baja del caso en siguientes seguimientos.
9. Si se han leído los datos en ASCII y se desea detectar los errores de formato,
   Asignar FORMAT = 1.
10. Si hay variables implicadas en las condiciones lógicas previas,
    Asignar la lista de variables implicadas a LVL.
11. Si V está dentro de un salto, Asignar a VS la variable que determina el salto.
    11.1. Asignar a XS el valor de VS que determina saltar. (Por defecto XS=0).
    11.2. Asignar a MVS la lista de valores desconocidos no recuperables de VS. (Por defecto MVS=9).
    11.3. Asignar a MVSr la lista de valores desconocidos recuperables de VS.
    11.4. Asignar a VD el valor de V si se salta. (Por defecto VD=vacío).
12. Asignar a C las variables que forman el identificador.
13. Asignar a CN la variable que contiene el número de secuencia de los casos. (Por defecto CN= @casenum).
14. Asignar a L el valor 0 si se desea omitir el listado de incidencias, 1 si se desea listar errores y valores desconocidos ó 2, si se desea listar errores pero no valores desconocidos. (Por defecto L=1).

Proceso (para cada caso):

Sea V_i el elemento i de la lista V y $@V_i$ la correspondiente variable auxiliar numérica.

15. Si $@V_i >= 50$ y $LVL \neq \varnothing$ (error en condiciones lógicas previas), Ir al paso 28.
16. $@V_i$ = SYSMIS.
17. Si $V_i = \varnothing$ ó V_i = MV, $@V_i = 0$.
18. Si $V_i = \varnothing$ y $\bullet V_i \neq$ blanco, $@V_i = 2$.
19. Si $V_i \in$ IDT, $@V_i = -4$.
20. Si $V_i \notin$ IDT, $@V_i = 3$.
21. Si VAR $\neq 0$ y VAR $\neq$ vacío y $(V_j - V_{j+1}) >$ VAR, $@V_i = 11$.
22. Si OUT$\neq$vacío VAR_{OUT} = XOut y Registro $_{j+1} \neq$ vacío, $@V_i = 12$.
23. Si VD$\neq$vacío y VS = XS y V_i = VD, $@V_i = -3$.
24. Si VD=vacío y VS = XS y V_i = VD, $@V_i = -2$.
25. Si VS = XS y $V_i \neq$ VD, $@V_i = 1$.
26. Si VS $\neq$ XS y VS $\neq$ XnS y VS $\neq$ MV, $@V_i = 1$.
27. Si VS = MVS y $V_i \neq$ blanco, $@V_i = 1$.
28. Si L>=1 y $@V_i >= 0$:
    28.1. Si LVL=@NULL, LVL = $\varnothing$.
    28.2. Si FORMAT = 1, Listar: CN, C, VS, $@V_i$, $\bullet V_i$, LVL.
    28.3. Si FORMAT = 0, Listar: CN, C, VS, $@V_i$, V_i, LVL.

Documentación 4-2: Parámetros de la macro !DRKey.

```
DEPURACIÓN DE DATOS: LISTA DE VALORES CONTENIDOS EN UNA TABLA DICCIONARIO
Creación 08.06.2000 Última revisión 07.07.2003
(c) A.Bonillo & JM. Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
!DRKey  V= Lista de las variables a verificar
     /IDT= Lista de los identificadores en el DICCIONARIO (por defecto, V)
   /TABLE= Nombre del archivo .SAV con el DICCIONARIO de códigos
    [/MV]= Lista de valores user missing (no recuperables) de las variables V
   [/MVr]= Lista de valores user missing (recuperables) de las variables V
[/FORMAT]= Verificación de formato (1=Sí, 0=No)      (por defecto,  0)
   [/LVL]= Lista de otras variables a listar
               La variable @V NO se inicializa a SYSMIS cuando se especifica
                nombres de variables o la palabra clave @NULL
               Si se omite el parámetro la variable @V se inicializa a SYSMIS
[/RENAME]= Lista de variables a renombrar            (por defecto ninguna)
  [/DROP]= Lista de variables a no añadir            (por defecto ninguna)
       /C= Lista de variables identificadoras del sujeto
  [/INDX]= Variable de índice
      /CN= Variable identificadora de la secuencia de los sujetos
                                                     (por defecto,@casenum)
      /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)
                                                     (por defecto,  1)
------------------------------------------------------------------------------
Parámetros para variables dentro de un salto:
    [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)
   [/XnS]= Lista de valores de VS que indican NO saltar (en formato SPSS)
                                                     (por defecto,  1)
    [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)
                                                     (por defecto,  0)
    [/VD]= Valor deducible si VS indica saltar       (en formato SPSS)
                                                     (por defecto, vacío)
   [/MVS]= Lista de valores user missing (no recuperables) de la variable VS
  [/MVSr]= Lista de valores user missing (recuperables) de la variable VS
------------------------------------------------------------------------------
Parámetros para variables implicadas en seguimientos:
   [/VAR]= Valor, o porcentaje, máximo de variación entre seguimientos
                                                     (en formato SPSS)
                                                     (por defecto,  0)
   [/OUT]= Variable indicadora de la baja del registro en futuros seguimientos
  [/XOUT]= Valor de OUT que indica la baja del registro en futuros seguimientos
------------------------------------------------------------------------------

Ejemplos de llamada:
      !DRKey V=CIE /TABLE="CIE9.SAV" /DROP=DES /MVr=" " /C=h caso.

      !DRKey V=CIE /TABLE="CIE9.SAV" /DROP=DES /MVr=" " /C=h caso.
```

En la parte inferior del Listado 4-11 aparecen las 7 incidencias detectadas al depurar la variable FUMA. El caso 133 tiene un código no válido (valor 7) en su primer registro y dos valores desconocidos en el segundo y tercero; el caso 14 tiene valor desconocido, el caso 16 tiene un error de formato (se ha grabado una “O”) y el caso 17 es menor de 11 años y afirma que fuma.

4.7.2.4 Depuración de variables categóricas a través de una tabla de claves

Cuando una variable categórica posee un gran número de valores válidos no consecutivos es recomendable depurarla a través de una tabla de claves (diccionario)

grabado en un archivo externo. El Algoritmo 4-3 muestra en pseudo-código el proceso para depurar variables categóricas a través de un diccionario.

Listado 4-12: Depuración de la variable categórica CIE.

```
*Condición @CIE=50: El código principal debe estar comprendido entre 390 y 459.
COMPUTE @cie=$SYSMIS.
IF NOT(RANGE(NUMBER(SUBSTR(cie,1,3),F3),390,459)) @CIE=50.

*Comprobación sistemática de la variable CIE.
!DRKey V=cie /TABLE='C:\...\Escritorio\CIE9.SAV' /DROP=des
  /MVr=' ' /VAR=HI /LVL=@NULL /C=h caso  /INDX=fr.
```

```
@CASENUM H CASO         FR @CIE                            CIE
       4  A   94 08.06.1993  0: Missing Rec
       9  A   14           .  0: Missing Rec
      10  A   14 13.07.1993  0: Missing Rec
      18  B   10 05.11.1993 50: Err Cond                   030.3
      20  B  103 29.11.1993  0: Missing Rec

Number of cases read:  5    Number of cases listed:  5
```

Existen varios motivos para realizar la depuración de una variable de este modo. En primer lugar, las categorías a depurar pueden ser tantas que escribirlas en una llamada a la macro es una estrategia poco eficiente, difícilmente revisable e, incluso, puede rebasar las limitaciones del software utilizado; en otras palabras, si hay un gran número de categorías el uso de llamadas conlleva una probabilidad de error tan alta que el chequeo requiere a su vez una validación. En segundo lugar, si los valores corresponden a clasificaciones universales, los códigos se acostumbran a actualizar periódicamente. Puesto que el organismo responsable de la codificación suministra habitualmente tablas ya digitalizadas el usuario se evita la tarea tediosa de introducir los valores.

Un ejemplo de clasificación universal de amplio uso en Ciencias de Salud es la CIE-9, que consta de 14002 códigos de patologías, 4280 de procedimientos médicos y 1225 de causas externas. Las tablas con estos códigos son fácilmente localizables tanto en los servicios de salud públicos como en páginas web. Otros ejemplos de tablas de claves se encuentra en el código postal, suministrado por Correos.

La macro !DRKey, cuyos parámetros se incluyen en la Documentación 4-2, permite comprobar que a cada valor de la variable a depurar le corresponde un registro en la tabla diccionario utilizada para la depuración.

La comprobación sistemática de la variable CIE se efectúa a través de la tabla de claves CIE9.SAV que contiene todos los códigos CIE9 válidos. Sin embargo, en este estudio esta variable tiene la particularidad de que sólo puede contener códigos diagnósticos comprendidos entre el 390.0 y el 459.9 (patologías del aparato circulatorio). Esta restricción se expresará como una condición lógica previa a la comprobación sistemática con la macro !DRKey. El proceso es análogo al expuesto para FUMA.

El Listado 4-12 recoge la secuencia que permite detectar las incidencias en la variable CIE. La condición lógica se comprueba asignando un código 50 a los sujetos con código diagnóstico diferente de los permitidos.

En la llamada a la macro !DRKey, el parámetro TABLE contiene el nombre y la ruta de la tabla diccionario de códigos CIE9. El proceso de depuración se basa en añadir

a los datos a depurar las variables contenidas en esta tabla de claves; el parámetro optativo DROP permite especificar las variables de la tabla diccionario que no se desean incorporar al archivo de datos. El parámetro LVL=@NULL impide que la macro inicialice la variable @CIE a SYSMIS. El parámetro MVr=“ ” indica que el valor *user missing* recuperable para esta variable es el blanco. En la llamada no se ha incluido el parámetro IDT=CIE (nombre de la variable de la tabla de claves que contiene los códigos CIE9) porque coincide con el nombre de la variable a depurar. Al parámetro VAR se le ha asignado el valor HI, ya que esta variable puede variar a lo largo de los seguimientos.

En la parte inferior del Listado 4-12 se listan las 5 incidencias detectadas al depurar la variable CIE. Los casos 94, 14 y 103 tienen valor desconocido. El caso 10, tiene un código diagnóstico que no corresponde a una patología del aparato circulatorio; se trata del código de la “Lepra dudosa” (030.3). Este listado omite el registro 12 porque corresponde a un duplicado que debe ser eliminado (véase apartado 4.8.1).

Documentación 4-3: Parámetros de la macro !DRF.

```
DEPURACIÓN DE DATOS: RANGO DE CAMPOS FECHA EXPRESADO EN FECHAS
Creación 30.11.1998 Última revisión 07.07.2003
(c) A.Bonillo & JM. Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
    !DRF V= Lista de campos fecha a verificar
     /FI]= Fecha inicial: día, mes, año
     /FS]= Fecha final  : día, mes, año
 /FORMAT]= Verificación de formato (1=Sí, 0=No)      (por defecto,  0)
    /LVL]= Lista de otras variables a listar
               La variable @V NO se inicializa a SYSMIS cuando se especifica
                nombres de variables o la palabra clave @NULL
               Si se omite el parámetro la variable @V se inicializa a SYSMIS
        /C= Nombre variable identificadora sujeto
   /INDX]= Variable de índice
       /CN= Variable identificadora de la secuencia de los sujetos
                                                    (por defecto,@casenum)
      /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)
                                                   (por defecto,  1)
------------------------------------------------------------------------------
Parámetros para variables dentro de un salto:
     /VS]= Nombre variable de salto (sólo si V está dentro de un salto)
    /XnS]= Lista de valores de VS que indican NO saltar (en formato SPSS)
                                                    (por defecto,  1)
     /XS]= Lista de valores de VS que indican saltar (en formato SPSS)
                                                    (por defecto,  0)
    /MVS]= Lista de valores user missing (no recuperables) de la variable VS
   /MVSr]= Lista de valores user missing (recuperables) de la variable VS
------------------------------------------------------------------------------
Parámetros para variables implicadas en seguimientos:
   /VAR]= Valor, en días, máximo de variación entre seguimientos
                                                   (en formato SPSS)
                                                   (por defecto,  0)
   /OUT]= Variable indicadora de la baja del registro en futuros seguimientos
  /XOUT]= Valor de OUT que indica la baja del registro en futuros seguimientos
------------------------------------------------------------------------------

Ejemplos de llamada:
       !DRF V=fr /FI=1,4,1993/ FS=30,12,1993 /FORMAT=1 /C=h caso.
```

4.7.3 Depuración de campos fecha

La fecha de un suceso no es más que un punto en el tiempo. Muchos programas estadísticos guardan internamente las fechas como el número de unidades de tiempo transcurridas desde una fecha de referencia. En SPSS estas unidades son segundos y la fecha de referencia es el 15.10.1582 (fecha del cambio de calendario juliano a gregoriano), y en SAS es el 01.01.1960. Los campos fecha no suelen tratarse estadísticamente, sino que se utilizan para generar el tiempo transcurrido entre dos eventos.

Existen dos formas de depurar una fecha: comprobar que sus valores están comprendidos dentro de un rango (es decir, dentro de un determinado intervalo de tiempo) o verificar el tiempo transcurrido respecto a otra fecha de referencia.

4.7.3.1 Depuración de fechas mediante un rango

Para depurar una fecha a través de un rango es necesario verificar que sus valores: a) no son desconocidos; b) no contienen errores de formato; c) están comprendidos entre dos fechas determinadas; d) si están dentro de un salto son congruentes con él; y f) si están han sido recogidas durante varios seguimientos que sus valores son consistente entre ellos, y g) no tienen incompatibilidades lógicas con otras variables.

El Algoritmo 4-4 muestra en pseudo-código el proceso para depurar una fecha a través de un rango.

La macro !DRF, cuyos parámetros se incluyen en la Documentación 4-3, permite realizar este tipo de depuración.

En el Listado 4-13 se presenta la llamada de la macro !DRF que permite depurar el campo FR (fecha respuesta). En este estudio FR sólo puede tener valores comprendidos entre el 1.4.1993 y el 30.12.1993.

El parámetro V recoge el nombre del campo a depurar. Los parámetros FI y FS recogen, respectivamente, la fecha inferior y superior del rango de fechas válidas; estos valores se han especificado en formato SPSS, separando con comas el día, el mes y el año. Puesto que se trata de una fecha que no debe ser constante, se ha especificado el VAR=HI; sólo serán listadas aquellas fechas que se hallen fuera del rango válido. Se ha especificado que se efectúe la depuración de formato.

En la parte inferior del Listado 4-13 se listan los casos con incidencias en la variable FR. El sujeto 14 presenta un error de formato y los casos 17 y 82 valores fuera del intervalo de fechas válidas.

Listado 4-13: Depuración del campo FR (fecha de respuesta).

```
!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /VAR=HI /FORMAT=1 /C=h caso /INDX=fr.
```

```
@CASENUM H CASO @FR                                              ·FR
      9  A   14  2: Err Formato                                  13.17.1993
     13  B   17  3: Fuera Rango                                  07.09.1994
     14  B   82  3: Fuera Rango                                  20.11.1962

Number of cases read:  3
```

Algoritmo 4-4: Depuración de una fecha mediante un rango.

Asignación de los parámetros:

1. Asignar a V la lista de variables a depurar.
2. Asignar a FI el límite inferior del rango válido de V.
3. Asignar a FS el límite superior del rango válido de V.
4. Asignar a VAR, si procede, el valor absoluto en días, máximo de variación entre dos seguimientos consecutivos. (Por defecto VAR=0).
5. Asignar a OUT, si procede, la variable que indica la baja del caso en siguientes seguimientos.
6. Asignar a XOut, si procede, el valor de OUT que implica la baja del caso en siguientes seguimientos.
7. Si se han leído los datos en ASCII y se desea detectar los errores de formato,
   Asignar FORMAT = 1.
8. Si hay variables implicadas en las condiciones lógicas previas,
   Asignar la lista de variables implicadas a LVL.
9. Asignar a ND, si procede, el número de decimales de V
10. Si V está dentro de un salto, Asignar a VS la variable que determina el salto.
    10.1. Asignar a XS el valor de VS que determina saltar. (Por defecto XS=0).
    10.2. Asignar a MVS la lista de valores desconocido no recuperables de VS. (Por defecto MVS=9).
    10.3. Asignar a MVSr la lista de valores desconocido recuperables de VS.
11. Asignar a C las variables que forman el identificador.
12. Asignar a CN la variable que contiene el número de secuencia de los casos. (Por defecto CN= @casenum).
13. Asignar a L el valor 0 si se desea omitir el listado de incidencias, 1 si se desea listar errores y valores desconocidos ó 2, si se desea listar errores pero no valores desconocidos. (Por defecto L=1).

Proceso (para cada caso):

Sea V_i el elemento i de la lista V y $@V_i$ la correspondiente variable auxiliar numérica.

14. Si $@V_i >= 50$ y $LVL \neq \varnothing$ (error en condiciones lógicas previas), Ir al paso 25.
15. $@V_i$ = SYSMIS.
16. Si $V_i = \varnothing$, $@V_i = 0$.
17. Si $V_i = \varnothing$ y $\bullet V_i \neq$ blanco, $@V_i = 2$.
18. Si $V_i \in (FI \div FS)$, $@V_i = -4$.
19. Si $V_i \notin (FI \div FS)$, $@V_i = 3$.
20. Si $VAR \neq 0$ y $VAR \neq$ vacío y $(V_j - V_{j+1}) > VAR$, $@V_i = 11$.
21. Si VAR= 0 y $(V_j \neq V_{j+1}) \neq 0$, $@V_i = 10$.
22. Si OUT≠vacío y VAR_{OUT} = XOut y Registro $_{j+1} \neq$ vacío, $@V_i = 12$.
23. Si $VS \neq XS$ y $VS \neq XnS$ y $VS \neq MV$, $@V_i = 1$.
24. Si VS = MVS y $V_i \neq$ vacía, $@V_i = 1$.
25. Si L>=1 y $@V_i >= 0$:
    25.1. Si LVL=@NULL, LVL = ∅.
    25.2. Si FORMAT = 1, Listar: CN, C, VS, $@V_i$, $\bullet V_i$, LVL.
    25.3. Si FORMAT = 0, Listar: CN, C, VS, $@V_i$, V_i, LVL.

Listado 4-14: Depuración del campo fecha FN.

```
!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /VAR=0 /FORMAT=1 /C=h caso
  /INDX=fr.

@CASENUM H CASO         FR @FN                                      ·FN               FR

      8  A  133 14.12.1993 10: Err Cnstant (Val. ant:25.05.1949)            14.12.1993
     12  B   16 14.12.1993  2: Err Formato                        16.10.     14.12.1993
     14  B   82 20.11.1962  4: Dif fuera Rango                    13.05.1993 20.11.1962
     15  B   82 01.08.1993 10: Err Cnstant (Val. ant:13.05.1993)  20.11.1962 01.08.1993

Number of cases read:  4    Number of cases listed:  4
```

4.7.3.2 Depuración de fechas mediante un intervalo de tiempo transcurrido

En ocasiones, la depuración de una fecha se debe efectuar comprobando el intervalo de tiempo que transcurre respecto a otra fecha de referencia. Esta comprobación tiene interés en campos fecha cuyos valores pueden ser muy distintos para los sujetos de un estudio, por ejemplo la fecha de nacimiento. Este campo puede depurarse fácilmente si se dispone de la fecha de respuesta y también utilizando la fecha actual.

El Algoritmo 4-5 muestra en pseudo-código el proceso para depurar un intervalo entre fechas. La Documentación 4-4 describe los parámetros de la macro !DDF que permite efectuar este tipo de depuración.

En el archivo de prueba la fecha de nacimiento (FN) puede depurarse comprobando que el tiempo transcurrido entre dicho valor y la fecha de respuesta es válido, lo cual en este ejemplo implica que los sujetos tengan entre 6 y 45 años y que la fecha debe permanecer constante a lo largo de los seguimientos. La parte superior del Listado 4-14 presenta la llamada de la macro !DDF que permite realizar esta comprobación.

El parámetro V contiene el nombre del campo fecha a depurar. El parámetro D contiene la diferencia (positiva) entre los campos fecha a depurar y fecha de referencia. Los parámetros MIN y MAX recogen el límite inferior y superior, en días, de esta diferencia de fechas. Puesto que se trata de un valor que debe ser constante, se especifica VAR=0.

En la parte inferior del Listado 4-14 se listan los casos con incidencias en FN. El caso 16 tiene un error de formato (le falta el año) y el caso 82 tiene un valor fuera de rango en el registro 14 (la fecha de respuesta es anterior a la fecha de nacimiento) y la fecha de nacimiento del registro siguiente es distinto. El caso 133 tiene valor desconocido en un seguimiento.

Es habitual que los campos de una lista de fechas contengan valores secuenciales (Cody, 1999). En este caso la macro se debe ejecutar repetidas veces tomando como referencia la fecha inmediatamente anterior. Por ejemplo, si en un estudio se dispone de la fecha de alta médica (FA), se debe comprobar que es posterior a la fecha de respuesta (FR), y esta última posterior a la de nacimiento (FN). Para ello, se podría especificar una condición lógica basada en FN<FR<FA que diera un error en los sujetos en los que el conjunto de fechas no estuviera ordenado. Sin embargo este tipo de comprobaciones globales atenta contra la lógica propuesta en este trabajo. En nuestro caso este chequeo

formaría parte de la depuración sistemática variable a variable y se efectuaría de acuerdo al siguiente esquema:

1. La fecha de nacimiento (FN) se comprueba utilizando la fecha de respuesta como referente, y se podría especificar un rango de edades válido.
2. La fecha de respuesta (FR) se comprueba, por ejemplo, a partir del rango de fechas en el que se han recogido los datos.
3. La fecha de alta (FA) se comprueba usando la fecha de respuesta como referente.

Algoritmo 4-5: Depuración de un intervalo transcurrido entre fechas.

Asignación de los parámetros:

1. Asignar a V la lista de variables a depurar.
2. Asignar a D el sentido de la diferencia entre fechas.
3. Asignar a MIN el límite inferior del rango válido de V.
4. Asignar a MAX el límite superior del rango válido de V.
5. Asignar a VAR, si procede, el valor absoluto en días, máximo de variación entre dos seguimientos consecutivos. (Por defecto VAR=0).
6. Asignar a OUT, si procede, la variable que indica la baja del caso en siguientes seguimientos.
7. Asignar a XOut, si procede, el valor de OUT que implica la baja del caso en siguientes seguimientos.
8. Si se han leído los datos en ASCII y se desea detectar los errores de formato, FORMAT = 1.
9. Si hay variables implicadas en las condiciones lógicas previas,
   Asignar la lista de variables implicadas a LVL.
10. Si V está dentro de un salto, Asignar a VS la variable que determina el salto.
    10.1. Asignar a XS el valor de VS que determina saltar. (Por defecto XS=0).
    10.2. Asignar a XnS el valor de VS que determina no saltar. (Por defecto XnS=1).
    10.3. Asignar a MVS la lista de valores desconocidos no recuperables de VS. (Por defecto MVS=9).
    10.4. Asignar a MVSr la lista de valores desconocidos recuperables de VS.
11. Asignar a C las variables que forman el identificador.
12. Asignar a CN la variable que contiene el número de secuencia de los casos. (Por defecto CN= @casenum).
13. Asignar a L el valor 0 si se desea omitir el listado de incidencias, 1 si se desea listar errores y valores desconocidos ó 2, si se desea listar errores pero no valores desconocidos. (Por defecto L=1).

Proceso (para cada caso):

Sea V_i el elemento i de la lista V, y $@V_i$ la correspondiente variable auxiliar numérica.

14. Si $@V_i >= 50$ y $LVL \neq \varnothing$ (error en condiciones lógicas previas), Ir al paso 23.
15. $@V_i$ = SYSMIS.
16. Si $V_i = \varnothing$, $@V_i = 0$.
17. Si $V_i = \varnothing$ y $\bullet V_i \neq$ blanco, $@V_i = 2$.
18. Si $V_i \in (MIN \div MAX)$, $@V_i = -4$.
19. Si $V_i \notin (MIN \div MAX)$, $@V_i = 3$.
20. Si OUT≠vacío y VAR_{OUT} = XOut y Registro $_{j+1} \neq$ vacío, $@V_i = 12$.
21. Si $VS \neq XS$ y $VS \neq XnS$ y $VS \neq MV$, $@V_i = 1$.
22. Si VS = MVS y $V_i \neq$ vacía, $@V_i = 1$.
23. Si L>=1 y $@V_i >= 0$:
    23.1. Si LVL=@NULL, $LVL = \varnothing$.
    23.2. Si FORMAT = 1, Listar: CN, C, VS, $@V_i$, $\bullet V_i$, LVL.
    23.3. Si FORMAT = 0, Listar: CN, C, VS, $@V_i$, V_i, LVL.

Documentación 4-4: Parámetros de la macro !DDF.

```
DEPURACIÓN DE DATOS: RANGO DE CAMPOS FECHA EXPRESADO EN DÍAS
Creación 30.11.1998 Última revisión 07.07.2003
(c) A.Bonillo & JM. Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
   !DDF V= Lista de campos fecha (inicial)
       /D= Diferencia entre los campos fecha a depurar y fecha de referencia
   [/MIN]= Valor mínimo del tiempo transcurrido (días)
   [/MAX]= Valor máximo del tiempo transcurrido (días)
[/FORMAT]= Verificación de formato (1=Sí, 0=No)      (por defecto,  0)
   [/LVL]= Lista de otras variables a listar
               La variable @V NO se inicializa a SYSMIS cuando se especifica
                nombres de variables o la palabra clave @NULL
               Si se omite el parámetro la variable @V se inicializa a SYSMIS
       /C= Nombre variable identificadora sujeto
  [/INDX]= Variable de índice
      /CN= Variable identificadora de la secuencia de los sujetos
                                                    (por defecto, casenum)
      /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)
                                                    (por defecto,  1)
------------------------------------------------------------------------------
Parámetros para variables dentro de un salto:
     [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)
    [/XnS]= Lista de valores de VS que indican NO saltar (en formato SPSS)
                                                     (por defecto,  1)
     [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)
                                                     (por defecto,  0)
    [/MVS]= Lista de valores user missing (no recuperables) de la variable VS
   [/MVSr]= Lista de valores user missing (recuperables) de la variable VS
------------------------------------------------------------------------------
Parámetros para variables implicadas en seguimientos:
   [/VAR]= Valor, o porcentaje, máximo de variación entre seguimientos
                                                    (en formato SPSS)
                                                    (por defecto,  0)
   [/OUT]= Variable indicadora de la baja del registro en futuros seguimientos
  [/XOUT]=  Valor de OUT que indica la baja del registro en futuros seguimientos
------------------------------------------------------------------------------

Ejemplos de llamada:
!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /VAR=0 /C=h caso.
!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /FORMAT=1 /VS=fuma /C=h caso.
```

4.7.4 Depuración de variables contenidas dentro de un salto

La depuración de variables afectadas por saltos posee algunas peculiaridades. En primer lugar, se debe tener presente que cuando se utilizan protocolos de recogida de datos estructurados en forma de árbol lógico las variables de las ramas no siempre se registran ya que pueden ser saltadas. Las variables que determinan la edición de las ramas se denominan “variables de salto” (o también de filtro o cribado). La depuración de estas variables no presenta ninguna problemática adicional ya que deben registrarse obligatoriamente para todos los sujetos.

Si se produce el salto, las variables contenidas en él pueden tomar valor “no aplicable”, valor desconocido o un valor concreto denominado “deducible” (Granero, 1999). El valor específico que se debe grabar depende de la causa que ha originado el salto. Esto implica que la depuración de una variable incluida dentro de un salto comporta el análisis de las posibles razones que pueden originar el salto y la comprobación de que los valores capturados son congruentes con dichas causas.

La Figura 4-3 ilustra este planteamiento para el ejemplo de salto contenido en el archivo de prueba. En esta figura se recogen los distintos valores que puede contener la variable de salto (FUMA), si el salto debe o no efectuarse en presencia de dichos valores y los códigos correctos permitidos para las variables contenidas en él (TAB y TIPTAB).

Si el sujeto es fumador (FUMA=1) el salto no se debe efectuar, TAB debe contener cualquier entero comprendido entre 1 y 80 (rango de valores aceptables), y TIPTAB debe contener cualquiera de los códigos válidos NE/RU/NR.

Si el sujeto no fuma (FUMA=0) el salto se debe efectuar y sólo tiene sentido que la variable TAB contenga el valor deducible 0 y TIPTAB valor no aplicable (vacío). El concepto de valor "deducible" siempre está asociado a valores de variables que están dentro de un salto y no debe confundirse con el concepto de campo calculado.

Si el sujeto no quiere contestar si es fumador (FUMA=9) el salto también se debe efectuar ya que no es pertinente preguntar ni el consumo diario de tabaco ni el tipo de tabaco. En esta situación, tanto la variable TAB como TIPTAB deben contener valor desconocido y por consiguiente deben quedar vacías. Es importante destacar que esta tipología de valor desconocido no denota un error de captura y es de carácter no recuperable porque el sujeto ya ha contestado "que no desea contestar".

Finalmente, cuando FUMA está vacía (FUMA=SYSMIS) la condición de salto es "no evaluable", y por lo tanto cualquier contenido en las variables TAB y TIPTAB debe destacarse como una incidencia.

La Tabla 4-2 amplía todas las posibilidades que se hubieran podido plantear en el ejemplo de la Figura 4-3 para las variables TAB y TIPTAB suponiendo que ambas hubieran permitido el código de *user missing* no sabe-no contesta. Por filas y sombreado aparecen todos los valores posibles que puede tomar la variable de salto. Por columnas y sombreado los valores posibles que podrían tener TAB y TIPTAB. En cada casilla aparecen los códigos resultantes de la depuración, y se han marcado en negrita aquellas situaciones correspondientes a valores consistentes que no requieren revisión. El significado de los códigos se encuentra en Tabla 4-1.

Otra característica de la comprobación de secuencias de saltos tiene que ver con las estructuras anidadas. En el caso de saltos anidados nuestra propuesta consiste en depurar sucesivamente cada variable respecto al salto que la determina.

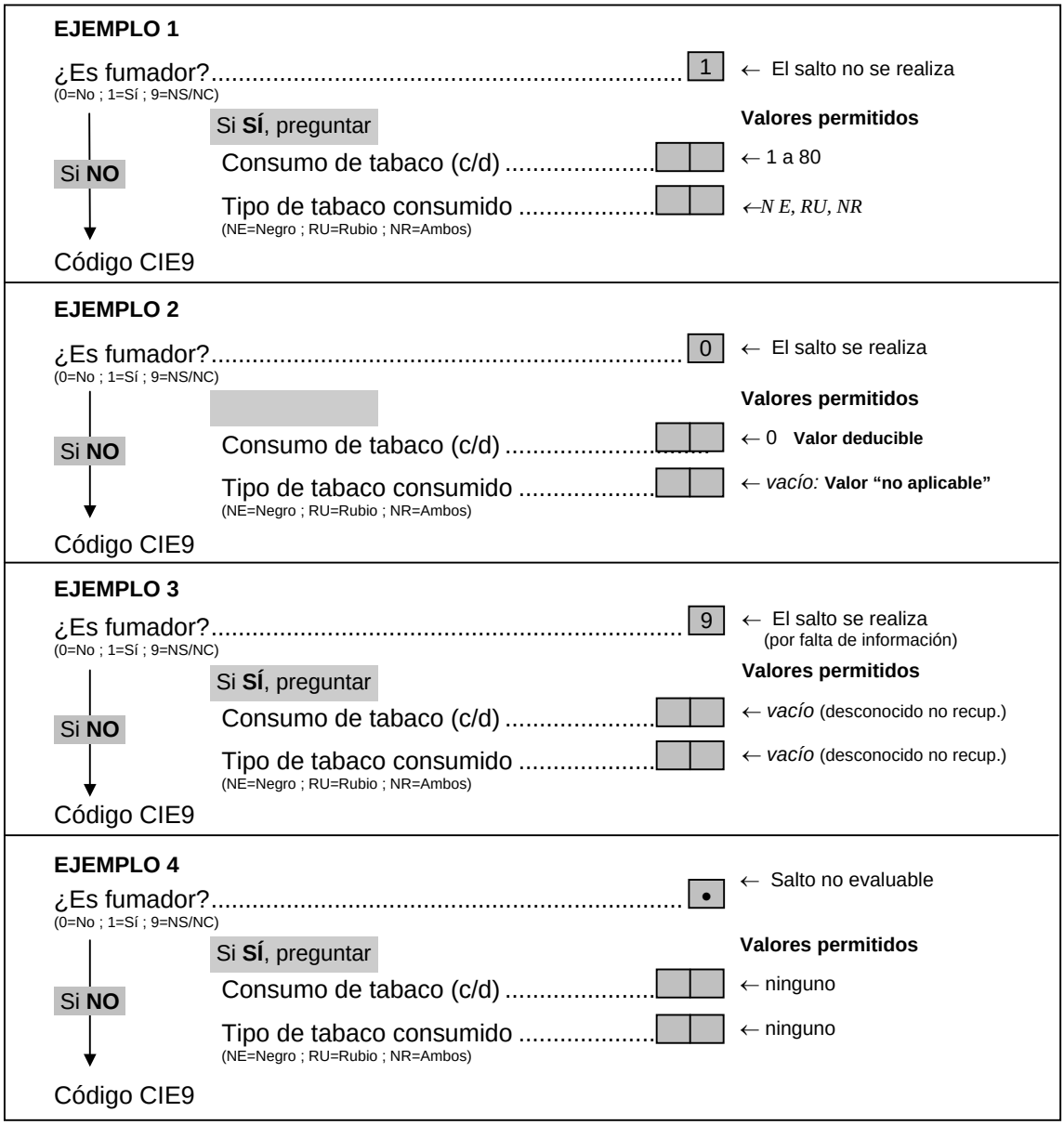


Figura 4-3: Valores permitidos de TAB y TIPTAB en función del valor de FUMA.

Tabla 4-2: Códigos resultantes de la depuración de TAB y TIPTAB.

	TAB (c/d)				
FUMA	0	1 a 80	Fuera rango	•	99 (NS/NC)
0 (No)	**–3**	1	1	1	1
1 (Sí)	1	**–4**	3	1	**–1**
9 (NS/NC)	1	1	1	**–1**	1
•	1	1	1	0	1

	TIPTAB			
FUMA	Vacío	NE,RU,NR	Código no válido	NS (NS/NC)
0 (No)	**–2**	1	1	1
1 (Sí)	1	**–4**	3	**–1**
9 (NS/NC)	**–1**	1	1	1
•	0	1	1	1

4.7.4.1 Depuración de una variable cuantitativa dentro de un salto

El Listado 4-15 muestra cómo detectar las incidencias en la variable TAB a través de la macro !DR. En la llamada se especifica que el rango de valores válidos oscila entre 1 y 80 (LV=1 thru 80). El parámetro ND=0 indica que el registro se ha efectuado con valores enteros (sin decimales). La variable de salto (FUMA) se transfiere a través del parámetro VS. El parámetro VD recoge el valor deducible para TAB, en este caso 0, cuando FUMA=0 (salto ejecutado para los sujetos no fumadores). El parámetro MVS recoge el código de la variable de salto que representa un valor desconocido no recuperable. El parámetro VAR toma valor 0, pues es una variable recogida al inicio del estudio. El parámetro XS toma valor 0, aunque podría haber sido omitido porque su valor por defecto ya es 0.

La parte inferior del Listado 4-15 muestra las incidencias detectadas en TAB. Los casos 133 y 14 tienen valor desconocido en la variable TAB y el caso 94 del hospital B un valor fuera de rango (-1). El resto de registro tienen valores para TAB que son inconsistentes con el contenido de la variable de salto FUMA. Por ejemplo, el caso 94 del hospital A corresponde a un no fumador (FUMA=0) cuyo supuesto consumo de tabaco es 15; el caso 12 tiene código "no sabe" en FUMA pero valor 0 en TAB; y el caso 133 se ha detectado porque tiene un valor fuera de rango en la variable de salto.

En la parte inferior del Listado 4-15 se aprecia que el caso 94 del hospital B tiene un valor fuera de rango y el caso 14 tiene un valor desconocido. El resto de casos listados tienen valores de TAB inconsistentes con el valor del salto, FUMA.

4.7.4.2 Depuración de una variable categórica dentro de un salto

La depuración de la variable categórica TIPTAB incluida dentro de un salto es semejante a la secuencia presentada para la variable TAB. La principal diferencia es que TIPTAB no tiene un código de valor deducible. En la Figura 4-3 aparecen los códigos válidos de esta variable en función del contenido de la variable de salto FUMA y en la Tabla 4-2 todas las situaciones posibles que pueden resultar del chequeo.

Listado 4-15: Detección de incidencias en la variable TAB, contenida en un salto.

```
!DR V=tab /LV=1 thru 80 /ND=0 /VS=fuma /MVS=9 /XS=0 /VD=0 /VAR=0 /FORMAT=1
   /C=h caso  /INDX=fr.
```

```
@CASENUM H CASO         FR FUMA @TAB                              ·TAB

      4  A   94 08.06.1993   0   1: Incons Salt                  15
      5  A   12 01.11.1993   9   1: Incons Salt                   0
      6  A  133 21.06.1993   7   1: Incons Salt
      7  A  133 14.08.1993   .   0: Missing Rec
      8  A  133 14.12.1993   .   0: Missing Rec
      9  A   14          .   .   0: Missing Rec
     10  A   14 13.07.1993   .   0: Missing Rec
     12  B   16 14.12.1993   .   1: Incons Salt                   0
     13  B   17 07.09.1994   1   1: Incons Salt                   0
     17  B   94 22.04.1993   1   3: Fuera Rango                  -1

Number of cases read:  10    Number of cases listed:  10
```

Listado 4-16: Detección de incidencias en la variable TIPTAB, contenida en un salto.

```
!DR V=tiptab /LV='NE','RU','NR' /MVr=' ' /VS=fuma /XS=0 /VAR=0 /MVS=9
   /C=h caso  /INDX=fr.
```

```
@CASENUM H CASO         FR FUMA @TIPTAB                          TIPTAB

      6  A  133 21.06.1993   7   1: Incons Salt
      7  A  133 14.08.1993   .   0: Missing Rec
      8  A  133 14.12.1993   .   0: Missing Rec
      9  A   14          .   .   1: Incons Salt                   RU
     10  A   14 13.07.1993   .   1: Incons Salt                   RU
     12  B   16 14.12.1993   .   0: Missing Rec
     13  B   17 07.09.1994   1   1: Incons Salt
     18  B   10 05.11.1993   1   3: Fuera Rango                   N

Number of cases read:  8    Number of cases listed:  8
```

El Listado 4-16 contiene la llamada a la macro !DR que permite depurar la variable TIPTAB. El parámetro LV recoge la lista de valores válidos de la variable. En el parámetro MVr se indica el código de valor desconocido recuperable (“blanco”). El parámetro VS recoge el nombre de la variable de salto. El parámetro MVS recoge el código de la variable de salto que representa un valor desconocido no recuperable. Los parámetros VD y XS son omitidos puesto que su valor de defecto (“vacío” y 0, respectivamente) coincide con los códigos utilizados en este test.

La parte inferior del Listado 4-16 contiene todas las incidencias detectadas. El caso 10 tiene un valor no válido (TIPTAB=”N”) y los casos 133 y 16 tienen valor desconocido. El resto de registros tienen valores en TIPTAB inconsistentes con el valor de la variable de salto FUMA. Por ejemplo, el caso 14 no tiene registrado si es fumador pero sí una marca de tabaco (“RU”), y el caso 17 es un supuesto fumador para el que no está registrado el tipo de tabaco.

4.8 DEPURACIÓN DE IDENTIFICADORES

En la secuencia del proceso de depuración, la corrección de los identificadores debe preceder a la del resto de variables. Sin embargo, esta etapa se expone posteriormente porque las variables que forman parte del identificador también requieren los chequeos que hemos presentado.

En concreto, la depuración de identificadores consiste en comprobar los tres siguientes criterios (Cody, 1999): a) los registros están unívocamente identificados, esto es, no existen registros duplicados ni con valores desconocidos en las variables identificadoras; b) los registros aparecen un número determinado de ocasiones (control aplicable cuando los archivos contienen más de una réplica por caso); y c) si el archivo está relacionado, su identificador está en la tabla principal.

Las tres primeras comprobaciones básicas se ilustran en la Figura 4-4, que contiene una tabla principal con el identificador de sujeto (CASO) y una tabla relacionada en la que cada sujeto debe tener obligatoriamente tres réplicas (identificadas con la variable SEC). El identificador de esta última tabla está formado por la concatenación de CASO+SEC.

La comprobación del primer criterio comporta crear la variable auxiliar N_IDT@ que contenga el número de veces que el identificador del caso está repetido. Por

ejemplo, la variable N_IDT@ toma valor 2 para los dos primeros registros del caso 137 porque tienen el mismo identificador.

La comprobación del segundo criterio comporta crear la variable auxiliar R_IDT@ que contenga el número réplicas por caso. Esta variable toma valor 2 en el caso 138 porque le falta la segunda secuencia.

La comprobación del tercer criterio conlleva crear la variable auxiliar I_IDT@, que toma valor –1 si el identificador del sujeto está incluido en la tabla principal y valor 3 si no lo está. Por ejemplo, las tres secuencias del caso 137 están marcadas con I_IDT@=3 porque este sujeto no existe en la tabla principal.

A estos tres criterios, nuestra propuesta añade un cuarto. Cuando una variable indique que el registro debe causar baja en seguimientos posteriores, debe comprobarse que efectivamente es así. Pese a que este supuesto puede comprobarse a través de la depuración de cualquier variable del estudio, es recomendable hacerlo durante la depuración del identificador.

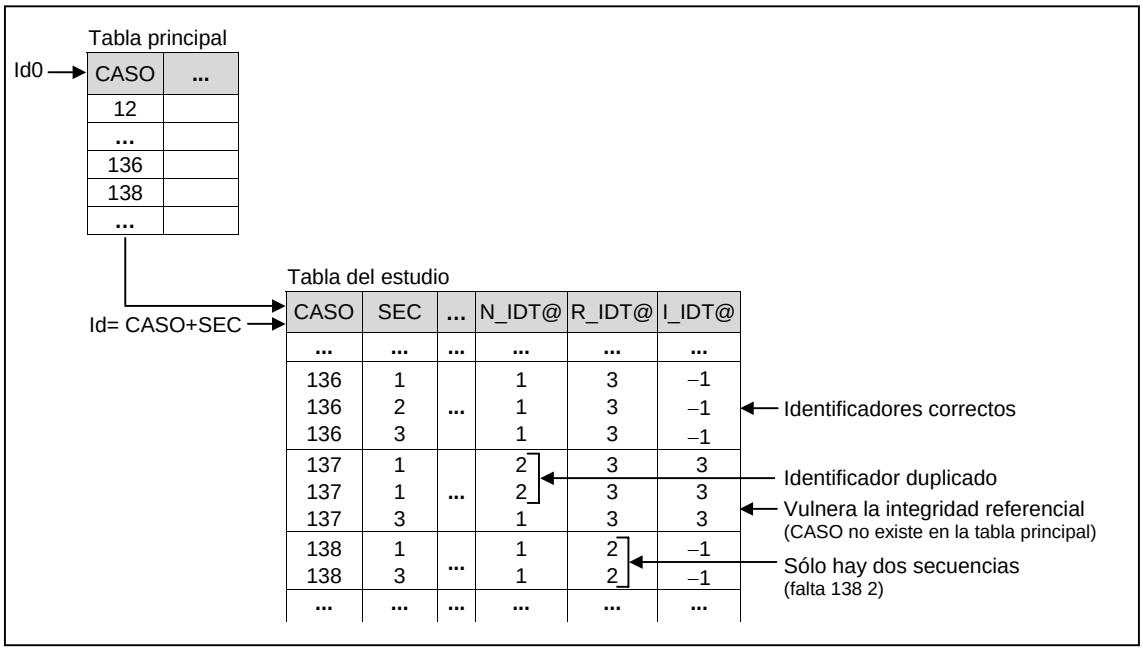


Figura 4-4: Comprobaciones básicas de identificadores.

Algoritmo 4-6: Depuración de identificadores: duplicados y réplicas.

Asignación de los parámetros:

1. Asignar a V la lista de variables que forman el identificador.
2. Asignar a R el número de registros que deben tener igual identificador. (Por defecto R=1).
3. Asignar a CN la variable que contiene el número de secuencia de los casos. (Por defecto CN= casenum).
4. Asignar a L el valor 0 si se desea omitir el listado de incidencias. (Por defecto L=1).
5. Asignar a OUT, si procede, la variable que indica la baja del caso en siguientes seguimientos.
6. Asignar a XOut, si procede, el valor de OUT que implica la baja del caso en siguientes seguimientos.

Proceso (para todos los casos):

7. Contar en N_IDT@ (ó en R_IDT@, Si R>1) el número de registros que comparten V

Proceso (para cada caso):

8. Si OUT≠vacío y VAR_{OUT} = XOut y Registro $_{j+1} \neq$ vacío, @V_i = 12.
9. Si L>=1 y N_IDT@ (ó R_IDT@, Si R>1) $\neq$ R:
   9.1. Listar: CN, V, N_IDT@ (ó R_IDT@ Si R>1).

Algoritmo 4-7: Depuración de la integridad referencial respecto a una tabla principal.

Asignación de los parámetros:

10. Asignar a V la lista de variables a depurar.
11. Asignar a TABLE el nombre (y ruta) de la tabla principal.
12. Asignar a DROP el nombre de las variables de TABLE que no se desean incorporar.
13. Asignar a C las variables que forman el identificador.
14. Asignar a CN la variable que contiene el número de secuencia de los casos. (Por defecto CN= @casenum).
15. Asignar a LVL, si procede, la lista de otras variables que desean ser listadas
16. Asignar a L el valor 0 si se desea omitir el listado de incidencias, 1 si se desea listar errores y valores desconocidos ó 2, si se desea listar errores pero no valores desconocidos. (Por defecto L=1).

Proceso (para cada caso):

Sea I_IDT@ la correspondiente variable auxiliar numérica y sea V_{TABLE} el identificador de la tabla principal.

17. I_IDT@ = SYSMIS.
18. Si V = $\varnothing$, I_IDT@ = 0.
19. Si V $\in$ V_{TABLE}, I_IDT@ = -1.
20. Si $V_i \notin V_{TABLE}$, I_IDT@ = 3.
21. Si L=1 y I_IDT@ >= 0 :
    21.1. Listar: CN, V, I_IDT@, LVL.

El Algoritmo 4-6 muestra en pseudo-código el proceso para comprobar los dos primeros criterios y el cuarto; el Algoritmo 4-7 muestra la secuencia para comprobar el tercero.

Documentación 4-5: Parámetros de la macro !IDT.

```
DEPURACIÓN DE DATOS: DUPLICADOS, RÉPLICAS E INTEGRIDAD REFERENCIAL
Creación 08.06.2000 Última revisión 07.07.2003
(c) A.Bonillo & JM.Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
   !IDT V= Lista de la variables que definen el identificador
       /R= Número de registros por caso             (por defecto, 1)
      /CN= Variable identificadora de la secuencia de los sujetos
                                                    (por defecto, @casenum)
     /LVL= Lista de otras variables a listar
      /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)
                                                    (por defecto,  1)
------------------------------------------------------------------------------
Parámetros para comparobar la integridad referencial:
ó[/TABLE]= Nombre (y ruta) de la tabla principal
   [/VID]= Lsita de variables identificadoras de la tabla principal
                                                    (por defecto, V)
  [/DROP]= Lista de variables a no añadir           (por defecto ninguna)
------------------------------------------------------------------------------
Parámetros para variables implicadas en seguimientos:
   [/OUT]= Variable indicadora de la baja del registro en futuros seguimientos
  [/XOUT]= Valor de OUT que indica la baja del registro en futuros seguimientos
------------------------------------------------------------------------------

 Ejemplos de llamada:
!IDT  V=h caso.
!IDT V=h caso fr /OUT=Exitus /XOut=1 /LVL=fn
   /TABLE='C:\...\Escritorio\Censal.SAV' /VID=h caso /DROP=nombre TO cpostal.
```

Listado 4-17: Detección de duplicados en los identificadores H y CASO y depuración de la integridad referencial del archivo de test.

```
!IDT V=h caso fr /OUT=Exitus /XOut=1 /LVL=fn
 /TABLE='C:\...\Escritorio\Censal.SAV' /VID=h caso /DROP=nombre TO cpostal.

@CASENUM H CASO         FR N_IDT@                     FN

     19  B  103 29.11.1993 Repetido 2 veces   05.11.1972
     20  B  103 29.11.1993 Repetido 2 veces   05.11.1972

Number of cases read:  2    Number of cases listed:  2

@CASENUM H CASO         FR I_IDT@                   FN

      1  A    . 11.07.1993  0: Missing Rec 12.04.1965
      3  A   21 17.09.1993 12:Err Baja cas 12.04.1965
      9  A   14          .  0: Missing Rec 30.01.1954
     11  A    5 21.05.1993  3: Falta Princ 11.06.1968
     17      94 22.04.1993  0: Missing Rec 20.10.1961

Number of cases read:  5    Number of cases listed:  5
```

La Documentación 4-5 describe los parámetros de la macro !IDT que permite depurar los identificadores.

En el fichero de test de que disponemos se debe comprobar que no existen duplicados, que el identificador del sujeto se incluye en la tabla principal (CENSAL.SAV) y que no hay valores desconocidos en el identificador. Se parte del supuesto que la tabla principal ya ha sido depurada.

La secuencia de depuración consistirá en comprobar los tres criterios antes especificados y a continuación efectuar los chequeos sistemáticos (y lógicos si los hubiese) propios de cada una de las variables que forman parte del identificador. Esta secuencia puede invertirse y se obtienen los mismos resultados.

El Listado 4-17 contiene la llamada a la macro !IDT para comprobar los cuatro criterios especificados. El parámetro V contiene el nombre de las variables que forman el identificador, incluyendo el identificador de réplica. Los parámetros OUT y XOut contienen, respectivamente, la variable y el valor, que indicaría que no deberían haberse registrados más seguimientos para ese caso. El parámetro LVL recoge la lista de otras variables que se desean listar junto al identificador. Esta opción resulta interesante para observar si dos registros que comparten identificador pertenecen realmente al mismo caso introducido dos veces o, por el contrario, se trata de dos registros distintos que han sido erróneamente identificados con los mismos valores.

Para comprobar la integridad referencial de los casos respecto a una tabla principal se utilizarán los siguientes parámetros. El parámetro TABLE incluye el nombre y la ruta de la tabla principal. El parámetro optativo DROP permite especificar las variables de la tabla principal que no se desean incorporar al archivo de trabajo actual. El parámetro opcional VID recoge la lista de variables del archivo de trabajo respecto a las que se desea comprobar la integridad referencial. Estarán compuestas por las variables identificadoras, a excepción de la que identifica la réplica. En el archivo de test presentado las variables identificadoras serían H, CASO y FR, siendo esta última la identificadora de réplica. Se ha omitido el parámetro R porque en este ejemplo no debe haber un número fijo de réplicas.

La parte superior del Listado 4-17 lista los sujetos con identificador repetido y también aquellos cuyo seguimiento no debería haberse registrado. Por un lado, el registro 3 corresponde al caso 21 y su seguimiento no debería haberse registrado, ya que en el registro anterior había sido dado de baja al tomar la variable EXITUS el valor 1. Por otro lado, los registros 19 y 20 tienen el mismo código de hospital y número de caso y comparten fecha de nacimiento, lo que hace sospechar que se trate del mismo registro introducido dos veces. Esta situación ilustra la necesidad de una variable con la secuencia de los sujetos en el archivo de datos (@CASENUM), ya que de otro modo sería imposible identificar y seleccionar uno de los registros con identificador duplicado.

En la parte inferior del Listado 4-17 se listan los sujetos que vulneran la integridad referencial. Los registros 1, 17 y 9 tienen valor 0 en la variable auxiliar porque se desconoce, respectivamente, el número de caso, el código de hospital y la fecha de respuesta; el registro 11 tiene valor 3 en I_IDT@ porque el sujeto no está incluido en la tabla principal (Listado 4-1).

Mediante la depuración efectuada hemos detectado duplicados, sujetos cuyo identificador no existe en la tabla principal (comprobación que incluye la detección de casos sin identificador) y seguimientos que, aparentemente, no deberían haberse registrado por indicarlo así una variable del estudio. Cuando sea necesario debe verificarse también que las variables que forman parte del identificador cumplen las condiciones de rango y lógicas.

En el archivo de test la variable H sólo puede tomar los valores ‘A’ o ‘B’ y la variable CASO debe estar comprendida entre 1 y 150, ser un número entero y no tener errores de formato. La parte superior del Listado 4-18 muestra las llamadas a la macro !DR que efectúan los chequeos sistemáticos de las variables H y CASO.

La parte inferior de este listado muestra los sujetos con incidencias detectados de las llamadas a !DR. El registro 18 tiene un código no válido en la variable hospital (H=“C”). El registro 11 tiene un error de precisión ya que el número de caso se había grabado con un decimal (CASO=5.1). El registro 14 contiene un error de formato (el mes 17 es imposible); los registros 13 y 14 están fuera del rango de las fechas especificadas. Las llamadas a las macros se realizan con el parámetro L=2 para indicar que no listen los valores desconocidos porque ya han sido detectados con la macro !IDT.

El registro 18 muestra que la integridad referencial y la depuración de rango de las variables del identificador son comprobaciones independientes. Este registro estaba presente en la tabla principal pero provenía de un hospital distinto a los admisibles en nuestro estudio.

Listado 4-18: Depuración de las variables que forman el identificador.

```
!DR V=h /LV='A','B' /MV=' ' /VAR=HI /C=h caso /INDX=fr /L=2.
!DR V=caso /LV=1 THRU 150 /ND=0 /VAR=HI /L=2 /FORMAT=1/C=h caso /INDX=fr /L=2.
!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /FORMAT=1 /VAR=HI /C=h caso /INDX=fr /L=2.
```

```
@CASENUM CASO         FR @H              H

     18    10 05.11.1993  3: Fuera Rango C

Number of cases read:  1    Number of cases listed:  1

@CASENUM H         FR @CASO           ·CASO

     11  A 21.05.1993  5: Err Num Dec 5.1

Number of cases read:  1    Number of cases listed:  1

@CASENUM H CASO         FR @FR                                      ·FR

      9  A   14          .  2: Err Formato                          13.17.1993
     13  B   17 07.09.1994  3: Fuera Rango                         07.09.1994
     14  B   82 20.11.1962  3: Fuera Rango                         20.11.1962

Number of cases read:  3    Number of cases listed:  3
```

Finalmente, el Listado 4-19 presenta el conjunto de incidencias detectadas en los identificadores.

Listado 4-19: Depuración completa del identificador.

```
!IDT V=h caso fr /OUT=Exitus /XOut=1 /LVL=fn
 /TABLE='C:\...\Escritorio\Censal.SAV' /VID=h caso /DROP=nombre TO cpostal.
```

```
@CASENUM H CASO         FR N_IDT@                      FN

     19  B  103 29.11.1993 Repetido 2 veces   05.11.1972
     20  B  103 29.11.1993 Repetido 2 veces   05.11.1972
Number of cases read:  2    Number of cases listed:  2

@CASENUM H CASO         FR I_IDT@                  FN

      1  A    . 11.07.1993  0: Missing Rec 12.04.1965
      3  A   21 17.09.1993 12:Err Baja cas 12.04.1965
      9  A   14           .  0: Missing Rec 30.01.1954
     11  A    5 21.05.1993  3: Falta Princ 11.06.1968
     17      94 22.04.1993  0: Missing Rec 20.10.1961
Number of cases read:  5    Number of cases listed:  5
```

```
!DR V=h /LV='A','B' /MV=' ' /C=caso.
!DR V=caso /LV=1 THRU 150 /ND=0 /L=2 /FORMAT=1 /C=h.
!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /L=2 /FORMAT=1 /VAR=HI /C=h caso.
```

```
@CASENUM CASO         FR @H              H

     18    10 05.11.1993  3: Fuera Rango C
Number of cases read:  1    Number of cases listed:  1

@CASENUM H         FR @CASO           ·CASO

     11  A 21.05.1993  5: Err Num Dec 5.1
Number of cases read:  1    Number of cases listed:  1
@CASENUM H CASO         FR @FR                                    ·FR

      9  A   14          .  2: Err Formato                        13.17.1993
     13  B   17 07.09.1994  3: Fuera Rango                       07.09.1994
     14  B   82 20.11.1962  3: Fuera Rango                       20.11.1962
Number of cases read:  3    Number of cases listed:  3
```

4.8.1 Corrección de incidencias en el identificador

El primer paso de la depuración consiste en detectar y corregir las incidencias de los identificadores. Este proceso debe efectuarse antes de depurar el resto de variables, ya que:

- Frente a casos duplicados, o que no debieran haberse registrado, es innecesario depurar sus variables y obtener las mismas incidencias en distintas ocasiones.
- Los casos no identificados requieren una decisión sobre su mantenimiento o exclusión de la base de datos. Si no se permite disponer de registros para los que no se conoce su origen, éstos deben ser eliminados y por tanto no es pertinente detectar si contienen otras incidencias.
- Finalmente, las correcciones de las variables con incidencias se facilitan conociendo el identificador. Si un registro carece de identificador o está equivocado su localización en las fuentes originales puede ser laboriosa e, incluso, imposible.

Para corregir las incidencias detectadas en los identificadores del archivo de test se dispone del Listado 4-19. En este test se ha comprobado que los registros 19 y 20 (esto es, con valor en @CASENUM 19 y 20) corresponden a un mismo sujeto introducido dos veces. Sin embargo, en la segunda ocasión se ha capturado de forma más completa, ya que se ha registrado la variable SEXO. El registro 3 no debería haberse registrado, ya que se ha comprobado que el paciente falleció tras el segundo seguimiento.

El primer registro sin número de caso corresponde a un paciente procedente del hospital A con número de caso igual a 21. Al registro 11 que tenía un error de precisión (CASO=5.1) le corresponde el valor 51 en CASO. Las fechas de respuesta de los registros 13 y 14, que eran erróneas, han sido corregidas. Finalmente, los registros 17 y 18 proceden ambos del hospital B.

Estas correcciones se traducen en las siguientes instrucciones, que deben utilizar como identificador de registro la variable @CASENUM:

```
SELECT IF (@casenum<> 3).     /*Eliminación del registro tras su exitus.
SELECT IF (@casenum<>19).     /*Eliminación del registro duplicado.
IF (@casenum= 1) caso= 21.    /*Corrección de incidencias.
IF (@casenum=11) caso= 51.
IF (@casenum=13) fr  = DATE.DMY(7,9,1993).
IF (@casenum=14) fr  = DATE.DMY(13,5,1993).
IF (@casenum=17) h   = 'B'.
IF (@casenum=18) h   = 'B'.
```

El Listado 4-20 muestra la secuencia del proceso de depuración de los identificadores. El procedimiento consiste en insertar antes de las llamadas a las macros de depuración de identificadores el anterior paquete de instrucciones de corrección. Seguidamente se ejecuta toda la secuencia desde el DATA LIST hasta el final. La depuración de identificadores finaliza porque tras las llamadas a las macros no se ha detectado ninguna incidencia.

Después de la corrección se han ejecutado las cuatro llamadas a las macros para asegurarnos que los cambios realizados no han introducido ni enmascarado errores. Podría haber sucedido, por ejemplo, que el caso 94 (@CASENUM=10) para el que no se conocía el hospital de procedencia hubiera sido erróneamente asignado al hospital A. En la segunda ocasión en que hubiéramos chequeado el identificador se hubiera detectado una nueva duplicidad porque sus identificadores hubieran coincidido con los del segundo registro.

Listado 4-20: Corrección de identificadores.

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='C:\...\Escritorio\TestNoDep.DAT'
  /h 2(A)
   caso   4-6(F)     ·caso  4-6(A)
   fr    8-17(EDATE) ·fr    8-17(A)
   fn   19-28(EDATE) ·fn    19-28(A)
   sexo   30(A)
   talla  32-35(F,2) ·talla 32-35(A)
   dpt    37(F)      ·dpt   37(A)
   dcs    39(F)      ·dcs   39(A)
   fuma   41(F)      ·fuma  41(A)
   tab    43-44(F)   ·tab   43-44(A)
   tiptab 46-47(A)
   cie    49-54(A).
   pad    56-58(F)   ·pad    56-58(A)
   pas    60-62(F)   ·pas    60-62(A)
   exitus 64   (F)   ·exitus 64  (A) .
SET ERRORS=NONE.      /*Desactiva el listado de errores de formato del DATA
LIST.
EXECUTE.
SET ERRORS=ON.        /*Activa de nuevo el listado de mensajes de error.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab cie.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.

*Corrección de identificadores.

SELECT IF (@casenum<> 3).        /*Eliminación del registro tras su exitus.
SELECT IF (@casenum<>19).        /*Eliminación del caso duplicado.
IF (@casenum= 1) caso= 21.       /*Corrección de incidencias.
IF (@casenum=11) caso= 51.
IF (@casenum=17) h   = 'B'.
IF (@casenum=18) h   = 'B'.
IF (@casenum=13) fr  = DATE.DMY(7,9,1993).
IF (@casenum=14) fr  = DATE.DMY(13,5,1993).
EXECUTE.

*Depuración del identificador.

!IDT V=h caso fr /OUT=Exitus /XOut=1 /LVL=fn
/TABLE='C:\...\Escritorio\Censal.SAV' /VID=h caso /DROP=nombre TO cpostal.

!DR V=h /LV='A','B' /MV=' ' /C=caso.

!DR V=caso /LV=1 THRU 150 /ND=0 /L=2 /FORMAT=1 /C=h.

!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /L=2 /FORMAT=1 /VAR=HI /C=h caso.
```

```
Macro !IDT V2003.07.07
Number of cases read:  0

Macro !DR V2003.07.07
Number of cases read:  0

Macro !DR V2003.07.07
Number of cases read:  0

Macro !DR V2003.07.07
Number of cases read:  0
```

Listado 4-21: Instrucciones para corregir las incidencias de las variables de salto

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='C:\...\Escritorio\TestNoDep.DAT'
  /h 2(A)
   caso   4-6(F)     ·caso  4-6   (A)
   fr    8-17(EDATE) ·fr    8-17  (A)
   fn   19-28(EDATE) ·fn    19-28 (A)
   sexo   30(A)
   talla  32-35(F,2) ·talla 32-35 (A)
   dpt    37(F)      ·dpt   37(A)
   dcs    39(F)      ·dcs   39(A)
   fuma   41(F)      ·fuma  41(A)
   tab    43-44(F)   ·tab   43-44 (A)
   tiptab 46-47(A)
   cie    49-54(A).
   pad    56-58(F)   ·pad    56-58(A)
   pas    60-62(F)   ·pas    60-62(A)
   exitus 64   (F)   ·exitus 64   (A).
EXECUTE.


*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab cie.
    COMPUTE var = UPCASE(var).
END REPEAT.
*Grabación del archivo no depurado para la estadística de calidad.
SAVE OUTFILE='C:\...\Escritorio\TabNoDep.SAV'.
*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6) fr fn (EDATE10).

*Corrección de los identificadores.
SELECT IF (@casenum<> 3).        /*Eliminación del registro tras su exitus.
SELECT IF (@casenum<>19).        /*Eliminación del caso duplicado.
IF (@casenum= 1) caso= 21.       /*Corrección de incidencias.
IF (@casenum=11) caso= 51.
IF (@casenum=17) h   = 'B'.
IF (@casenum=18) h   = 'B'.
IF (@casenum=13) fr  = DATE.DMY(7,9,1993).
IF (@casenum=14) fr  = DATE.DMY(13,5,1993).
EXECUTE.

*Corrección de la variable de salto.
IF (h='A' AND caso =133) fuma =$sysmis./*Corrección variables de salto.
IF (h='A' AND caso =133) ·fuma  = ' '.
IF (h='A' AND caso = 14) fuma =$sysmis.
IF (h='B' AND caso = 16) fuma =0.
IF (h='B' AND caso = 17) fuma =0.
.....................................

*Depuración del identificador.
!IDT V=h caso /OUT=Exitus /XOut=1 /LVL=fn /VID=h caso
/TABLE='C:\...\Escritorio\Censal.SAV' /DROP=nombre to cpostal.


!DR V=h /LV='A','B' /MV=' ' /C=caso.
!DR V=caso /LV=1 THRU 150 /ND=0 /L=2 /FORMAT=1 /C=h.
!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /L=2 /FORMAT=1 /VAR=HI /C=h caso.

*Depuración de variables de salto.
.....................................
*Depuración del resto de variables del estudio.
!DR V=sexo /LV='M','F' /MVr=' ' /C=h caso.
.....................................
```

4.8.2 Corrección de incidencias en las variables de salto

Las instrucciones que permiten corregir las variables de salto o cribado se sitúan, en la ventana de sintaxis, tras las instrucciones de depuración y corrección de los identificadores. Esta propuesta jerárquica de la secuencia de las correcciones pretende ser consistente con la estructura jerárquica de los estudios. Así, las variables de mayor rango jerárquico son los identificadores, puesto que de ellas se derivan el resto de valores. En segundo lugar jerárquico es ocupado por las variables de salto, ya que de ellas dependen todas las variables de los subformularios incluídas en él.

Las instrucciones que permiten corregir las variables de salto del ejemplo presentado son:

```
*Corrección de la variable de salto.
IF (h='A' AND caso =133) fuma =$sysmis./*Corrección variables de salto.
IF (h='A' AND caso =133) ·fuma  = ' '.
IF (h='A' AND caso = 14) fuma =$sysmis.
IF (h='B' AND caso = 16) fuma =0.
IF (h='B' AND caso = 17) fuma =0.
```

De los casos 133 y 14 no ha podido averiguarse si fuman o no, por lo que los errores han sido asignados a valor desconocido. De los casos 16 y 17 sí se ha averiguado que no fuman.

Documentación 4-6: Parámetros de la macro !INCIDEN.

```
DEPURACIÓN DE DATOS: Informe de incidencias
Creación 30.11.1998 Última revisión 07.07.2003
(c) A.Bonillo & JM. Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
!INCIDEN V= Lista de variables auxiliares a verificar
[/EXCLUDE]= Lista de valores (nº registro y variables) que no se desea listar
        /C= Lista de variables identificadoras del sujeto
   [/INDX]= Variable de índice
       /CN= Variable identificadora de la secuencia de los sujetos
                                                (por defecto,@casenum)

Ejemplos de llamada:
!INCIDEN LV=@fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie/C=H CASO.
!INCIDEN LV=@fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie
   /EXCLUDE=2 @cie, 3 @dcs, 4 @talla @fuma @tab @tiptab /C=H CASO.
```

4.9 INFORME DE INCIDENCIAS

Una vez finalizado el chequeo de las variables se debe obtener un informe de las incidencias detectadas para corregirlas. Los listados obtenidos en los chequeos anteriores presentan las incongruencias variable a variable. Sin embargo es preferible disponer de un listado sujeto a sujeto porque facilita la identificación de los errores y su corrección.

La Documentación 4-6 describe los parámetros de la macro !INCIDEN que permite realizar el informe de incidencias.

La parte superior del Listado 4-22 presenta la llamada a la macro !INCIDEN para el archivo de prueba. El parámetro V contiene la lista completa de variables @.

La parte inferior del Listado 4-22 contiene la estadística de incidencias listadas junto al listado completo de errores y valores desconocidos a recuperar del archivo de prueba.

En la estadística de incidencias se comprueba que se han detectado un total de 28 errores, que representan un 12% respecto al total de datos chequeados. El total de valores desconocidos recuperables es de 27 (11.5%).

En el listado de incidencias se observa que, por ejemplo, el sujeto con identificador H="A" y CASO=94 (registro número 4) tiene un error porque el contenido de la variable TAB es incongruente con un salto, y también un valor desconocido en la variable CIE.

El listado obtenido con la macro !INCIDEN permite corregir las incidencias contenidas en los datos. Para este ejemplo, junto a cada error y/o valor desconocido recuperable se ha escrito el supuesto valor correcto que figuraba en el formulario de papel (Tabla 4-3). El parámetro EXCLUDE se utiliza para no listar las incidencias en las variable de salto (fuma) que no han podido ser recuperadas.

El Listado 4-22 y la Tabla 4-3 omiten el registro 19 porque corresponde a un duplicado que debe ser eliminado (véase apartado 4.8.1).

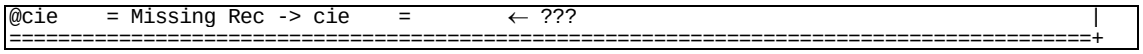
	H	CASO	FR	FN	SEXO	TALLA	DPT	DCS	FUMA	TAB	TIPTAB	CIE	PAD	PAS	EXITUS
1	A	**21**	11.07.1993	12.04.1965	M	1.69	1	1	1	20	NE	398.90	104	162	0
2	A	21	30.07.1993	12.04.1965	**M**	1.69	1	1	1	20	NE	398.90	90	168	1
4	A	94	08.06.1993	15.07.1966	F	1.79	2	3	0	**0**			94	182	0
5	a	12	01.11.1993	07.11.1966	**M**	1.70	**1**		9	•		432	102	154	0
6	A	133	21.06.1993	25.05.1949	F		3	1	•			415.11	86	104	0
7	A	133	14.08.1993	25.05.1949	**F**		3	1				415.11	84	**130**	0
8	A	133	14.12.1993		F		3	1				415.11	**90**	104	0
9	A	14	**13.7.1993**	30.01.1954	M	**1.75**	2	2	•		***vacío***		96	178	0
10	A	14	13.07.1993	30.01.1954	M	1.75	2	2	•		***vacío***		96	178	0
11	A	**51**	21.05.1993	11.06.1968	M	1.73			1	15	RU	435.9		138	0
12	B	16	14.12.1993		F	1.77	1	3	**0**	0		423.9	94	150	0
13	B	17	**7.09.1993**	**01.11.1987**	F	**1.05**		2	**0**	**0**		411.0	88	132	0
14	B	82	**13.5.1993**	**20.11.1962**	M		2	1	0	0		398.90	100	156	0
15	B	82	01.08.1993	20.11.1962	M		2	1	0	0		398.90	90	150	0
16	B	82	05.11.1993	20.11.1962	M		2	1	0	0		422.9	95	130	0
17	**B**	94	22.04.1993	20.10.1961	**M**	1.78	2	2	1	1	RU	429.9	86	162	0
18	**B**	10	05.11.1993	03.04.1952	M	1.66	3	3	1	10	N	**411.0**	94	138	0
20	B	103	29.11.1993	05.11.1972	F	1.58	3	3	0	0			88	138	0

Tabla 4-3: Datos corregidos del archivo de test TabNoDep.DAT.

Listado 4-22: Informe de incidencias por caso obtenidas con la macro !INCIDEN.

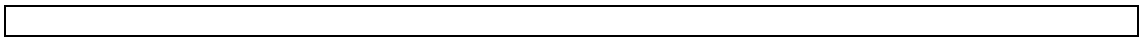
```
!INCIDEN V=@fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie @pad @pas
         @exitus /EXCLUDE=6 @fuma, 7 @fuma, 8 @fuma, 9 @fuma /C=H CASO FR.
```

```
ESTADÍSTICA DE INCIDENCIAS LISTADAS ------------------------------------------------
Total ERRORES ..... =      28 (11.966 %)
Total MISSING recuperables =      27 (11.538 %)
Total valores correctos... =     179 (76.496 %)
=============================================================================================

Identificador caso: H= A  ; CASO=  21  ; FR= 30.07.1993
Número de Registro:      2 ;         Número de incidencias: Error=   1 ;   Missing=   0
---------------------------------------------------------------------------------------------+
@sexo     =   Err   Cnstant->   sexo   =                                ←   Es   hombre     (M)
|
=============================================================================================+

Identificador caso: H= A  ; CASO=  94  ; FR= 08.06.1993
Número de Registro:      4 ;         Número de incidencias: Error=   1 ;   Missing=   1
---------------------------------------------------------------------------------------------+
@tab    = Incons Salt -> tab    = 15   ← No fuma   (0)
     |
@cie    = Missing Rec -> cie    =      ← ???                                                 |
=============================================================================================+

dentificador caso: H= A ; CASO= 12   ; FR= 01.11.1993
Número de Registro:      3;          Número de incidencias: Error=   2;   Missing=   2
---------------------------------------------------------------------------------------------+
@sexo   = Missing Rec -> sexo   =    ← Es mujer (F)
     |
@dpt    = Fuera Rango -> dpt    = 0  ← No hace deporte  (1)                                   |
@dcs    = Missing Rec -> dcs    = .  ← ???                                                    |
@tab    = Incons Salt -> tab    =    ← ???  (vacío)                                          |
=============================================================================================+

Identificador caso: H= A  ; CASO= 133  ; FR= 21.06.1993
Número de Registro:      6 ;         Número de incidencias: Error=   3 ;   Missing=   1
---------------------------------------------------------------------------------------------+
@talla  = Missing Rec -> talla  = .  ← ???                                                    |
@fuma   = Fuera Rango -> fuma   = 7  ← ???                                                    |
@tab    = Incons Salt -> tab    = .  ← ???    (vacío)
     |
@tiptab = Incons Salt -> tiptab =    ← ???    (vacío)
     |
=============================================================================================+

Identificador caso: H= A  ; CASO= 133  ; FR= 14.08.1993
Número de Registro:      7 ;         Número de incidencias: Error=   1 ;   Missing=   4
---------------------------------------------------------------------------------------------+
@talla  = Missing Rec -> talla  = .  ← ???                                                    |
@fuma   = Fuera Rango -> fuma   = 7  ← ???                                                    |
@tab    = Incons Salt -> tab    = .  ← ???    (vacío)
     |
@tiptab = Incons Salt -> tiptab =    ← ???    (vacío)
     |
@pas    = Excso Varcn-> pas      = 184  ← 130
     |
=============================================================================================+

Identificador caso: H= A  ; CASO= 133  ; FR= 14.12.1993
Número de Registro:      8 ;         Número de incidencias: Error=   1 ;   Missing=   5
---------------------------------------------------------------------------------------------+
@fn     = Err Cnstant -> fn       =          .  ← 25.5.49
|
@talla  = Missing Rec -> talla  = .  ← ???                                                    |
@fuma   = Fuera Rango -> fuma   = 7  ← ???                                                    |
@tab    = Incons Salt -> tab    = .  ← ???    (vacío)
     |
@tiptab = Incons Salt -> tiptab =    ← ???    (vacío)
                          |
@pad    = Excso Varcn-> pad      =  20 ← 90                                                   |
=============================================================================================+

Identificador caso: H= A  ; CASO=  14  ; FR= 13.07.1993
Número de Registro:      9 ;         Número de incidencias: Error=   2 ;   Missing=   3
---------------------------------------------------------------------------------------------+
@talla  = Fuera Rango -> talla  = 2.75   ← 1.75                                               |
@fuma   = Missing Rec -> fuma   = .      ← ???                                                |
@tab    = Missing Rec -> tab    = .      ← ???  (vacío)                                       |
@tiptab = Incons Salt -> tiptab = RU     ← ???  (vacío)                                       |
```


@cie = Missing Rec -> cie = ← ???

Listado 4-22: Informe de incidencias por caso obtenidas con la macro !INCIDEN. (continuación)

```
Identificador caso: H= A  ; CASO=  14  ; FR= 13.07.1993
Número de Registro:     10 ;         Número de incidencias: Error=   1 ;   Missing=   3
-------------------------------------------------------------------------------------+
@fuma   = Missing Rec -> fuma   = .      ← ???                                       |
@tab    = Missing Rec -> tab    = .      ← ???  (vacío)                              |
@tiptab = Incons Salt -> tiptab = RU     ← ???  (vacío)                              |
@cie    = Missing Rec -> cie    =        ← ???                                       |
=====================================================================================+

Identificador caso: H= A  ; CASO=  51  ; FR= 21.05.1993
Número de Registro:     11 ;         Número de incidencias: Error=   0 ;   Missing=   3
-------------------------------------------------------------------------------------+
@dpt    = Missing Rec -> dpt    = .      ← ???                                       |
@dcs    = Missing Rec -> dcs    = .      ←???                                        |
@pad    = Missing Rec-> pad     =   .    ←???
         |
=====================================================================================+

Identificador caso: H= B  ; CASO=  16  ; FR= 14.12.1993
Número de Registro:     12 ;         Número de incidencias: Error=   3 ;   Missing=   1
-------------------------------------------------------------------------------------+
@fn     = Err Formato -> ·fn    = 16.10. ← ???                                       |
@fuma   = Err Formato -> ·fuma  = O      ← No fuma                                   |
@tab    = Incons Salt -> tab    =  0     ← Está bien porque no fuma                  |
@tiptab  =  Missing  Rec  ->  tiptab  =                ←  Está  bien  porque  no  fuma
     |
=====================================================================================+

Identificador caso: H= B  ; CASO=  17  ; FR= 07.04.1994
Número de Registro:     13 ;         Número de incidencias: Error=   4 ;   Missing=   1
-------------------------------------------------------------------------------------+
@talla  = Err Num Dec -> talla  =   .98  ← Mide 1.05 metros                          |
@dpt    = Missing Rec -> dpt    = .      ← ???                                       |
@fuma   = Err Cond 11 -> fuma   = 1      ← No fuma                                   |
@tab    = Incons Salt -> tab    =  0     ← Está bien porque no fuma                  |
@tiptab = Incons Salt -> tiptab =        ← Está bien porque no fuma                  |
=====================================================================================+

Identificador caso: H= B  ; CASO=  82  ; FR= 13.05.1993
Número de Registro:     14 ;         Número de incidencias: Error=   2 ;   Missing=   0
-------------------------------------------------------------------------------------+
@fn     = Dif fuera rango-> fn   = 13.05.1993  ← Están cambiadas las fechas de nacimiento
y la de respuesta |
@talla  = Err Cnstant-> talla    =   .         ← ???                                  |
=====================================================================================+

Identificador caso: H= B  ; CASO=  82  ; FR= 01.08.1993
Número de Registro:     15 ;         Número de incidencias: Error=   0 ;   Missing=   1
-------------------------------------------------------------------------------------+
@talla  = Missing Rec-> talla    =   .          ← ???                                 |
=====================================================================================+

Identificador caso: H= B  ; CASO=  82  ; FR= 05.11.1993
Número de Registro:     16 ;         Número de incidencias: Error=   0 ;   Missing=   1
-------------------------------------------------------------------------------------+
@talla  = Missing Rec-> talla    =   .          ← ???                                 |
=====================================================================================+

Identificador caso: H= B  ; CASO=  94  ; FR= 22.04.1993
Número de Registro:     17 ;         Número de incidencias: Error=   2 ;   Missing=   0
-------------------------------------------------------------------------------------+
@sexo   = Fuera Rango -> sexo   = V      ← Es hombre (M)                             |
@tab    = Fuera Rango -> tab    = -1     ← 15 c / d                                  |
=====================================================================================+

Identificador caso: H= B  ; CASO=  10  ; FR= 05.11.1993
Número de Registro:     18 ;         Número de incidencias: Error=   2 ;   Missing=   0
-------------------------------------------------------------------------------------+
@tiptab = Fuera Rango-> tiptab   = N       ← Fuma negro (NE)
         |
@cie    = Fuera Rango -> cie    = 030.3  ← 411.0                                     |
=====================================================================================+

Identificador caso: H= B  ; CASO= 103  ; FR= 29.11.1993
Número de Registro:     20 ;         Número de incidencias: Error=   0 ;   Missing=   1
-------------------------------------------------------------------------------------+
@cie    = Missing Rec -> cie    =        ← ???                                       |
=====================================================================================+
```



Listado 4-23: Instrucciones para corregir las incidencias de las variables del fichero de test.

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='TestNoDep.DAT'
  /h 2(A)
   caso   4-6(F)     ·caso  4-6(A)
   fr    8-17(EDATE) ·fr    8-17(A)
   fn   19-28(EDATE) ·fn    19-28(A)
   sexo   30(A)
   talla  32-35(F,2) ·talla 32-35(A)
   dpt    37(F)      ·dpt   37(A)
   dcs    39(F)      ·dcs   39(A)
   fuma   41(F)      ·fuma  41(A)
   tab    43-44(F)   ·tab   43-44(A)
   tiptab 46-47(A)
   cie    49-54(A).
   pad    56-58(F)   ·pad    56-58(A)
   pas    60-62(F)   ·pas    60-62(A)
   exitus 64   (F)   ·exitus 64  (A) .
EXECUTE.
SET ERRORS=ON.        /*Activa de nuevo el listado de mensajes de error.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.

*Corrección de identificadores.
SELECT IF (@casenum<> 3).       /*Eliminación del registro tras su exitus.
SELECT IF (@casenum<>19).       /*Eliminación del caso duplicado.
IF (@casenum= 1) caso= 21.      /*Corrección de incidencias.
IF (@casenum=11) caso= 51.
IF (@casenum=17) h   = 'B'.
IF (@casenum=18) h   = 'B'.
IF (@casenum=13) fr  = DATE.DMY(7,9,1993).
IF (@casenum=14) fr  = DATE.DMY(13,5,1993).
EXECUTE.

*Corrección de la variable de salto.
IF (h='A' AND caso =133) fuma =$sysmis./*Corrección variables de salto.
IF (h='A' AND caso =133) •fuma  = ' '.
IF (h='A' AND caso = 14) fuma =$sysmis.
IF (h='B' AND caso = 16) fuma =0.
IF (h='B' AND caso = 17) fuma =0.

*Corrección del resto de variables del estudio.
IF (h='A' AND caso = 21) sexo   = 'M'.
IF (h='A' AND caso = 94) tab   = 0.
IF (h='A' AND caso = 12) sexo  = 'M'.
IF (h='A' AND caso = 12) dpt   = 1.
IF (h='A' AND caso = 12) tab   = $sysmis.
IF (h='A' AND caso =133) sexo  = 'F'.
IF (h='A' AND caso =133) tab   = $sysmis.
IF (h='A' AND caso = 14) fr    = DATE.DMY(13,7,1993).
IF (h='A' AND caso = 14) talla = 1.75.
IF (h='A' AND caso = 14) tiptab= ' '.
IF (h='B' AND caso = 16) ·fn    =' '.
IF (h='B' AND caso = 17) fr    = DATE.DMY(7,9,1993).
IF (h='B' AND caso = 17) talla = 1.05.
IF (h='B' AND caso = 82) fr    = DATE.DMY(13,5,1993).
IF (h='B' AND caso = 82) fn    = DATE.DMY(20,11,1962).
IF (h='B' AND caso = 94) sexo  = 'M'.
IF (h='B' AND caso = 94) tab   = 15 .
IF (h='B' AND caso = 10) tiptab= 'NE'.
IF (h='B' AND caso = 10) cie   = '422.90'.
```

```
EXECUTE.
```

Listado 4-23: Instrucciones para corregir las incidencias de las variables del fichero de test. (continuación)

```
*Depuración del identificador.
!IDT V=h caso fr /OUT=Exitus /XOut=1 /LVL=fn
/TABLE='C:\Documents and Settings\Albert\Escritorio\Spss\Censal.SAV' /VID=h
caso /DROP=nombre TO cpostal.

!DR V=h /LV='A','B' /MVr=' ' /C=caso  /INDX=fr /L=2.
!DR V=caso /LV=1 THRU 150 /ND=0 /FORMAT=1 /C=h  /INDX=fr /L=2.
!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /FORMAT=1 /VAR=HI /C=h caso /INDX=fr /L=2.


*Depuración de la(s) variable(s) de salto.
*Condición @FUMA=50: Los fumadores deben tener más de 10 años.
COMPUTE @fuma=$SYSMIS.
IF (fuma=1 AND CTIME.DAYS(fr-fn)<(365.25*11)) @fuma=50.
*Comprobación variable FUMA.
!DR V=fuma /LV=0,1 /MV=9 /VAR=0 /LVL=fn fr  /FORMAT=1 /C=h caso  /INDX=fr.

*Depuración de las variables del estudio.

!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /VAR=0 /FORMAT=1 /C=h caso
/INDX=fr.

!DR V=sexo /LV='M','F' /VAR=0 /MVr=' ' /C=h caso  /INDX=fr.

!DR V=talla /LV=0.95 THRU 2 /VAR=0 /ND=2 /VAR=0 /FORMAT=1 /C=h caso  /INDX=fr.

!DR V=dpt dcs /LV=1,2,3 /VAR=0 /FORMAT=1 /C=h caso  /INDX=fr.


!DR V=tab /LV=1 thru 80 /ND=0 /VS=fuma /MVS=9 /XS=0 /VD=0 /VAR=0 /FORMAT=1 /C=h
caso  /INDX=fr.
!DR V=tiptab /LV='NE','RU','NR' /MVr=' ' /VS=fuma /XS=0 /VAR=0 /MVS=9 /C=h caso
/INDX=fr.


*Comprobación variable CIE.
COMPUTE @cie=$SYSMIS.
IF NOT(RANGE(NUMBER(SUBSTR(cie,1,3),F3),390,459)) @cie=50.
!DRKey V=cie /TABLE='C:...\Escritorio\CIE9.SAV' /DROP=des
  /MVr=' ' /VAR=HI /LVL=@NULL /C=h caso  /INDX=fr.


!DR V=pad /LV=20 THRU 300 /ND=0 /VAR=30% /FORMAT=1 /C=h caso  /INDX=fr.

!DR V=pas /LV=50 THRU 400 /ND=0 /VAR=30% /FORMAT=1 /C=h caso  /INDX=fr.

!DR V=exitus /LV=0,1 /VAR=HI /FORMAT=1 /C=h caso  /INDX=fr.


*Informe de incidencias.
!INCIDEN V=@fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie /C=H CASO
/INDX=fr /EXCLUDE=2 @cie, 3 @dcs, 4 @talla @fuma @tab @tiptab @cie, 5 @fuma @tab
@tiptab @dcs @cie, 6 @dpt @dcs @talla @tab @fuma @tiptab, 7 @fn @talla @tab @fuma
@tiptab, 8 @fn @dpt @talla @tab @fuma @tiptab, 9 @talla @tab @fuma @tiptab @cie,
10 @tab @fuma @tiptab @cie, 11 @dpt @dcs, 12 @fn, 13 @dpt, 14 @talla, 15 @talla,
16 @talla, 20 @cie.
```

```
ESTADÍSTICA DE INCIDENCIAS LISTADAS ----------------------------------------------------
Total ERRORES ............ =      1 (  .556 %)
Total MISSING recuperables =      0 (  .000 %)
           no recuperados  =     45 (25.000 %)
Total valores correctos... =    134 (74.444% )
=============================================================================================

Identificador caso: H= B  ; CASO=  17 ; fr= 07.09.1993
Número de Registro:     13 ;         Número de incidencias: Error=   1 ;   Missing=   0
---------------------------------------------------------------------------------------+
@fn     = Dif fuera rango-> fn   = 01.11.1987                                          |
=======================================================================================+
```

4.10 CORRECCIÓN DE LAS INCIDENCIAS

Cuando no es posible recurrir a las fuentes originales, los errores detectados se deben pasar de forma sistemática a valor desconocido. Si se trata de un error por inconsistencia entre variables, todos los campos afectados deben asignarse a *vacío* ya que no es posible asegurar cuál de ellos es válido y cuál no.

Cuando se dispone de las fuentes de información originales se deben comprobar las incidencias detectadas y corregirlas en la base de datos.

En ambas situaciones, es indispensable que quede constancia de todas las correcciones efectuadas en un archivo histórico de cambios para que se pueda retornar a los valores anteriores y se pueda someter a auditoria el proceso de corrección.

4.10.1 Corrección de incidencias en las variables

Si se disponen de los valores correctos, la corrección de las incidencias se efectúa escribiendo una instrucción para introducir cada uno de los valores correctos. Si no es posible recuperar los valores correctos las incidencias deben ser asignadas a valor desconocido. En los dos subapartados siguientes se mostrarán ambas situaciones.

4.10.1.1 Introducción de los valores correctos

El Listado 4-22 contiene todas las incidencias detectadas en el archivo de prueba y la Tabla 4-3 recoge en negrita los valores correctos localizados en los formularios originales de papel. El Listado 4-23 contiene las instrucciones SPSS que permiten efectuar estos cambios. En este caso no se debe utilizar como identificador la variable @CASENUM sino el identificador original porque el listado de estas instrucciones constituirá el documento que registra el conjunto de cambios realizados sobre los datos.

El proceso de depuración es análogo al efectuado para los identificadores y las variables de salto. Se trata de insertar a continuación de las instrucciones de corrección de identificadores y de las de las variables de salto, las de corrección de datos. Compruebe que cuando los datos originales están grabados en ASCII, se han detectado errores de formato y no ha sido posible localizar los valores correctos en las fuentes originales, la correspondiente variable auxiliar $\cdot V_i$ se ha asignado a blanco para que esta incidencia no vuelva a ser detectada como un error en sucesivos chequeos. Por ejemplo, del caso 16 no ha sido posible averiguar su año de nacimiento y por esta razón la variable ·fn se ha asignado a vacío (“blanco”).

En el Listado 4-23 la llamada a la macro !INCIDEN se ha modificado incluyendo el parámetro adicional EXCLUDE para indicar que no liste nuevamente los registros con valores desconocidos que no han podido ser recuperados (por ejemplo, del segundo registro no se ha averiguado su código CIE y del tercero el valor en DCS).

Se trata de ejecutar la secuencia completa desde el inicio (DATA LIST). En este caso la depuración no finaliza porque el listado de incidencias contiene un error de rango para el caso 17, que no cumple la edad mínima para participar en el estudio (6 años). Recordemos que en la fase de chequeos previa se había detectado que este sujeto tenía errónea la fecha de respuesta. Al corregir esta incidencia se ha descubierto un nuevo error. Tras acudir nuevamente a las fuentes originales se comprueba que la fecha de nacimiento de este sujeto es el 01.11.1986.

Listado 4-24: Grabación del archivo original en formato SPSS.

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='TestNoDep.DAT'
  /h 2(A)
   caso   4-6(F)     ·caso  4-6(A)
   fr    8-17(EDATE) ·fr    8-17(A)
   fn   19-28(EDATE) ·fn    19-28(A)
   sexo   30(A)
   talla  32-35(F,2) ·talla 32-35(A)
   dpt    37(F)      ·dpt   37(A)
   dcs    39(F)      ·dcs   39(A)
   fuma   41(F)      ·fuma  41(A)
   tab    43-44(F)   ·tab   43-44(A)
   tiptab 46-47(A)
   cie    49-54(A).
   pad    56-58(F)   ·pad    56-58(A)
   pas    60-62(F)   ·pas    60-62(A)
   exitus 64   (F)   ·exitus 64  (A) .
SET ERRORS=NONE.      /*Desactiva el listado de errores de formato del DATA
LIST.
EXECUTE.
SET ERRORS=ON.        /*Activa de nuevo el listado de mensajes de error.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Grabación del archivo no depurado necesario para la estadística de calidad.
SAVE OUTFILE='TestNoDep.SAV'.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.
```

Para corregir esta nueva incidencia detectada se debe agregar al conjunto de instrucciones con las correcciones para las variables un nuevo cambio:

```
IF (h='B' AND caso = 17) fn= DATE.DMY(1,11,1986).
```

Tras ejecutar nuevamente todas las instrucciones desde el DATA LIST, no se detectan nuevas incidencias y por lo tanto la depuración finaliza. Los datos están revisados y se deben grabar en un nuevo archivo con la instrucción:

```
SAVE OUTFILE = 'TestDep.SAV' /KEEP = h TO exitus.
```

4.10.1.2 Asignación automática de las incidencias a valor desconocido

En caso de que no se dispongan de los valores correctos o de medios para recuperarlos, los errores detectados deben ser asignados a valor desconocido de manera automática. Este proceso se efectúa con la macro !CORREC, cuya documentación puede consultarse en la Documentación 4-7. La operativa de esta macro es sencilla: detecta, mediante las variables auxiliares $@V_i$, los casos que contienen errores y asigna éstos a valor desconocido. Es oportuno conservar el listado de incidencias obtenido con la macro !INCIDEN ya que representa el documento que registra el conjunto de cambios realizados sobre los datos. Sólo así es posible la auditación de los datos y el *roll-back*. La auditación de los datos se entiende en este contexto como la posibilidad de realizar una monitorización del valor desde que es emitido por el sujeto del estudio

hasta que es utilizado en el análisis estadístico. Cualquier cambio realizado, así como el motivo del mismo, debe constar documentalmente. Sólo este rigor permitirá el *roll-back* en caso de que sea necesario, es decir, retornar el valor a su guarismo original si posteriormente se ha averiguado que la corrección ha sido errónea.

La llamada a la macro !CORREC que se habría realizado para el archivo de test en caso de poder disponer de los valores correctos seria muy sencilla: incluiría el nombre de todas las variables auxiliares.

Documentación 4-7: Parámetros de la macro !CORREC.

```
DEPURACIÓN DE DATOS: Asignación automática a missing de incidencias
Creación 08.06.2003 Última revisión 07.07.2003
(c) A.Bonillo & JM. Doménech
Email: MacrosSPSS@metodo.uab.es

Llamada de la Macro:
!CORREC V= Lista de las variables auxiliares
       /C= Lista de variables identificadoras del sujeto
  [/INDX]= Variable de índice
      /CN= Variable identificadora de la secuencia de los sujetos
                                                (por defecto,@casenum)

Ejemplos de llamada:
!CORREC V= @h   @caso @fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie
          @pad @pas  @exitus /C=h caso.
```

Algoritmo 4-8: Asignación automática a valor desconocido de las incidencias detectadas.

Asignación de los parámetros:

1. Asignar a V la lista de variables auxiliares que contienen los códigos de error.
2. Asignar a C las variables que forman el identificador.
3. Asignar a CN la variable que contiene el número de secuencia de los casos.
   (Por defecto CN= casenum).

Proceso (para cada caso):

Sea $@V_i$ el elemento i de la lista V y V_i la variable original numérica.

4. Si $@V_i$ >= 1:
   4.1. Asignar V_i a vacío.

4.11 VALORACIÓN DE LA CALIDAD DE LOS DATOS

Finalizada la corrección de las incidencias se debe efectuar una valoración de la calidad de los datos, obteniendo el porcentaje de errores por sujeto y por variable detectados durante la depuración. Otro aspecto de la calidad es el porcentaje final de valores desconocidos por sujeto y por variable que no han podido ser recuperados.

El porcentaje de errores detectados durante el proceso de depuración es un índice de las inconsistencias contenidas en la matriz de datos original y de los valores que se han olvidado grabar. Este porcentaje se obtiene comparando la matriz de datos originales con la matriz de datos depurados. Toda discordancia entre un dato original y el correspondiente dato depurado se considera un error.

El porcentaje final de valores desconocidos en la matriz depurada se obtiene dividiendo el total de datos desconocidos que no han podido ser recuperados entre el total de datos excluyendo los datos ”no aplicables”.

No debe sorprender que las estadísticas obtenidas reproduzcan, por un lado, los errores que contenía la matriz original, mientras que por otro reproduzcan los valores desconocidos de la matriz final. Cabe señalar que ambas medidas no son en absoluto independientes. Es probable que muchos de los errores detectados hayan tenido que ser asignados a valor desconocido al no haberse recuperado el valor original. Obtener una estadística de los valores desconocidos de la matriz original no sería indicativo de la calidad de los datos, ya que su número se verá incrementado con los errores no recuperados.

La macro !QUALITY, cuyos parámetros se recogen en la Documentación 4-8, permite realizar de forma automática ambas valoraciones de calidad.

Documentación 4-8: Parámetros de la macro !QUALITY.

```
DEPURACIÓN DE DATOS: Valoración de la calidad de los datos
Creación 08.06.2001 Última revisión 07.07.2003
(c) A.Bonillo & JM. Doménech
Email: MacrosSPSS@metodo.uab.es
Llamada de la Macro:
!QUALITY V= Lista de las variables depuradas
  [/BREAK]= Lista de variables categóricas que definen subpoblaciones
     /FILE= Nombre (y ruta) del archivo sin depurar en formato SPSS (.SAV)

Ejemplos de llamada:
!QUALITY V=h caso fr fn sexo talla dpt dcs fuma tab tiptab cie pad pas exitus
     /FILE='TestNoDep.SAV.
```

La parte superior del Listado 4-25 contiene la llamada a la macro !QUALITY para el archivo de prueba. El parámetro V contiene la lista completa con los nombres de las variables depuradas. El parámetro FILE debe contener el nombre del archivo no depurado en formato SPSS (aunque el archivo original estuviera grabado en ASCII o DBF). Este archivo se debe grabar al inicio del proceso de depuración en el punto indicado en el Listado 4-24.

El parámetro BREAK permite definir subpoblaciones para las cuales se desean obtener estadísticas distintas. Si además obtener las estadísticas para el conjunto el archivo se deseara obtener estadísticas de incidencias para cada hospital se debería añadir /BREAK=H. Obtener estadísticas de incidencias por subpoblaciones es especialmente útil cuando se sospecha una alta variabilidad en la generación de incidencias entre los responsables de la captura de los datos. Si, retomando el ejemplo del archivo de test, se cree que algunos hospitales pueden haber sido poco diligentes con la captura, o bien en otro contexto, que algunos operadores son muy descuidados, puede ser interesante obtener estadísticas independientes para cada uno de ellos y en función de los resultados tomar decisiones. Por ejemplo, si en el archivo de test un hospital fuera el responsable de la mayoría de las incidencias se podría tomar la decisión de excluirlo del estudio o de sustituirlo por otro hospital más eficiente.

La parte inferior del Listado 4-25 recoge los resultados de esta llamada. En primer lugar se presentan los errores por caso, por variable y global. La tabla “Errores por caso” indica que hay 5 sujetos sin errores, 8 con un error, etc. La tabla “Errores por variable” muestra que los campos con mayor proporción de errores son FR (28%) y TAB, SEXO, FUMA, TIPTAB y FN (17%), y que las variables EXITUS, PAS, PAD y

DCS han sido grabadas sin ningún error. La tabla "Estadística de datos con error" indica en el total de los 522 datos había 53 errores que representan un 10.2%. Finalmente, se indica que se han eliminado 2 registros de los 20 iniciales (10.7%), de los cuales uno era un duplicado, mientras el otro no debiera haberse registrado por haber sido dado de baja.

Listado 4-25: Estadística de valores desconocidos que no han sido recuperados y errores con la macro !QUALITY.

```
!QUALITY V=h caso fr fn sexo talla dpt dcs fuma tab tiptab cie pad pas exitus
       /FILE='C:\...\Escritorio\TestNoDep.SAV'.
```

Errores por caso

	Frecuencia	Porcentaje
0	5	25.0
1	8	40.0
2	2	10.0
3	4	20.0
4	1	5.0

Errores por variable

	N	Suma	Prop.
FR	18	5	.2778
TAB	18	3	.1667
SEXO	18	3	.1667
FUMA	18	3	.1667
FN	18	3	.1667
TIPTAB	18	3	.1667
CASO	18	2	.1111
TALLA	18	2	.1111
H	18	2	.1111
DPT	18	1	.0556
CIE	18	1	.0556
EXITUS	18	0	.0000
PAS	18	0	.0000
PAD	18	0	.0000
DCS	18	0	.0000

Estadística de datos con error

	Frecuencia	Porcentaje
Errores	53	10.2
Correctos	469	89.8
Total	522	100.0

Registros eliminados: 2/20 (10.00%)

Valores missing por caso

	Frecuencia	Porcentaje
0	4	22.2
1	8	44.4
3	1	5.6
4	4	22.2
5	1	5.6

Valores missing por variable

	N	Suma	Prop.
TIPTAB	11	5	.4545
TALLA	18	6	.3333
TAB	18	5	.2778
FUMA	18	5	.2778
CIE	18	4	.2222
DCS	18	2	.1111
DPT	18	2	.1111
FN	18	2	.1111
PAD	18	1	.0556
EXITUS	18	0	.0000
PAS	18	0	.0000
SEXO	18	0	.0000
FR	18	0	.0000
CASO	18	0	.0000
H	18	0	.0000

Estadística de datos con valor missing

	N	Suma	Prop.
Valores missing recuperables	263	32	**.1217**
Valores missing no recuperables	263	3	.0114

A continuación se presenta la estadística de datos con valor desconocido del archivo depurado. La tabla "Valores missing por caso" indica que sólo 4 registros están completos (22%), 8 tienen un valor desconocido (44%), etc. La tabla "Valores missing por variable" muestra que el campo con mayor proporción de valores desconocidos es TIPTAB (45%) seguido de TALLA (33%). Finalmente, la tabla "Estadística de datos con valor missing" indica que del total de 263 valores aplicables, 32 no han sido recuperados (12.2%).

4.12 SÍNTESIS DEL PROCEDIMIENTO DE DEPURACIÓN

La Figura 4-5 muestra el diagrama de flujo del proceso de depuración propuesto, que consta de las siguientes fases:

1. Comprobación de identificadores
2. Corrección de las incidencias de los identificadores
3. Comprobación de las variables de salto
4. Corrección de las incidencias en las variables de salto
5. Comprobación del resto de variables
6. Informe de incidencias
7. Corrección de las incidencias de las variables
8. Estadísticas de valores desconocidos en los datos depurados y de errores detectados durante la depuración
9. Grabación de los datos depurados

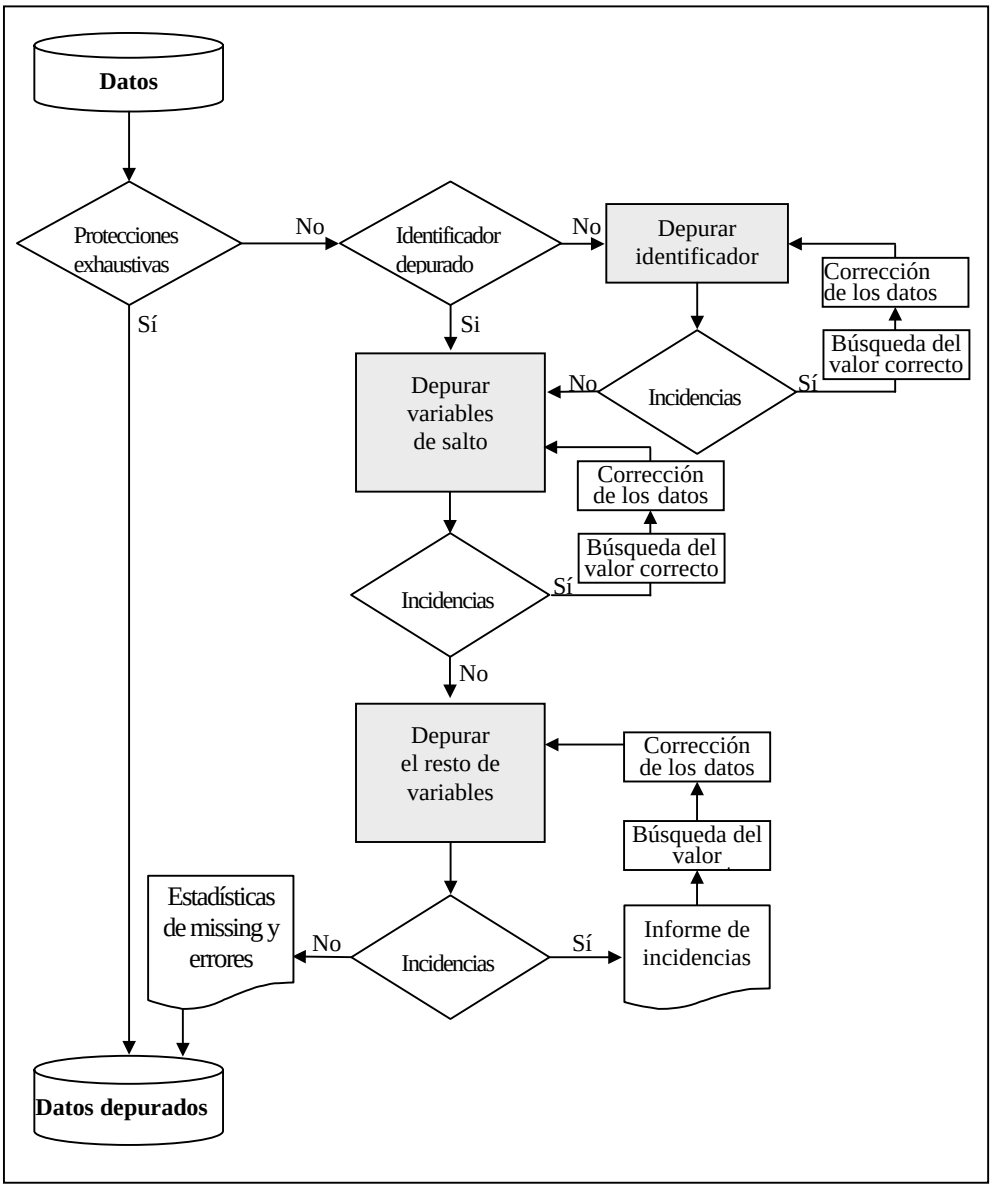


Figura 4-5: Secuencia del proceso de depuración.

Listado 4-26: Proceso completo de depuración del archivo de test.

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='TestNoDep.DAT'
  /h 2(A)
   caso   4-6(F)     ·caso  4-6(A)
   fr    8-17(EDATE) ·fr    8-17(A)
   fn   19-28(EDATE) ·fn    19-28(A)
   sexo   30(A)
   talla  32-35(F,2) ·talla 32-35(A)
   dpt    37(F)      ·dpt   37(A)
   dcs    39(F)      ·dcs   39(A)
   fuma   41(F)      ·fuma  41(A)
   tab    43-44(F)   ·tab   43-44(A)
   tiptab 46-47(A)
   cie    49-54(A).
   pad    56-58(F)   ·pad    56-58(A)
   pas    60-62(F)   ·pas    60-62(A)
   exitus 64   (F)   ·exitus 64  (A) .
EXECUTE.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = h sexo tiptab.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Grabación del archivo no depurado necesario para la estadística de calidad.
SAVE OUTFILE='TestNoDep.SAV'.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.


*Corrección de identificadores.
SELECT IF (@casenum<> 3).        /*Eliminación del registro tras su exitus.
SELECT IF (@casenum<>19).        /*Eliminación del caso duplicado.
IF (@casenum= 1) caso= 21.       /*Corrección de incidencias.
IF (@casenum=11) caso= 51.
IF (@casenum=17) h   = 'B'.
IF (@casenum=18) h   = 'B'.
IF (@casenum=13) fr  = DATE.DMY(7,9,1993).
IF (@casenum=14) fr  = DATE.DMY(13,5,1993).
EXECUTE.
*Corrección de la variable de salto.
IF (h='A' AND caso =133) fuma =$sysmis./*Corrección variables de salto.
IF (h='A' AND caso =133) •fuma  = ' '.
IF (h='A' AND caso = 14) fuma =$sysmis.
IF (h='B' AND caso = 16) fuma =0.
IF (h='B' AND caso = 17) fuma =0.

*Corrección del resto de variables del estudio.
IF (h='A' AND caso = 21) sexo   = 'M'.
IF (h='A' AND caso = 94) tab    = 0.
IF (h='A' AND caso = 12) sexo   = 'M'.
IF (h='A' AND caso = 12) dpt    = 1.
IF (h='A' AND caso = 12) tab    = $sysmis.
IF (h='A' AND caso =133) sexo   = 'F'.
IF (h='A' AND caso =133) tab    = $sysmis.
IF (h='A' AND caso = 14) talla = 1.75.
IF (h='A' AND caso = 14) tiptab= ' '.
IF (h='B' AND caso = 16) ·fn    =' '.
IF (h='B' AND caso = 17) fn     = DATE.DMY(1,11,1986)./* Nueva corrección.
IF (h='B' AND caso = 17) talla = 1.05.
IF (h='B' AND caso = 82) fr     = DATE.DMY(13,5,1993).
IF (h='B' AND caso = 82) fn     = DATE.DMY(20,11,1962).
IF (h='B' AND caso = 94) sexo   = 'M'.
IF (h='B' AND caso = 94) tab    = 15 .
IF (h='B' AND caso = 10) tiptab= 'NE'.
IF (h='B' AND caso = 10) cie    = '422.90'.

EXECUTE.
```

Listado 4-26: Proceso completo de depuración del archivo de test.

```
*Depuración del identificador.
!IDT V=h caso fr /OUT=Exitus /XOut=1 /TABLE='C:\...\Escritorio\Censal.SAV'
     /DROP=nombre to cpostal.
!DR V=h /LV='A','B' /MVr=' ' /C=caso  /INDX=fr /L=2.
!DR V=caso /LV=1 THRU 150 /ND=0 /FORMAT=1 /C=h  /INDX=fr /L=2.
!DRF V=fr /FI=1,4,1993 /FS=30,12,1993 /FORMAT=1 /VAR=HI /C=h caso /INDX=fr /L=2.


*Depuración de la(s) variable(s) de salto.
COMPUTE @fuma=$SYSMIS.      /*Condición lógica @fuma=50.
IF (fuma=1 AND CTIME.DAYS(fr-fn)<(365.25*11)) @fuma=50.
!DR V=fuma /LV=0,1 /MV=9 /VAR=0 /LVL=fn fr  /FORMAT=1 /C=h caso  /INDX=fr.


*Depuración de las variables del estudio.
!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /VAR=0 /FORMAT=1 /C=h caso
/INDX=fr.
!DR V=sexo /LV='M','F' /VAR=0 /MVr=' ' /C=h caso  /INDX=fr.
!DR V=talla /LV=0.95 THRU 2 /VAR=0 /ND=2 /VAR=0 /FORMAT=1 /C=h caso  /INDX=fr.
!DR V=dpt dcs /LV=1,2,3 /VAR=0 /FORMAT=1 /C=h caso  /INDX=fr.

!DR V=tab /LV=1 thru 80 /ND=0 /VS=fuma /MVS=9 /XS=0 /VD=0 /VAR=0 /FORMAT=1 /C=h
caso  /INDX=fr.
!DR V=tiptab /LV='NE','RU','NR' /MVr=' ' /VS=fuma /XS=0 /VAR=0 /MVS=9 /C=h caso
/INDX=fr.

*Comprobación variable CIE.
COMPUTE @cie=$SYSMIS.
IF NOT(RANGE(NUMBER(SUBSTR(cie,1,3),F3),390,459)) @cie=50.
!DRKey V=cie /TABLE='C:...\Escritorio\CIE9.SAV' /DROP=des
  /MVr=' ' /VAR=HI /LVL=@NULL /C=h caso  /INDX=fr.
!DR V=pad /LV=20 THRU 300 /ND=0 /VAR=30% /FORMAT=1 /C=h caso  /INDX=fr.
!DR V=pas /LV=50 THRU 400 /ND=0 /VAR=30% /FORMAT=1 /C=h caso  /INDX=fr.
!DR V=exitus /LV=0,1 /VAR=HI /FORMAT=1 /C=h caso  /INDX=fr.


*Informe de incidencias.
!INCIDEN V=@fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie /C=H CASO
/INDX=fr /EXCLUDE=2 @cie, 3 @dcs, 4 @talla @fuma @tab @tiptab @cie, 5 @fuma @tab
@tiptab @dcs @cie, 6 @dpt @dcs @talla @tab @fuma @tiptab, 7 @fn @talla @tab @fuma
@tiptab, 8 @fn @dpt @talla @tab @fuma @tiptab, 9 @talla @tab @fuma @tiptab @cie,
10 @tab @fuma @tiptab @cie, 11 @dpt @dcs, 12 @fn, 13 @dpt, 14 @talla, 15 @talla,
16 @talla, 20 @cie.

*Estadísticas de calidad.
!QUALITY V=h caso fr fn sexo talla dpt dcs fuma tab tiptab cie pad pas exitus
       /FILE='C:\...\Escritorio\TestNoDep.SAV'.

*Grabar los datos depurados en formato SPSS.
SAVE OUTFILE = 'C:\...\Escritorio\TestDep.SAV' /KEEP = h TO cie.

*Salir de este proceso sin grabar la ventana de datos.
```

El Listado 4-26 muestra el proceso completo para depurar el archivo de prueba. Este documento contiene las instrucciones de todos los cambios realizados en los datos junto a las comprobaciones efectuadas.

5
APLICACIÓN A ESTUDIOS CON GRANDES VOLÚMENES DE DATOS

5.1 PRESENTACIÓN

El proceso de depuración expuesto en el capítulo anterior ha sido desarrollado en varias etapas. En este capítulo se ilustran tres casos prácticos, siendo los dos primeros realizados con versiones anteriores de las macros presentadas. La experiencia obtenida con estas aplicaciones ha servido para optimizar el proceso de depuración.

Los tres principales argumentos que justifican la inclusión de estas aplicaciones en este trabajo son: 1) se trata de tres experiencias muy enriquecedoras con aportaciones importantes a nuestra propuesta de depuración; 2) ilustran la funcionalidad del análisis realizado y de las herramientas facilitadas para su ejecución; y 3) muestran la flexibilidad del proceso elaborado, ya que su adaptación para cubrir las necesidades particulares de cada estudio ha resultado muy sencilla.

5.2 ENCUESTA SOCIODEMOGRÁFICA

La primera aplicación que se expone en este capítulo corresponde a depurar la Encuesta Sociodemográfica (en adelante, ESD; INE, 1993). Esta depuración se efectuó como respuesta a la solicitud efectuada por el Instituto de Ciencias para la Familia de la Universidad de Navarra.

La recogida de datos con el ESD se llevó a cabo al mismo tiempo que el censo de 1991, seleccionando una muestra representativa de la población española de aquel año. La ESD ha sido utilizada para validar el censo de 1991 y con fines de investigación en el ámbito sociológico.

La ESD recoge información sociodemográfica de los individuos (IEA, 1997): biografía familiar, lugares de residencia, educación y actividad. Está estructurada en forma de árbol lógico y consta de 80 fichas que se agrupan en las 11 categorías presentadas en la

Tabla 5-1. Las fichas corresponden a campos de repetición de las 11 categorías principales; por ejemplo, en la categoría “Hermanos” cada sujeto tiene un máximo de 6 fichas.

El total de fichas disponibles para cada sujeto depende de lo extensa que sea su familia, de los cambios de residencia que ha efectuado o de la cantidad de trabajos distintos que ha desempeñado a lo largo de su vida. En la encuesta realizada en 1991 se recogió información sobre 157.154 individuos con edades iguales o superiores a 10 años.

El Instituto Nacional de Estadística facilitó el fichero en formato ASCII objeto de depuración. En este archivo cada ficha de cada sujeto ocupaba un registro, y en total se disponía de unos 3.5 millones de registros. Los datos habían sido previamente grabados con doble entrada, y tras su captura se habían sometido a un estudio de consistencia.

En esta aplicación la depuración se efectuó de forma estructurada y se organizó en función del tipo de categoría. En concreto, se elaboró un paquete de llamadas a las macros de comprobación para cada categoría y un bucle aplicaba estos mismos chequeos para todas las fichas. Por ejemplo, para la categoría Hermanos se elaboraron las llamadas a las macros y se implementó un bucle que las aplicaba de forma sistemática a todos los hermanos.

La depuración final consistió en comprobar un total de 1635 variables, empleando un tiempo de procesamiento total de 265 horas.

El Listado 5-1 recoge la estadística de errores por caso de la ESD. No hemos reproducido la estadística de valores desconocidos porque en muchas variables no era distinguible el *vacío* del no aplicable y se sobreestimaría el valor real.

En el Anexo 3 se adjunta el material elaborado para depurar la ESD. Se debe tener presente que algunas de las macros utilizadas tienen el mismo nombre pero corresponden a versiones anteriores de las presentadas en el Anexo 1.

Algunas de las particularidades del trabajo realizado son las siguientes:

- Puesto que los datos estaban grabados en formato ASCII, existía la posibilidad de que se hubieran producido movimientos de códigos en las columnas durante la captura de algunos registros. Como la mayoría de variables eran de tipo categórico y compartían un mismo conjunto de valores válidos, era poco probable detectar estos desplazamientos disponiendo únicamente de la acumulación de errores por caso que aporta la depuración. Por esta razón se optó por detectar este tipo de incidencia previamente al inicio de la depuración de las variables. La estrategia utilizada consistió en comprobar que las columnas que debían estar en blanco en cada registro (porque actuaban como separadores entre campos), realmente lo estaban.
- Puesto que se requiere mucho tiempo de procesamiento para realizar cualquier operación sobre 157.000 registros, se decidió utilizar la opción de visualización “borrador” en la ventana de resultados. Esto suponía que los resultados de los chequeos se mostraban en ASCII y no en el formato habitual de objetos del SPSS, reduciendo considerablemente el tiempo de procesamiento requerido.
- El proceso de depuración utilizado permite identificar tipologías de errores que, en caso de repetir el trabajo, permitirían mejorar la calidad de la información registrada. En las estadísticas de errores mostradas en el Listado 5-1 puede sorprender que 84470 casos (un 53.8% del total) tengan 4 errores. El motivo de esta situación es que se trata de un error por inconsistencia entre variables que están dentro de un salto respecto a la variable de salto. Esto provoca que todo el contenido del subformulario sea correcto o que se presenten cuatro errores. En concreto, a los informantes se les pregunta sobre si el sujeto vive en una vivienda familiar o en solitario. Muchos sujetos que contestan que viven en soledad responden luego a preguntas como “¿Cuál es su relación con la persona/s con la/s que convive?”, “¿Cuál de uds. era el propietario de la vivieda?” o “¿Continúa uds.

conviviendo con la/s misma/s persona/s?. Este error puede deberse a un error en como los entrevistadores plantean la pregunta o a una confusión en los términos en que está redactada, ya que da a entender (y el protocolo de recogida así lo especifica) que si no se convive con otras personas las preguntas sobre la convivencia no deberían contestarse. Gracias a la detección de la acumulación de un error concreto en los datos, en estudios siguientes podría entrenarse a los entrevistadores en este tipo de preguntas o reformular la redacción de las mismas.

- Para evitar que las irregularidades en el suministro eléctrico nos hicieran perder el trabajo realizado y acumulado, se diseñó una aplicación compilada específica que grababa tanto los datos como los resultados periódicamente (cada 15 minutos).
- Finalmente, debido a la gran cantidad de correcciones que se debían realizar sobre los datos, se programó una macro que generaba automáticamente un fichero de sintaxis con las instrucciones de cambios.

Tabla 5-1: Categorías de la ESD y número máximo de campos de repetición (fichas).

Categorías ESD	Campos de repetición (fichas)
Miembros del Hogar	16
Padres	4
Hermanos	6
Matrimonios y uniones maritales estables	4
Hijos	6
Lugares de residencia	16
Viviendas	8
Estudios académicos	7
Otros estudios	8
Biografía de actividad	4
Actividad actual	1

Tabla 5-2: Variables del CMDBAH.

Código de hospital	Fecha de admisión	Otros Diagnósticos 2	Otros Procedimientos 3
Número de historia clínica	Circunstancias de admisión	Otros Diagnósticos 3	Tiempo de gestación
Número de asistencia	Fecha de alta	Código CIE9 tipo E	Peso del primer neonato
Fecha de nacimiento	Circunstancias de alta	Procedimiento principal	Peso del segundo neonato
Sexo	Código del centro de traslado	Otros Procedimientos 1	Sexo del primer neonato
Código de residencia	Diagnóstico principal	Otros Procedimientos 2	Sexo del segundo neonato
Régimen económico	Otros Diagnósticos 1		

Listado 5-1: Estadísticas de errores por caso de la ESD.

```
!INCIDEN LV= @v00111 to @v20038 /OUTPUT=ERROR.
```

Número de errores por caso

	Frecuencia	Porcentaje
0	42059	26.8
1	20030	12.7
2	3677	2.3
3	4997	3.2
4	84470	53.8
5	1339	.9
6	360	.2
7	43	.0
8	71	.0
9	19	.0
10	24	.0
11	5	.0
12	5	.0
18	1	.0
Total	157100	100.0

5.3 CONJUNTO MÍNIMO DE DATOS BÁSICOS DE ALTA HOSPITALARIA

La segunda aplicación que presentamos en este trabajo corresponde al Conjunto Mínimo Básico de datos de Alta Hospitalaria, (en adelante, CMDBAH). Se trata de una base de datos clínico-administrativa que recoge información del alta de los episodios que originaron la hospitalización (Librero, Ordiñana y Peiro, 1998). Constituye una adaptación del *Uniform Hospital Discharge Data Set* (UHDDS), creado en 1972 en Estados Unidos por el *National Committee on Vital and Health Statistics*.

En nuestro país el CMDBAH se graba en cada hospital en soporte magnético por personal entrenado de las unidades de documentación clínica. Tras su envío al INSALUD o al servicio de salud autonómico, los datos de los diversos hospitales son incorporados a un único CMDBAH estatal dependiente del Consejo Interterritorial del Sistema Nacional de Salud (SNS).

El bajo coste de registrar el CMDBH (ya que se recogen únicamente 26 variables) y su amplia difusión lo convierten en la principal herramienta para construir indicadores relacionados con la atención hospitalaria (por ejemplo, mortalidad, reingresos, complicaciones, etc.) y poder comparar los distintos centros. Además, el CMDBAH también puede utilizarse para organizar sistemas de financiación y gestión hospitalaria. El CMDBAH resulta tan revelante que distintos trabajos han hecho hincapié en la importancia de minimizar sus errores (Corn, 1991; Guilabert, Peres, Almela y Company, 1995; Librero, Ordiñana y Peiro, 1998), pese a que ninguno ha expuesto ni la metodología a seguir ni los algoritmos a implementar.

A solicitud del Servei Català de la Salut (en adelante, SCS) se procedió a construir un programa que permitiera depurar regularmente el CMDBAH (Tabla 5-2). En el caso del sistema público sanitario de Cataluña, los centros hospitalarios remiten trimestralmente un fichero con las altas producidas y registradas en el CMDBAH durante ese período. Cuando el SCS recibe este archivo efectúa una depuración para detectar inconsistencias y valores desconocidos utilizando nuestra propuesta.

Seguidamente se crean algunos indicadores, como la edad del paciente y la estancia hospitalaria. Finalmente, los datos libres de inconsistencias se incorporan a un fichero histórico y se retorna a cada centro un archivo con los datos corregidos y una hoja resumen de la información válida y no válida para cada una de las variables.

El procedimiento seguido para elaborar el programa de depuración consistió en realizar un análisis de las condiciones lógicas que debía garantizar la información grabada, y a continuación se elaboraron las llamadas a las macros de depuración para detectar cualquier incidencia.

- Se realizaron algoritmos que permitieran la obtención de estadísticas de errores por grupos de variables (en lugar de por variable) para facilitar su interpretación (véase Listado 5-2). Se debe tener presente que uno de los objetivos de la depuración por parte del SCS es reenviar a los hospitales estadísticas de las inconsistencias detectadas con el objetivo de poder minimizarlas en futuras ocasiones. Esto hacía indispensable la obtención de estadísticas personalizadas y fácilmente inteligibles por personal lego.
- Dado que las comprobaciones que se debían realizar eran idénticas para todos los ficheros llegados de los distintos hospitales, una vez verificado el correcto funcionamiento de las llamadas de chequeo, éstas se introdujeron en una nueva macro que las automatizaba. Esta macro final permitía efectuar el conjunto de comprobaciones sobre diferentes ficheros simultáneamente.
- Finalmente, puesto que el uso del CMDBAH requiere la integración de los datos de los diferentes hospitales, se diseñó una sencilla macro para evitar que se produjeran errores como consecuencia de esta operación. Esta macro permite fusionar en un único fichero los datos de un trimestre de los 184 hospitales y clínicas catalanas que trimestralmente envían sus datos al SCS.

De este trabajo nos gustaría destacar algunas particularidades:

- Puesto que en la actualidad no existe posibilidad de corregir las incidencias detectadas, se optó por asignar de forma automática cualquier inconsistencia en los datos a valor desconocido.

El Listado 5-2 presenta la estadística de errores por variable del fichero de test del CMDBAH facilitado por el SCS para el desarrollo del proceso de depuración.

En el Anexo 4 se adjunta el material elaborado para depurar el CMDBAH. Debe tener presente que algunas de las macros utilizadas tienen el mismo nombre pero corresponden a versiones anteriores de las presentadas en el Anexo 1.

Listado 5-2: Estadísticas de errores del archivo de test utilizado para desarrollar el proceso de depuración del CMDBAH.

```
!CMDBAH ARXIU="6KL9904.TXT"  /HOSP="H08000999" /DI=1,9,1999 /DF=31,12,1999.
```

Identificadors

	Correcte		Incorrecte	
	N	%	N	%
Codi de l'hospital	1607	99.9%	2	.1%
Numero d'història clínica	1609	100.0%		
Número d'assistència	1609	100.0%		

Sociodemogràfiques

	Correcte		Error Format		Missing		Incorrecte	
	N	%	N	%	N	%	N	%
Sexe	1606	99.8%			3	.2%		
Data de naixement	1144	71.1%	462	28.7%			3	.2%
Codi de residència	1597	99.3%			12	.7%		

Administratives

	Correcte		Error Format		Missing		Incorrecte	
	N	%	N	%	N	%	N	%
Règim econòmic	1604	99.7%			5	.3%		
Data d'admissió	1399	86.9%	203	12.6%			7	.4%
Data d'alta	1607	99.9%	2	.1%				

Dades pacient

	Correcte		Missing		Incorrecte		Incong. amb salt	
	N	%	N	%	N	%	N	%
Circumstàncies admissió	1607	99.9%	1	.1%	1	.1%		
Circumstàncies d'alta	1609	100.0%						
Codi del centre trasllat	31	93.9%	1	3.0%			1	3.0%

Incong. amb salt: manca el codi de trasllat a les circumstàncies d'alta

Diagnòstics

	Correcte		Missing	
	N	%	N	%
Diagnòstic principal	1556	96.7%	53	3.3%
Altres diagnòstics-1	1609	100.0%		
Altres diagnòstics-2	1609	100.0%		
Altres diagnòstics-3	1609	100.0%		

Procediments

	Correcte	
	N	%
Procediment principal	1609	100.0%
Altres procediments-1	1609	100.0%
Altres procediments-2	1609	100.0%
Altres procediments-3	1609	100.0%

Perinatals

	Correcte		Missing	
	N	%	N	%
Temps de gestació	1556	96.7%	53	3.3%
Pes del primer nadó	1595	100.0%	14	0.9%
Sexe del primer nadó	1597	100.0%	12	0.7%

Causes Externes

	Correcte		Missing	
	N	%	N	%
Codi E	1556	96.7%	53	3.3%

5.4 DEPURACIÓN DE LA HISTORIA CLÍNICA ELECTRÓNICA

La Historia Clínica Electrónica es una herramienta informática implementada por el Servei Català de la Salut (en adelante SCS). El objetivo de este instrumento es recoger todas las actuaciones que la sanidad pública realice sobre un paciente, así como la información básica que sobre él disponga. En este registro se incluye, por mencionar algunas áreas, toda la información recogida en la asistencia primaria, las hospitalaciones programadas y las analíticas que se le han practicado.

Con el registro informatizado se pretende lograr dos objetivos. Por un lado, maximizar la calidad del registro de la información y ponerla a disposición de todos los centros para que dispongan de ella. Este aspecto es fundamental para dar una asistencia de calidad, ya que actualmente un paciente sólo dispone de su historial clínico en el hospital o centro de asistencia primaria al que está asignado. Cuando la Historia Clínica Electrónica se halle plenamente implantada en todo el territorio catalán, cosa que se preveé que ocurra en unos 5 años, se podrá acceder a su historial clínico completo desde cualquier punto del sistema sanitario, aunque el paciente no esté administrativamente asignado él. Por otro lado, el SCS pretende fomentar la investigación en sus profesionales, prestando especial atención al uso de muestras compuestas por sujetos de nuestra región. Para facilitar la investigación, se pretende centralizar la captura de los datos, maximizando su calidad, y gestionar los mismos desde un organismo central dedicado a ello a tiempo completo. Así, el investigador sólo debería descargar desde una página web las variables que necesite para la población que seleccione, pudiéndose despreocupar de todo el preproceso. Se cree, justificadamente, que esto facilitará la investigación entre los profesionales sanitarios.

La Historia Clínica Electrónica se compone de una serie de bases de datos, facilitando la captura un conjunto de formularios diseñados a tal efecto. Se han implentado protecciones para evitar que las variables recogidas en un mismo momento tomen valores incoherentes entre sí. Ahora bien, puesto que los términales que permiten introducir los datos no acceden a la base de datos histórica se producen valores incoherentes respecto a información recogida en anteriores visitas.

Debido a que la Historia Clínica Electrónica registra un número enorme de variables, el SCS confeccionó un archivo de test con una selección de las mismas para aplicar los algoritmos propuestos previamente a hacerlo sobre la base de datos histórica. La aplicación de los algoritmos sobre la base de datos histórica aún no se ha realizado.

Al igual que en los ejemplos anteriores, el procedimiento seguido para elaborar el programa de depuración consistió en realizar un análisis de las condiciones lógicas que debía garantizar la información grabada, y a continuación se elaboraron las llamadas a las macros de depuración para detectar cualquier incidencia.

De este trabajo nos gustaría destacar algunas particularidades:

- La depuración es un proceso útil para detectar tipologías de error. Aunque su propósito principal es hallar errores y corregirlos en datos que han sido capturados sin protecciones exhaustivas, también puede ser útil para detectar tipologías de error e implementar protecciones en los formularios de captura. En este trabajo éste es uno de los objetivos básicos: comprobar si determinadas tipologías de

errores se dan e implementar protecciones en los formulario de captura que los minimice.

- Este trabajo ha permitido analizar la tipología de errores que deben detectarse en datos que contienen seguimientos. A partir de esta aplicación, se confeccionaron controles que permitieran detectar variaciones excesivas en variables cuantitativas, variaciones no esperables en variables categóricas cuyo valor no puede variar, entre otros.

En el Anexo 5 se adjunta el material elaborado para depurar la Historia Clínica Electrónica. En la Tabla 5-3 puede verse la lista de variables, y su descripción, que componían el archivo de test. En el Listado 5-3 se presenta la estadística de errores del archivo de test utlizado para comprobar el correcto funcionamiento de los algoritmos.

Tabla 5-3: Lista y descripción de variables del archivo de test de la Historia Clínica Electrónica.

Nombre	Descripción de la variable
TAS	Tensión arterial sistólica
A8991	Continente de origen del paciente (1=europeo;2=magrebí;3=subsahariano;4=americano;5=asiático)
PES	Peso del paciente en Kg.
TAD	Tensión arterial diastólica
C	Número de piezas dentales cariadas
TALLA	Talla (m.)
P	Proyecciones craniales realizadas
A	Número de piezas dentales ausentes
ASET	Alcohol puro ingerido a la semana (gr.)
APCT	Alcohol puro ingerido en el día de ayer (gr.)
H	Habitage (0: adecuado, 1: Inadecuado;2: Inaceptable)
IMC	Índice de masa corporal (kg/m^2)
FC	Frecuencia cardíaca (latidos/minuto)
N	Número de Enfermedades anteriores destacables
D	Toma de diuréticos (1: Sí; 2: No)
COLTOT	Colesterol total en sangre (mg/dl)
TG	Triglicéridos en ayunas (mg/dl)
GLU	Glucemia en plasma venoso en ayunas (mg/dl)
E	Frecuencia en la práctica de deporte (0=No;1=Inicio;2=Moderado.;3=Alta frecuencia)

Listado 5-3: Estadísticas de errores del archivo de test utilizado para desarrollar el proceso de depuración de la Historia Clínica Electrónica.

```
!QUALITY V=a8991 apct aset a c coltot d e fc glu h imc n p pes tad tas talla tg
     /FILE='C:\...\Escritorio\HClinOr.SAV'.
```

Estadística de datos con error

	Frecuencia	Porcentaje
Errores	883099297	24.5
Correctos	2726651275	75.5
Total	3609750572	100.0

Errores por caso

		Frecuencia	Porcentaje
Válidos	0	89413	64.2
	1	45519	32.7
	2	4283	3.08
	Total	139215	100.0

Proporción de errores por variable

	Suma	Prop.
TAS	33119.00	.2379
A8991	28368.00	.2038
PES	19038.00	.1368
TAD	17806.00	.1279
C	15313.00	.1100
TALLA	13684.00	.0983
P	12155.00	.0873
A	11883.00	.0854
ASET	10929.00	.0785
APCT	9795.00	.0704
H	9687.00	.0696
IMC	6405.00	.0460
FC	6352.00	.0456
N	5946.00	.0427
D	5556.00	.0399
COLTOT	5343.00	.0384
TG	4540.00	.0326
GLU	4455.00	.0320
E	3971.00	.0285

6
CONCLUSIONES Y DISCUSIÓN

El diseño de depuración propuesto en este trabajo se organiza en las siguientes fases: 1) lectura de los datos grabados; 2) depuración del identificador, garantizando que cada registro está unívocamente identificado y que se adecúa a las formas normales de integridad referencial de la teoría relacional; 3) corrección de las incidencias detectadas en el indentificador 4) depuración de las variables de salto, 5) corrección de las incidencias detectadas en las variables de salto, 6) depuración del resto de variables del estudio, detectando las incidencias contenidas en los datos grabados que son consecuencia de inconsistencias y valores desconocidos; 7) corrección de las incidencias detectadas, introduciendo siempre que sea posible el valor correcto o asignando a valor desconocido cuando no se disponga de éste; y 8) obtención de una estadística de los errores detectados por la depuración y de los valores desconocidos contenidos en los datos finales.

Nuestra propuesta de depuración constituye un proceso sistemático, integral y acumulativo. Las fases de chequeo y corrección se deben realizar de forma iterativa hasta que las únicas incidencias detectadas sean valores desconocidos no recuperables. Asimismo, este proceso de depuración debe acompañarse de un historial de cambios que permita conocer todas las modificaciones efectuadas a partir de los datos originales. Los controles sistemáticos de este diseño deben integrarse a través de macros (por ejemplo en lenguaje SPSS o SAS) que los automaticen y garanticen su funcionalidad. En este trabajo hemos elaborado las macros en sintaxis SPSS para efectuar la depuración; los algoritmos contenidos en estas macros son fácilmente transportables a otros paquetes estadísticos.

Quisiéramos destacar que nuestra propuesta de depuración se ha mostrado sólida y flexible en las aplicaciones realizadas con datos reales. Sólida porque ha permitido depurar tres bases de datos con problemáticas muy diferentes. La mayor complejidad de depurar la ESD estriba en su estructura, ya que es un protocolo muy estructurado que cuenta con una gran cantidad de campos. En la ESD es posible hallar ejemplos de prácticamente todo tipo de campos y de subestructuras, saltos anidados, grupos de repetición y condiciones lógicas que implican múltiples variables. En el caso del CMDBAH la mayor complejidad reside en las incompatibilidades lógicas entre variables que registran códigos diagnósticos y datos censales. En ambas aplicaciones el procedimiento de depuración propuesto permitió detectar inconsistencias y datos desconocidos con la misma sistemática y de forma muy eficiente. A la solidez del

diseño se debe añadir su flexibilidad ya que fue posible particularizar los requisitos de cada estudio de forma sencilla. En el caso de la Historia Clínica Electrónica, su principal dificultad estriba en la detección de errores a lo largo de los seguimientos.

En cualquier caso, a pesar de las buenas propiedades demostradas por el proceso de depuración elaborado, se requiere comprobar su funcionalidad con otras bases de datos para detectar si existen nuevas estructuras de tipo sistemático que pudieran ser implementadas en nuestra propuesta.

Por otro lado, sería interesante implementar controles de tipo probabilístico a los controles ya expuestos. Queda pendiente el análisis de las técnicas que sean de uso más frecuentes en el ámbito psicológico.

REFERENCIAS

Abelson, R.P. (1998). *La estadística razonada: reglas y principios*. Barcelona: Paidós. (Edición original: Erlbaum, 1995).

Arnau, J. (1996). *Model general d'investigació psicológica*. Barcelona: Fundació per la UOC.

Bannert, M., y Kunkel, K. (1991). The design of computer-based diagnosis systems: what can be learned from research in human-computer interaction? *Revue Européene de Psychologie Appliquée, 41,* 271–278.

Barton, C., Hatcher, C., Schurig, K., Marciano, P., Wilcox, K., y Brooks, L. (1991). Managing data entry of a large-scale interview project with optical scanning hardware and software. *Behavior Research Methods, Instruments, and Computers, 23,* 214–218.

Batra, D., y Srinivasan, A. (1992). A review and analysis of the usability of data management environments. *International Journal of Man-Machine Studies, 36,* 395–417.

Bayés, R. (1984). *Una introducción al método científico en Psicología*. Barcelona: Martínez Roca.

Benford, F. (1938). The law of anomalous numbers. *Proceedings of American Philosophical Society, 78,* 551–572.

Bernstein, F. (1998). The law of anomalous numbers. *Business Week, 10-07-1998*.

Berton, L. (1995). He's got their number: scholar uses math to foil financial fraud. *The wall street journal, 10-07-1995*.

Bienias, J.L. (1995). *Methods of Outlier Detection for the Quaterly Financial Report*. Nass Research Report RD 97-04. Nationatl Agricultural Statistics Service: Washinton, DC.

Bitton D. y DeWitt, D.J. (1983) Duplicate Record Elimination in Large Data Files, *ACM Transactions on Database Systems 8*,(*2),* 255-265.

Bobrowski, M., Marré, M. y Yankelevich, D. (1999). *Measuring Data Quality*. Report n.: 99-002. Acceso Internet: ftp://zorzal.dc.uba.ar/pub/tr/1999/99-002.pdf

BOE Real Decreto 561/1993, (de 16 de abril), por el que establece los requisitos para la realización de ensayos clínicos con medicamentos. *Boletín Oficial del Estado* (Publicación: 13/05/1993).

Bonillo, A., Doménech, J.M. y Granero, R. (2000). *Macros SPSS para análisis de datos en Ciencias de la Salud.* Barcelona: Signo.

Booth, P.A. (1991). Errors and theory in human-computer interaction. *Acta Psychologica, 78,* 69–96.

Boyle, J. (1994). An appllication of Fourier series to most significant digit problem. *The American Mathematical Monthly, 101,* 879–886.

Browne, M. (1998). Following Benford's Law. *The New York Times, 04-08-1998*.

Bureau, M.S. y Sistla, M. (1986). A comparison of the different imputation techniques for quantitative data. Working paper, BSMD.

Butcher, J.N. (1994). Psychological assessment by computer: potential gains and problems to be avoid. *Psychiatric Annals, 24,* 20–24.

Cardiff Software Inc. (1998a). *TELEform. Versión 6* [Programa para ordenador]. San Marcos, CA: Autor.

Cardiff Software Inc. (1998b). *TELEform Standard User Guide. Versión 6*. San Marcos, CA: Autor.

Carslaw, C. (1988). Anomalies in Income Numbers: Evidence of Goal Oriented Behaviour. *The Accounting Review, 63,* 321–327.

Carroll, J.M. (1993). Creating a design science of human-computer interaction. *Interacting with Computers, 5,* 3–12.

Clarke, P. A. (1993). Data validation. En R.K. Rondel, S.A Varley y C.F. Weeb (Eds.) *Clinical data management* (pp. 189-212). Chichester: John Wiley & Sons.

Christian, C. y Gupta, S. (1993). New evidence on "Secondary Evasion". *The Journal of the American Taxation Association,* 72–93.

Cobos, A. (1995). El síndrome GIGO. *JANO, 49,* 481–482.

Codd, E.F. (1985a). Does your DBMS run by the rules?. *Computerworld, 21-10-1985.*

Codd, E.F. (1985b). Is your DBMS really relational?. *Computerworld, 14-10-1985.*

Cody, R. (1999). *Cody's Data Cleaning Techniques Using Sas Software*. Cary, NC: SAS Institute Inc.

Cohen, D. (1976). An explanation of the first digit phenomenon. *Journal of Combinational Theory, 20,* 367-370.

Connett, J.E., y Lee, W.W. (1990). Estimation of the coefficient of variation from laboratory analysis of split specimens for quality control in clinical trials. *Controlled Clinical Trials, 11,* 24–36.

Conthe, M. (2001). Hoyo en 1. *Expansión, 10-07-2001*.

Corn, R.F. (1991). The sensitivity of prospective hospital reimbursement to errors in patient data. *Inquiry, 18,* 351–360.

Crawford, S.L., Teenstedt, S.L., y McKinlay, J.B. (1995). A comparison of analytic methods for non-random missingness of outcome data. *Journal of Clinical Epidemiology, 48,* 209–219.

Cronbach, L.J., (1970). *Essentials of psychological testing.* Harper & Row: New York.

Dasu, T. y Johnson, T. (2003). *Exploratory data mining and data cleaning*. John Wiley & Sons: New Jersey.

Date, C.J. (1999). *An introduction to Data Base Systems (7ª ed.).* Reading, MA: Addison Wesley.

Department of Health, Education, and Welfare. National Committee on Vital and Health Statistics (1972). *Uniform Hospital Discharge Data Minimum Data Set. DHEW Pub. No. (PHS) 80-1157*. Hyattsville, MD: U.S. Department of Health, Education, and Welfare.

DISA (Defense Information System Agency) (2001). DOD guidelines on data quality management. Dirección Internet: *http://www-datadmn.itsi.disa.mil/dqpaper.pdf.*

Dix, M., Finlay, J., Abowd, G., y Beale, R. (1993). *Human-Computer interaction.* New York: Prentice-Hall.

Doménech, J.M. (2001). *Proceso de datos sanitarios con el Sistema SPSS.* Campus de Bellaterra, Barcelona: Laboratori d'Estadística Aplicada i de Modelització. Universitat Autònoma de Barcelona.

Doménech, J.M., Losilla, J.M., y Portell, M. (1998). La verificació aleatòria: una estratègia per millorar i avaluar la qualitat de l'entrada de dades. *QÜESTIO. Quaderns d'Estadística i Investigació Operativa, 22,* 493-510.

Drasgow, F., Levine, M.V. y McLaughlin, M.E. (1985). Appropiateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology, 38,* 67-86.

Eason, K.D. (1991). Ergonomic perspectives on advances in human-computer interaction. *Ergonomics, 34*, 721–741.

Efron, B. (1994). Missing data, imputation, and the bootstrap (with discussion). *Journal of the American Statistical Association, 89,* 463–478.

Ellis, J.L. y van der Wollenberg (1993). Local homogeneity in latent trait models. A characterization of the homogeneous monotone IRT model. *Psychometrika, 58,* 417-429.

Ember, C.R. (1986). The quality and quantity of data for cross-cultural studies. *Behavior Science Research, 20,* 1–16.

Espeland, M.A., Byington, R.P., Hire, D., Davis, V.G., Hartwell, T., y Probstield, J. (1992). Analysis strategies for serial multivariate ultrasonographic data that are incomplete. *Statistics in Medicine, 11,* 1041–1056.

Espinosa, E., Zamora, P., y Feliu, J. (1996). Reflexiones sobre la ley de ensayos clínicos. *Medicina Clínica (Barc), 106,* 24-26.

Feekin, A.y Chen, Z. (2000). Duplicate Detection Using K-way Sorting Method. *Applied Computing 2000, Proceedings of the 2000 ACM Symposium on Applied Computing,* 323-327

Fellegi, I. P. y Sunter, A. B. (1969). A Theory of Record Linkage. *Journal of the American Statistical Association, 64,* 1183-1210.

Flehinger, B. (1966). On the probability that random integer has inicial digit "A". *American Mathematical Monthly, 73,* 1056-1061.

Fowler, F.J. (1993). *Survey research methods.* Newbury Park: SAGE Publications.

Freedland, K.E., y Carney, R.M. (1992). Data management and accountability in behavioral and biomedical research. *American Psychologist, 47,* 640–645.

Furry, W. y Hurwitz, H. (1945). Distribution of Numbers and Distribution of Significant Figures. *Nature, 155,* 52–53.

Galhardas, H., Florescu, D., Shasha, D. y Simon, E. (2000a). Extensible framework for data cleaning. *Proceedings International Conference on Data Engineering* 312: 312.

Galhardas, H., D. Florescu, , Shasha, D. y Simon, E. (2000b). AJAX: An extensible data cleaning tool. *Sigmod Record 29(2),* 590-590.

García, J.F. (1993). Impacto de la normativa legal en la calidad de los ensayos clínicos realizados en España. *Medicina Clínica, 100,* 770-777.

Gassman, J.J., Owen, W.W., Kuntz, T.E., Martin, J.P., y Amoroso, W.P. (1995). Data quality assurance, monitoring and reporting. *Controlled Clinical Trials, 16 (2 Suppl.),* 104S–136S.

Gibson, D., Harvey, A.J., Everett, V., y Parmar, M.K. (1994). Is doble data entry necessary? The CHART trials: Continuous, Huperfractionated, Accelerated Radiotherapy. *Controlled Clinical Trials, 15,* 482–488.

Gilabert A, Perez López JJ, Almela V, Company V. (1995). Caracterización de la cirugia mayor ambulatoria en un hospital. *Revista de Calidad Asistencial, 5,* 287-293.

González, H. (1993). Psicología de las interfaces: usuario-sistema: teorías y métodos. *Revista Intercontinental de Psicología y Educación, 6,* 35–61.

Goudsmit, S.A. y Furry, W. (1944). Significant Figures of Numbers on Statistical Tables. *Nature, 154,* 800–801.

Granero, R. (1999). Mejora de la calidad de la gestión de datos clínicos a través de sistemas computerizados. Aplicación a la informatización de la entrevista diagnóstica estructurada DICA-IV. Tesis Doctoral. Universitat Autònoma de Barcelona.

Granero, R. y Doménech, J.M. (1997, septiembre). *Calidad del proceso de datos en la evaluación psicopatológica: problemática de la presencia de valores "missing" y no aplicables.* Ponencia presentada al V Congreso de Metodología de las CC Humanas y Sociales, Sevilla, España.

Granero, R. y Doménech, J.M. (2001). Captura de datos clínicos con verificación aleatoria: una nueva técnica para controlar y mejorar la calidad del registro. *Psicothema, 13* (1), 166-172.

Granero, R., Doménech, J.M. y Bonillo, A. (2001). Estudio de la eficacia de la técnica de verificación aleatoria como alternativa a la doble entrada de datos: ventajas y limitaciones. *Metodología de Encuestas, 3,* 1-13.

Greenland, S., y Finkley, W. (1995). A critical look at methods for handling missing covariates in epidemiologic regression analyses. *American Journal of Epidemiology, 142,* 1255−1264.

Groves, R.M. (1989). *Survey errors and survey costs.* New York: John Wiley & Sons.

Guilabert, A., Pérez López, J.J., Almela, V. y Company, V. (1995). Calidad de datos y grupos relacionados con el diagnóstico. *Prevista de Calidad Asistencial, 5,* 287-293.

Hamming, R. (1970). On the distribution of numbers. *The bell system technical journal, 49(8),* 1609−1625.

Harnisch, D.L., y Linn, R.L. (1981). Analysis of item responser patterns: Questionable test data and dissimilar curriculum practices. *Journal of Educational Measurement, 18,* 133-146.

Hartson, H.R., y Hix, D. (1989). Toward empirically derived methodologies and tools for human-computer interface development. *International Journal of Man-Machine Studies, 31,* 477−499.

Haviland, M.G. (1990). Yate's correction for continuity and the analysos of 2×2 contingency tables. *Statistics in Medicine, 9,* 363-383.

Hernandez, M.A. y Stolfo, S.J. (1995). The Merge/Purge Problem for Large Databases. En Michael J. y Schneider, D. (editors): Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data. Acceso Internet: http://citeseer.nj.nec.com/stolfo95mergepurge.html

Hernandez, M.A. y Stolfo, S.J. (1998). Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. *Data Mining and Knowledge Discovery, 2,* 1-31.

Hidiroglou, M.A. y Berthelot, J.M. (1986). Statistical editing and imputation for periodic business surveys. *Survey Methodology, 111(1),* 73-83.

Hill, T. P. (1996a). A note on distributions of true versus fabricated data. *Perceptual and Motor Skills, 83,* 776-778.

Hill, T. P. (1996b). A statistical derivation of the significant-digit law. *Statistical Science,10,* 354-363.

Hill, T. P. (1996c). The first-digit phenomenon. *American Scientists, 86,* 358-363.

Hill, T. P. (1997). Benford's Law. *Encyclopedia of Mathematics Supplement.1,* 102-105.

Hill, T. P. (1999). The difficulty of faking data. *Chance, (26),* 8-13.

Holland, P. W. (1990). On the sampling theory foundations of item response theory models. Psychometrika,*55,* 577-601.

Huber, P.J. (1981). *Robust statistics*. Nueva York: John Wiley.

Huberty, C.J., y Julian, M.W. (1995). An ad hoc analysis strategy with missing data. *Journal of Experimental Education, 63*, 333-342.

ICH (International Conference on Harmonization) (1994). Note for Guidance on Good Clinical Safety Data Management: The Standards for Expedited Reporting. Dirección Internet: *http://www.emea.eu.int/pdfs/human/ich/037795en.pdf.* (CPMP/ICH/377/95).

ICH (International Conference on Harmonization) (1996). Note for Guidance on Clinical Safety Data Management: Periodic Safety Update Reports for Marketed Drugs. Dirección Internet: *http://www.emea.eu.int/pdfs/human/ich/028895en.pdf* (CPMP/ICH/288/95).

ICH (International Conference on Harmonization) (1998). Guidance on Statistical Principles for Clinical Trials. Dirección Internet: *http://www.emea.eu.int/pdfs/human/ich/036396en.pdf.* CPMP/ICH/363/96).

IEA (Instituto de Estadística de Andalucía) (1997). *Un análisis de la ESD de 1991.* Sevilla: Autor.

INE (Instituto Nacional de Estadística) (1993). *Encuesta Sociodemográfica 1991. Metodología,* Madrid: Autor.

Jáñez, L. (1989). *Fundamentos de Psicología matemática*. Madrid: Pirámide.

Jennings, R. (1997). *ACCESS 97. Edición especial.* Madrid: Prentice Hall.

Johnson, P. E. (1992). *Human Computer Interaction: Psychology, Task Analysis and Software Engineering*. Londres: McGraw-Hill.

Johnson, W.D., George, V.T., Shahane, A., y Fuchs, G.J. (1992). Fitting growth curve models to longitudinal data with missing observations. *Human Biology, 64,* 243–253.

Kochhar, S. (1991). Development of diagnostic data in the 10-percent sample of disabled SSI recipients. *Social Security Bulletin, 54,* 10–21.

Lewis-Beck, M.S. (1995). *Data analysis: an introduction*. Beverly Hills, CA: Sage.

Levine, M.V. y Drasgow, F. (1983). Appropriateness measurement: Validating studies and variable ability modelss. En D.J. Weiss (Ed.) *New horizons in testing: latent trait test theory and computerized adaptative testing* (pp. 109-131). New York: Academic Press.

Librero, J., Ordiñana, R. y Peiró, S. (1998). Análisis automatizado de la calidad del conjunto mínimo de datos básicos. Implicaciones para los sistemas de ajuste de riesgos. *Gaceta Sanitaria, 12,* 9-21.

Little, R.J.A., y Rubin, D.B. (1987). *Statistical analysis with missing data*. Nueva York: John Wiley.

Losada, J.L. y López-Feal, R. (2003). *Métodos de investigación en Ciencias Humanas y Sociales*. Madrid: Thomson.

Lowgren, J., y Lauren, U. (1993). Supporting the use of guideliness and style guides in professional user interface design. *Interacting with Computer, 5,* 385–396.

Magnusson, D., y Bergman, L.R. (1990). *Data quality in longitudinal research.* Cambridge: Cambridge University Press.

Maletic, J. y Marcus, A. (2001). *Supporting Program Comprehension Using Semantic and Structural Information*. International Conference on Software Engineering", 103-112.

Maney, K. (2000). Baffled by math. USA Today, *18-10-2000*.

Marcus, S.C., y Robins, L.N. (1998). Detecting errors in scoring program: a method of double diagnosis using a computer-generated sample. *Social Psychiatry and Psychiatric Epidemiology, 33,* 258–262.

Martin, M.P., y Fuerst, W.L. (1987). Using computer knowledge in the design of interactive systems. *International Journal of Man-Machine Studies, 26,* 333–342.

Martinez-Arias, R. (1996). *Psicometría: Teoría de los Tests Psicológicos y Educativos*. Madrid: Editorial Síntesis.

Melgratti, H. y Yankelevich, D. (2000). Tools for Data Quality. *Technical Report 99-005* Dirección Internet: *http://www.dc.uba.ar/people/proyinv/arte/trabajos.htm*

Ministerio de Sanidad y Consumo (1989). *Clasificación Internacional de Enfermedades, 9ª revisión. Modificación Clínica*. Madrid: Autor.

Molina, L.C. (2000). *Data mining: una introducción.* Barcelona: Universitat Oberta de Catalunya.

Monge, A. E. (1998). Adaptive detection of approximately duplicate database records and the database integration approach to information discovery. Tesis Doctoral: University Of California San Diego.

Monge, A. (2000a). An adaptative and efficient algorithm for detecting approximately duplicate database records. *Bulletin of the IEEE Computer Society Technical Committee of Data Engineering, 23(4),* 14-20. Dirección Internet: *http://www.research.microsoft.com/research/db/debull*

Monge, A. (2000b). Matching Algorithms within a Duplicate Detection System. *Bulletin of the IEEE Computer Society Technical Committee of Data Engineering, 23(5),* 25-40. Dirección Internet: *http://www.research.microsoft.com/research/db/debull*

Monge, A. y Elkan, C. (2001). The field matching problem: Algorithms and applications. *Bulletin of the IEEE Computer Society Technical Committee od Data Engineering, 113,* 14-20. Dirección Internet: *http://www.research.microsoft.com/research/db/debull*

Mora, R. (1999). Cómo mejorar la calidad estadística de los artículos presentados a revistas biomédicas: lista de comprobación para los autores. *Medicina Clínica, 113,* 138-149.

Moritz, T.E., Ellis, N.K., Villanueva, C.B., Steeger, J.E., Ludwig, S.T., y Deegan, N.I. (1995). Development of an interactive data base management system for capturing large volumes of data. *Medical Care, 33* (10 Suppl.).

Mullooly, J.P. (1990). The effects of data entry error: an analysis of partial verification. *Computers and Biomedical Research, 23,* 259–267.

Nacional Agricultural Statistics Service (1999). *Edits methods to error.* Autor

Neaton, J.D., Duchene, A.G., Svendsen, K.H., y Wentworth, D. (1990). An examination of the efficiency of some quality assurance methods commonly employed in clinical trials. *Statistics in Medicine, 9,* 115–124.

Newcomb, S. (1881). Note of the frequency of use of the different digits in natural numbers. *American Journal of Mathematics, 4,* 39–40.

Nigrini, M. (1994). Using digital frequencies to detect fraud. *The white paper*, 3-6.

Nigrini, M. (1996). The taxpayer compliance application of Benford's Law. *The Journal of the American Taxation Association, 18,* 72-91.

Nigrini, M. (2000). *Data Analysis Using Benford's Law*. Global Audit Publication: Vancouver.

Norman, G.R., y Streiner, D.L. (1996). *Bioestadística*. Barcelona: Mosby/Doyma. (Edición original: Mosby, 1994).

Norman, M.A., y Thomas, P.J. (1991). Informing HCI design through conversation analysis. *International Journal of Man-Machine Studies, 35,* 235–250.

Orme, J.G., y Reis, J. (1991). Multiple regression with missing data. *Journal of Social Service Research, 15,* 61-91.

Practical Bioinformatics (2000). Lecture 6: Sequence Alignment: I. Pairwise Alignments. Autor: Acceso Internet: http://warta.bio.psu.edu/htt_doc/WebPage/html/Courses/BIOL497D/slides/day1.pdf

Perera, M. Y Ayllón, J. (1999) El primer dígito significativo. *Epsilon, 45* (3), 339-351.

Piatetsky-Shapiro, G., y Frawley, W.J. (Eds.). (1991). *Knowledge Discovery in Databases*. Menlo Park, CA: AAAI Press and MIT Press.

Pinkham, R. (1961). On the Distribution of First Significant Digits. *CAnnals of Mathematical Statistics, 32,* 1223–1230.

Pocius, K.E. (1991). Personality factors in human-computer interaction: a review of the literature. *Computers in Human Behavior, 7,* 103–135.

Rath, A., y Brown, D.E. (1995). Conceptions of human-computer interaction: a model for understanding student errors. *Journal of Educational Computing Research, 12,* 395–409.

Redman, T.C. (1992). *Data quality. Management and technology.* New York: Bantam Books.

Reynolds-Haertle, R.A., y McBride, R. (1992). Single vs doble data entry in CAST. *Controlled Clinical Trials, 13,* 487–494.

Raimi, R. (1969a). Mathematical support for Benford's Law using Banach and other scale invariant measures. *American Mathematical Monthly, 76,* 342–348.

Raimi, R. (1969b). The peculiar Distribution of First Digits. *Scientific American, Diciembre,* 109–120.

Reise, S.P. y Flannery, W.P. (1996). Assessing person-fit on measures of typical performance. *Applied measurement in education, 9(1),* 9-26.

Rondel, R.K., Varley, S.A., y Weeb, C.F. (Eds.) (1999). *Clinical data management.* 2nd Edition. Chichester: John Wiley & Sons.

Rowley, G. (1989). Reliability and other indicators of data quality in classroom observation research. *Journal of Classroom Interaction, 24,* 22–29.

Rubin, D.B. (1987). *Multiple imputation for nonresponse in surveys.* New York: John Wiley & Sons.

Rubin, D.B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association, 91,* 473–489.

Rubin, D.B., y Schenker, N. (1991). Multiple imputation in health-care databases: an overview and some applications. *Statistics in Medicine, 10,* 585–598.

Sánchez, M.E. (1992). Effects of questionnaire design on the quality of survey data. *Public Opinion Quarterly, 56,* 206–217.

Sato, T. (1975). *The construction and interpretation of S-P tables.* Tokyo: Meiji Tosho.

Saris, W.E. (1991). *Computer-assisted interviewing.* Newbury Park: SAGE Publications.

SAS Institute Inc. (1999). *SAS OnlineDoc®, Version 8*. Cary, NC: Autor.

Schafer, J.L. (1997). *Analysis of incomplete multivariate data*. Londres: Chapman & Hall.

Schneiderman, B. (1992). *Designing the user interface: strategies for effective human-computer interaction.* Reading, MA: Addison-Wesley.

Senay, H. (1992). Fuzzy command grammars for intelligent interface design. *IEEE Transactions on Systems, Man, and Cybernetics, 22,* 1124–1131.

Sigman, R. y Wagner, D. (1997). *Algorithms for adjusting survey data that fail balance edits*. Proceedings of the survey research methods section. American Statistical Association: Alexandria, VA.

Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular. Biology, 147,* 195-197.

Sonquist, J.A., y Dunkelberg, W.C. (1977). *Survey and opinion research, procedures for processing and analysis.* Englewood Clifs, New Jersey: Prentice-Hall.

SPSS Inc. (1999). *Syntax Reference Guide, Version 10.0*. Chicago: Autor.

StataCorp. (2001). *Statistical Software: Release 7.0*. College Station, TX: Autor.

Stellman, S.D. (1989). Brief reports. The case of the missing eights. An object lesson in data quality assurance. *American Journal of Epidemiology, 129*, 857–860.

Taylor, M.A., y Amir, N. (1994). The problem of missing clinical data for research in psychopathology. Some solution guidelines. *The Journal of Nervous and Mental Disease, 182,* 222–229.

Tenn, Joseph S. (1987). Simon Newcomb: A Famous and Forgotten American Astronomer. *Griffith Observer 51* (*11),* 2.

Tellegen, A. y Atkinson, G. (1974). Openness to absorbing and self-altering experiences ("absorption"), a trait related to hypnotic susceptibility. *Journal of Abnormal Psychology, 83*, 268-277.

Thompson, K. y Della Rocca, G. (1998). Statistical methods for developing ratio edit tolerantes for economic data. *Journal of Official Statistics, 2*, 5-15

Todaro, T. (1997). *Evaluation of the SPEER Automatic Edit and Imputation System.* Nass Research Report RD 97-04. Nationatl Agricultural Statistics Service: Washinton, DC.

Vach, W., y Blettner, M. (1991). Biased estimation of the odds ration in case-control studies due to the use of ad hoc methods of correcting for missing falues for confounding variables. *American Journal of Epidemiology, 134,* 895–907.

Villan, I. y Bravo, M. (1990). *Procedimiento de depuración de datos estadísticos.* Instituto Vasco de Estadística: Zarautz.

Varian, H. (1972). Benford's Law. *The American Statistician, 23*, 65–66.

Wallace, M.D., y Anderson, T.J. (1993). Approaches to interface design. *Interacting with Computers, 5*, 259–278.

Wang R. y Madnick, S.E. (1989). The Inter-Database Instance Identification Problem in Integrating Autonomous Systems. *IEEE Computer Society: Proceedings of the Fifth International Conference on Data Engineering*. Los Angeles, California.

Wei, G.C.G., y Tanner, M.A. (1991). Applications of multiple imputation to the analysis of censored regression data. *Biometrics, 47,* 1297–1309.

Werner, A. (1996). Importance of the quality of human-software interaction in expert systems. *Behaviour and Information Technology, 15,* 331–335.

Westmeyer, H., y Hageböck, J. (1992). Computer-Assisted Assessment: a normative perspective, *European Journal of Psychological Assessment, 8,* 1–16.

Whitehead, J.C. (1994). Item response in contingent valuation: should CV researchers impute values for missing independent variables? *Journal of Leisure Research, 26,* 296-303.

Winkler, W. y Draper, L.R. (1994). *Application of the SPEER edit system*. Research paper. US. Bureau of the Census: Washington, DC.

Wolf, R.M. (1993). Data quality and norms in international studies. Special Issue: Mandatory testing: Issues in policy-driven assessment. *Measurement and Evaluation in Counseling and Development, 26,* 35–40.

Wright, B.D. y Stone, M.H. (1979). Best test design. Rasch measurement. Chicago: Mesa Press.

Yeoh, C., y Davies, H. (1993). Clinical coding: completeness in accuracy when doctors take it on. *British Medical Journal, 306,* 972.

Anexo 1:
LISTADO DE LAS MACROS SPSS

Macro !IDT

```
PRESERVE.
SET ERRORS=NONE.
DO IF ($casenum=1).
+   PRINT /'Macro !IDT V2003.07.07 cargada. Para imprimir la documentación ejecutar: !IDT HELP.'/.
END IF.
DEFINE !IDT ( V=!CHAREND('/')
             /R=!CHAREND('/')!DEFAULT('1')
           /LVL=!CHAREND('/')
         /TABLE=!CHAREND('/')
           /VID=!CHAREND('/')
          /DROP=!CHAREND('/')
           /OUT=!CHAREND('/')
          /XOut=!CHAREND('/')
            /CN=!CHAREND('/')!DEFAULT('@casenum')
             /L=!CHAREND('/')!DEFAULT('1')
         /HELP=!CHAREND('/')!DEFAULT(1) ).
PRESERVE.
SET PRINT=NONE ERRORS=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
*Impresión de la cabecera.
DO IF ($casenum=1).
+  PRINT /'Macro !IDT V2003.07.07 (c)A.Bonillo & JM.Doménech' / 'LISTADO DE CASOS CON IDENTIFICADOR
ERRÓNEO'/.
END IF.
!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*****************************************************************************'
           /'* DEPURACIÓN DE DATOS: DUPLICADOS, RÉPLICAS E INTEGRIDAD REFERENCIAL         *'
           /'* Creación 08.06.2000 Última revisión 07.07.2003                            *'
           /'* (c) A.Bonillo & JM.Doménech                                               *'
           /'* Email: MacrosSPSS@metodo.uab.es                                          *'
           /'*                                                                          *'
           /'* Llamada de la Macro:                                                     *'
           /'*    !IDT V= Lista de la variables que definen el identificador           *'
           /'*      [/R]= Número de registros por caso             (por defecto, 1)     *'
           /'*    [/LVL]= Lista de otras variables a listar                             *'
           /'*  [/TABLE]= Nombre (y ruta) de la tabla principal                         *'
           /'*   [/DROP]= Lista de variables a no añadir           (por defecto ninguna) *'
           /'*       /CN= Variable identificadora de la secuencia de los sujetos       *'
           /'*                                                     (por defecto,@casenum) *'
           /'*       /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)*'
           /'*                                                     (por defecto,  1)    *'
           /'*                                                                          *'
           /'* Ejemplos de llamada:                                                     *'
           /'*        !IDT V=h caso.                                                    *'
           /'*        !IDT V=h caso /TABLE='CENSAL.SAV' /DROP=nombre TO cpostal.        *'
           /'*****************************************************************************'.
+    END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE.
+  !IF ( (!L<>1 !AND !L<>0 !AND !L<>2) !OR (!R<=0 !OR !R>9 !OR !INDEX(!R,'.')<>0) ) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Parámetro(s) erróneo(s)'.
+    END IF.
+    EXECUTE.
+  !ELSE.
*    Nombre de la variable auxiliar.
+    !IF (!R=1) !THEN
+      !LET !NAME=N_IDT@.
+    !ELSE
+      !LET !NAME=R_IDT@.
+    !IFEND.
*    Borrado de la variable n_idt@ si existía.
+    MATCH FILES /FILE=* /DROP=!NAME.
*    Creación de variables auxiliares V@.
+    !LET !V@=!NULL.
+    !DO !I !IN (!V).
+      !LET !V@=!CONCAT(!I,'@ ',!V@).
+    !DOEND.
```

```
*    Creación de variables numéricas usar RANK.
+    AUTORECODE VARIABLES =!V /into !V@.

*    Conteo de registros.
+    RANK !HEAD(!V@) BY !V@ /N into !NAME /PRINT=NO.
+    VARIABLE LABEL !NAME !CONCAT ('Número de registros que comparten ',!V).

*    Borrado de las variables auxiliares.
+    MATCH FILES /FILE=* /DROP=!V@.

*    Asignación de las variables de MATCH: si existe VID se asigna, sino V.
+    !IF (!VID=!NULL) !THEN
+      !LET !VMATCH=!V.
+    !ELSE.
+      !LET !VMATCH=!VID.
+    !IFEND.
+    !IF (!TABLE<>!NULL) !THEN
+      MATCH FILES /FILE=* /DROP= I_IDT@.
+      SORT CASES BY !V.
+      MATCH FILES /FILE=*  /TABLE=!TABLE /IN=I_IDT@  /BY !VMATCH
       !IF (!DROP<>!NULL) !THEN
                           /DROP =!DROP
       !IFEND
       .
+      SORT CASES BY !CN.
+      FORMATS I_IDT@(F2).
+      VARIABLE I_IDT@ (NOMINAL).
+      RECODE I_IDT@  (0=3) (1=-4).
+      VARIABLE LABEL I_IDT@ !CONCAT ('Integridad referencial de ',!V).
+    !IFEND.
+    EXECUTE.
+    DO REPEAT V=!V.
+      IF MISSING(V) !NAME=!R.
+      IF (V=' ')    !NAME=!R.
+    END REPEAT.
+    !IF (!TABLE<>!NULL) !THEN
*      Missings del identificador.
+      DO REPEAT V=!V.
+        IF MISSING(V) I_IDT@=0.
+        IF (V=' ')    I_IDT@=0.
+      END REPEAT.
+     RENAME VARIABLES (I_IDT@=@).
+    !IFEND.
*   Error de seguimiento.
+   !LET !IDLAG=!NULL.
+   !DO !J !IN (!VID).
+     !LET !IDLAG=!CONCAT(!IDLAG,' AND ', !J,'=LAG(',!J,')'  ).
+   !DOEND.
+   !LET !IDLAG=!SUBSTR(!IDLAG,5).
+    IF LAG(!OUT)=!XOut AND !EVAL(!IDLAG) @=12.
+    !IF (!L>=1) !THEN
*     Asignación de la @ a una variable temporal.
+     RENAME VARIABLES (!NAME=@@).
+     TEMPORARY.
+       SELECT IF (@@<>!R).
+       STRING !NAME (A18).
+       IF (@@<10) !NAME=CONCAT('Repetido ',STRING(@@,F1),' veces').
+       IF (@@>=10 AND @@<100) !NAME=CONCAT('Repetido ',STRING(@@,F2),' veces').
+       IF (@@>=100 AND @@<1000) !NAME=CONCAT('Repetido ',STRING(@@,F3),' veces').
+       LIST !CN !V !NAME !LVL.
+       RENAME VARIABLES (@@=!NAME).
+     TEMPORARY.
+       !IF (!TABLE<>!NULL) !THEN
+        DO IF (!L=1).
+          SELECT IF (@>=0).
+        ELSE IF (!L=2).
+          SELECT IF (@>=1).
+        END IF.
*         Variable auxiliar cadena para listar.
+         STRING I_IDT@ (A15).
+         IF (@= 0) I_IDT@=' 0: Missing Rec'.
+         IF (@= 3) I_IDT@=' 3: Falta Princ'.
+         IF (@=12) I_IDT@='12:Err Baja cas'.
*         Asignación de la variable temporal a la auxiliar @.
+         LIST !CN !V I_IDT@ !LVL.
+         RENAME VARIABLES (@=I_IDT@).
+         VALUE LABEL I_IDT@ -4 '-4: Correcto' 0 '0: Missing Rec' 3 '3: Falta Princ' 12 '12:Err
Baja cas'.
+       !IFEND.
+    !IFEND.
+    RENAME VARIABLES (@=I_IDT@).
+    VALUE LABEL I_IDT@ -4 '-4: Correcto' 0 '0: Missing Rec' 3 '3: Falta Princ' 12 '12:Err Baja
cas'.
+  !IFEND.
+  !IFEND.
!IFEND.
EXECUTE.
RESTORE.
```

```
!ENDDEFINE.
```

Macro !DR

```
DO IF ($casenum=1).
+   PRINT /'Macro !DR V2003.07.07 cargada. Para imprimir la documentación ejecutar: !DR HELP.'/.
END IF.
DEFINE !DR ( V=!CHAREND('/')
            /C=!CHAREND('/')
          /INDX=!CHAREND('/')
            /LV=!CHAREND('/')!DEFAULT('ELSE')
            /ND=!CHAREND('/')
            /MV=!CHAREND('/')
           /MVr=!CHAREND('/')!DEFAULT('SYSMIS')
       /FORMAT=!CHAREND('/')!DEFAULT(0)
       /CADENA=!CHAREND('/')!DEFAULT(0)
           /VS=!CHAREND('/')
           /XS=!DEFAULT(0)!CHAREND('/')
           /VD=!CHAREND('/')
          /MVS=!CHAREND('/')
         /MVSr=!CHAREND('/')
          /VAR=!CHAREND('/')
          /OUT=!CHAREND('/')
         /XOut=!CHAREND('/')
          /LVL=!CHAREND('/')
           /CN=!CHAREND('/')!DEFAULT('@casenum')
            /L=!CHAREND('/')!DEFAULT(1)
         /HELP=!CHAREND('/')!DEFAULT(1) ).
PRESERVE.
SET PRINT=NONE ERRORS=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
*Impresión de la cabecera.
DO IF ($casenum=1).
+  PRINT /'Macro !DR V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.

!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*****************************************************************************'
           /'* DEPURACIÓN DE DATOS: RANGO DE VARIABLES NUMERICAS Y CADENA                *'
           /'* Creación 30.11.1998 Última revisión 07.07.2003                           *'
           /'* (c) A.Bonillo & JM.Doménech                                              *'
           /'* Email: MacrosSPSS@metodo.uab.es                                          *'
           /'*                                                                          *'
           /'* Llamada de la Macro:                                                     *'
           /'* !DR     V= Lista de variables a verificar                                *'
           /'*     [/LV]= Lista de valores válidos (en formato SPSS)                    *'
           /'*     [/ND]= Número máximo de cifras decimales de los valores              *'
           /'*     [/MV]= Lista de valores user missing (no recuperables) de variables V  *'
           /'*    [/MVr]= Lista de valores user missing (recuperables) de las variables V *'
           /'* [/FORMAT]= Verificación de formato (1=Sí, 0=No)      (por defecto, 0)      *'
           /'*    [/LVL]= Lista de otras variables a listar                             *'
           /'*              La variable @V NO se inicializa a SYSMIS cuando se especifica *'
           /'*               nombres de variables o la palabra clave @NULL              *'
           /'*              Si se omite el parámetro la variable @V se inicializa a SYSMIS*'
           /'*        /C= Lista de variables identificadoras del sujeto                 *'
           /'*   [/INDX]= Variable de índice                                            *'
           /'*       /CN= Variable identificadora de la secuencia de los sujetos        *'
           /'*                                                    (por defecto, @casenum)*'
           /'*                                                                          *'
           /'*       /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)*'
           /'*                                                    (por defecto,  1)     *'
           /'*--------------------------------------------------------------------------*'
           /'* Parámetros para variables dentro de un salto:                           *'
           /'*     [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)  *'
           /'*     [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)   *'
           /'*                                                    (por defecto,  0)     *'
           /'*     [/VD]= Valor deducible si VS indica saltar     (en formato SPSS)     *'
           /'*                                                    (por defecto, vacío)  *'
           /'*    [/MVS]= Lista de valores user missing (no recuperables) de variable VS  *'
           /'*                                                    (por defecto,  9)     *'
           /'*   [/MVSr]= Lista de valores user missing (recuperables) de la variable VS  *'
           /'* ------------------------------------------------------------------------*'
           /'* Parámetros para variables implicadas en seguimientos:                   *'
           /'*    [/VAR]= Valor, o porcentaje, máximo de variación entre seguimientos   *'
           /'*                                                    (en formato SPSS)     *'
           /'*    [/OUT]= Variable indicadora de la baja del registro en futuros seguim.  *'
           /'*   [/XOUT]= Valor de OUT que indica la baja del registro en futuros seguim. *'
           /'*--------------------------------------------------------------------------*'
           /'* Ejemplos de llamada:                                                     *'
           /'*       !DR V=tab /LV=1 thru 80 /ND=0 /VS=fuma /VD=0 /FORMAT=1 /MVS=9.     *'
           /'*       !DR V=tiptab /LV="NE","RU","NR" /MVr=" " /VS=fuma /C=h caso.       *'
           /'*       !DR V=sexo /LV="M","F" /MVr=" " /C=h caso.                         *'
           /'*       !DR V=dpt dcs /LV=1,2,3 /FORMAT=1 /C=h caso.                       *'
           /'*       !DR V=nombre  /MVr=" " /C=h caso.                                  *'
           /'*****************************************************************************'.
+  END IF.
```

```
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE.

+  !IF ( (!FORMAT<>1 !AND !FORMAT<>0) !OR (!L<>1 !AND !L<>0 !AND !L<>2)
         !OR (!ND<>!NULL !AND (!ND<0 !OR !ND>9 !OR !INDEX(!ND,'.')<>0))     ) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Parámetro(s) erróneo(s)'.
+    END IF.
+    EXECUTE.
+  !ELSE.
+  !DO !I !IN (!V).
*   Creación del nombre de variable cadena "hermana".
+   !IF (!FORMAT=1) !THEN
+     !LET !@FORMAT = !CONCAT('·',!I).
+   !ELSE
+     !LET !@FORMAT = !EVAL(!I).
+   !IFEND.
*   Creación del nombre de variable cadena "hermana".
+   !IF (!CADENA=1) !THEN
+     RENAME VARIABLES (!i= !CONCAT('·',!I) ).
+     COMPUTE !i=NUMBER(!CONCAT('·',!I),F16).
+     !LET !@FORMAT = !CONCAT('·',!I).
+   !IFEND.

*    Nombre de la variable auxiliar.
+   !LET !@V = !CONCAT('@',!I).
*     Si LVL existe debemos comprobar sólo los casos que
      cumplido la condición.
+   !IF (!LVL<>!NULL) !THEN
+     DO IF (MISSING(!@V)).
+   !ELSE.
+     COMPUTE !@V=$SYSMIS.
+   !IFEND.

+   !IF (!VS=!NULL) !THEN
*     Comprobación rango o lista de valores.
+     RECODE !I (MISSING,!MVr=0) (!LV=-4) (ELSE=3) INTO !@V.
+     IF (!I=!MV) !@V=-1.
+   !ELSE
+      !IF (!MVS<>!NULL) !THEN
+        DO IF NOT ( ANY(!VS,!XS,!MVS) ).
*          Comprobación rango o lista de valores si NO se salta.
+          RECODE !I (!MVr,!VD=1) (!LV=-4) (ELSE=3) INTO !@V.
*          El NS/NC debe marcarse como correcto.
+          IF (!I=!MV) !@V=-1.
+          IF ( SYSMIS(!VS) ) !@V=-1.
+        END IF.
+      !ELSE
+        DO IF NOT ( ANY(!VS,!XS) ).
*          Comprobación rango o lista de valores si NO se salta.
+          RECODE !I (!MVr,!VD=1) (!LV=-4) (ELSE=3) INTO !@V.
*          El NS/NC debe marcarse como correcto.
+          IF (!I=!MV) !@V=-1.
+          IF ( SYSMIS(!VS) ) !@V=-1.
+        END IF.
+      !IFEND.

*     Si el salto es NS/NC V debe estar vacía.
+     IF (ANY(!VS,!MVS) AND SYSMIS(!I)) !@V=-1.
+     IF (ANY(!VS,!MVS) AND !I=!MVr) !@V=-1.
*     Comprobar que SÍ se salta, V vale VD.
+     !IF (!VD=!NULL) !THEN
+        IF (ANY(!VS,!XS) AND !I=!MVr) !@V=-2.
+        IF (ANY(!VS,!XS) AND SYSMIS(!I)) !@V=-2.
+     !ELSE
+        IF (ANY(!VS,!XS) AND !I=!VD)  !@V=-3.
+        IF (ANY(!VS,!XS) AND !I<>!VD)  !@V=1.
+     !IFEND.
*     Si el salto está o es no evaluable, V debe estar vacía.
+     IF SYSMIS(!VS) AND SYSMIS(!I) !@V=0.
+     IF SYSMIS(!VS) AND ANY(!I,!MVr) !@V=0.
+     IF SYSMIS(!VS) AND (!I<>!MVR) !@V=1.
+     IF SYSMIS(!VS) AND NOT(SYSMIS(!I)) !@V=1.
+     IF ANY(!VS,!MVSr) AND SYSMIS(!I) !@V=0.
+     IF ANY(!VS,!MVSr) AND ANY(!I,!MVr) !@V=0.
*     En el resto de situaciones V es inconsistente con el salto.
+     IF MISSING(!@V) !@V=1.
+   !IFEND.

*   Comprobar que V no tiene exceso de precisión.
+   !IF (!ND<>!NULL) !THEN
+     COMPUTE #MOD=MOD(!I*10**!ND,1).
```

```
+     IF (#MOD>0.5) #MOD=1-#MOD.
+     IF (#MOD>(10**-6)) !@V=5.
+   !IFEND.
*   Error de formato: comprobar que si V es missing la variable F debe estar vacía.
+   IF (!@FORMAT<>' ' AND !@V=0 AND SYSMIS(!I) ) !@V=2.
*   Error de seguimiento.
+   !LET !IDLAG=!NULL.
+   !DO !J !IN (!C).
+     !LET !IDLAG=!CONCAT(!IDLAG,' AND ', !J,'=LAG(',!J,')'  ).
+   !DOEND.
+   !LET !IDLAG=!SUBSTR(!IDLAG,5).
+   !IF (!VAR=0) !THEN
+     IF (!EVAL(!IDLAG) AND (!I<>LAG(!I) OR NMISS(!I,LAG(!I))=1) ) !@V=10.
+   !ELSE.
+     !IF (!VAR<>0 !AND !VAR<>HI) !THEN
+       !IF (!INDEX(!VAR,'%')=0) !THEN
+         IF (!EVAL(!IDLAG) AND ABS(!I-LAG(!I)) >!VAR ) !@V=11.
+       !ELSE.
+         IF (!EVAL(!IDLAG) AND ABS( (LAG(!I)-!I)/LAG(!I)*100 ) >!SUBSTR(!VAR,1,2)  ) !@V=11.
+       !IFEND.
+     !IFEND.
+   !IFEND.
*   Etiquetas de la variable auxiliar.
+   ADD VALUE LABEL !@V -4 'Correcto'    -3 'Deducible'
                        -2 'No Apl Corr' -1 'Miss No Rec'
                         0 'Missing Rec'  1 'Incons Salt'
                         2 'Err Formato'  3 'Fuera Rango'
                         5 'Err Num Dec'
                        10 'Err Constnt' 11 'Excso Varcn' 12 '12:Err Baja caso'.
+   !IF (!LVL<>!NULL) !THEN
+     END IF.
+   !IFEND.
+   FORMATS !@V (F2).
+   VARIABLE LEVEL !@V (NOMINAL).
+   !IF (!VAR<>!NULL) !THEN
+     STRING @@@ (A3).
+     COMPUTE @@@=STRING(LAG(!I),F3).
+     COMPUTE @@@=LAG(!I).
+   !IFEND.
*   Listado de casos con error.
+   !IF (!L>=1) !THEN
*     Asignación de la @ a una variable temporal.
+     RENAME VARIABLES (!@V=@).
+     TEMPORARY.
*     Cambio de formato en V si se depura precisión.
+     !IF (!ND<>!NULL) !THEN
+       !LET !A=!CONCAT('F8.',!LENGTH(!CONCAT(!BLANKS(!ND),!BLANKS(1)))).
+       FORMATS !I (!A).
+     !IFEND.
+     DO IF (!L=1).
+       SELECT IF (@>=0).
+     ELSE IF (!L=2).
+       SELECT IF (@>=1).
+     END IF.
*    Variable auxiliar cadena para listar.
+     !IF (!VAR<>!NULL) !THEN
+       STRING !@V (A30).
+     !ELSE
+       STRING !@V (A15).
+     !IFEND.
+     IF (@=-4) !@V='-4: Correcto   '.
+     IF (@=-3) !@V='-3: Deducible  '.
+     IF (@=-2) !@V='-2: No Apl Corr'.
+     IF (@=-1) !@V='-1: Miss No Rec'.
+     IF (@= 0) !@V=' 0: Missing Rec'.
+     IF (@= 1) !@V=' 1: Incons Salt'.
+     IF (@= 2) !@V=' 2: Err Formato'.
+     IF (@= 3) !@V=' 3: Fuera Rango'.
+     IF (@= 5) !@V=' 5: Err Num Dec'.
+     IF (@=10) !@V=CONCAT( '10: Err Cnstant (Val. ant:',@@@,')' ).
+     IF (@=11) !@V=CONCAT( '11: Excso Varcn (Seg. ant:',@@@,')' ) .
+     IF (@=12) !@V='12:Err Baja caso'.
+     IF (@>=50)!@V=CONCAT(STRING(@,F2),': Err Cond').
+     LIST !CN !C !INDX !VS !@V !@FORMAT !LVL.
*    Asignación de la variable temporal a la auxiliar @.
+     RENAME VARIABLES (@=!@V).
+   !IFEND.
+   !IF (!VAR<>!NULL) !THEN
+     MATCH FILE /FILE=* /DROP=@@@.
+     EXECUTE.
+   !IFEND.
+   !IF (!CADENA=1) !THEN
+     MATCH FILE /FILE=* /DROP=!I.
+     RENAME VARIABLES (!CONCAT('·',!I)=!i).
+     EXECUTE.
+   !IFEND.
+  !DOEND.
+  !IFEND.
+  !IFEND.
```

```
!IFEND.
RESTORE.
!ENDDEFINE.
```

Macro !DRKey

```
DO IF ($casenum=1).
+  PRINT /'Macro !DRKey V2003.07.07 cargada. Para imprimir la documentación ejecutar: !DRKey
HELP.'/.
END IF.
DEFINE !DRKey ( V=!CHAREND('/')
             /IDT=!CHAREND('/')
           /TABLE=!CHAREND('/')
              /MV=!CHAREND('/')
             /MVr=!CHAREND('/')!DEFAULT('SYSMIS')
          /FORMAT=!CHAREND('/')!DEFAULT(0)
              /VS=!CHAREND('/')
              /XS=!CHAREND('/')!DEFAULT(0)
              /VD=!CHAREND('/')
             /MVS=!CHAREND('/')!DEFAULT(9)
            /MVSr=!CHAREND('/')
             /VAR=!CHAREND('/')
             /OUT=!CHAREND('/')
            /XOut=!CHAREND('/')
             /LVL=!CHAREND('/')
              /CN=!CHAREND('/')!DEFAULT('@casenum')
               /C=!CHAREND('/')
            /INDX=!CHAREND('/')
          /RENAME=!CHAREND('/')
            /DROP=!CHAREND('/')
               /L=!CHAREND('/')!DEFAULT(1)
            /HELP=!CHAREND('/')!DEFAULT(1) ).
PRESERVE.
SET PRINT=NONE ERRORS=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
DO IF ($casenum=1).
+  PRINT /'Macro !DRKey V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.
!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*****************************************************************************'
           /'* DEPURACIÓN DE DATOS: LISTA DE VALORES CONTENIDOS EN UNA TABLA DICCIONARIO  *'
           /'* Creación 08.06.2000 Última revisión 07.07.2003                            *'
           /'* (c) A.Bonillo & JM. Doménech                                              *'
           /'* Email: MacrosSPSS@metodo.uab.es                                          *'
           /'*                                                                          *'
           /'* Llamada de la Macro:                                                     *'
           /'*!DRKey  V= Lista de las variables a verificar                           *'
           /'*     /IDT= Lista de los identificadores en el DICCIONARIO (por defecto, V)  *'
           /'*   /TABLE= Nombre del archivo .SAV con el DICCIONARIO de códigos        *'
           /'*    [/MV]= Lista de valores user missing (no recuperables)de las variables V*'
           /'*   [/MVr]= Lista de valores user missing (recuperables) de las variables V  *'
           /'*[/FORMAT]= Verificación de formato (1=Sí, 0=No)      (por defecto,  0)      *'
           /'*   [/LVL]= Lista de otras variables a listar.                             *'
           /'*[/RENAME]= Lista de variables a renombrar            (por defecto ninguna)  *'
           /'*  [/DROP]= Lista de variables a no añadir            (por defecto ninguna)  *'
           /'*       /C= Lista de variables identificadoras del sujeto                  *'
           /'*      /CN= Variable identificadora de la secuencia de los sujetos         *'
           /'*                                                    (por defecto,@casenum) *'
           /'*      /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno) *'
           /'*                                                    (por defecto,  1)      *'
           /'*--------------------------------------------------------------------------*'
           /'* Parámetros para variables dentro de un salto                             *'
           /'*    [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)     *'
           /'*   [/XnS]= Lista de valores de VS que indican NO saltar (en formato SPSS)   *'
           /'*                                                    (por defecto,  1)      *'
           /'*    [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)      *'
           /'*                                                    (por defecto,  0)      *'
           /'*    [/VD]= Valor deducible si VS indica saltar      (en formato SPSS)      *'
           /'*   [/MVS]= Lista de valores user missing (no recuperables) de la variable VS*'
           /'*  [/MVSr]= Lista de valores user missing (recuperables) de la variable VS   *'
           /'*--------------------------------------------------------------------------*'
           /'*                                                                          *'
           /'* Ejemplos de llamada:                                                     *'
           /'*         !DRKey V=CIE /TABLE="COD.SAV" /DROP=DES /MVr=" " /C=h caso.      *'
           /'*****************************************************************************'.
+  END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL !OR !TABLE=!NULL) !THEN
+     DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE.
+  !IF ( (!FORMAT<>1 !AND !FORMAT<>0 !AND !FORMAT<>!NULL) !OR (!L<>1 !AND !L<>0 !AND !L<>2) ) !THEN
+    DO IF ($casenum=1).
```

```
+      PRINT /'ERROR: Parámetro(s) erróneo(s)'.
+    END IF.
+    EXECUTE.
+  !ELSE.

+    !LET !SBLANK=!NULL.
+    !LET !BLANK=' '.
+    !DO !I !IN (!V).
+      !LET !SBLANK=!CONCAT(!SBLANK,!BLANK).
*      Conteo de variables para construir los nombres.
+      !LET !COUNT=!LENGTH(!SBLANK).
+      !LET !RENAME2=!NULL.
*      Nombres a renombrar.
+      !DO !J !IN (!RENAME).
+        !LET !RENAME2=!CONCAT(!RENAME2,' ',!CONCAT(!J,!COUNT)).
+      !DOEND.
*      Nombre de variable cadena "hermana".
+      !IF (!FORMAT=1) !THEN
+        !LET !@FORMAT = !CONCAT('·',!I).
+      !ELSE
+        !LET !@FORMAT = !EVAL(!I).
+      !IFEND.
*      Asignación de los identificadores.
+      !IF (!IDT=!NULL) !THEN
+        !LET !TIDT=!EVAL(!I).
+      !ELSE
+        !LET !TIDT=!EVAL(!IDT).
+      !IFEND.

+      SORT CASES BY !I.
*      Nombre de la variable auxiliar.
+      !LET !@V = !CONCAT('@',!I).

*      Si LVL existe debemos asignamos los errores lógicos a una variable temporal.
+      !IF (!LVL<>!NULL) !THEN
+         COMPUTE @=!@V.
+      !IFEND.
*      Si la variable @ existe es borrada.
+      MATCH FILES /FILE=*  /DROP=!@V.
*      Añadido del diccionario.
+      MATCH FILES /FILE=*  /TABLE=!TABLE
       !IF (!IDT<>!NULL) !THEN
                            /RENAME= (!TIDT=!I)
       !IFEND
       !IF (!RENAME<>!NULL) !THEN
                            /RENAME= (!RENAME=!RENAME2)
       !IFEND
                            /IN=!@V  /BY !I
       !IF (!DROP<>!NULL) !THEN
                            /DROP =!DROP
       !IFEND
       .
+      EXECUTE.
+     !IF (!VS=!NULL) !THEN
*        La variable auxiliar es cambiada a nuestra propuesta.
+        RECODE !@V (0=3) (1=-4).
+        RECODE !I (!MVr=0) into !@V.
+        IF (!I=!MV) !@V=-1.
+     !ELSE

+        !IF (!MVS<>!NULL) !THEN
+           DO IF NOT ( ANY(!VS,!XS,!MVS) ).
*             Comprobación rango o lista de valores si NO se salta.
+             RECODE !@V (0=3) (1=-4).
+             RECODE !I (!MVr,!VD=1) INTO !@V.
*             El NS/NC debe marcarse como correcto.
+             IF (!I=!MV) !@V=-1.
+             IF ( SYSMIS(!VS) ) !@V=-1.
+           END IF.
+        !ELSE
+           DO IF NOT ( ANY(!VS,!XS) ).
*             Comprobación rango o lista de valores si NO se salta.
+             RECODE !@V (0=3) (1=-4).
+             RECODE !I (!MVr,!VD=1) INTO !@V.
*             El NS/NC debe marcarse como correcto.
+             IF (!I=!MV) !@V=-1.
+             IF ( SYSMIS(!VS) ) !@V=-1.
+           END IF.
+        !IFEND.

*        Si el salto es NS/NC V debe estar vacía.
+        IF (ANY(!VS,!MVS) AND SYSMIS(!I)) !@V=-1.
+        IF (ANY(!VS,!MVS) AND !I=!MVr) !@V=-1.

*        Condición -2: Comprobar que si VS=XS la variable vale VD.
+        !IF (!VD=!NULL) !THEN
+           IF (ANY(!VS,!XS) AND !I=!MVr) !@V=-2.
+           IF (ANY(!VS,!XS) AND SYSMIS(!I)) !@V=-2.
+        !ELSE
```

```
+           IF (ANY(!VS,!XS) AND !I=!VD)  !@V=-3.
+           IF (ANY(!VS,!XS) AND !I<>!VD)  !@V=1.
+        !IFEND.

*        Si el salto está  o es no evaluable, V debe estar vacía.
+        IF SYSMIS(!VS) AND SYSMIS(!I) !@V=0.
+        IF SYSMIS(!VS) AND ANY(!I,!MVr) !@V=0.
+        IF SYSMIS(!VS) AND (!I<>!MVR) !@V=1.
+        IF SYSMIS(!VS) AND NOT(SYSMIS(!I)) !@V=1.
+        IF ANY(!VS,!MVSr) AND SYSMIS(!I) !@V=0.
+        IF ANY(!VS,!MVSr) AND ANY(!I,!MVr) !@V=0.
*        Si el sato es erróneo V es incongruente.
+        IF NOT (ANY(!VS,!XS,!XnS,!MVS)) !@V=1.
+        IF (ANY(!VS,!XS) AND !@V=0) !@V=1.
+     !IFEND.

*      Si LVL existe debemos asignamos los guardados en una variable temporal a la var @.
+      !IF (!LVL<>!NULL) !THEN
+        COMPUTE !@V=MAX(@,!@V).
+        MATCH FILES /FILE=* /DROP=@.
+      !IFEND.

+      !IF (!VAR<>!NULL) !THEN
+        STRING @@@ (A3).
+        COMPUTE @@@=STRING(LAG(!I),F3).
+        COMPUTE @@@=LAG(!I).
+      !IFEND.

*     Error de seguimiento.
+     !LET !IDLAG=!NULL.
+     !DO !J !IN (!C).
+       !LET !IDLAG=!CONCAT(!IDLAG,' AND ', !J,'=LAG(',!J,')'  ).
+     !DOEND.
+     !LET !IDLAG=!SUBSTR(!IDLAG,5).

+     !IF (!VAR=0) !THEN
+       SORT CASES BY !CN.
+       IF (!EVAL(!IDLAG) AND (!I<>LAG(!I) OR NMISS(!I,LAG(!I))=1) ) !@V=10.
+     !IFEND.

*      Etiquetas de la variable auxiliar.
+      FORMATS !@V (F2).
+      VARIABLE LEVEL !@V (NOMINAL).
+      ADD VALUE LABEL !@V -4 'Correcto'    -3 'Deducible'
                           -2 'No Apl Corr' -1 'Miss No Rec'
                            0 'Missing Rec'  1 'Incons Salt'
                            2 'Err Formato'  3 'Fuera Rango'
                            5 'Err Num Dec'
                           10 'Err Constnt' 11 'Excso Varcn' 12 '12:Err Baja caso'.

*      Listado de casos con error.
+      !IF (!L>=1) !THEN
*        Asignación de la @ a una variable temporal.
+        RENAME VARIABLES (!@V=@).
+        SORT CASES BY !CN.
+        TEMPORARY.
+        DO IF (!L=1).
+          SELECT IF (@>=0).
+        ELSE IF (!L=2).
+          SELECT IF (@>=1).
+        END IF.
*        Variable auxiliar cadena para listar.
+       !IF (!VAR<>!NULL) !THEN
+         STRING !@V (A30).
+       !ELSE
+         STRING !@V (A15).
+       !IFEND.

+        IF (@=-4) !@V='-4: Correcto   '.
+        IF (@=-3) !@V='-3: Deducible  '.
+        IF (@=-2) !@V='-2: No Apl Corr'.
+        IF (@=-1) !@V='-1: Miss No Rec'.
+        IF (@= 0) !@V=' 0: Missing Rec'.
+        IF (@= 1) !@V=' 1: Incons Salt'.
+        IF (@= 2) !@V=' 2: Err Formato'.
+        IF (@= 3) !@V=' 3: Fuera Rango'.
+        IF (@=10) !@V=CONCAT( '10: Err Cnstant (Val. ant:',@@@,')' ).
+        IF (@=11) !@V=CONCAT( '11: Excso Varcn (Seg. ant:',@@@,')' ) .
+        IF (@=12) !@V='12:Err Baja caso'.
+        IF (@>=50)!@V=CONCAT(STRING(@,F2),': Err Cond').
+        LIST !CN !C !INDX !VS !@V !@FORMAT.
*        Asignación de la variable temporal a la auxiliar @.
+        RENAME VARIABLES (@=!@V).
+      !IFEND.
+      !IF (!VAR<>!NULL) !THEN
+        MATCH FILE /FILE=* /DROP=@@@.
+        EXECUTE.
+      !IFEND.
+    !DOEND.
```

```
+  !IFEND.
+  !IFEND.
!IFEND.
RESTORE.
!ENDDEFINE.
```

Macro !DDF

```
DO IF ($casenum=1).
+   PRINT /'Macro !DDF V2003.07.07 cargada. Para imprimir la documentación ejecutar: !DDF HELP.'/.
END IF.
DEFINE !DDF ( V=!CHAREND('/')
          /INDX=!CHAREND('/')
            /D=!CHAREND('/')
           /MIN=!CHAREND('/')
           /MAX=!CHAREND('/')
        /FORMAT=!CHAREND('/')!DEFAULT(0)
            /VS=!CHAREND('/')
           /XS=!CHAREND('/')!DEFAULT(0)
          /MVS=!CHAREND('/')!DEFAULT(9)
         /MVSr=!CHAREND('/')
          /NUM=!CHAREND('/')!DEFAULT(0)
        /CADENA=!CHAREND('/')!DEFAULT(0)
          /VAR=!CHAREND('/')
          /OUT=!CHAREND('/')
         /XOut=!CHAREND('/')
          /LVL=!CHAREND('/')
            /C=!CHAREND('/')
           /CN=!CHAREND('/')!DEFAULT('@casenum')
            /L=!CHAREND('/')!DEFAULT(1)
         /HELP=!CHAREND('/')!DEFAULT(1) ) .
PRESERVE.
SET PRINT=NONE ERRORS=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
DO IF ($casenum=1).
+   PRINT /'Macro !DDF V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.
EXECUTE.

!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*****************************************************************************'
          /'* DEPURACIÓN DE DATOS: RANGO DE CAMPOS FECHA EXPRESADO EN DIAS               *'
          /'* Creación 30.11.1998 Última revisión 07.07.2003                            *'
          /'* (c) A.Bonillo & JM. Doménech                                              *'
          /'* Email: MacrosSPSS@metodo.uab.es                                           *'
          /'*                                                                           *'
          /'* Llamada de la Macro:                                                      *'
          /'* !DDF V= Lista de campos fecha                                             *'
          /'*     /D= Diferencia entre los campos fecha a depurar y fecha de referencia  *'
          /'* [/MIN]= Valor mínimo del tiempo transcurrido (días)                       *'
          /'* [/MAX]= Valor máximo del tiempo transcurrido (días)                       *'
          /'*[/FORMAT]= Verificación de formato (1=Sí, 0=No)   (por defecto,  0)        *'
          /'* [/LVL]= Lista de otras variables a listar.                              *'
          /'*     /C= Lista de variables identificadoras del sujeto                   *'
          /'*    /CN= Variable identificadora de la secuencia de los sujetos          *'
          /'*                                                (por defecto,@casenum)   *'
          /'*    /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)   *'
          /'*                                                (por defecto,  1)        *'
          /'*---------------------------------------------------------------------------*'
          /'* Parámetros para variables dentro de un salto                             *'
          /'*  [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)      *'
          /'* [/XnS]= Lista de valores de VS que indican NO saltar (en formato SPSS)    *'
          /'*                                                (por defecto,  1)        *'
          /'*  [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)       *'
          /'*                                                (por defecto,  0)        *'
          /'* [/MVS]= Lista de valores user missing (no recuperables) de la variable VS  *'
          /'*[/MVSr]= Lista de valores user missing (recuperables) de la variable VS    *'
          /'*---------------------------------------------------------------------------*'
          /'*                                                                           *'
          /'* Ejemplos de llamada:                                                      *'
          /'*!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /C=h caso.                *'
          /'*!DDF V=fn /D=fr-fn /MIN=365.25*6 /MAX=365.25*45 /FORMAT=1 /VS=fuma/C=h caso.*'
          /'*****************************************************************************'.
+  END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL !OR !D=!NULL) !THEN
+     DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE.

+  !IF ( (!FORMAT<>1 !AND !FORMAT<>0 !AND !FORMAT<>!NULL) !OR  (!L<>1 !AND !L<>0 !AND !L<>2) ) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Parámetro(s) erróneo(s)'.
+    END IF.
+    EXECUTE.
+  !ELSE.

+  !DO !I !IN (!V).
*    Nombre de la variable cadena "hermana".
```

```
+    !IF (!FORMAT=1 !OR !NUM=1) !THEN
+      !LET !@FORMAT = !CONCAT('·',!I).
+    !ELSE
+      !LET !@FORMAT = !EVAL(!I).
+    !IFEND.

*     Detección de la diferencia.
+     !LET !V1 = !HEAD(!d).
+     !LET !V2 = !TAIL(!TAIL(!d)).
+     !IF (!head(!v)=!v1) !THEN
+       !LET !DD= !concat(!i,'-',!V2).
+     !ELSE.
+       !LET !DD= !concat(!V1,'-',!i).
        !LET !V2= !V1.
+     !IFEND.

+   !IF (!NUM=1) !THEN
+     !LET !@FORMAT = !CONCAT('·',!I).
+     RENAME VARIABLES (!i !V2= !CONCAT('·',!I) !CONCAT('·',!V2) ).
+     COMPUTE
!i=DATE.DMY(TRUNC(!CONCAT('·',!I)/1000000),MOD(TRUNC(!CONCAT('·',!I)/10000),100),MOD(!CONCAT('·',!I),10000)).
+     COMPUTE
!V2=DATE.DMY(TRUNC(!CONCAT('·',!V2)/1000000),MOD(TRUNC(!CONCAT('·',!V2)/10000),100),MOD(!CONCAT('·',!V2),10000)).
+     FORMATS !i !V2 (EDATE).
+   !ELSE
+   !IF (!CADENA=1) !THEN
+     !LET !@FORMAT = !CONCAT('·',!I).
+     RENAME VARIABLES (!i !V2= !CONCAT('·',!I) !CONCAT('·',!V2) ).
+     COMPUTE !i=DATE.DMY(NUMBER(SUBSTR(!CONCAT('·',!I),1,2), F2),NUMBER(SUBSTR(!CONCAT('·',!I) ,4,2),
F2),NUMBER(SUBSTR(!CONCAT('·',!I) ,7), F4)).
+     COMPUTE !V2=DATE.DMY(NUMBER(SUBSTR(!CONCAT('·',!V2) ,1,2), F2),NUMBER(SUBSTR(!CONCAT('·',!V2) ,4,2),
F2),NUMBER(SUBSTR(!CONCAT('·',!V2) ,7), F4)).
+     FORMATS !i !V2 (EDATE).
+   !IFEND.
+   !IFEND.

*     Nombre de la variable auxiliar.
+     !LET !@V = !CONCAT('@',!I).

*     Si LVL existe debemos comprobar sólo los casos que hayan cumplido la condición.
+     !IF (!LVL<>!NULL) !THEN
+       DO IF (MISSING(!@V)).
+     !ELSE.
*       Si no existe LVL la variable @ es inicializada.
+       COMPUTE !@V=$SYSMIS.
+     !IFEND.

*     Comprobación de rango y missing.
+     IF RANGE(CTIME.DAYS(!EVAL(!DD)),!MIN, !MAX) OR MISSING(!V2) !@V=-4.
+     IF NOT RANGE(CTIME.DAYS(!EVAL(!DD)),!MIN, !MAX) !@V=4.
+     IF MISSING(!I)  !@V=0.

*     Condición -2: Comprobar que si VS=XS la variable vale VD.
+     IF (ANY(!VS,!XS) AND !@V=0)  !@V=-2. /*No aplicable.
+     IF (ANY(!VS,!XS) AND !I=!VD) !@V=-3. /*Deducible.

*     Si el salto es NS/NC V debe estar vacía.
+     IF (ANY(!VS,!MVS) AND SYSMIS(!I)) !@V=-1.
+     IF (ANY(!VS,!MVS) AND NOT(SYSMIS(!I)) ) !@V=1.

*     Condición 4: Comprobar que si VS=XS la variable V debe estar vacía.
+     IF (ANY(!VS,!XS) AND !@V<>-2) !@V=1.

*     Condición 1: Comprobar que si V es missing la variable F debe estar vacía.
+     IF (!@FORMAT<>' ' AND !@V=0 AND MISSING(!I) ) !@V=2.
+     IF (NOT(MISSING(!@FORMAT)) AND !@V=0 AND MISSING(!I) ) !@V=2.

*     Error de seguimiento.
+    !LET !IDLAG=!NULL.
+    !DO !J !IN (!C).
+      !LET !IDLAG=!CONCAT(!IDLAG,' AND ', !J,'=LAG(',!J,')'  ).
+    !DOEND.
+    !LET !IDLAG=!SUBSTR(!IDLAG,5).

+    !IF (!VAR=0) !THEN
+      IF (!EVAL(!IDLAG) AND (!I<>LAG(!I) OR NMISS(!I,LAG(!I))=1) ) !@V=10.
+    !ELSE.
+      !IF (!VAR<>0 !AND !VAR<>HI) !THEN
+        IF (!EVAL(!IDLAG) AND ABS(CTIME.DAYS(!I-LAG(!I))) >!VAR ) !@V=11.
+      !IFEND.
+    !IFEND.

*    Etiquetas de la variable auxiliar.
+    ADD VALUE LABEL !@V -4 'Correcto'    -3 'Deducible'
                        -2 'No Apl Corr' -1 'Miss No Rec'
                         0 'Missing Rec'  1 'Incons Salt'
                         2 'Err Formato'  4 'Dif fuera Rango'
                        10 'Err Constnt' 11 'Excso Varcn' 12 '12:Err Baja caso'.
```

```
+     !IF (!LVL<>!NULL) !THEN
+       END IF.
+     !IFEND.
+     FORMATS !@V (F2).
+     VARIABLE LEVEL !@V (NOMINAL).

+     !IF (!VAR<>!NULL) !THEN
+       STRING @@@ (A10).
+       COMPUTE @@@=STRING(LAG(!I),EDATE10).
+       COMPUTE @@@=LAG(!I).
+     !IFEND.

*     Listado de casos con error.
+     !IF (!L>=1) !THEN
*       Asignación de la @ a una variable temporal.
+       RENAME VARIABLES (!@V=@).
+       TEMPORARY.
+       DO IF (!L=1).
+         SELECT IF (@>=0).
+       ELSE IF (!L=2).
+         SELECT IF (@>=1).
+       END IF.

*       Variable auxiliar cadena para listar.
+       !IF (!VAR<>!NULL) !THEN
+         STRING !@V (A40).
+       !ELSE
+         STRING !@V (A15).
+       !IFEND.
+       IF (@=-4) !@V='-4: Correcto   '.
+       IF (@=-3) !@V='-3: Deducible  '.
+       IF (@=-2) !@V='-2: No Apl Corr'.
+       IF (@=-1) !@V='-1: Miss No Rec'.
+       IF (@= 0) !@V=' 0: Missing Rec'.
+       IF (@= 1) !@V=' 1: Incons Salt'.
+       IF (@= 2) !@V=' 2: Err Formato'.
+       IF (@= 4) !@V=' 4: Dif fuera Rango'.
+       IF (@=10) !@V=CONCAT( '10: Err Cnstant (Val. ant:',@@@,')' ).
+       IF (@=11) !@V=CONCAT( '11: Excso Varcn (Seg. ant:',@@@,')' ) .
+       IF (@=12) !@V='12:Err Baja caso'.
+       IF (@>=50)!@V=CONCAT(STRING(@,EDATE10),': Err Cond').
+       LIST !CN !C !INDX !VS !@V !@FORMAT !V2 !LVL.
*       Asignación de la variable temporal a la auxiliar @.
+       RENAME VARIABLES (@=!@V).
+     !IFEND.

+   !IF (!VAR<>!NULL) !THEN
+     MATCH FILE /FILE=* /DROP=@@@.
+     EXECUTE.
+   !IFEND.

+   !IF (!NUM=1 !OR !CADENA=1) !THEN
+     MATCH FILE /FILE=* /DROP=!I !V2 .
+     RENAME VARIABLES (!CONCAT('·',!I) !CONCAT('·',!V2)=!i !V2).
+     EXECUTE.
+   !IFEND.

+  !DOEND.
+  !IFEND.
+  !IFEND.
!IFEND.
RESTORE.
!ENDDEFINE.
RESTORE.
```

Macro !DRF

```
PRESERVE.
SET ERRORS=NONE.
DO IF ($casenum=1).
+   PRINT /'Macro !DRF V2003.07.07 cargada. Para imprimir la documentación ejecutar: !DRF HELP.'/.
END IF.
EXECUTE.
DEFINE !DRF ( V=!CHAREND('/')
            /FI=!CHAREND('/')
            /FS=!CHAREND('/')
        /FORMAT=!CHAREND('/')!DEFAULT(0)
        /NUM=!CHAREND('/')!DEFAULT(0)
        /CADENA=!CHAREND('/')!DEFAULT(0)
            /VS=!CHAREND('/')
            /XS=!CHAREND('/')!DEFAULT(0)
           /MVS=!CHAREND('/')!DEFAULT(9)
          /MVSr=!CHAREND('/')
           /VAR=!CHAREND('/')
           /OUT=!CHAREND('/')
          /XOut=!CHAREND('/')
           /LVL=!CHAREND('/')
             /C=!CHAREND('/')
          /INDX=!CHAREND('/')
            /CN=!CHAREND('/')!DEFAULT('@casenum')
             /L=!CHAREND('/')!DEFAULT(1)
         /HELP=!CHAREND('/')!DEFAULT(1) ).
PRESERVE.
SET PRINT=NONE ERRORS=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
*Impresión de la cabecera.
DO IF ($casenum=1).
+   PRINT /'Macro !DRF V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.

!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*****************************************************************************'
          /'* DEPURACIÓN DE DATOS: RANGO DE CAMPOS FECHA EXPRESADO EN FECHAS            *'
          /'* Creación 30.11.1998 Última revisión 07.07.2003                           *'
          /'* (c) A.Bonillo & JM. Doménech                                             *'
          /'* Email: MacrosSPSS@metodo.uab.es                                          *'
          /'*                                                                          *'
          /'* Llamada de la Macro:                                                     *'
          /'* !DRF V= Lista de campos fecha a verificar                                *'
          /'*  [/FI]= Fecha inicial: día, mes, año                                     *'
          /'*  [/FS]= Fecha final  : día, mes, año                                     *'
          /'*[/FORMAT]= Verificación de formato (1=Si, 0=No)   (por defecto,  0)       *'
          /'* [/LVL]= Lista de otras variables a listar.                              *'
          /'*     /C= Lista de variables identificadoras del sujeto                    *'
          /'*    /CN= Variable identificadora de la secuencia de los sujetos           *'
          /'*                                                (por defecto,@casenum)    *'
          /'*    /L = Listado de incidencias (2=Errores,1=Errores y missing,0=Ninguno)  *'
          /'*                                                (por defecto,  1)         *'
          /'*--------------------------------------------------------------------------*'
          /'* Parámetros para variables dentro de un salto                            *'
          /'*  [/VS]= Nombre variable de salto (sólo si V está dentro de un salto)     *'
          /'* [/XnS]= Lista de valores de VS que indican NO saltar (en formato SPSS)   *'
          /'*                                                  (por defecto,  1)       *'
          /'*  [/XS]= Lista de valores de VS que indican saltar (en formato SPSS)      *'
          /'*                                                  (por defecto,  0)       *'
          /'* [/MVS]= Lista de valores user missing (no recuperables) de la variable VS *'
          /'*[/MVSr]= Lista de valores user missing (recuperables) de la variable VS   *'
          /'*--------------------------------------------------------------------------*'
          /'*                                                                          *'
          /'* Ejemplos de llamada:                                                     *'
          /'*        !DRF V=fr /FI=1,4,1993/ FS=30,12,1993 /FORMAT=1 /C=h caso.        *'
          /'*****************************************************************************'.
+  END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE.

+  !IF ( (!FORMAT<>1 !AND !FORMAT<>0 !AND !FORMAT<>!NULL) !OR  (!L<>1 !AND !L<>0 !AND !L<>2) ) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Parámetro(s) erróneo(s)'.
+    END IF.
+    EXECUTE.

+  !ELSE.

+ !DO !I !IN (!V).
```

```
*   Creación del nombre de variable cadena hermana.
+   !IF (!FORMAT=1 !OR !NUM=1) !THEN
+     !LET !@FORMAT = !CONCAT('·',!I).
+   !ELSE
+     !LET !@FORMAT = !EVAL(!I).
+   !IFEND.

+   !IF (!NUM=1) !THEN
+     !LET !@FORMAT = !CONCAT('·',!I).
+     RENAME VARIABLES (!i= !CONCAT('·',!I) ).
+     COMPUTE
!i=DATE.DMY(TRUNC(!CONCAT('·',!I)/1000000),MOD(TRUNC(!CONCAT('·',!I)/10000),100),MOD(!CONCAT('·',!I),10000)).
+     FORMATS !i (EDATE).
+   !ELSE
+   !IF (!CADENA=1) !THEN
+     !LET !@FORMAT = !CONCAT('·',!I).
+     RENAME VARIABLES (!i= !CONCAT('·',!I) ).
+     COMPUTE !i=DATE.DMY(NUMBER(SUBSTR(!CONCAT('·',!I),1,2), F2),NUMBER(SUBSTR(!CONCAT('·',!I) ,4,2),
F2),NUMBER(SUBSTR(!CONCAT('·',!I) ,7), F4)).
+     FORMATS !i (EDATE).
+   !IFEND.
+   !IFEND.

*    Nombre de la variable auxiliar.
+   !LET !@V = !CONCAT('@',!I).

*     Si LVL existe debemos comprobar sólo los casos que
      cumplido la condición.
+   !IF (!LVL<>!NULL) !THEN
+     DO IF (MISSING(!@V)).
+   !ELSE.
*     Si no existe LVL la variable @ es inicializada.
+     COMPUTE !@V=$SYSMIS.
+   !IFEND.

*    Comprobación del intervalo rango y missing.
+    IF (RANGE(!I, DATE.DMY(!FI), DATE.DMY(!FS))) !@V=-4.
+    IF (NOT(RANGE(!I, DATE.DMY(!FI), DATE.DMY(!FS)))) !@V=3.
+    IF (MISSING(!I)) !@V=0.

*    Condición -2: Comprobar que si VS=XS la variable vale VD.
+    IF (ANY(!VS,!XS) AND !@V=0) !@V=-2.  /*No aplicable.
+    IF (ANY(!VS,!XS) AND !I=!VD) !@V=-3. /*Deducible.

*    Si el salto es NS/NC V debe estar vacía.
+    IF (ANY(!VS,!MVS) AND SYSMIS(!I)) !@V=-1.
+    IF (ANY(!VS,!MVS) AND NOT(SYSMIS(!I)) ) !@V=1.

*    Si el sato es erróneo V es incongruente.
+    IF NOT (ANY(!VS,!XS,!XnS,!MVS)) !@V=1.

*    Condición 4: Comprobar que si VS=XS la variable V debe estar vacía.
+    IF (ANY(!VS,!XS) AND !@V<>-2) !@V=1.

*    Error de formato.
+    IF (!@FORMAT<>' ' AND !@V=0 AND MISSING(!I) ) !@V=2.
+    IF (NOT(MISSING(!@FORMAT)) AND !@V=0 AND MISSING(!I) ) !@V=2.

*   Error de seguimiento.
+   !LET !IDLAG=!NULL.
+   !DO !J !IN (!C).
+     !LET !IDLAG=!CONCAT(!IDLAG,' AND ', !J,'=LAG(',!J,')'  ).
+   !DOEND.
+   !LET !IDLAG=!SUBSTR(!IDLAG,5).

+   !IF (!VAR=0) !THEN
+     IF (!EVAL(!IDLAG) AND (!I<>LAG(!I) OR NMISS(!I,LAG(!I))=1) ) !@V=10.
+   !ELSE.
+     !IF (!VAR<>0 !AND !VAR<>HI) !THEN
+       IF (!EVAL(!IDLAG) AND ABS(CTIME.DAYS(!I-LAG(!I))) >!VAR ) !@V=11.
+     !IFEND.
+   !IFEND.

*    Etiquetas de la variable auxiliar.
+    ADD VALUE LABEL !@V -4 'Correcto'    -3 'Deducible'
                         -2 'No Apl Corr' -1 'Miss No Rec'
                          0 'Missing Rec'  1 'Incons Salt'
                          2 'Err Formato'  3 'Fuera Rango'
                         10 'Err Constnt' 11 'Excso Varcn' 12 '12:Err Baja caso'.

+    !IF (!LVL<>!NULL) !THEN
+      END IF.
+    !IFEND.
+    FORMATS !@V (F2).
+    VARIABLE LEVEL !@V (NOMINAL).

+   !IF (!VAR<>!NULL) !THEN
+     STRING @@@ (A10).
+     COMPUTE @@@=STRING(LAG(!I),EDATE10).
```

```
+     COMPUTE @@@=LAG(!I).
+   !IFEND.

*    Listado de casos con error.
+    !IF (!L>=1) !THEN
*      Asignación de la @ a una variable temporal.
+      RENAME VARIABLES (!@V=@).
+      TEMPORARY.
+      DO IF (!L=1).
+        SELECT IF (@>=0).
+      ELSE IF (!L=2).
+        SELECT IF (@>=1).
+      END IF.
*      Variable auxiliar cadena para listar.
+     !IF (!VAR<>!NULL) !THEN
+       STRING !@V (A40).
+     !ELSE
+       STRING !@V (A15).
+     !IFEND.
+      IF (@=-4) !@V='-4: Correcto   '.
+      IF (@=-3) !@V='-3: Deducible  '.
+      IF (@=-2) !@V='-2: No Apl Corr'.
+      IF (@=-1) !@V='-1: Miss No Rec'.
+      IF (@= 0) !@V=' 0: Missing Rec'.
+      IF (@= 1) !@V=' 1: Incons Salt'.
+      IF (@= 2) !@V=' 2: Err Formato'.
+      IF (@= 3) !@V=' 3: Fuera Rango'.
+      IF (@=10) !@V=CONCAT( '10: Err Cnstant (Val. ant:',@@@,')' ).
+      IF (@=11) !@V=CONCAT( '11: Excso Varcn (Seg. ant:',@@@,')' ) .
+      IF (@=12) !@V='12:Err Baja caso'.
+      IF (@>=50)!@V=CONCAT(STRING(@,EDATE10),': Err Cond').
+      LIST !CN !C !INDX !VS !@V !@FORMAT !LVL.
*      Asignación de la variable temporal a la auxiliar @.
+      RENAME VARIABLES (@=!@V).
+    !IFEND.

+   !IF (!VAR<>!NULL) !THEN
+     MATCH FILE /FILE=* /DROP=@@@.
+     EXECUTE.
+   !IFEND.

+   !IF (!NUM=1 !OR !CADENA=1) !THEN
+     MATCH FILE /FILE=* /DROP=!I.
+     RENAME VARIABLES (!CONCAT('·',!I)=!i).
+     EXECUTE.
+   !IFEND.

+  !DOEND.
+  !IFEND.
+  !IFEND.
!IFEND.
RESTORE.
!ENDDEFINE.
```

Macro !INCIDEN

```
DO IF ($casenum=1).
+   PRINT /'Macro !INCIDEN V2003.07.07 cargada. Para imprimir la documentación ejecutar: !INCIDEN
HELP.'/.
END IF.
EXECUTE.
DEFINE !INCIDEN ( V=!CHAREND('/')
          /EXCLUDE=!CHAREND('/')
                /C=!CHAREND('/')
             /INDX=!CHAREND('/')
               /CN=!CHAREND('/') !DEFAULT('@casenum')
             /HELP=!CHAREND('/')!DEFAULT(1) ).
PRESERVE.
SET ERRORS=NONE PRINTBACK=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
DO IF ($casenum=1).
+   PRINT /'Macro !INCIDEN V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.
!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*************************************************************************'
          /'* DEPURACIÓN DE DATOS: Informe de incidencias                           *'
          /'* Creación 30.11.1998 Última revisión 07.07.2003                        *'
          /'* (c) A.Bonillo & JM. Doménech                                          *'
          /'* Email: MacrosSPSS@metodo.uab.es                                       *'
          /'*                                                                       *'
          /'* Llamada de la Macro:                                                  *'
          /'* !INCIDEN V= Lista de variables auxiliares a verificar                 *'
          /'* [/EXCLUDE]= Lista de valores (registro y variables) que no se desea listar *'
          /'*          /C= Lista de variables identificadoras del sujeto            *'
          /'*        /CN= Variable identificadora de la secuencia de los sujetos    *'
          /'*                                              (por defecto,@casenum)   *'
          /'*                                                                       *'
          /'*Ejemplos de llamada:                                                   *'
          /'*!INCIDEN V=@fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie/C=H CASO. *'
          /'*!INCIDEN V=@fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab @cie      *'
          /'*  /EXCLUDE=2 @cie,                                                     *'
          /'*           3 @dcs,                                                     *'
          /'*           4 @talla @fuma @tab @tiptab                                 *'
          /'*        /C=H CASO.                                                     *'
          /'*************************************************************************'.
+  END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL ) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE
+     TEMPORARY.
+     COMPUTE a=1.
+     RANK A BY A /N into @ /PRINT=NO.
+    !IF (!EXCLUDE<>!NULL) !THEN
+     TEMPORARY.
*     Desactivar los casos de EXCLUDE.
+     !LET !ID=2.
+     !DO !i !IN (!EXCLUDE).
+      !IF (!ID=2) !THEN
+       !LET !ID=1.
+        DO IF (@CASENUM=!I).
+      !ELSE.
+        !IF (!ID=1 !AND !I<>',') !THEN
+           COMPUTE !I=-9.
+        !ELSE.
+           !IF (!ID=1 !AND !I=',') !THEN
+             !LET !ID=2.
+              END IF.
+           !IFEND.
+        !IFEND.
+      !IFEND.
+     !DOEND.
+     END IF.
+    !IFEND.
+     COUNT #NER=!V (1 thru HI).
+     COUNT #NMr=!V (0).
+     COUNT #NM =!V (-9).
+     COUNT #NC =!V (-4 thru -1).
+     COMPUTE #NERcum=#NERcum+#NER.
+     COMPUTE #NMRcum=#NMRcum+#NMr.
+     COMPUTE #NMcum =#NMcum+#NM.
+     COMPUTE #NCcum =#NCcum+#NC.
+     COMPUTE #TOT=SUM(#NERcum,#NMRcum,#NMcum,#NCcum).
+     FORMATS #NERcum #NMRcum #NMcum #NCcum #TOT (F8).
+     COMPUTE #PCTNER=100*#NERcum/#TOT.
+     COMPUTE #PCTNMr=100*#NMrcum/#TOT.
```

```
+     COMPUTE #PCTNM =100*#NMcum/#TOT.
+     COMPUTE #PCTNC =100*#NCcum/#TOT.
+     FORMATS #PCTNER #PCTNM #PCTNMR #PCTNC (F6.3).
+     COMPUTE #@@=#@@+1.
+     DO IF (#@@=@).
+       PRINT /'ESTADÍSTICA DE INCIDENCIAS LISTADAS ----------------------------------------------
------'.
+       PRINT /'Total ERRORES ..... ='#NERcum '(' #PCTNER '%)'.
+       PRINT /'Total MISSING recuperables ='#NMRcum '(' #PCTNMR '%)'.
+       !IF (!EXCLUDE<>!NULL) !THEN
+         PRINT /'           no recuperados  ='#NMcum  '(' #PCTNM '%)'.
+       !IFEND.
+       PRINT /'Total valores correctos... ='#NCcum  '(' #PCTNC '%)'.
+       PRINT
/'=============================================================================================='/.
+     END IF.
+     EXECUTE.
+    !IF (!EXCLUDE<>!NULL) !THEN
+     TEMPORARY.
*      Desactivar los casos de EXCLUDE.
+     !LET !ID=2.
+     !DO !i !IN (!EXCLUDE).
+      !IF (!ID=2) !THEN
+       !LET !ID=1.
+        DO IF (@CASENUM=!I).
+      !ELSE.
+        !IF (!ID=1 !AND !I<>',') !THEN
+           COMPUTE !I=-9.
+        !ELSE.
+           !IF (!ID=1 !AND !I=',') !THEN
+             !LET !ID=2.
+              END IF.
+           !IFEND.
+        !IFEND.
+      !IFEND.
+     !DOEND.
+     END IF.
+    !IFEND.
+     !LET !CAB=!NULL.
+     !LET !x=!CONCAT(!C,' ',!INDX).
+     !DO !i !IN (!x).
+       !IF (!CAB=!NULL) !THE
+         !LET !CAB=!CONCAT(!CAB,!QUOTE(!CONCAT(!i,'= ')), !i).
+       !ELSE.
+         !LET !CAB=!CONCAT(!CAB,!QUOTE(!CONCAT(' ; ',!i,'= ')), !i).
+       !IFEND.
+     !DOEND.
*     Errores y missing.
+     COUNT #NER=!V (1 thru HI).
+     COUNT #NM =!V(0).
+     COUNT #NC =!V (-4 thru -1).
+     FORMATS #NER #NM #NC (F4).
*     Impresión de las cabeceras.
+     DO IF (#NER+#NM>0).
+       PRINT  /'Identificador caso: ' !CAB.
+       PRINT  /'Número de Registro: '!CN ';           Número de incidencias: Error=' #NER ';
Missing='#NM.
+       PRINT  /'----------------------------------------------------------------------------------
-------+'.
+     END IF.
*     Impresión de valores.
+     !DO !i !IN (!V).
+       !LET !VV=!SUBSTR(!i,2).
+       !LET !V·=!CONCAT('·',!VV).
+        DO IF (!i=0).
+          PRINT /!QUOTE(!i) ' = '8 'Missing Rec-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=1).
+          PRINT /!QUOTE(!i) ' = '8 'Incons Salt-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=2).
+          PRINT /!QUOTE(!i) ' = '8 'Err Formato-> '   !QUOTE(!V·) ' = '33 !V· '|'90.
+        ELSE IF (!i=3).
+          PRINT /!QUOTE(!i) ' = '8 'Fuera Rango-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=4).
+          PRINT /!QUOTE(!i) ' = '8 'Dif fuera rango-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=5).
+          PRINT /!QUOTE(!i) ' = '8 'Err Num Dec-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=10).
+          PRINT /!QUOTE(!i) ' = '8 'Err Cnstant-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=11).
+          PRINT /!QUOTE(!i) ' = '8 'Excso Varcn-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i=12).
+          PRINT /!QUOTE(!i) ' = '8 'Err Baja caso-> '   !QUOTE(!VV) ' = '33 !VV '|'90.
+        ELSE IF (!i>=50).
+          PRINT /!QUOTE(!i) ' = '8 'Err Cond '!i '-> '!QUOTE(!VV) ' = '33 !VV '|'90.
+        END IF.
+     !DOEND.

*      Impresión del pie de las cabeceras.
+      DO IF (#NER+#NM>0).
```

```
+        PRINT
/'=============================================================================================+'/.
+      END IF.
+      EXECUTE.
+      MATCH FILE /FILE=* /DROP=@.
+      EXECUTE.
+  !IFEND.
!IFEND.
RESTORE.
!ENDDEFINE.
```

Macro !CORREC

```
DO IF ($casenum=1).
+   PRINT /'Macro !CORREC V2003.07.07 cargada. Para imprimir la documentación ejecutar: !CORREC
HELP.'/.
END IF.
EXECUTE.
DEFINE !CORREC ( V=!CHAREND('/')
                 /C=!CHAREND('/')
               /CN=!CHAREND('/') !DEFAULT('@casenum')
             /HELP=!CHAREND('/') !DEFAULT(1) ).
PRESERVE.
SET ERRORS=NONE PRINTBACK=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
DO IF ($casenum=1).
+   PRINT /'Macro !CORREC V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.

!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*************************************************************************'
           /'* DEPURACIÓN DE DATOS: Asignación automática a missing de incidencias        *'
           /'* Creación 08.06.2003 Última revisión 07.07.2003                            *'
           /'* (c) A.Bonillo & JM. Doménech                                              *'
           /'* Email: MacrosSPSS@metodo.uab.es                                           *'
           /'*                                                                           *'
           /'* Llamada de la Macro:                                                      *'
           /'* !CORREC V= Lista de las variables auxiliares                              *'
           /'*        /C= Lista de variables identificadoras del sujeto                  *'
           /'*       /CN= Variable identificadora de la secuencia de los sujetos         *'
           /'*                                                  (por defecto,@casenum)   *'
           /'*                                                                           *'
           /'* Ejemplos de llamada:                                                      *'
           /'* !CORREC V= @h   @caso @fr @fn @sexo @talla @dpt @dcs @fuma @tab @tiptab    *'
           /'*             @cie @pad @pas  @exitus /C=h caso.                            *'
           /'*************************************************************************'.
+  END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL ) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE

+     TEMPORARY.
+     COMPUTE a=1.
+     RANK A BY A /N into @ /PRINT=NO.

+     COUNT #NER=!V (1 thru HI).
+     COUNT #NMr=!V (0).
+     COUNT #NM =!V (-9).
+     COUNT #NC =!V (-4 thru -1).
+     COMPUTE #NERcum=#NERcum+#NER.
+     COMPUTE #NMRcum=#NMRcum+#NMr.
+     COMPUTE #NMcum =#NMcum+#NM.
+     COMPUTE #NCcum =#NCcum+#NC.
+     COMPUTE #TOT=SUM(#NERcum,#NMRcum,#NMcum,#NCcum).
+     FORMATS #NERcum #NMRcum #NMcum #NCcum #TOT (F8).

+     COMPUTE #PCTNER=100*#NERcum/#TOT.
+     COMPUTE #PCTNMr=100*#NMrcum/#TOT.
+     COMPUTE #PCTNM =100*#NMcum/#TOT.
+     COMPUTE #PCTNC =100*#NCcum/#TOT.
+     FORMATS #PCTNER #PCTNM #PCTNMR #PCTNC (F6.3).

+     COMPUTE #@@=#@@+1.
+     DO IF (#@@=@).
+       PRINT /'ESTADÍSTICA DE VALORES ASIGNADOS A MISSING ---------------------------------------
------'.
+       PRINT /'Total ERRORES ..... ='#NERcum '(' #PCTNER '%)'.
+       PRINT /'Total MISSING recuperables ='#NMRcum '(' #PCTNMR '%)'.
+       PRINT /'Total valores correctos... ='#NCcum  '(' #PCTNC '%)'.
+       PRINT
/'=========================================================================================='/.
+       PRINT /'LISTADO DE VALORES ASIGNADOS A MISSING -------------------------------------------
------'.
+     END IF.
+     EXECUTE.

+     !LET !CAB=!NULL.
+     !LET !x=!CONCAT(!C,' ',!INDX).
+     !DO !i !IN (!x).
+       !IF (!CAB=!NULL) !THE
+         !LET !CAB=!CONCAT(!CAB,!QUOTE(!CONCAT(!i,'= ')), !i).
+       !ELSE.
```

```
+         !LET !CAB=!CONCAT(!CAB,!QUOTE(!CONCAT(' ; ',!i,'= ')), !i).
+       !IFEND.
+     !DOEND.

*     Errores y missing.
+     COUNT #NER=!V (1 thru HI).
+     COUNT #NM =!V(0).
+     COUNT #NC =!V (-4 thru -1).
+     FORMATS #NER #NM #NC (F4).

*     Impresión de las cabeceras.
+     DO IF (#NER>0).
+       PRINT  /'Identificador caso: ' !CAB.
+       PRINT  /'Número de Registro: '!CN ';                                    Número de
error(es)=' #NER.
+       PRINT  /'Variable y valor(es) erróneo(s) ----------------------------------------------------
-------+'.
+     END IF.

*     Impresión de valores.
+     !DO !i !IN (!V).
+       !LET !VV=!SUBSTR(!i,2).
+       !LET !V·=!CONCAT('·',!VV).
+        DO IF (!i>=1 and !i<>2).
+          PRINT /!QUOTE(!vv) ' = '9 !VV  '|'90.
+        END IF.
+     !DOEND.

+     !DO !i !IN (!V).
+       !LET !VV=!SUBSTR(!i,2).
+        IF (!i>=1) !VV=$SYSMIS.
+        IF (!i>=1) !VV=' '.
+     !DOEND.

*      Impresión del pie de las cabeceras.
+      DO IF (#NER>0).
+        PRINT
/'=============================================================================================+'/.
+      END IF.
+      EXECUTE.
+      MATCH FILE /FILE=* /DROP=@.
+      EXECUTE.
+  !IFEND.
!IFEND.
RESTORE.
!ENDDEFINE.
```

Macro !QUALITY

```
DO IF ($casenum=1).
+   PRINT /'Macro !QUALITY V2003.07.07 cargada. Para imprimir la documentación ejecutar: !QUALITY
HELP.'/.
END IF.
EXECUTE.
DEFINE !QUALITY ( V=!CHAREND('/')
              /FILE=!CHAREND('/')
              /SPLIT=!CHAREND('/')
              /HELP=!CHAREND('/')!DEFAULT(1) ).
PRESERVE.
SET ERRORS=NONE PRINTBACK=NONE LENGTH=NONE WIDTH=132 MESSAGES=NONE.
DO IF ($casenum=1).
+   PRINT /'Macro !QUALITY V2003.07.07 (c)A.Bonillo & JM.Doménech'/.
END IF.

!IF (!HELP<>1) !THEN
*Impresión de la documentación.
+  DO IF ($casenum=1).
+    PRINT /'*****************************************************************************'
           /'* DEPURACIÓN DE DATOS: Valoración de la calidad de los datos                *'
           /'* Creación 08.06.2001 Última revisión 07.07.2003                           *'
           /'* (c) A.Bonillo & JM. Doménech                                             *'
           /'* Email: MacrosSPSS@metodo.uab.es                                          *'
           /'*                                                                          *'
           /'* Llamada de la Macro:                                                     *'
           /'* !QUALITY V= Lista de las variables depuradas                             *'
           /'*      /FILE= Nombre (y ruta) del archivo sin depurar en formato SPSS (.SAV) *'
           /'*   [/SPLIT]= Variable(s) de segmentación para realizar la estadística     *'
           /'*                                                                          *'
           /'* Ejemplos de llamada:                                                     *'
           /'* !QUALITY V=h caso fr fn sexo talla dpt dcs fuma tab tiptab cie            *'
           /'*              /FILE='TestNoDep.SAV'.                                      *'
           /'*****************************************************************************'.
+  END IF.
EXECUTE.
!ELSE.
*   Comprobación de parámetros.
+  !IF (!V=!NULL !OR !FILE=!NULL) !THEN
+    DO IF ($casenum=1).
+      PRINT /'ERROR: Falta parámetro'.
+    END IF.
+    EXECUTE.
+  !ELSE

+    !LET !@V=!NULL.
+    !LET !V@=!NULL.
+    !LET !·V=!NULL.
*    Construcción de nombres.
+    !DO !i !IN (!V).
+      !LET !@V=!CONCAT(!@V,' ','@',!I).
+      !LET !V@=!CONCAT(!V@,' ',!I,'@').
+      !LET !·V=!CONCAT(!·V,' ','·',!I).
+    !DOEND.

+    TEMPORARY.
+      COMPUTE a=1.
+      RANK A BY A /N into @ /PRINT=NO.

******************Errores.
+      ADD FILES /FILE=*  /FILE=!FILE  /IN=@ID@.
+      EXECUTE.

+    NUMERIC !V@.
*    Etiquetado de variables.
+    !DO !i !IN (!V).
+       VARIABLE LABEL !CONCAT(!I,'@') !QUOTE(!UPCASE(!I)).
+       VARIABLE LABEL !CONCAT('@',!I) !QUOTE(!UPCASE(!I)).
+    !DOEND.

+      IF (@ID@=1) #CASENUM=#CASENUM+1.
+      IF (@ID@=1) @CASENUM=#CASENUM.
+      SORT CASES BY @CASENUM.

+      *Marcar como 1 los errores y 0 los aciertos.
+      DO REPEAT VAR=!V /VAR2=!V@ /VAR3=!·V.
+        COMPUTE var = UPCASE(var).
+        DO IF (@CASENUM=LAG(@CASENUM)).
+          COMPUTE VAR2=(VAR<>LAG(VAR)).
+          IF (NMISS(VAR,LAG(VAR))=1) VAR2=1.
+          IF (NMISS(VAR,LAG(VAR))=2) VAR2=0.
+          IF (SYSMIS(VAR) AND VAR3<>' ') VAR2=1.
+        END IF.
+      END REPEAT.

+      DO IF (@ID@=1).
+        COUNT NER=!V@ (1).
```

```
+        COUNT NC =!V@ (0).
+        COMPUTE TOT=SUM(NER+NC).
+      END IF.
+      EXECUTE.

+      SORT CASES BY !SPLIT.
+      SPLIT FILE LAYERED BY !SPLIT.

+    TEMPORARY.
+      SELECT IF (@ID@=1).
*      Número de errores por caso.
+      FORMATS NER (F4).
+      VARIABLE LABELS NER 'Errores por caso'.
+      FREQUENCIES VARIABLES=NER /STATISTICS SUM.

+    TEMPORARY.
+      SELECT IF (@ID@=1).
+      TITLE 'Errores por variable'.
+      FORMATS !V@ (F4.2).
+      DESCRIPTIVES VARIABLES=!V@ /STATISTICS=MEAN SUM /SORT=MEAN (D).

*    Estadística de datos con error.
+      DO IF (@ID@=1).
+        COMPUTE #NER=NER+#NER.
+        COMPUTE NER=#NER.
+        COMPUTE #NC=NC+#NC.
+        COMPUTE NC=#NC.
+        COMPUTE #TOT=TOT+#TOT.
+        COMPUTE TOTAL=#TOT.
+      END IF.

+      DO IF (@CASENUM>=@).
+       COMPUTE NC=LAG(NER).
+       COMPUTE A=0.
+      END IF.
+      IF (LAG(A)=0) A=1.

+      TEMP.
+      SELECT IF ANY(A,0,1).
+      WEIGHT by NC.
+        VAR LABELS A 'Estadística de datos con error'.
+        VAL LABELS A 0 'Errores' 1 'Correctos'.
+        FREQUENCIES VARIABLES=A /STATISTICS SUM.
+      WEIGHT  OFF.

*      Duplicados.
+      RANK VARIABLES= @ID@  BY @ID@  /N INTO @nseg /PRINT=NO.
+      SORT CASES BY @nseg (a).
+      IF ( @nseg>lag(@nseg) ) #nseg = @nseg-lag(@nseg).

+      DO IF (#nseg >0 AND $CASENUM=(@nseg+1) ).
+        COMPUTE #DUPPCT=100*#nseg /@nseg.
+        DO IF (@CASENUM<100).
+          PRINT/'Registros repetidos: ' #nseg (F2) '/' @nseg(F2) ' (' #DUPPCT(F5.2) '%)'.
+        ELSE IF (@CASENUM<1000).
+          PRINT/'Registros repetidos: ' #nseg (F3) '/' @nseg(F3) ' (' #DUPPCT(F5.2) '%)'.
+        ELSE IF (@CASENUM<10000).
+          PRINT/'Registros repetidos: ' #nseg (F4) '/' @nseg(F4) ' (' #DUPPCT(F5.2) '%)'.
+        ELSE IF (@CASENUM<100000).
+          PRINT/'Registros repetidos: ' #nseg (F5) '/' @nseg(F8) ' (' #DUPPCT(F5.2) '%)'.
+        ELSE IF (@CASENUM<1000000).
+          PRINT/'Registros repetidos: ' #nseg (F6) '/' @nseg(F8) ' (' #DUPPCT(F5.2) '%)'.
+        ELSE IF (@CASENUM<10000000).
+          PRINT/'Registros repetidos: ' #nseg (F7) '/' @nseg(F8) ' (' #DUPPCT(F5.2) '%)'.
+        END IF.
+      END IF.
+      NUMERIC @@.
+      SORT CASES BY @CASENUM.

******************Missing.
+      SELECT IF @ID@=0.
+      MATCH FILE /FILE=* /DROP=@ to @@.

+      SORT CASES BY !SPLIT.
+      SPLIT FILE LAYERED BY !SPLIT.

*    Estadísticas de missing.
+    TEMPORARY.
*      Número de missing por caso.
+      COUNT NM=!@V(0).
+      FORMATS NM (F4).
+      VAR LABELS NM 'Valores missing por caso'.
+      FREQUENCIES VARIABLES=NM /STATISTICS SUM.

+      SPLIT FILE LAYERED BY !SPLIT.

+    TEMPORARY.
+      TITLE 'Valores missing por variable'.
+      RECODE  !@V (0=1) (-2=SYSMIS) (ELSE=0).
```

```
+      FORMATS !@V (F4.2).
+      DESCRIPTIVES VARIABLES=!@V /STATISTICS =MEAN SUM /SORT=MEAN(D).

+    TEMPORARY.
+      TITLE 'Estadística de datos con valor missing'.
+      COUNT NM =!@V(-1).
+      COUNT NMr=!@V(0).
+      COUNT RESTO=!@V(-4 THRU -3,1 THRU HI).
+      COMPUTE TOT=SUM(NM,NMr,RESTO).
+      WEIGHT by TOT.
+      COMPUTE NM=NM/TOT.
+      COMPUTE NMR=NMR/TOT.
+      FORMATS NM NMR (F4.2).
+      VAR LABELS NM  'Valores missing no recuperables'
                 /NMR 'Valores missing recuperables'.
+      DESCRIPTIVES VARIABLES=NMr NM /STATISTICS=MEAN SUM.
+      SPLIT FILE OFF.
+  !IFEND.
!IFEND.
RESTORE.
!ENDDEFINE.
RESTORE.
PRESERVE.
SET ERRORS=NONE.
EXECUTE.
RESTORE.
```

Anexo 2:
PLANTILLA PARA EFECTUAR LA DEPURACIÓN

```
*Doble lectura de datos ASCII: con formato original y con formato cadena.
DATA LIST FILE='NoDepur.DAT'
  /Vident    X-X(A)
   Vcadena   X-X(A)
   Vnum1     X-X(F)     ·Vnum1    X-X(A)
   Vnum2     X-X(F)     ·Vnum2    X-X(A)
   Vfecha1   X-X(F)     ·Vfecha1  X-X(A)
   Vfecha2   X-X(F)     ·Vfecha2  X-X(A).
SET ERRORS=NONE.      /*Desactiva el listado de errores de formato del DATA LIST.
EXECUTE.
SET ERRORS=ON.        /*Activa de nuevo el listado de mensajes de error.

*Transformación del contenido de las variables cadena a mayúscula.
DO REPEAT var = Vident to Vcadena1.
    COMPUTE var = UPCASE(var).
END REPEAT.

*Grabación del archivo no depurado necesario para la estadística de calidad.
SAVE OUTFILE='NoDepur.SAV'.

*Creación de la variable con el orden secuencial de los sujetos.
COMPUTE @casenum= $casenum.
FORMATS @casenum (F6).
EXECUTE.

*Corrección de identificadores.
SELECT IF (@casenum<>XX).    /* Eliminación de un registro duplicado.
IF (@casenum= X) Vident= X.
EXECUTE.

*Correcciones.
IF (Vident='X') Vnum = X.
EXECUTE.

*Depuración del identificador.
!IDT V= Vident.   /TABLE='TPrincipal.SAV'.

*Comprobación variable cadena (identificador).
!DR V=Vident /LV='X','Y' /MVr=' '.

*Comprobación variable cuantitativa.
!DR V=Vnum1 /LV=X THRU Y /ND=NúmeroDecimales /FORMAT=1 /C=Vident.

*Comprobación variable Vfecha1.
!DRF V=Vfecha1 /FI=día,mes,año /FS= día,mes,año /FORMAT=1 /C=Vident.

*Comprobación rango entre Vfecha1 y Vfecha2.
!DDF V=Vfecha2 /D=Vfecha1-Vfecha2 /MIN=X /MAX=Y /FORMAT=1 /C=Vident.

*Comprobación variable Vnum2 implicada en una condición lógica.
COMPUTE @Vnum2=$SYSMIS.
IF (Vnum2=X AND Vcadena='Y') @Vnum2=50.
*Comprobación variable a través de un diccionario.
!DRKey V=Vnum2 /TABLE='Dccnario.SAV' /LVL=Vcadena /MVr=' ' /FORMAT=1 /C=Vident.

*Comprobación variable cadena (afectada por un salto).
!DR V=Vcadena /LV='X','Y' /VS=VarSalto /FORMAT=1 /C=Vident.

*Informe de incidencias.
!INCIDEN V=@Vcadena @Vnum1 @Vnum2 @Vfecha1 @Vfecha2 /C=Vident.
!INCIDEN V=@Vcadena @Vnum1 @Vnum2 @Vfecha1 @Vfecha2 /C=Vident /EXCLUDE=X @Vnum1.

*Estadísticas de errores y missing.
!QUALITY V=Vident Vcadena Vnum1 Vnum2 Vfecha1 Vfecha2 /FILE='NoDepur.SAV'.

*Grabar los datos depurados en formato SPSS.
SAVE OUTFILE = 'Depur.SAV' /KEEP = Vident TO Vfecha2.
```

Anexo 3:
SINTAXIS PARA DEPURAR LA E.S.D.

```
*Lectura y depuración del fichero de datos con todas las variables a utilizar,
        ya hemos eliminado las variables aprox y blancos intermedios.
DEFINE !TF ().
       *1era parte: Lectura de datos.
FILE TYPE GROUPED FILE
 'C:\WINDOWS\Escritorio\esd\sdorden.dat'
 CASE ident 1-7 RECORD ficha 8-10 MISSING=NOWARN WILD=NOWARN
        DUPLICATE=WARN ORDERED=NO.
!DO !i=1 !TO 1.
*Esta orden convierte el formato de la ficha (i, una posición) en un
 formato con tres posiciones,ejemplo, de ser valor 1 pasa a ser valor 001.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-12 !CONCAT (!viii, '13') 13-15
 !CONCAT (!viii, '16') 16 !CONCAT (!viii, '17') 17-18
 !CONCAT (!viii, '19') 19-21 !CONCAT (!viii, '22') 22-30 (5)
 !CONCAT (!viii, '31') 31-38 (4).
       !DOEND.
!DO !i=2 !TO 2.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11  !CONCAT (!viii, '14') 14-15
 !CONCAT (!viii, '16') 16-18 !CONCAT (!viii, '24') 24
 !CONCAT (!viii, '25') 25    !CONCAT (!viii, '26') 26
 !CONCAT (!viii, '27') 27-28 !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '32') 32 !CONCAT (!viii, '35') 35
 !CONCAT (!viii, '36') 36-37.
       !DOEND.
!DO !i=3 !TO 17.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '16') 16-18
 !CONCAT (!viii, '24') 24 !CONCAT (!viii, '25') 25 !CONCAT (!viii, '26') 26
 !CONCAT (!viii, '27') 27-28 !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '32') 32 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34
 !CONCAT (!viii, '36') 36-37 .
       !DOEND.
```

```
!DO !i=18 !TO 21.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '13') 13 (a) !CONCAT (!viii, '14') 14-15
 !CONCAT (!viii, '17') 17-18 !CONCAT (!viii, '20') 20
 !CONCAT (!viii, '21') 21 (a) !CONCAT (!viii, '22') 22-23
 !CONCAT (!viii, '25') 25-26 !CONCAT (!viii, '28') 28
 !CONCAT (!viii, '29') 29-31  !CONCAT (!viii, '33') 33-35
 !CONCAT (!viii, '37') 37 !CONCAT (!viii, '38') 38
 !CONCAT (!viii, '39') 39-41 !CONCAT (!viii, '43') 43 !CONCAT (!viii, '44') 44
 !CONCAT (!viii, '45') 45 !CONCAT (!viii, '46') 46
 !CONCAT (!viii, '47') 47 !CONCAT (!viii, '48') 48
 !CONCAT (!viii, '49') 49 !CONCAT (!viii, '50') 50
 !CONCAT (!viii, '51') 51 !CONCAT (!viii, '52') 52-54 (a)
 !CONCAT (!viii, '55') 55 !CONCAT (!viii, '56') 56-57.
       !DOEND.

!DO !i=22 !TO 37.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12  (a)
 !CONCAT (!viii, '13') 13-14  !CONCAT (!viii, '16') 16-17
 !CONCAT (!viii, '19') 19 !CONCAT (!viii, '20') 20-21
 !CONCAT (!viii, '23') 23-24 !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '28') 28 !CONCAT (!viii, '29') 29-31
 !CONCAT (!viii, '33') 33-35 !CONCAT (!viii, '37') 37
 !CONCAT (!viii, '38') 38-40 !CONCAT (!viii, '42') 42.
       !DOEND.

!DO !i=38 !TO 41.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
       !LET !viii=!CONCAT('v',!iii).

DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12-13
 !CONCAT (!viii, '15') 15-16  !CONCAT (!viii, '18') 18
 !CONCAT (!viii, '19') 19-21  !CONCAT (!viii, '23') 23-25
 !CONCAT (!viii, '27') 27 !CONCAT (!viii, '28') 28
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '32') 32 !CONCAT (!viii, '33') 33
 !CONCAT (!viii, '34') 34-35 !CONCAT (!viii, '37') 37-38
 !CONCAT (!viii, '40') 40 !CONCAT (!viii, '41') 41
 !CONCAT (!viii, '42') 42-43 !CONCAT (!viii, '44') 44
 !CONCAT (!viii, '45') 45 !CONCAT (!viii, '46') 46
 !CONCAT (!viii, '47') 47 !CONCAT (!viii, '48') 48
 !CONCAT (!viii, '49') 49-51 (a)  !CONCAT (!viii, '52') 52
 !CONCAT (!viii, '53') 53 !CONCAT (!viii, '54') 54
 !CONCAT (!viii, '55') 55 !CONCAT (!viii, '56') 56
 !CONCAT (!viii, '57') 57.
       !DOEND.
```

```
!DO !i=42 !TO 57.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '13') 13 !CONCAT (!viii, '14') 14 (a)
 !CONCAT (!viii, '15') 15-16 !CONCAT (!viii, '18') 18-19
 !CONCAT (!viii, '21') 21 !CONCAT (!viii, '22') 22 (a)
 !CONCAT (!viii, '23') 23-24 !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '31') 31
 !CONCAT (!viii, '32') 32-33 !CONCAT (!viii, '35') 35-36
 !CONCAT (!viii, '38') 38 !CONCAT (!viii, '39') 39-40
 !CONCAT (!viii, '42') 42 !CONCAT (!viii, '43') 43.
       !DOEND.

!DO !i=58 !TO 58.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

RECORD TYPE !i.
       !LET !viii=!CONCAT('v',!iii).

DATA LIST /!CONCAT (!viii, '11') 11-13 !CONCAT (!viii, '14') 14-15
 !CONCAT (!viii, '16') 16-17 !CONCAT (!viii, '18') 18-24
 !CONCAT (!viii, '25') 25  !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '29') 29-30  !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33-34 !CONCAT (!viii, '35') 35-36
 !CONCAT (!viii, '37') 37-38 !CONCAT (!viii, '39') 39
 !CONCAT (!viii, '41') 41-42 !CONCAT (!viii, '43') 43-44 (a)
 !CONCAT (!viii, '45') 45-46 (a)
 !CONCAT (!viii, '47') 47-48 (a) !CONCAT (!viii, '49') 49
 !CONCAT (!viii, '50') 50 !CONCAT (!viii, '51') 51
 !CONCAT (!viii, '52') 52 !CONCAT (!viii, '53') 53
 !CONCAT (!viii, '54') 54 !CONCAT (!viii, '55') 55
 !CONCAT (!viii, '56') 56 !CONCAT (!viii, '57') 57-58
 !CONCAT (!viii, '59') 59-60 (a) !CONCAT (!viii, '61') 61-62.
       !DOEND.

!DO !i=59 !TO 59.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-13 !CONCAT (!viii, '14') 14-15
 !CONCAT (!viii, '16') 16-17 !CONCAT (!viii, '18') 18-24
 !CONCAT (!viii, '25') 25  !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33-34 !CONCAT (!viii, '35') 35-36
 !CONCAT (!viii, '37') 37-38 !CONCAT (!viii, '39') 39
 !CONCAT (!viii, '41') 41-42
 !CONCAT (!viii, '43') 43-44 !CONCAT (!viii, '45') 45-46.
       !DOEND.

!DO !i=60 !TO 73.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
```

```
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-13 !CONCAT (!viii, '14') 14-15
 !CONCAT (!viii, '16') 16-17 !CONCAT (!viii, '18') 18-24
 !CONCAT (!viii, '25') 25  !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33-34 !CONCAT (!viii, '35') 35-36
 !CONCAT (!viii, '37') 37-38 !CONCAT (!viii, '39') 39
 !CONCAT (!viii, '41') 41-42.
       !DOEND.

!DO !i=74 !TO 81.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-12 !CONCAT (!viii, '13') 13
 !CONCAT (!viii, '14') 14 !CONCAT (!viii, '15') 15
 !CONCAT (!viii, '16') 16  !CONCAT (!viii, '17') 17
 !CONCAT (!viii, '18') 18-19 !CONCAT (!viii, '21') 21-22
 !CONCAT (!viii, '24') 24  !CONCAT (!viii, '25') 25
 !CONCAT (!viii, '26') 26 !CONCAT (!viii, '27') 27-28
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '31') 31-32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34
 !CONCAT (!viii, '35') 35 !CONCAT (!viii, '36') 36
 !CONCAT (!viii, '37') 37.
       !DOEND.

!DO !i=82 !TO 82.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '13') 13-14
 !CONCAT (!viii, '16') 16  !CONCAT (!viii, '17') 17
 !CONCAT (!viii, '18') 18-20 !CONCAT (!viii, '21') 21-22
 !CONCAT (!viii, '23') 23 !CONCAT (!viii, '24') 24
 !CONCAT (!viii, '25') 25-26
 !CONCAT (!viii, '28') 28 !CONCAT (!viii, '29') 29
 !CONCAT (!viii, '30') 30-32 !CONCAT (!viii, '33') 33-34
 !CONCAT (!viii, '35') 35 !CONCAT (!viii, '36') 36
 !CONCAT (!viii, '37') 37-38
 !CONCAT (!viii, '40') 40 !CONCAT (!viii, '41') 41
 !CONCAT (!viii, '42') 42.
       !DOEND.

!DO !i=83 !TO 83.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '13') 13 !CONCAT (!viii, '14') 14
 !CONCAT (!viii, '15') 15 !CONCAT (!viii, '16') 16
 !CONCAT (!viii, '17') 17 !CONCAT (!viii, '18') 18
 !CONCAT (!viii, '19') 19 !CONCAT (!viii, '20') 20.
       !DOEND.

!DO !i=84 !TO 84.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
```

```
!IFEND.
!IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11  !CONCAT (!viii, '16') 16-17
 !CONCAT (!viii, '18') 18-20 !CONCAT (!viii, '22') 22-23
 !CONCAT (!viii, '25') 25-26 !CONCAT (!viii, '27') 27
 !CONCAT (!viii, '28') 28-29
 !CONCAT (!viii, '31') 31 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34-36
 !CONCAT (!viii, '38') 38-39 !CONCAT (!viii, '41') 41
 !CONCAT (!viii, '42') 42 !CONCAT (!viii, '43') 43-44
 !CONCAT (!viii, '45') 45-46.
       !DOEND.

!DO !i=85 !TO 85.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST / !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '16') 16-17 !CONCAT (!viii, '18') 18-20
 !CONCAT (!viii, '22') 22-23 !CONCAT (!viii, '25') 25-26
 !CONCAT (!viii, '27') 27 !CONCAT (!viii, '28') 28-29
 !CONCAT (!viii, '31') 31 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34-36
 !CONCAT (!viii, '38') 38-39
 !CONCAT (!viii, '41') 41  !CONCAT (!viii, '42') 42
 !CONCAT (!viii, '43') 43-44 !CONCAT (!viii, '45') 45-46.
       !DOEND.

!DO !i=86 !TO 90.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '13') 13-15 !CONCAT (!viii, '16') 16-17
 !CONCAT (!viii, '18') 18-20 !CONCAT (!viii, '22') 22-23
 !CONCAT (!viii, '25') 25-26 !CONCAT (!viii, '27') 27
 !CONCAT (!viii, '28') 28-29 !CONCAT (!viii, '31') 31 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34-36
 !CONCAT (!viii, '38') 38-39 !CONCAT (!viii, '41') 41
 !CONCAT (!viii, '42') 42 !CONCAT (!viii, '43') 43-44
 !CONCAT (!viii, '45') 45-46 .
       !DOEND.

!DO !i=91 !TO 98.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).

!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-14 !CONCAT (!viii, '15') 15-16
 !CONCAT (!viii, '18') 18-19 !CONCAT (!viii, '21') 21-22
 !CONCAT (!viii, '24') 24-25 !CONCAT (!viii, '27') 27
 !CONCAT (!viii, '28') 28 !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '31') 31 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34
 !CONCAT (!viii, '35') 35 !CONCAT (!viii, '36') 36
 !CONCAT (!viii, '37') 37 !CONCAT (!viii, '38') 38.
       !DOEND.

!DO !i=99 !TO 99.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
```

```
+   !LET !iii=!i.
!IFEND.
!IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '13') 13-14 !CONCAT (!viii, '16') 16-17
 !CONCAT (!viii, '19') 19-20 !CONCAT (!viii, '22') 22-23
 !CONCAT (!viii, '25') 25 !CONCAT (!viii, '26') 26
 !CONCAT (!viii, '27') 27 !CONCAT (!viii, '28') 28
 !CONCAT (!viii, '29') 29 !CONCAT (!viii, '30') 30.
       !DOEND.

!DO !i=100 !TO 103.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-12
 !CONCAT (!viii, '14') 14-15 !CONCAT (!viii, '17') 17-18
 !CONCAT (!viii, '20') 20 !CONCAT (!viii, '21') 21-23 (a)
 !CONCAT (!viii, '24') 24-26 (a) !CONCAT (!viii, '27') 27
 !CONCAT (!viii, '28') 28 !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '31') 31-32 !CONCAT (!viii, '33') 33
 !CONCAT (!viii, '34') 34 !CONCAT (!viii, '35') 35
 !CONCAT (!viii, '36') 36 !CONCAT (!viii, '37') 37
 !CONCAT (!viii, '38') 38 !CONCAT (!viii, '39') 39-40
 !CONCAT (!viii, '42') 42-43 !CONCAT (!viii, '45') 45
 !CONCAT (!viii, '46') 46  !CONCAT (!viii, '49') 49
 !CONCAT (!viii, '50') 50 !CONCAT (!viii, '51') 51-52
 !CONCAT (!viii, '54') 54-55 !CONCAT (!viii, '57') 57-58
 !CONCAT (!viii, '60') 60-61 !CONCAT (!viii, '63') 63.
       !DOEND.

!DO !i=104 !TO 104.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12-13
 !CONCAT (!viii, '15') 15-16 !CONCAT (!viii, '18') 18-19
 !CONCAT (!viii, '20') 20-22 !CONCAT (!viii, '23') 23-24
 !CONCAT (!viii, '25') 25 !CONCAT (!viii, '26') 26
 !CONCAT (!viii, '27') 27  !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '31') 31-32 !CONCAT (!viii, '33') 33-34
 !CONCAT (!viii, '35') 35-36 !CONCAT (!viii, '37') 37
 !CONCAT (!viii, '38') 38 !CONCAT (!viii, '39') 39.
       !DOEND.

!DO !i=105 !TO 105.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12-13
 !CONCAT (!viii, '15') 15-16  !CONCAT (!viii, '18') 18
 !CONCAT (!viii, '19') 19-20  !CONCAT (!viii, '22') 22-23
 !CONCAT (!viii, '25') 25-26
 !CONCAT (!viii, '27') 27-28 !CONCAT (!viii, '29') 29
 !CONCAT (!viii, '30') 30 !CONCAT (!viii, '31') 31
 !CONCAT (!viii, '32') 32 !CONCAT (!viii, '33') 33.
       !DOEND.
```

```
!DO !i=106 !TO 108.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST / !CONCAT (!viii, '12') 12-13
 !CONCAT (!viii, '15') 15-16  !CONCAT (!viii, '18') 18
 !CONCAT (!viii, '19') 19-20  !CONCAT (!viii, '22') 22-23
 !CONCAT (!viii, '25') 25-26
 !CONCAT (!viii, '27') 27-28 !CONCAT (!viii, '29') 29
 !CONCAT (!viii, '30') 30 !CONCAT (!viii, '31') 31
 !CONCAT (!viii, '32') 32 !CONCAT (!viii, '33') 33.
       !DOEND.

!DO !i=109 !TO 109.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12
 !CONCAT (!viii, '13') 13 !CONCAT (!viii, '14') 14 !CONCAT (!viii, '15') 15
 !CONCAT (!viii, '16') 16 !CONCAT (!viii, '17') 17 !CONCAT (!viii, '18') 18
 !CONCAT (!viii, '19') 19 !CONCAT (!viii, '20') 20-22 (a)
 !CONCAT (!viii, '23') 23
 !CONCAT (!viii, '24') 24-27 !CONCAT (!viii, '28') 28-32
 !CONCAT (!viii, '33') 33-34 !CONCAT (!viii, '35') 35 !CONCAT (!viii, '36') 36
 !CONCAT (!viii, '37') 37 !CONCAT (!viii, '38') 38 !CONCAT (!viii, '39') 39
 !CONCAT (!viii, '40') 40-41 !CONCAT (!viii, '42') 42 !CONCAT (!viii, '43') 43
 !CONCAT (!viii, '44') 44-45 !CONCAT (!viii, '46') 46-47 !CONCAT (!viii,'48') 48
 !CONCAT (!viii, '49') 49  !CONCAT (!viii, '50') 50.
       !DOEND.

!DO !i=118 !TO 130.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '16') 16-18
 !CONCAT (!viii, '24') 24 !CONCAT (!viii, '25') 25 !CONCAT (!viii, '26') 26
 !CONCAT (!viii, '27') 27-28 !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '32') 32 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34
 !CONCAT (!viii, '36') 36-37 .
       !DOEND.

!DO !i=138 !TO 145.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11 !CONCAT (!viii, '12') 12  (a)
 !CONCAT (!viii, '13') 13-14  !CONCAT (!viii, '16') 16-17
 !CONCAT (!viii, '19') 19 !CONCAT (!viii, '20') 20-21
 !CONCAT (!viii, '23') 23-24 !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '28') 28 !CONCAT (!viii, '29') 29-31
 !CONCAT (!viii, '33') 33-35 !CONCAT (!viii, '37') 37
 !CONCAT (!viii, '38') 38-40 !CONCAT (!viii, '42') 42.
       !DOEND.
```

```
!DO !i=174 !TO 175.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-13 !CONCAT (!viii, '14') 14-15
 !CONCAT (!viii, '16') 16-17 !CONCAT (!viii, '18') 18-24
 !CONCAT (!viii, '25') 25  !CONCAT (!viii, '26') 26-27
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33-34 !CONCAT (!viii, '35') 35-36
 !CONCAT (!viii, '37') 37-38 !CONCAT (!viii, '39') 39
 !CONCAT (!viii, '41') 41-42.
       !DOEND.

!DO !i=182 !TO 183.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.

RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-12 !CONCAT (!viii, '13') 13
 !CONCAT (!viii, '14') 14 !CONCAT (!viii, '15') 15
 !CONCAT (!viii, '16') 16  !CONCAT (!viii, '17') 17
 !CONCAT (!viii, '18') 18-19 !CONCAT (!viii, '21') 21-22
 !CONCAT (!viii, '24') 24  !CONCAT (!viii, '25') 25
 !CONCAT (!viii, '26') 26 !CONCAT (!viii, '27') 27-28
 !CONCAT (!viii, '29') 29-30 !CONCAT (!viii, '31') 31-32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34
 !CONCAT (!viii, '35') 35 !CONCAT (!viii, '36') 36
 !CONCAT (!viii, '37') 37.
       !DOEND.

!DO !i=199 !TO 200.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
RECORD TYPE !i.
!LET !viii=!CONCAT('v',!iii).
DATA LIST /!CONCAT (!viii, '11') 11-14 !CONCAT (!viii, '15') 15-16
 !CONCAT (!viii, '18') 18-19 !CONCAT (!viii, '21') 21-22
 !CONCAT (!viii, '24') 24-25 !CONCAT (!viii, '27') 27
 !CONCAT (!viii, '28') 28 !CONCAT (!viii, '29') 29-30
 !CONCAT (!viii, '31') 31 !CONCAT (!viii, '32') 32
 !CONCAT (!viii, '33') 33 !CONCAT (!viii, '34') 34
 !CONCAT (!viii, '35') 35 !CONCAT (!viii, '36') 36
 !CONCAT (!viii, '37') 37 !CONCAT (!viii, '38') 38.
       !DOEND.

END FILE TYPE.
       EXECUTE.
```

```
*2ª parte: Depuración.
!DO !i=1 !TO 1.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).
!DR V=!CONCAT (!viii,'11') /LV=1 thru 52 /C=ident.
!DR V=!CONCAT (!viii,'13') /LV=0 thru 999 /C=ident.
!DR V=!CONCAT (!viii,'16') /LV=0,1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'17') /LV=1 thru 21 /C=ident.
!DR V=!CONCAT (!viii,'19') /LV=1 thru 313 /C=ident.
       !DOEND.

!DO !i=2 !TO 2.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).
!DR V=!CONCAT (!viii,'11')  /LV=1,6 /C=ident.
*         1,2,3,4,5,6,7,8,9,10,11 y 12 (v00214).
!DR V=!CONCAT (!viii,'14')  /LV=1,2,3,4,5,6,7,8,9,10,11,12 /C=ident.
*         900><992 (v00216).
!DR V=!CONCAT (!viii,'16') /LV=901 thru 991 /C=ident.
*         1 y 6 (v00224).
!DR V=!CONCAT (!viii,'24') /LV=1,6 /C=ident.
*         Comprobación lógica entre estado civil y edad del sujeto:
            edad(variable que hemos de crear a partir del año de
          nacimiento y de la fecha de realización de la encuesta);
            una persona menor de 14 años sólo puede estar soltera
          vedad<= 14 y v00225 es diferente a 1 es un error @=3.
COMPUTE !CONCAT ('@',!viii,'25')=$sysmis.
       IF ((1991-1000+!CONCAT(!viii,'16'))<15 AND !CONCAT(!viii,'25')<>1)
!CONCAT('@',!viii,'25')=3.

*         1,2,3,4 y 5 (v00225).
!DR V=!CONCAT (!viii,'25') /LV=1,2,3,4,5 /C=ident /LVL=!CONCAT (!viii,'16').
*         1,2 y 3 (v00226).
!DR V=!CONCAT (!viii,'26') /LV=1,2,3 /C=ident.
*         PAises  (v00227).
!DR V=!CONCAT (!viii,'27')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
       81,82,83,84,91,92,93,94,95 /C=ident.

*         Comprobación lógica entre año de la última llegada a España (v00229)
             y  año de nacimiento:
             No puede haber llegado a España en una fecha anterior a la
          de su año de nacimiento, Ojo porque el año de nacimiento está expresado
          con tres dígitos, mientras que el año de la última llegada a España está
               expresado con dos dígitos.

COMPUTE !CONCAT ('@',!viii,'29')=$sysmis.
IF (1900+!CONCAT (!viii,'29')<1000+!CONCAT(!viii, '16'))
 !CONCAT('@',!viii,'29')=3.
*         De 0 a 91 (v00229).
!DR V=!CONCAT (!viii,'29') /LV=0 thru 91 /C=ident /LVL=!CONCAT (!viii, '16').
*         1 (v00232).
!DR V=!CONCAT (!viii,'32') /LV=1  /C=ident.
*         1,2,3,4,5 y 6 (v00235).
!DR V=!CONCAT (!viii,'35') /LV=1,2,3,4,5, 6  /C=ident.
*         1 (v00236).
!DR V=!CONCAT (!viii,'36') /LV=1  /C=ident.
       !DOEND.

!DO !i=3 !TO 17.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
```

```
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).


*Los tipos de depuración en esta segunda fase van a ser los siguientes
     1) Lista de valores
     1 y 6 (v00211).
!DR V=!CONCAT (!viii,'11')  /LV=1,6 /C=ident.
*         900><992 (v00216).
!DR V=!CONCAT (!viii,'16') /LV=901 thru 991 /C=ident.
*         1 y 6 (v00224).
!DR V=!CONCAT (!viii,'24') /LV=1,6 /C=ident.

*         Comprobación lógica entre estado civil y edad del sujeto:
            edad(variable que hemos de crear a partir del año de
          nacimiento y de la fecha de realización de la encuesta);
            una persona menor de 14 años sólo puede estar soltera
          vedad<= 14 y v00225 es diferente a 1 es un error @=3.
COMPUTE !CONCAT ('@',!viii,'25')=$sysmis.
IF ((1991-1000+!CONCAT(!viii,'16'))<15 AND !CONCAT(!viii,'25')<>1)
 !CONCAT('@',!viii,'25')=3.
*         1,2,3,4 y 5 (v00225).
!DR V=!CONCAT (!viii,'25') /LV=1,2,3,4,5 /C=ident /LVL=!CONCAT (!viii,'16').
*         1,2 y 3 (v00226).
!DR V=!CONCAT (!viii,'26') /LV=1,2,3 /C=ident.
*         01,06,21,28, 29,30,32,49,55,57,67,69,82,90,91,92,93,94 y 95 (v00227).
!DR V=!CONCAT (!viii,'27')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
 81,82,83,84,91,92,93,94,95 /C=ident.
*         Comprobación lógica entre año de la última llegada a España (v00229)
             y  año de nacimiento:
             No puede haber llegado a España en una fecha anterior a la
          de su año de nacimiento, Ojo porque el año de nacimiento está expresado
          con tres dígitos, mientras que el año de la última llegada a España está
          expresado con dos dígitos.
COMPUTE !CONCAT ('@',!viii,'29')=$sysmis.
IF (1900+!CONCAT (!viii,'29')<1000+!CONCAT(!viii, '16'))
 !CONCAT('@',!viii,'29')=3.
*         De 0 a 91 (v00229).
!DR V=!CONCAT (!viii,'29') /LV=0 thru 91 /C=ident /LVL=!CONCAT (!viii, '16').
      *       1 (v00232).

*Edad y relación de parentesco 1.
COMPUTE !CONCAT ('@',!viii,'32')=$sysmis.
*La diferencia de edad entre abuelo o abuela y ego ha de ser mayor de 46 años.
IF ((1000+!CONCAT(!viii,'16')-1000+v00216)<46  AND !CONCAT(!viii,'32')=3)
 !CONCAT('@',!viii,'32')=8.
*La diferencia de edad entre padre o madre e hijo ha de
 ser mayor de 15 años.
IF ((1000+!CONCAT(!viii,'16')-1000+v00216)<15  AND !CONCAT(!viii,'32')=2)
 !CONCAT('@',!viii,'32')=7.
*una persona nacida después de 1945 no puede ser abuelo o
 abuela del sujeto entrevistado error @=6.
IF ((1000+!CONCAT(!viii,'16'))>1951 AND !CONCAT(!viii,'32')=3)
 !CONCAT('@',!viii,'32')=6.
*una persona nacida después de 1960 no puede ser padre o
 madre del sujeto entrevistado error @=5.
IF ((1000+!CONCAT(!viii,'16'))>1966 AND !CONCAT(!viii,'32')=2)
 !CONCAT('@',!viii,'32')=5.
*una persona menor de 46 años no puede ser abuelo o abuela del sujeto
 entrevistado (mayor de 15 años) vedad <=46 y v00332 igual a 3 es un error @=4.
IF ((1991-1000+!CONCAT(!viii,'16'))<46 AND !CONCAT(!viii,'32')=3)
 !CONCAT('@',!viii,'32')=4.
*            Una persona menor de 31 años no puede ser padre o
             madre del sujeto entrevistado (mayor de 15 años)
             vedad<= 31 y v00332 igual a 2 es un error @=3.
IF ((1991-1000+!CONCAT(!viii,'16'))<31 AND !CONCAT(!viii,'32')=2)
 !CONCAT('@',!viii,'32')=3.
!DR V=!CONCAT (!viii,'32') /LV=2,3,4,,5,6,7,8, 9  /C=ident
       /LVL=!CONCAT(!viii,'16') v00216.

COMPUTE !CONCAT ('@',!viii,'33')=$sysmis.
*             Relación de parentesco 1 y 2, no puede darse simultaneamente
          contestación en las variables v00332 y v00333: es un error @=5.
IF (NVALID(!CONCAT (!viii,'32'),!CONCAT (!viii,'33'))=2)
       !CONCAT ('@',!viii,'33')=5.

*            Una persona menor de 16 años no puede estar trabajando en el hogar
               como servicio doméstico: es un error @=4.
IF ((1991-1000+!CONCAT(!viii,'16'))<16 AND !CONCAT(!viii,'33')=7)
 !CONCAT('@',!viii,'33')=4.
*            Una persona menor de 31 años no puede ser padre o
             madre del cónyuge del sujeto entrevistado (mayor de 15 años)
             vedad<= 31 y v00332 igual a 2 es un error @=3.
```

```
IF ((1991-1000+!CONCAT(!viii,'16'))<31 AND !CONCAT(!viii,'33')=2)
       !CONCAT('@',!viii,'33')=3.

!DR V=!CONCAT (!viii,'33') /LV=1,2,3,4,5,6,7,8, 9  /C=ident /LVL= !CONCAT (!viii,'16')
       !CONCAT (!viii,'32').

!DR V=!CONCAT (!viii,'34') /LV=1  /C=ident.


!DR V=!CONCAT (!viii,'36') /LV=2 thru 30  /C=ident.
       !DOEND.

!DO !i=18 !TO 21.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

*Definición de los valores missing en las variables cadena.
MISSING VALUES !CONCAT (!viii,'13') !CONCAT (!viii,'21')
       !CONCAT (!viii,'52') (' ') !CONCAT (!viii,'56') ('99') .

!DR V=!CONCAT (!viii,'11')  /LV=1,2,3 /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV=0,1   /C=ident.
       !DR V=!CONCAT (!viii,'13')  /LV='1','&' /C=ident.


*b)año de nacimiento del padre de ego (v01829) y el año de comienzo
 de convivencia de ego y su padre (v01814): debe haber quince o más años de diferencia.
COMPUTE !CONCAT ('@',!viii,'14')=$sysmis.
IF (1900+!CONCAT (!viii,'14')-(1000+!CONCAT (!viii,'29'))<15)
 !CONCAT('@',!viii,'14')=3.
!DR V=!CONCAT (!viii,'14')  /LV=0 thru 91  /C=ident /LVL=!CONCAT (!viii,'29').
!DR V=!CONCAT (!viii,'17')  /LV=0 thru 80  /C=ident.
!DR V=!CONCAT (!viii,'20')  /LV=0,1   /C=ident.
!DR V=!CONCAT (!viii,'21')  /LV='1','&' /C=ident.

*Año de comienzo de convivencia del padre con ego (v01814)
 y año de cese de la convivencia entre el padre y el ego (v01822),
 El primer año debe de ser anterior al segundo.
COMPUTE !CONCAT ('@',!viii,'22')=$sysmis.
IF (1900+!CONCAT (!viii,'22') < 1900+!CONCAT (!viii,'14'))
 !CONCAT('@',!viii,'22')=3.
!DR V=!CONCAT (!viii,'22')  /LV=0 thru 91 /C=ident
       /LVL=!CONCAT (!viii,'14').

*Edad de comienzo de convivencia ha de ser menor que la edad de fin de
 convivencia.
COMPUTE !CONCAT ('@',!viii,'25')=$sysmis.
IF (!CONCAT (!viii,'17') >!CONCAT (!viii,'25')) !CONCAT('@',!viii,'25')=3.
!DR V=!CONCAT (!viii,'25')  /LV=0 thru 83  /C=ident /LVL =!CONCAT (!viii,'17').
!DR V=!CONCAT (!viii,'28')  /LV=1,6  /C=ident.
!DR V=!CONCAT (!viii,'29')  /LV=800 thru 971 /C=ident.
!DR V=!CONCAT (!viii,'33')  /LV=23 thru 151 /C=ident.
!DR V=!CONCAT (!viii,'37')  /LV=1,2,3,4,5,6,7  /C=ident.
       !DR V=!CONCAT (!viii,'38')  /LV=1,6  /C=ident.

*Relación entre variables: el año de fin de convivencia debe ser menor o igual
 que el año de fallecimiento del padre.
*vs= v01838 (si vive actualmente no puede tener año de fallecimiento).
COMPUTE !CONCAT ('@',!viii,'39')=$sysmis.
IF (1900+!CONCAT (!viii,'22')>1000+!CONCAT (!viii,'39'))
 !CONCAT('@',!viii,'39')=3.
!DR V=!CONCAT (!viii,'39')  /LV=879 thru 991 /C=ident /VS=!CONCAT (!viii,'38')
       /XS=1 /LVL= !CONCAT (!viii,'22').

*Si no vive , vs=6, no puede haber contestado sobre su situación actual.
!DR V=!CONCAT (!viii,'43')  /LV=1,2,3,4,5,6  /C=ident /VS=!CONCAT (!viii,'38')
 /XS=6.
!DR V=!CONCAT (!viii,'44')  /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'43') /XS=1.
!DR V=!CONCAT (!viii,'45')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'38') /XS=6.
!DR V=!CONCAT (!viii,'46')  /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'38')/XS=6.
!DR V=!CONCAT (!viii,'47')  /LV=1,2,3,4  /C=ident /VS=!CONCAT (!viii,'46')
 /XS=1,3.
!DR V=!CONCAT (!viii,'48')  /LV=1,2,3,4,5  /C=ident /VS=!CONCAT (!viii,'46')
 /XS=1.
       !DR V=!CONCAT (!viii,'49')  /LV=1,2,3,4,5,6,7  /C=ident.

!DR V=!CONCAT (!viii,'50')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'49')
       /XS=2,3,4,5,6,7.

!DR V=!CONCAT (!viii,'51')  /LV=1,2,3,4  /C=ident.
```

```
*    Ocupaciones (v01852).
!DR V=!CONCAT (!viii,'52')  /LV='011','012','013','014','021',
 '022','023','024','031','032','033','034','035',
 '036','037','038','039','041','042','049','051','052','053','054','055','056',
 '057','058','059','061','062','071','072','073','074','081','082','083','084',
 '091','101','111','121','122','131','141','151','152','153','154','161','171',
 '181','191','192','193','201','211','212','221','222','231','241','242','243',
 '244','245','251','252','253','254','261','262','263','264','271','272','273',
 '281','282','283','284','285','291','301','311','312','313','314','315',
 '321','331',
 '332','333','334','341','342','343','344','345','346','347','348','351','352',
 '361','362','363','364','365','366','371','372','373','374','375','376','377',
 '378','381','382','383','391','401','411','412','421','431','432','01','02',
 '03','04','05','06','07','08','09','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28',
 '29','30','31','32','33','34','35','36','37','38','39','40','41',
 '42','43','91','92','93','94','95','96','99'   /C=ident
       /VS=!CONCAT (!viii,'51') /XS=3,4.
!DR V=!CONCAT (!viii,'55')  /LV=1,2,3,4,5,6,7,8,9  /C=ident
       /VS=!CONCAT (!viii,'51') /XS=3,4.
!DR V=!CONCAT (!viii,'56')  /LV=1,2,3,4,5,6,7,8,9,10,11,12,13  /C=ident
  /VS=!CONCAT (!viii,'51') /XS=3,4.
      !DOEND.
!DO !i=22 !TO 37.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
      !LET !viii=!CONCAT('v',!iii).
      MISSING VALUES !CONCAT (!viii,'12') (' ') .
!DR V=!CONCAT (!viii,'11')  /LV=1,2,3,4 /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV='1','&' /C=ident.
!DR V=!CONCAT (!viii,'13')  /LV=0 thru 150 /C=ident.
!DR V=!CONCAT (!viii,'16')  /LV=0 thru 150 /C=ident.
!DR V=!CONCAT (!viii,'19')  /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'20')  /LV=00 thru 91 /C=ident /VS=!CONCAT(!viii,'19')
       /XS=1.
*Edad de comienzo de convivencia ha de ser menor que la edad de fin de
 convivencia.
COMPUTE !CONCAT ('@',!viii,'23')=$sysmis.
IF (!CONCAT (!viii,'16') > !CONCAT (!viii,'23')) !CONCAT('@',!viii,'23')=3.
!DR V=!CONCAT (!viii,'23')  /LV=0 thru 150 /C=ident  /VS=!CONCAT(!viii,'19')
       /XS=1 /LVL=!CONCAT (!viii,'16').
!DR V=!CONCAT (!viii,'26')  /LV=0 thru 30 /C=ident.
!DR V=!CONCAT (!viii,'28')  /LV=1,6 /C=ident.
*         a)año de nacimiento del hermano de ego (v02229) y el año de comienzo
          de convivencia de ego y su hermano (v02213), el primer año debe de ser
          anterior al segundo.
COMPUTE !CONCAT ('@',!viii,'29')=$sysmis.
IF (1900+!CONCAT (!viii,'13') < 1000+!CONCAT (!viii,'29'))
 !CONCAT('@',!viii,'29')=3.
!DR V=!CONCAT (!viii,'29')  /LV=872 thru 991 /C=ident /LVL=!CONCAT (!viii,'13').
!DR V=!CONCAT (!viii,'33')  /LV=0 thru 150 /C=ident.
!DR V=!CONCAT (!viii,'37')  /LV=1,6 /C=ident.
*año de nacimiento del hermano de ego (v02229) y el año de fallecimiento
el hermano de ego (v02238), el primer año debe de ser anterior al segundo.
COMPUTE !CONCAT ('@',!viii,'38')=$sysmis.
IF (1000+!CONCAT (!viii,'29') > 1000+!CONCAT (!viii,'38'))
 !CONCAT('@',!viii,'38')=3.
!DR V=!CONCAT (!viii,'38')  /LV=879 thru 991 /C=ident /VS=!CONCAT(!viii,'37')
 /XS=1 /LVL=!CONCAT (!viii,'29').
!DR V=!CONCAT (!viii,'42')  /LV=1,2,3,4,5 /C=ident  /VS=!CONCAT(!viii,'37')
 /XS=6.
      !DOEND.
!DO !i=38 !TO 41.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
```

```
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).

MISSING VALUES !CONCAT (!viii,'49') (' ') .


!DR V=!CONCAT (!viii,'11')  /LV=1,6,9 /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV=0 thru 91 /C=ident.
!DR V=!CONCAT (!viii,'15')  /LV=15 thru 94 /C=ident.
!DR V=!CONCAT (!viii,'18')  /LV=1,2,3,4,5,6  /C=ident.

*Año de nacimiento de la pareja de ego (v03819) y el año de comienzo
 de convivencia de ego y su pareja (v03812), el primer año debe de ser
 15 años mayor que el segundo.
COMPUTE !CONCAT ('@',!viii,'19')=$sysmis.
IF (1900+!CONCAT (!viii,'12')- 1000+!CONCAT (!viii,'19')<15)   !CONCAT('@',!viii,'19')=3.
!DR V=!CONCAT (!viii,'19')  /LV=876 thru 977 /C=ident /LVL=!CONCAT (!viii,'12').
!DR V=!CONCAT (!viii,'23')  /LV=15 thru 125 /C=ident.
!DR V=!CONCAT (!viii,'27')  /LV=1,2,3,4,5,6,7  /C=ident.
!DR V=!CONCAT (!viii,'28')  /LV=1,6 /C=ident.

*año de nacimiento del cónyuge de ego (v03819) y su año de
 fallecimiento (v03829), el primer año debe de ser anterior al segundo .
COMPUTE !CONCAT ('@',!viii,'29')=$sysmis.
IF (1900+!CONCAT (!viii,'29')-1000+!CONCAT (!viii,'19')<15)
 !CONCAT('@',!viii,'29')=3.
!DR V=!CONCAT (!viii,'29')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'28')
       /XS=1 /LVL=!CONCAT (!viii,'19').

!DR V=!CONCAT (!viii,'32')  /LV=1,6 /C=ident /VS=!CONCAT (!viii,'28')
       /XS=1.

!DR V=!CONCAT (!viii,'33')  /LV=1,6 /C=ident /VS=!CONCAT (!viii,'28')
       /XS=6.

!DR V=!CONCAT (!viii,'34')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'33')
       /XS=1.

*Edad de comienzo de convivencia ha de ser menor que la edad de fin de
 convivencia.
COMPUTE !CONCAT ('@',!viii,'37')=$sysmis.
IF (!CONCAT (!viii,'15')> !CONCAT (!viii,'37')) !CONCAT('@',!viii,'37')=3.
!DR V=!CONCAT (!viii,'37')  /LV=15 thru 125 /C=ident /VS=!CONCAT (!viii,'33')
       /XS=1 /LVL=!CONCAT(!viii,'15').

COMPUTE !CONCAT ('@',!viii,'40')=$sysmis.
IF (!CONCAT (!viii,'11') =6 AND NOT(MISSING(!CONCAT (!viii,'40'))) )
 !CONCAT ('@',!viii,'40')=3.
!DR V=!CONCAT (!viii,'40')  /LV=1,2,3,4,5,6,7  /C=ident /VS=!CONCAT (!viii,'33')
       /XS=1 /LVL=!CONCAT (!viii,'11').

!DR V=!CONCAT (!viii,'41')  /LV=1,2,3,4 /C=ident /VS=!CONCAT (!viii,'33')  /XS=1.
!DR V=!CONCAT (!viii,'42')  /LV=0 thru 30 /C=ident.

COMPUTE SEPHIJOS=!CONCAT (!viii,'42')>0 AND (!CONCAT (!viii,'33')=6 OR
 !CONCAT (!viii,'32')=6 ).
*Situación especial: la condición viene de tres variables: haber tenido
hijos y estar separado (este o no viva la pareja).
!DR V=!CONCAT (!viii,'44')  /LV=1,6 /C=ident /VS=SEPHIJOS.
!DR V=!CONCAT (!viii,'45')  /LV=1,2,3,4,5,6,7  /C=ident /VS=!CONCAT (!viii,'44')
       /XS=6.

*Situación especial: la condición viene de dos variables: no haber fallecido y
 que permanezca la unión o haber fallecido.
COMPUTE PARACT=NOT(!CONCAT (!viii,'28')=6 OR (!CONCAT (!viii,'28')=1
       AND !CONCAT (!viii,'33')=6 )).

!DR V=!CONCAT (!viii,'46')  /LV=1,2,3,4,5,6,7  /C=ident /VS=PARACT.

!DR V=!CONCAT (!viii,'47')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'46')
 /XS=2,3,4,5,6,7.
!DR V=!CONCAT (!viii,'48')  /LV=1,2,3,4,5,6,7,8  /C=ident /VS=PARACT.

*Ocupaciones (v03849).
!DR V=!CONCAT (!viii,'49')  /LV='011','012','013','014','021',
 '022','023','024','031','032','033','034','035',
 '036','037','038','039','041','042','049','051','052','053','054','055','056',
 '057','058','059','061','062','071','072','073','074','081','082','083','084',
 '091','101','111','121','122','131','141','151','152','153','154','161','171',
 '181','191','192','193','201','211','212','221','222','231','241','242','243',
 '244','245','251','252','253','254','261','262','263','264','271','272','273',
 '281','282','283','284','285','291','301','311','312','313','314','315',
 '321','331',
 '332','333','334','341','342','343','344','345','346','347','348','351','352',
 '361','362','363','364','365','366','371','372','373','374','375','376','377',
 '378','381','382','383','391','401','411','412','421','431','432','01','02',
 '03','04','05','06','07','08','09','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28',
 '29','30','31','32','33','34','35','36','37','38','39','40','41',
 '42','43','91','92','93','94','95','96','99'   /C=ident
       /VS=!CONCAT (!viii,'48') /XS=4,5,6,7,8.
```

```
!DR V=!CONCAT (!viii,'52')  /LV=1,2,3,4,5,6,7,8,9  /C=ident
       /VS=!CONCAT (!viii,'48') /XS=4,5,6,7,8.

!DR V=!CONCAT (!viii,'53')  /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'54')  /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'55')  /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'56')  /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'57')  /LV=1,2,3,4 /C=ident.

      !DOEND.


!DO !i=42 !TO 57.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).

*Definición de los valores missing en las variables cadena.
MISSING VALUES !CONCAT (!viii,'14') !CONCAT (!viii,'22') (' ') .

!DR V=!CONCAT (!viii,'11')  /LV=0 /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV=0,1,2,3,4 /C=ident.
!DR V=!CONCAT (!viii,'13')  /LV=1,2,3,4 /C=ident.
!DR V=!CONCAT (!viii,'14')  /LV='1','&' /C=ident.
!DR V=!CONCAT (!viii,'15')  /LV=0 thru 91 /C=ident.
!DR V=!CONCAT (!viii,'18')  /LV=0 thru 99 /C=ident.
!DR V=!CONCAT (!viii,'21')  /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'22')  /LV='1','&' /C=ident.
!DR V=!CONCAT (!viii,'23')  /LV=0 thru 91 /C=ident.

*Edad de comienzo de convivencia ha de ser menor que la edad de fin de convivencia.
COMPUTE !CONCAT ('@',!viii,'26')=$sysmis.
IF (!CONCAT (!viii,'18')> !CONCAT (!viii,'26')) !CONCAT('@',!viii,'26')=3.
      !DR V=!CONCAT (!viii,'26')  /LV=0 thru 95 /C=ident /LVL=!CONCAT (!viii,'18').

*vs= v04221 (el hijo nunca convivió con el sujeto) vis= v04229 xs= 1.
!DR V=!CONCAT (!viii,'29')  /LV=02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17
 18,19,20,21,22,23,24,25,26,27,28,29 /C=ident /VS=!CONCAT (!viii,'21') /XS=1.
      !DR V=!CONCAT (!viii,'31')  /LV=1,6 /C=ident.

COMPUTE !CONCAT ('@',!viii,'32')=$sysmis.
*La diferencia de edad entre padre o madre e hijo ha de ser mayor de 15 años.
IF ((1000+v00216-1900+!CONCAT(!viii,'32'))<15) !CONCAT('@',!viii,'32')=4.
*año de nacimiento del hijo de ego (v04232) y el año de comienzo
 de convivencia de ego y su hijo (v04215), el primer año debe de ser
 anterior al segundo.
IF (1900+!CONCAT (!viii,'15') < 1900+!CONCAT (!viii,'32'))
 !CONCAT('@',!viii,'32')=3.
      !DR  V=!CONCAT  (!viii,'32')    /LV=0  thru  91  /C=ident  /LVL=!CONCAT  (!viii,'15')  !CONCAT
(!viii,'16').

!DR V=!CONCAT (!viii,'35')  /LV=0 thru 95 /C=ident.
      !DR V=!CONCAT (!viii,'38')  /LV=1,6 /C=ident.

*b)año de nacimiento del hijo de ego (v04232) y su año de fallecimiento
          (v04239), el primer año debe de ser anterior al segundo.
COMPUTE !CONCAT ('@',!viii,'39')=$sysmis.
IF (1900+!CONCAT (!viii,'32') > 1900+!CONCAT (!viii,'39'))
       !CONCAT('@',!viii,'39')=3.

*vs= v04238 (vive o ha fallecido) vis= v04239  xs= 1.
!DR V=!CONCAT (!viii,'39')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'38')
       /XS=1 /LVL=!CONCAT (!viii,'32').

*c)edad (v04235) y estado civil (v04242)
 Asumimos que una persona menor de 14 años sólo puede estar soltera.
COMPUTE !CONCAT ('@',!viii,'42')=$sysmis.
IF (1991-1900+!CONCAT (!viii,'32')<14 AND !CONCAT (!viii,'42')<>1)
 !CONCAT('@',!viii,'42')=3.
*vs= v04238 (vive o ha fallecido)  vis= v04242, v04243 xs= 6.
!DR V=!CONCAT (!viii,'42')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=6 /LVL=!CONCAT (!viii,'32').
!DR V=!CONCAT (!viii,'43')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=6.
      !DOEND.

!DO !i=58 !TO 58.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
      !LET !viii=!CONCAT('v',!iii).
```

```
MISSING VALUES !CONCAT (!viii,'33') ('99') !CONCAT (!viii,'43') !CONCAT (!viii,'45')
       !CONCAT (!viii,'47')  !CONCAT (!viii,'59') (' ').

!DR V=!CONCAT (!viii,'11')  /LV=0 thru 999  /C=ident.
!DR V=!CONCAT (!viii,'14')  /LV= 1,2,3,4,5,6,7,8,9,10,
 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,
 51,52,60 /C=ident.




!DR V=!CONCAT (!viii,'16')
  /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
 81,82,83,84,91,92,93,94,95  /C=ident.
      !DR V=!CONCAT (!viii,'25')  /LV=1,6 /C=ident.

!DR V=!CONCAT (!viii,'26')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'29')  /LV=0 thru 98 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'32')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'33')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'35')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'37')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'39')  /LV=1 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'41')  /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,
       17,18 /C=ident.

!DR V=!CONCAT (!viii,'43')
 /LV=' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28','29','30',
 '31','32','33','34','35','36','37','38','39','40','41','42','43','44','45',
 '46','47','48','49','50','51','52','53','54','55','56','57','58','59','60',
 '61','62','63','64','65','66','67','68','69','70','71','72','73','74','75',
 '76','77','78','79','80','81','82','83','84','85','86','87','88','89','90',
 '91','92','93','94','95','96','97','98','99','--' /C=ident
 /VS=!CONCAT (!viii,'25')  /XS=1.
!DR V=!CONCAT (!viii,'45')
 /LV=' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28','29','30',
 '31','32','33','34','35','36','37','38','39','40','41','42','43','44','45',
 '46','47','48','49','50','51','52','53','54','55','56','57','58','59','60',
 '61','62','63','64','65','66','67','68','69','70','71','72','73','74','75',
 '76','77','78','79','80','81','82','83','84','85','86','87','88','89','90',
 '91','92','93','94','95','96','97','98','99','--' /C=ident
       /VS=!CONCAT (!viii,'25')  /XS=1.

!DR V=!CONCAT (!viii,'47')
 /LV=' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28','29','30',
 '31','32','33','34','35','36','37','38','39','40','41','42','43','44','45',
 '46','47','48','49','50','51','52','53','54','55','56','57','58','59','60',
 '61','62','63','64','65','66','67','68','69','70','71','72','73','74','75',
 '76','77','78','79','80','81','82','83','84','85','86','87','88','89','90',
 '91','92','93','94','95','96','97','98','99','--' /C=ident
       /VS=!CONCAT (!viii,'25')  /XS=1.

!DR V=!CONCAT (!viii,'49')  /LV=1,2 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'50')  /LV=1,2 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'51')  /LV=1,2,3,4,5 /C=ident .
!DR V=!CONCAT (!viii,'52')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'53')  /LV=1,2,3,4,5 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'54')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'55')  /LV=1,2,3,4 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'56')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'57')  /LV=0 thru 98 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.

!DR V=!CONCAT (!viii,'59')
 /LV=' 1',' 2',' 3',' 4',' 5',' 6',' 7',' 8',' 9','10','11','12','13','14','15',
```

```
 '16','17','18','19','20','21','22','23','24','25','26','27','28','29','30',
 '31','32','33','34','35','36','37','38','39','40','41','42','43','44','45',
 '46','47','48','49','50','51','52','53','54','55','56','57','58','59','60',
 '61','62','63','64','65','66','67','68','69','70','71','72','73','74','75',
 '76','77','78','79','80','81','82','83','84','85','86','87','88','89','90',
 '91','92','93','94','95','96','97','98','99','--' /C=ident
       /VS=!CONCAT (!viii,'25')  /XS=1.

!DR V=!CONCAT (!viii,'61')  /LV=0 thru 95 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
       !DOEND.


!DO !i=59 !TO 59.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'33') ('99') .

!DR V=!CONCAT (!viii,'11')  /LV=0 thru 999  /C=ident.
!DR V=!CONCAT (!viii,'14')  /LV= 1,2,3,4,5,6,7,8,9,10,
 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,
       51,52,60  /C=ident.

!DR V=!CONCAT (!viii,'16')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
 81,82,83,84,91,92,93,94,95   /C=ident.
!DR V=!CONCAT (!viii,'25')  /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'26')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'29')  /LV=0 thru 95 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'32')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'33')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'35')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'37')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'39')  /LV=1 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'41')  /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,
       17,18 /C=ident.

!DR V=!CONCAT (!viii,'43')  /LV=0 thru 95 /C=ident.
       !DR V=!CONCAT (!viii,'45')  /LV=0 thru 95 /C=ident.

       !DOEND.

!DO !i=60 !TO 73.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

MISSING VALUES !CONCAT (!viii,'33') ('99') .
!DR V=!CONCAT (!viii,'11')  /LV=0 thru 999  /C=ident.
!DR V=!CONCAT (!viii,'14')  /LV= 1,2,3,4,5,6,7,8,9,10,
 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,
       51,52,60  /C=ident.

!DR V=!CONCAT (!viii,'16')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
       81,82,83,84,91,92,93,94,95 /C=ident.

!DR V=!CONCAT (!viii,'25')  /LV=1,6 /C=ident.
```

```
!DR V=!CONCAT (!viii,'26')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'29')  /LV=0 thru 95 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'32')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'33')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'35')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'37')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
       /XS=1.

!DR V=!CONCAT (!viii,'39')  /LV=1 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'41')  /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,
       17,18 /C=ident.

!DR V=!CONCAT (!viii,'43')  /LV=0 thru 95 /C=ident.
!DR V=!CONCAT (!viii,'45')  /LV=0 thru 95 /C=ident.
      !DOEND.

!DO !i=74 !TO 81.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'27') ('99') .

!DR V=!CONCAT (!viii,'11')  /LV=01,02,03,04,05,06,07,08,09,
 10,11,12,13,14,15,16,17,18  /C=ident.
       !DR V=!CONCAT (!viii,'13')  /LV=1,6  /C=ident.

       *vs= v07413 (vivienda colectiva) vis= v07414, v07415, v7416, v07417 xs=6.

!DR V=!CONCAT (!viii,'14')  /LV=1,2,3,4,5  /C=ident /VS =!CONCAT (!viii,'13')
 /XS=6.
!DR V=!CONCAT (!viii,'15')  /LV=1,2,3,4,5,9  /C=ident /VS =!CONCAT (!viii,'13')
 /XS=6.
!DR V=!CONCAT (!viii,'16')  /LV=1,2,3,4,5  /C=ident /VS =!CONCAT (!viii,'15')
 /XS=1,2,3,4,5.
!DR V=!CONCAT (!viii,'17')  /LV=1,2,3,4,5,6,7,8  /C=ident
 /VS =!CONCAT (!viii,'15') /XS=9.
       !DR V=!CONCAT (!viii,'18')  /LV=0 thru 99 /C=ident.

COMPUTE !CONCAT ('@',!viii,'21')=$sysmis.
IF (1900+!CONCAT (!viii,'18') > 1900+!CONCAT (!viii,'21'))
 !CONCAT('@',!viii,'21')=3.
       !DR V=!CONCAT (!viii,'21')  /LV=0 thru 91 /C=ident /LVL=!CONCAT (!viii,'18').

!DR V=!CONCAT (!viii,'24')  /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'25')  /LV=1,6 /C=ident /VS =!CONCAT (!viii,'24')
 /XS=1.
!DR V=!CONCAT (!viii,'26')  /LV=1,2,3 /C=ident /VS =!CONCAT (!viii,'24')
       /XS=1.

!DR V=!CONCAT (!viii,'27')  /LV=01,02,10,11,12,13,14,15,16,17,18,
 19,20,21,22,23,30,31,32,33,34,40,41,42,43,44,45,46 /C=ident
       /VS =!CONCAT (!viii,'24') /XS=1.

!DR V=!CONCAT (!viii,'29')  /LV=01,02,10,11,12,13,14,15,16,17,18,
       19,20,21,22,23,30,31,32,33,34,40,41,42,43,44,45,46 /C=ident.

!DR V=!CONCAT (!viii,'31')  /LV=01,02,10,11,12,13,14,15,16,17,18,
       19,20,21,22,23,30,31,32,33,34,40,41,42,43,44,45,46 /C=ident.

       !DR V=!CONCAT (!viii,'33')  /LV=1 /C=ident.

!DR V=!CONCAT (!viii,'34')  /LV=1,2,3,4,5,9  /C=ident.
!DR V=!CONCAT (!viii,'35')  /LV=1,2,3,4,5,9  /C=ident.
!DR V=!CONCAT (!viii,'36')  /LV=1,2,3,4,5,9  /C=ident.
!DR V=!CONCAT (!viii,'37')  /LV=1,2,3,4,5,9  /C=ident.
       !DOEND.

!DO !i=82 !TO 82.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
```

```
       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'21') ('99') .

!DR V=!CONCAT (!viii,'11')  /LV=1,6  /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'13')  /LV=0 thru 91  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'16')  /LV=1,2,3  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'17')  /LV=1,2,3  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.


!DR V=!CONCAT (!viii,'18')  /LV=0 thru 999  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'21')
 /LV= 1,2,3,4,5,6,7,8,9,10,
 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,
        51,52,60  /C=ident /VS=!CONCAT (!viii,'12')  /XS=6.
!DR V=!CONCAT (!viii,'23')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'24')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'25')  /LV=81 thru 91  /C=ident /VS=!CONCAT (!viii,'24')
 /XS=1.
!DR V=!CONCAT (!viii,'28')  /LV=1,2,3  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'29')  /LV=1,2,3  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'30')  /LV=0 thru 999  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'33')  /LV=1 thru 52 /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'35')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'36')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'37')  /LV=81 thru 91  /C=ident /VS=!CONCAT (!viii,'36')
 /XS=1.
!DR V=!CONCAT (!viii,'40')  /LV=1,2,3  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'41')  /LV=1,2,3,4,5  /C=ident.
       !DR V=!CONCAT (!viii,'42')  /LV=1,2,3,4,5,6  /C=ident.

       !DOEND.

!DO !i=83 !TO 83.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

!DR V=!CONCAT (!viii,'11')  /LV=1,6  /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'13')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'14')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'15')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'16')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'17')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=6.
!DR V=!CONCAT (!viii,'18')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'19')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.
!DR V=!CONCAT (!viii,'20')  /LV=1  /C=ident /VS=!CONCAT (!viii,'12')
 /XS=6.

       !DOEND.

!DO !i=84 !TO 84.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
```

```
       !IFEND.

       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'43') ('99') .

!DR V=!CONCAT (!viii,'11') /LV=1,2,3,4,5 /C=ident.
!DR V=!CONCAT (!viii,'16')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
        81,82,83,84,91,92,93,94,95 /C=ident.



!DR V=!CONCAT (!viii,'18') /LV=893 thru 991 /C=ident.
!DR V=!CONCAT (!viii,'22') /LV=0 thru 95 /C=ident.
!DR V=!CONCAT (!viii,'25') /LV=0 thru 40 /C=ident.
!DR V=!CONCAT (!viii,'27') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'28') /LV=1 thru 58 /C=ident /VS=!CONCAT (!viii,'27')
 /XS=1.
!DR V=!CONCAT (!viii,'31') /LV=1,2,3 /C=ident.
!DR V=!CONCAT (!viii,'32') /LV=0,1,2,3,4,5,6,7,8,9 /C=ident
 /VS=!CONCAT (!viii,'31') /XS=1,2.
!DR V=!CONCAT (!viii,'33') /LV=0,1,2,3,4,5,6,7,8,9 /C=ident
        /VS=!CONCAT (!viii,'31') /XS=1.


*El año de inicio de estudios no puede ser posterior al año de finalización.
COMPUTE !CONCAT ('@',!viii,'34')=$sysmis.
IF (1000+!CONCAT (!viii,'18')> 1000+!CONCAT (!viii,'34'))
 !CONCAT('@',!viii,'34')=3.
       !DR V=!CONCAT (!viii,'34') /LV=898 thru 991 /C=ident /LVL=!CONCAT (!viii,'18').

*La edad de inicio de estudios no puede ser superior a la de finalización.
COMPUTE !CONCAT ('@',!viii,'38')=$sysmis.
IF (!CONCAT (!viii,'22')> !CONCAT (!viii,'38')) !CONCAT('@',!viii,'38')=3.
!DR V=!CONCAT (!viii,'38') /LV=0 thru 95 /C=ident /LVL=!CONCAT (!viii,'22').
!DR V=!CONCAT (!viii,'41') /LV=1,2,3,4,5,6 /C=ident.
       !DR V=!CONCAT (!viii,'42') /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'41') /XS=3.

!DR V=!CONCAT (!viii,'43') /LV=10,11,12,20,30,40,50,60,70,80 /C=ident
        /VS=!CONCAT (!viii,'42') /XS=1,2.

!DR V=!CONCAT (!viii,'45') /LV=10,11,12,20,30,40,50,60,70,80 /C=ident
        /VS=!CONCAT (!viii,'42') /XS=1,2.

       !DOEND.

!DO !i=85 !TO 85.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT(!viii,'43') ('99') .

!DR V=!CONCAT (!viii,'12') /LV=1,2,3,4,5,6,7,8 /C=ident.
!DR V=!CONCAT (!viii,'16')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
        81,82,83,84,91,92,93,94,95 /C=ident.

!DR V=!CONCAT (!viii,'18') /LV=900 thru 991 /C=ident.
!DR V=!CONCAT (!viii,'22') /LV=0 thru 95 /C=ident.
!DR V=!CONCAT (!viii,'25') /LV=0 thru 40 /C=ident.
!DR V=!CONCAT (!viii,'27') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'28') /LV=1 thru 41 /C=ident /VS=!CONCAT (!viii,'27')
 /XS=1.
!DR V=!CONCAT (!viii,'31') /LV=1,2,3 /C=ident.
!DR V=!CONCAT (!viii,'32') /LV=0,1,2,3,4,5,6,7,8,9 /C=ident
 /VS=!CONCAT (!viii,'31') /XS=1,2.
!DR V=!CONCAT (!viii,'33') /LV=0,1,2,3,4,5,6,7,8,9 /C=ident
        /VS=!CONCAT (!viii,'31') /XS=1.


*El año de inicio de estudios no puede ser posterior al año de finalización.
COMPUTE !CONCAT ('@',!viii,'34')=$sysmis.
IF (1000+!CONCAT (!viii,'18')> 1000+!CONCAT (!viii,'34'))
 !CONCAT('@',!viii,'34')=3.
       !DR V=!CONCAT (!viii,'34') /LV=900 thru 991 /C=ident /LVL=!CONCAT (!viii,'18').

*La edad de inicio de estudios no puede ser superior a la de finalización.
COMPUTE !CONCAT ('@',!viii,'38')=$sysmis.
IF (!CONCAT (!viii,'22')> !CONCAT (!viii,'38')) !CONCAT('@',!viii,'38')=3.
!DR V=!CONCAT (!viii,'38') /LV=0 thru 95 /C=ident /LVL=!CONCAT (!viii,'22').
```

```
!DR V=!CONCAT (!viii,'41') /LV=1,2,3,4,5,6 /C=ident.
       !DR V=!CONCAT (!viii,'42') /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'41') /XS=3.

!DR V=!CONCAT (!viii,'43') /LV=10,11,12,20,30,40,50,60,70,80 /C=ident
        /VS=!CONCAT (!viii,'42') /XS=1,2.

!DR V=!CONCAT (!viii,'45') /LV=10,11,12,20,30,40,50,60,70,80 /C=ident
        /VS=!CONCAT (!viii,'42') /XS=1,2.

       !DOEND.

*Comprobación de que los años en los que inició los diferentes estudios
 son consecutivos.
IF (v08418<v08518 or v08518<v08618 or v08618<v08718 or v08718<v08818 or
        v08818<v08918 or v08918<v09018) @v09018=3.



*Comprobación de que los años en los que finalizó los diferentes estudios
 son consecutivos.
IF (v08434<v08534 or v08534<v08634 or v08634<v08734 or v08734<v08834 or
        v08834<v08934 or v08934<v09034) @v09034=3.

!DO !i=86 !TO 90.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

!LET !viii=!CONCAT('v',!iii).

MISSING VALUES !CONCAT (!viii,'43') ('99') .

!DR V=!CONCAT (!viii,'13')  /LV=131,199,221,231,241,251,299,301,302,303,
 311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,
 331,341,351,352,361,362,363,364,365,366,367,368,369,370,371,372,373,374,
 375,376,377,378,379,380,381,382,399,411,412,413,414,415,416,421,422,423,
 431,432,441,442,443,451,452,453,461,462,463,464,465,499,511,512,513,514,515,
 516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,541,542,
        543,551,552,553,554,555,561,562,591,599,611,621,629 /C=ident.

!DR V=!CONCAT (!viii,'16')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
 81,82,83,84,91,92,93,94,95 /C=ident.
!DR V=!CONCAT (!viii,'18') /LV=900 thru 991 /C=ident.
!DR V=!CONCAT (!viii,'22') /LV=0 thru 95 /C=ident.
!DR V=!CONCAT (!viii,'25') /LV=0 thru 40 /C=ident.
!DR V=!CONCAT (!viii,'27') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'28') /LV=1 thru 40 /C=ident /VS=!CONCAT (!viii,'27')
 /XS=1.
!DR V=!CONCAT (!viii,'31') /LV=1,2,3 /C=ident.
!DR V=!CONCAT (!viii,'32') /LV=0,1,2,3,4,5,6,7,8,9 /C=ident
 /VS=!CONCAT (!viii,'31') /XS=1,2.
!DR V=!CONCAT (!viii,'33') /LV=0,1,2,3,4,5,6,7,8,9 /C=ident
        /VS=!CONCAT (!viii,'31') /XS=1.

*El año de inicio de estudios no puede ser posterior al año de finalización.
COMPUTE !CONCAT ('@',!viii,'34')=$sysmis.
IF (1000+!CONCAT (!viii,'18')> 1000+!CONCAT (!viii,'34'))
 !CONCAT('@',!viii,'34')=3.
       !DR V=!CONCAT (!viii,'34') /LV=900 thru 991 /C=ident /LVL=!CONCAT (!viii,'18').

*La edad de inicio de estudios no puede ser superior a la de finalización.
COMPUTE !CONCAT ('@',!viii,'38')=$sysmis.
IF (!CONCAT (!viii,'22')> !CONCAT (!viii,'38')) !CONCAT('@',!viii,'38')=3.
!DR V=!CONCAT (!viii,'38') /LV=0 thru 95 /C=ident /LVL=!CONCAT (!viii,'22').
!DR V=!CONCAT (!viii,'41') /LV=1,2,3,4,5,6 /C=ident.
       !DR V=!CONCAT (!viii,'42') /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'41') /XS=3.

!DR V=!CONCAT (!viii,'43') /LV=10,11,12,20,30,40,50,60,70,80 /C=ident
        /VS=!CONCAT (!viii,'42') /XS=1,2.

!DR V=!CONCAT (!viii,'45') /LV=10,11,12,20,30,40,50,60,70,80 /C=ident
 /VS=!CONCAT (!viii,'42') /XS=1,2.
       !DOEND.

!DO !i=91 !TO 98.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).
!DR V=!CONCAT (!viii,'11') /LV=0110,0120,0131,0132,0133,0141,
```

```
 0142,0151,0161,0162,0210,0220,0230,0310,0320,0330,0340,0350,
 0410,0420,0430,0440,1100,1200,1300,
 1400,2100,2200,2300,2400,2500,2600,2700,2800,2900,3000,3100,
 3200,3300,3400,3500,3600,3700,3800,4100,4200,4300,4400,4500,
 4600,4700,4800,4900,5100,5200,5300,5400,5500,5600,5700,6100,
 6200,6300,6400,7101,7102,7103,7104,7105,7106,7107,8101,8102,
        8103,8104,8105,8201,8202,8203,8204,8301,8302,9100,9200,9900   /C=ident.

!DR V=!CONCAT (!viii,'15') /LV=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
 34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,
 59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,
 74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,99  /C=ident.
!DR V=!CONCAT (!viii,'18') /LV=4 thru 95  /C=ident.



*Año de comienzo de curso ha de ser menor que el año de fin de curso.
COMPUTE !CONCAT ('@',!viii,'21')=$sysmis.
IF (1900+!CONCAT (!viii,'15')> 1900+!CONCAT (!viii,'21'))
 !CONCAT('@',!viii,'21')=3.
!DR V=!CONCAT (!viii,'21') /LV=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
 34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,
 59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,
 74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,99
        /C=ident /LVL=!CONCAT (!viii,'15').

*La edad de comienzo de curso ha de ser menor que la edad de fin de convivencia.
COMPUTE !CONCAT ('@',!viii,'24')=$sysmis.
IF (!CONCAT (!viii,'18')> !CONCAT (!viii,'24')) !CONCAT('@',!viii,'24')=3.
       !DR V=!CONCAT (!viii,'24') /LV=4 thru 95  /C=ident /LVL=!CONCAT (!viii,'18').

!DR V=!CONCAT (!viii,'27') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'28') /LV=1,2,3,4,5,6,7  /C=ident.
!DR V=!CONCAT (!viii,'29') /LV=11,12,13,14,19,21,22,29,31,32,39,41,49
 /C=ident.
!DR V=!CONCAT (!viii,'31') /LV=1,2,3 /C=ident /C=ident /VS=!CONCAT (!viii,'35')
 /XS=6.
!DR V=!CONCAT (!viii,'32') /LV=1,2,3,4 /C=ident /VS=!CONCAT (!viii,'35')
 /XS=6.
!DR V=!CONCAT (!viii,'33') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'35') /XS=1.
!DR V=!CONCAT (!viii,'34') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'35') /XS=1.
!DR V=!CONCAT (!viii,'35') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'36') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'37') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
       !DR V=!CONCAT (!viii,'38') /LV=1,2,3,4,5,6,7,8,9  /C=ident.

       !DOEND.

!DO !i=99 !TO 99.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

       !LET !viii=!CONCAT('v',!iii).

!DR V=!CONCAT (!viii,'11') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'12') /LV=1,6 /C=ident /VS= !CONCAT (!viii,'11') /XS=1.
!DR V=!CONCAT (!viii,'13') /LV=1 thru 91 /C=ident /VS=!CONCAT(!viii,'12')/XS=6.
       !DR V=!CONCAT (!viii,'16') /LV=4 thru 95 /C=ident /VS=!CONCAT(!viii,'12')/XS=6.

*Año de comienzo de búsqueda de trabajo ha de ser anterior al año de fin de búsqueda.
COMPUTE !CONCAT ('@',!viii,'19')=$sysmis.
IF (1900+!CONCAT (!viii,'13')> 1900+!CONCAT (!viii,'19'))
 !CONCAT('@',!viii,'19')=3.
!DR V=!CONCAT (!viii,'19') /LV=1 thru 91 /C=ident /VS=!CONCAT(!viii,'12')/XS=6
        /LVL=!CONCAT (!viii,'13').

*Edad de comienzo de búsqueda de trabajo ha de ser menor que la edad de fin de búsqueda.
COMPUTE !CONCAT ('@',!viii,'22')=$sysmis.
IF (!CONCAT (!viii,'16')> !CONCAT (!viii,'22')) !CONCAT('@',!viii,'22')=3.
!DR V=!CONCAT (!viii,'22') /LV=4 thru 95 /C=ident /VS=!CONCAT(!viii,'12')/XS=6
        /LVL=!CONCAT (!viii,'16').

!DR V=!CONCAT (!viii,'25') /LV=1 /C=ident /VS=!CONCAT(!viii,'12')/XS=6.
!DR V=!CONCAT (!viii,'26') /LV=1,2,3,4,5,6,7 /C=ident /VS=!CONCAT(!viii,'12')
 /XS=1.
!DR V=!CONCAT (!viii,'27') /LV=1,2,3,4,5,6,7 /C=ident /VS=!CONCAT(!viii,'12')
 /XS=1.
!DR V=!CONCAT (!viii,'28') /LV=1,2,3,4,5,6,7 /C=ident /VS=!CONCAT(!viii,'12')
 /XS=1.
!DR V=!CONCAT (!viii,'29') /LV=1,2,3,4,5 /C=ident.
!DR V=!CONCAT (!viii,'30') /LV=1,2,3,4,5 /C=ident.
       !DOEND.

*Comprobación de que los años en los que inició el periodo de actividad
 son consecutivos.
```

```
       IF (v10014<v10114 or v10114<v10214 or v10214<v10314) @v10314=3.

!DO !i=100 !TO 103.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

*Definición de valores misising de las variables cadena.
       MISSING VALUES !CONCAT (!viii,'21') !CONCAT (!viii,'24') (' ').


*En esta fase vamos a controlar la información de cada una de las variables
 que integran la ficha, Posteriormente compararemos y depuraremos aquellas
 variables que impliquen a dos fichas.
!DR V=!CONCAT (!viii,'11') /LV=0 thru 99 /C=ident.
!DR V=!CONCAT (!viii,'14') /LV=0 thru 91 /C=ident.
!DR V=!CONCAT (!viii,'17') /LV=5 thru 81 /C=ident.
       !DR V=!CONCAT (!viii,'20') /LV=1 /C=ident.

!DR V=!CONCAT (!viii,'21')  /LV='011','012','013','014','021',
 '022','023','024','031','032','033','034','035',
 '036','037','038','039','041','042','049','051','052','053','054','055','056',
 '057','058','059','061','062','071','072','073','074','081','082','083','084',
 '091','101','111','121','122','131','141','151','152','153','154','161','171',
 '181','191','192','193','201','211','212','221','222','231','241','242','243',
 '244','245','251','252','253','254','261','262','263','264','271','272','273',
 '281','282','283','284','285','291','301','311','312','313','314','315',
 '321','331',
 '332','333','334','341','342','343','344','345','346','347','348','351','352',
 '361','362','363','364','365','366','371','372','373','374','375','376','377',
 '378','381','382','383','391','401','411','412','421','431','432','01','02',
 '03','04','05','06','07','08','09','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28',
 '29','30','31','32','33','34','35','36','37','38','39','40','41',
 '42','43','91','92','93','94','95','96','99'   /C=ident /VS=!CONCAT (!viii,'20')
       /XS=1.

!DR V=!CONCAT (!viii,'24')  /LV='011','012','013','014','021',
 '022','023','024','031','032','033','034','035',
 '036','037','038','039','041','042','049','051','052','053','054','055','056',
 '057','058','059','061','062','071','072','073','074','081','082','083','084',
 '091','101','111','121','122','131','141','151','152','153','154','161','171',
 '181','191','192','193','201','211','212','221','222','231','241','242','243',
 '244','245','251','252','253','254','261','262','263','264','271','272','273',
 '281','282','283','284','285','291','301','311','312','313','314','315',
 '321','331',
 '332','333','334','341','342','343','344','345','346','347','348','351','352',
 '361','362','363','364','365','366','371','372','373','374','375','376','377',
 '378','381','382','383','391','401','411','412','421','431','432','01','02',
 '03','04','05','06','07','08','09','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28',
 '29','30','31','32','33','34','35','36','37','38','39','40','41',
 '42','43','91','92','93','94','95','96','99'   /C=ident /VS=!CONCAT (!viii,'20')
       /XS=1.

!DR V=!CONCAT (!viii,'27') /LV=1,2,3,4,5,6,7,8,9 /C=ident /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'28') /LV=1,2,3,4,5,6,7,8,9 /C=ident /VS=!CONCAT (!viii,'20')
       /XS=1.

!DR V=!CONCAT (!viii,'29') /LV=1,2,3,4,5,6,7,8,9,10,11,12,13 /C=ident /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'31') /LV=1,2,3,4,5,6,7,8,9,10,11,12,13 /C=ident /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'33') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'34') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'35') /LV=1,6 /C=ident /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'36') /LV=1,6 /C=ident  /VS=!CONCAT (!viii,'20')
 /XS=1.
!DR V=!CONCAT (!viii,'37') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'20')
       /XS=1.

       !DR V=!CONCAT (!viii,'38') /LV=1,6 /C=ident.

*Año de comienzo de periodo de actividad ha de ser anterior al de fin de actividad.
COMPUTE !CONCAT ('@',!viii,'39')=$sysmis.
IF (1900+!CONCAT (!viii,'14')> 1900+!CONCAT (!viii,'39'))
 !CONCAT('@',!viii,'39')=3.
!DR V=!CONCAT (!viii,'39') /LV=0 thru 91 /C=ident  /VS=!CONCAT (!viii,'38')
       /XS=1 /LVL=!CONCAT (!viii,'14').

*Edad de comienzo de periodo de actividad ha de ser menor que la edad de fin
 de actividad.
```

```
COMPUTE !CONCAT ('@',!viii,'42')=$sysmis.
IF (!CONCAT (!viii,'17')> !CONCAT (!viii,'42')) !CONCAT('@',!viii,'42')=3.
!DR V=!CONCAT (!viii,'42') /LV=5 thru 92 /C=ident /VS=!CONCAT (!viii,'38')
       /XS=1 /LVL=!CONCAT (!viii,'17').

!DR V=!CONCAT (!viii,'45') /LV=1,2,3,4,5,6,7,8 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=1.
!DR V=!CONCAT (!viii,'46') /LV=1,2,3,4,5,6,7,8 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=1.
!DR V=!CONCAT (!viii,'47') /LV=1,2,3,4,5,6,7,8 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=1.
!DR V=!CONCAT (!viii,'48') /LV=1,2,3,4,5,6,7,8 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=1.
!DR V=!CONCAT (!viii,'49') /LV=1,6 /C=ident /VS=!CONCAT (!viii,'38')
 /XS=1.
!DR V=!CONCAT (!viii,'50') /LV=1,6 /C=ident /VS=!CONCAT (!viii,'49')
 /XS=1.

!DR V=!CONCAT (!viii,'51') /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'50')
 /XS=6.
!DR V=!CONCAT (!viii,'54') /LV=5 thru 68 /C=ident /VS=!CONCAT (!viii,'50')
 /XS=6.
!DR V=!CONCAT (!viii,'57') /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'50')
 /XS=6.
!DR V=!CONCAT (!viii,'60') /LV=5 thru 68 /C=ident /VS=!CONCAT (!viii,'50')
 /XS=6.
!DR V=!CONCAT (!viii,'63') /LV=1 /C=ident /VS=!CONCAT (!viii,'49')
       /XS=1.

       !DOEND.

!DO !i=104 !TO 104.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'23') ('99') .

*Los tipos de depuración en esta segunda fase van a ser los siguientes.
!DR V=!CONCAT (!viii,'11')  /LV=1,2,3  /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV=0 thru 91  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'15')  /LV=0 thru 95  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'18')  /LV=1 thru 99  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'17')  /LV=1,2,3  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'20')  /LV=1 thru 999 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'23') /LV= 1,2,3,4,5,6,7,8,9,10,
 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,
 51,52,60 /C=ident /VS=!CONCAT (!viii,'11')  /XS=1.
!DR V=!CONCAT (!viii,'25')  /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'26')  /LV=1 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'27')  /LV=1 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'29')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
* El año de comienzo del servicio militar ha de ser anterior al de fin.
COMPUTE !CONCAT ('@',!viii,'31')=$sysmis.
IF (1900+!CONCAT (!viii,'29')>1900+!CONCAT(!viii, '31'))
 !CONCAT('@',!viii,'31')=3.
!DR V=!CONCAT (!viii,'31')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'11')
       /XS=1 /LVL=!CONCAT (!viii,'29').

!DR V=!CONCAT (!viii,'33')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
* El año de comienzo de la movilización militar ha de ser anterior al de fin.
COMPUTE !CONCAT ('@',!viii,'35')=$sysmis.
IF (1900+!CONCAT (!viii,'33')>1900+!CONCAT(!viii, '35'))
 !CONCAT('@',!viii,'35')=3.
!DR V=!CONCAT (!viii,'35')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'11')
       /XS=1 /LVL=!CONCAT (!viii,'33').

!DR V=!CONCAT (!viii,'37')  /LV=1,2,3,4,5,6,7,8  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'38')  /LV=1,2,3,4 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
!DR V=!CONCAT (!viii,'39')  /LV=1,2,3,4,5,6,7,8,9  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=1.
       !DOEND.
```

```
*Comprobación de que los años en los que inició el periodo de labores del hogar son consecutivos.
       IF (v10512<v10612 or v10612<v10712 or v10712<v10812) @v10812=3.

!DO !i=105 !TO 105.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).
!DR V=!CONCAT (!viii,'11')  /LV=1,6  /C=ident .
!DR V=!CONCAT (!viii,'12')  /LV=0 thru 91  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=6.

!DR V=!CONCAT (!viii,'15')  /LV=5 thru 95  /C=ident/VS=!CONCAT (!viii,'11')
 /XS=6.
!DR V=!CONCAT (!viii,'18')  /LV=1,6  /C=ident/VS=!CONCAT (!viii,'11')
       /XS=6.

* El año de comienzo del periodo sl ha de ser anterior al de fin.
COMPUTE !CONCAT ('@',!viii,'19')=$sysmis.
IF (1900+!CONCAT (!viii,'12')>1900+!CONCAT(!viii, '19'))
 !CONCAT('@',!viii,'19')=3.
!DR V=!CONCAT (!viii,'19')  /LV=0 thru 91  /C=ident /VS=!CONCAT (!viii,'18')
       /XS=6 /LVL=!CONCAT (!viii,'12').

* La edad de comienzo del periodo sl ha de ser anterior al de fin.
COMPUTE !CONCAT ('@',!viii,'22')=$sysmis.
IF (1900+!CONCAT (!viii,'15')>1900+!CONCAT(!viii, '22'))
 !CONCAT('@',!viii,'22')=3.
!DR V=!CONCAT (!viii,'22')  /LV=5 thru 95 /C=ident
       /VS=!CONCAT (!viii,'18') /XS=6 /LVL=!CONCAT (!viii,'15').

!DR V=!CONCAT (!viii,'25')  /LV=10,11,12,13,14,20,21,22 /C=ident
       /VS=!CONCAT (!viii,'18') /XS=6.

!DR V=!CONCAT (!viii,'27')  /LV=10,11,12,13,14,20,21,22 /C=ident
 /VS=!CONCAT (!viii,'18') /XS=6.
!DR V=!CONCAT (!viii,'29')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'18')
 /XS=6.
!DR V=!CONCAT (!viii,'30')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'18')
 /XS=6.
!DR V=!CONCAT (!viii,'31')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'18')
 /XS=6.
!DR V=!CONCAT (!viii,'32')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'18')
       /XS=6.

      !DR V=!CONCAT (!viii,'33')  /LV=1,2,3,4,5,6,7,8,9 /C=ident.

      !DOEND.

!DO !i=106 !TO 108.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
      !LET !viii=!CONCAT('v',!iii).

!DR V=!CONCAT (!viii,'12')  /LV=0 thru 91  /C=ident /VS=!CONCAT (!viii,'11')
 /XS=6.
!DR V=!CONCAT (!viii,'15')  /LV=5 thru 95  /C=ident/VS=!CONCAT (!viii,'11')
 /XS=6.
!DR V=!CONCAT (!viii,'18')  /LV=1,6  /C=ident/VS=!CONCAT (!viii,'11')
       /XS=6.

* El año de comienzo del periodo sl ha de ser anterior al de fin.
COMPUTE !CONCAT ('@',!viii,'19')=$sysmis.
IF (1900+!CONCAT (!viii,'12')>1900+!CONCAT(!viii, '19'))
 !CONCAT('@',!viii,'19')=3.
!DR V=!CONCAT (!viii,'19')  /LV=0 thru 91  /C=ident /VS=!CONCAT (!viii,'18')
       /XS=6 /LVL=!CONCAT (!viii,'12').

* La edad de comienzo del periodo sl ha de ser anterior al de fin.
COMPUTE !CONCAT ('@',!viii,'22')=$sysmis.
IF (1900+!CONCAT (!viii,'15')>1900+!CONCAT(!viii, '22'))
 !CONCAT('@',!viii,'22')=3.
!DR V=!CONCAT (!viii,'22')  /LV=5 thru 95 /C=ident
       /VS=!CONCAT (!viii,'18') /XS=6 /LVL=!CONCAT (!viii,'15').

!DR V=!CONCAT (!viii,'25')  /LV=10,11,12,13,14,20,21,22 /C=ident
       /VS=!CONCAT (!viii,'18') /XS=6.

!DR V=!CONCAT (!viii,'27')  /LV=10,11,12,13,14,20,21,22 /C=ident
 /VS=!CONCAT (!viii,'18') /XS=6.
!DR V=!CONCAT (!viii,'29')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'18')
 /XS=6.
```

```
!DR V=!CONCAT (!viii,'30')  /LV=1,6  /C=ident /VS=!CONCAT (!viii,'18')
 /XS=6.
!DR V=!CONCAT (!viii,'31')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'18')
 /XS=6.
!DR V=!CONCAT (!viii,'32')  /LV=1,2,3,4,5 /C=ident /VS=!CONCAT (!viii,'18')
        /XS=6.
       !DR V=!CONCAT (!viii,'33')  /LV=1,2,3,4,5,6,7,8,9 /C=ident.
       !DOEND.









!DO !i=109 !TO 109.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

*Definición de los valores missing de las variables cadena.
MISSING VALUES !CONCAT (!viii, '20') (' ').
!DR V=!CONCAT (!viii,'11') /LV=1,2,3,4 /C=ident.
!DR V=!CONCAT (!viii,'12') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'13') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'14') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'15') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'16') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'17') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'18') /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'19') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'20') /LV='011','012','013','014','021',
 '022','023','024','031','032','033','034','035',
 '036','037','038','039','041','042','049','051','052','053','054','055','056',
 '057','058','059','061','062','071','072','073','074','081','082','083','084',
 '091','101','111','121','122','131','141','151','152','153','154','161','171',
 '181','191','192','193','201','211','212','221','222','231','241','242','243',
 '244','245','251','252','253','254','261','262','263','264','271','272','273',
 '281','282','283','284','285','291','301','311','312','313','314','315',
 '321','331',
 '332','333','334','341','342','343','344','345','346','347','348','351','352',
 '361','362','363','364','365','366','371','372','373','374','375','376','377',
 '378','381','382','383','391','401','411','412','421','431','432','01','02',
 '03','04','05','06','07','08','09','10','11','12','13','14','15',
 '16','17','18','19','20','21','22','23','24','25','26','27','28',
 '29','30','31','32','33','34','35','36','37','38','39','40','41',
 '42','43','91','92','93','94','95','96','99'   /C=ident
        /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'23') /LV=1,2,3,4,5,6,7,8,9  /C=ident
       /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'24') /LV=101,102,201,202,203,301,302,303,304,305,306,307,
 308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,
 327,401,501,502,503,504,505,506,601,701,702,703,704,705,801,802,901,1001,1101,
 1102,1201,1202,1203,1204,1205,1206,1207,1208,1209,1210,1211,1301 /C=ident
       /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'28') /LV=1 thru 99999  /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'33') /LV=11,21,22,23,24,25,31,32,33,34,41,42,43,44,45,46,
51,52,53,80,81,82,83,84,85,86,87,88,89,90  /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'35') /LV=1,2,3,4,5 /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'36') /LV=1,2,3,4 /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'37') /LV=1,2,3,4,5,6 /C=ident /VS=!CONCAT (!viii,'11')
 /XS=4.
!DR V=!CONCAT (!viii,'38') /LV=1,6 /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'39') /LV=1,6 /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'40') /LV=0  thru 99 /C=ident
 /VS=!CONCAT (!viii,'11') /XS=4.
!DR V=!CONCAT (!viii,'42') /LV=1,2,3 /C=ident.
!DR V=!CONCAT (!viii,'43') /LV=1,2,3 /C=ident.
!DR V=!CONCAT (!viii,'44') /LV=1,2,3,4,5,8,9,10,11,12,13,14,15 /C=ident.
!DR V=!CONCAT (!viii,'46') /LV=1,2,3,4,5,8,9,10,11,12,13,14,15 /C=ident.
!DR V=!CONCAT (!viii,'48') /LV=1 /C=ident .
```

```
!DR V=!CONCAT (!viii,'49') /LV=2 /C=ident .
!DR V=!CONCAT (!viii,'50') /LV=3 /C=ident .
       !DOEND.

!DO !i=118 !TO 130.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).
*Los tipos de depuración en esta segunda fase van a ser los siguientes
 1) Lista de valores 1 y 6 (v00211).


!DR V=!CONCAT (!viii,'11')  /LV=1,6 /C=ident.
*         900><992 (v00216).
!DR V=!CONCAT (!viii,'16') /LV=901 thru 991 /C=ident.
*         1 y 6 (v00224).
       !DR V=!CONCAT (!viii,'24') /LV=1,6 /C=ident.

*         Comprobación lógica entre estado civil y edad del sujeto:
            edad(variable que hemos de crear a partir del año de
          nacimiento y de la fecha de realización de la encuesta);
            una persona menor de 14 años sólo puede estar soltera
          vedad<= 14 y v00225 es diferente a 1 es un error @=3.
COMPUTE !CONCAT ('@',!viii,'25')=$sysmis.
IF ((1991-1000+!CONCAT(!viii,'16'))<15 AND !CONCAT(!viii,'25')<>1)
 !CONCAT('@',!viii,'25')=3.
*         1,2,3,4 y 5 (v00225).
!DR V=!CONCAT (!viii,'25') /LV=1,2,3,4,5 /C=ident /LVL=!CONCAT (!viii,'16').
*         1,2 y 3 (v00226).
!DR V=!CONCAT (!viii,'26') /LV=1,2,3 /C=ident.
*         01,06,21,28, 29,30,32,49,55,57,67,69,82,90,91,92,93,94 y 95 (v00227).
!DR V=!CONCAT (!viii,'27')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
 81,82,83,84,91,92,93,94,95 /C=ident.
*Comprobación lógica entre año de la última llegada a España (v00229)
             y  año de nacimiento:
             No puede haber llegado a España en una fecha anterior a la
          de su año de nacimiento, Ojo porque el año de nacimiento está expresado
          con tres dígitos, mientras que el año de la última llegada a España está
          expresado con dos dígitos.
COMPUTE !CONCAT ('@',!viii,'29')=$sysmis.
IF (1900+!CONCAT (!viii,'29')<1000+!CONCAT(!viii, '16'))
 !CONCAT('@',!viii,'29')=3.
*         De 0 a 91 (v00229).
!DR V=!CONCAT (!viii,'29') /LV=0 thru 91 /C=ident /LVL=!CONCAT (!viii, '16').
       *       1 (v00232).

*Edad y relación de parentesco 1.
COMPUTE !CONCAT ('@',!viii,'32')=$sysmis.
*La diferencia de edad entre abuelo o abuela y ego ha de ser mayor de 46 años.
IF ((1000+!CONCAT(!viii,'16')-1000+v00216)<46  AND !CONCAT(!viii,'32')=3)
 !CONCAT('@',!viii,'32')=8.
*La diferencia de edad entre padre o madre e hijo ha de ser mayor de 15 años.
IF ((1000+!CONCAT(!viii,'16')-1000+v00216)<15  AND !CONCAT(!viii,'32')=2)
 !CONCAT('@',!viii,'32')=7.

*una persona nacida después de 1945 no puede ser abuelo o
 abuela del sujeto entrevistado error @=6.
IF ((1000+!CONCAT(!viii,'16'))>1945 AND !CONCAT(!viii,'32')=3)
 !CONCAT('@',!viii,'32')=6.
*una persona nacida después de 1960 no puede ser padre o
 madre del sujeto entrevistado error @=5.
IF ((1000+!CONCAT(!viii,'16'))>1960 AND !CONCAT(!viii,'32')=2)
 !CONCAT('@',!viii,'32')=5.
*una persona menor de 46 años no puede ser abuelo o
 abuela del sujeto entrevistado (mayor de 15 años)
 vedad <=46 y v00332 igual a 3 es un error @=4.
IF ((1991-1000+!CONCAT(!viii,'16'))<46 AND !CONCAT(!viii,'32')=3)
 !CONCAT('@',!viii,'32')=4.
*Una persona menor de 31 años no puede ser padre o
 madre del sujeto entrevistado (mayor de 15 años)
 vedad<= 31 y v00332 igual a 2 es un error @=3.
IF ((1991-1000+!CONCAT(!viii,'16'))<31 AND !CONCAT(!viii,'32')=2)
 !CONCAT('@',!viii,'32')=3.
!DR V=!CONCAT (!viii,'32') /LV=2,3,4,,5,6,7,8, 9  /C=ident
       /LVL=!CONCAT(!viii,'16') v00216.

COMPUTE !CONCAT ('@',!viii,'33')=$sysmis.
*Relación de parentesco 1 y 2, no puede darse simultáneamente
 contestación en las variables v00332 y v00333: es un error @=5.
IF (NVALID(!CONCAT (!viii,'32'),!CONCAT (!viii,'33'))=2)
```

```
        !CONCAT ('@',!viii,'33')=5.

*Una persona menor de 16 años no puede estar trabajando en el hogar
 como servicio doméstico: es un error @=4.
IF ((1991-1000+!CONCAT(!viii,'16'))<16 AND !CONCAT(!viii,'33')=7)
 !CONCAT('@',!viii,'33')=4.
*Una persona menor de 31 años no puede ser padre o
 madre del cónyuge del sujeto entrevistado (mayor de 15 años)
vedad<= 31 y v00332 igual a 2 es un error @=3.
IF ((1991-1000+!CONCAT(!viii,'16'))<31 AND !CONCAT(!viii,'33')=2)
        !CONCAT('@',!viii,'33')=3.

       !DR V=!CONCAT (!viii,'33') /LV=1,2,3,4,,5,6,7,8, 9  /C=ident.

!DR V=!CONCAT (!viii,'34') /LV=1  /C=ident.
!DR V=!CONCAT (!viii,'36') /LV=2 thru 30  /C=ident.
       !DOEND.



!DO !i=138 !TO 145.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'12') (' ') .

!DR V=!CONCAT (!viii,'11')  /LV=1,2,3,4 /C=ident.
!DR V=!CONCAT (!viii,'12')  /LV='1','&' /C=ident.
!DR V=!CONCAT (!viii,'13')  /LV=0 thru 150 /C=ident.
!DR V=!CONCAT (!viii,'16')  /LV=0 thru 150 /C=ident.
!DR V=!CONCAT (!viii,'19')  /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'20')  /LV=00 thru 91 /C=ident /VS=!CONCAT(!viii,'19')
        /XS=1.

*Edad de comienzo de convivencia ha de ser menor que la edad de fin de
 convivencia.
COMPUTE !CONCAT ('@',!viii,'23')=$sysmis.
IF (!CONCAT (!viii,'16') > !CONCAT (!viii,'23')) !CONCAT('@',!viii,'23')=3.
!DR V=!CONCAT (!viii,'23')  /LV=0 thru 150 /C=ident  /VS=!CONCAT(!viii,'19')
        /XS=1 /LVL=!CONCAT (!viii,'16').

!DR V=!CONCAT (!viii,'26')  /LV=0 thru 30 /C=ident.
       !DR V=!CONCAT (!viii,'28')  /LV=1,6 /C=ident.

*año de nacimiento del hermano de ego (v02229) y el año de comienzo
 de convivencia de ego y su hermano (v02213), el primer año debe de ser
 anterior al segundo.
COMPUTE !CONCAT ('@',!viii,'29')=$sysmis.
IF (1900+!CONCAT (!viii,'13') < 1000+!CONCAT (!viii,'29'))
 !CONCAT('@',!viii,'29')=3.
       !DR V=!CONCAT (!viii,'29')  /LV=872 thru 991 /C=ident /LVL=!CONCAT (!viii,'13').

!DR V=!CONCAT (!viii,'33')  /LV=0 thru 150 /C=ident.
       !DR V=!CONCAT (!viii,'37')  /LV=1,6 /C=ident.

*año de nacimiento del hermano de ego (v02229) y el año de fallecimiento
 el hermano de ego (v02238), el primer año debe de ser anterior al segundo.
COMPUTE !CONCAT ('@',!viii,'38')=$sysmis.
IF (1000+!CONCAT (!viii,'29') > 1000+!CONCAT (!viii,'38'))
 !CONCAT('@',!viii,'38')=3.
!DR V=!CONCAT (!viii,'38')  /LV=879 thru 991 /C=ident /VS=!CONCAT(!viii,'37') /XS=1
 /LVL=!CONCAT (!viii,'29').
!DR V=!CONCAT (!viii,'42')  /LV=1,2,3,4,5 /C=ident  /VS=!CONCAT(!viii,'37') /XS=6.
       !DOEND.

*Comprobación de que los años en los que se produjo un cambio de
        residencia son consecutivos.

IF (v05826<v05926 or v05926<v06026 or v06026<v06126 or v06126<v06226 or
 v06226<v06326 or v06326<v06426 or v06426<v06526 or v06526<v06626 or
 v06626<v06726 or v06726<v06826 or v06826<v06926 or v06926<v07026 or
 v07026<v07126 or v07126<v07226 or v07226<v07326 or v07326<v17426 or
        v17426<v17526) @v17526=3.

!DO !i=174 !TO 175.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
       !LET !viii=!CONCAT('v',!iii).

MISSING VALUES !CONCAT (!viii,'33') ('99') .
```

```
!DR V=!CONCAT (!viii,'11')  /LV=0 thru 999  /C=ident.
!DR V=!CONCAT (!viii,'14')  /LV= 1,2,3,4,5,6,7,8,9,10,
 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,
        51,52,60  /C=ident.

!DR V=!CONCAT (!viii,'16')
 /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,
 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
 61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
        81,82,83,84,91,92,93,94,95 /C=ident.

!DR V=!CONCAT (!viii,'25')  /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'26')  /LV=0 thru 91 /C=ident /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'29')  /LV=0 thru 95 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'32')  /LV=1,2,3 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.

!DR V=!CONCAT (!viii,'33')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
        /XS=1.

!DR V=!CONCAT (!viii,'35')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
        /XS=1.

!DR V=!CONCAT (!viii,'37')  /LV=01,02,10,11,12,13,14,15,16,17,18,19,20,21,22,
 23,24,25,26,27,28,29,30,31,32,33,34 /C=ident  /VS=!CONCAT (!viii,'25')
        /XS=1.

!DR V=!CONCAT (!viii,'39')  /LV=1 /C=ident  /VS=!CONCAT (!viii,'25')
 /XS=1.
!DR V=!CONCAT (!viii,'41')  /LV=01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,
        17,18 /C=ident.

!DR V=!CONCAT (!viii,'43')  /LV=0 thru 95 /C=ident.
!DR V=!CONCAT (!viii,'45')  /LV=0 thru 95 /C=ident.
       !DOEND.

*Comprobación de que los años en los que comenzó a vivir las distintas viviendas
 sean consecutivos.
IF (v07418<v07518 or v07518<v07618 or v07618<v07718 or v07718<v07818 or
        v07818<v07918 or v08018<v08118 or v08118<v18218 or v18218<v18318) @v18318=3.

*Comprobación de que los años en los que dejo a vivir las distintas viviendas
 sean consecutivos.
IF (v07421<v07521 or v07521<v07621 or v07621<v07721 or v07721<v07821 or
        v07821<v07921 or v08021<v08121 or v08121<v18221 or v18221<v18321) @v18321=3.

!DO !i=182 !TO 183.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
       !IFEND.

       !LET !viii=!CONCAT('v',!iii).

       MISSING VALUES !CONCAT (!viii,'27') ('99') .

!DR V=!CONCAT (!viii,'11')  /LV=01,02,03,04,05,06,07,08,09,
 10,11,12,13,14,15,16,17,18  /C=ident.
       !DR V=!CONCAT (!viii,'13')  /LV=1,6  /C=ident.

*vs= v07413 (vivienda colectiva)
 vis= v07414, v07415, v7416, v07417
        xs= 6.

!DR V=!CONCAT (!viii,'14')  /LV=1,2,3,4,5  /C=ident /VS =!CONCAT (!viii,'13')
 /XS=6.
!DR V=!CONCAT (!viii,'15')  /LV=1,2,3,4,5,9  /C=ident /VS =!CONCAT (!viii,'13')
 /XS=6.
!DR V=!CONCAT (!viii,'16')  /LV=1,2,3,4,5  /C=ident /VS =!CONCAT (!viii,'15')
 /XS=1,2,3,4,5.
!DR V=!CONCAT (!viii,'17')  /LV=1,2,3,4,5,6,7,8  /C=ident
 /VS =!CONCAT (!viii,'15') /XS=9.
       !DR V=!CONCAT (!viii,'18')  /LV=0 thru 99 /C=ident.

COMPUTE !CONCAT ('@',!viii,'21')=$sysmis.
IF (1900+!CONCAT (!viii,'18') > 1900+!CONCAT (!viii,'21'))
 !CONCAT('@',!viii,'21')=3.
       !DR V=!CONCAT (!viii,'21')  /LV=0 thru 91 /C=ident /LVL=!CONCAT (!viii,'18').

!DR V=!CONCAT (!viii,'24')  /LV=1 /C=ident.
!DR V=!CONCAT (!viii,'25')  /LV=1,6 /C=ident /VS =!CONCAT (!viii,'24')
 /XS=1.
!DR V=!CONCAT (!viii,'26')  /LV=1,2,3 /C=ident /VS =!CONCAT (!viii,'24')
        /XS=1.

!DR V=!CONCAT (!viii,'27')  /LV=01,02,10,11,12,13,14,15,16,17,18,
```

```
 19,20,21,22,23,30,31,32,33,34,40,41,42,43,44,45,46 /C=ident
       /VS =!CONCAT (!viii,'24') /XS=1.

!DR V=!CONCAT (!viii,'29')  /LV=01,02,10,11,12,13,14,15,16,17,18,
       19,20,21,22,23,30,31,32,33,34,40,41,42,43,44,45,46 /C=ident.

!DR V=!CONCAT (!viii,'31')  /LV=01,02,10,11,12,13,14,15,16,17,18,
       19,20,21,22,23,30,31,32,33,34,40,41,42,43,44,45,46 /C=ident.

      !DR V=!CONCAT (!viii,'33')  /LV=1 /C=ident.

!DR V=!CONCAT (!viii,'34')  /LV=1,2,3,4,5,9  /C=ident.
!DR V=!CONCAT (!viii,'35')  /LV=1,2,3,4,5,9  /C=ident.
!DR V=!CONCAT (!viii,'36')  /LV=1,2,3,4,5,9  /C=ident.
!DR V=!CONCAT (!viii,'37')  /LV=1,2,3,4,5,9  /C=ident.
      !DOEND.

*Comprobación de que los años en los que inició los diferentes estudios
 no académicos son consecutivos.
IF (v09115<v09215 or v09215<v09315 or v09315<v09415 or v09415<v09515 or
 v09515<v09615 or v09615<v09715 or v09715<v09815 or v09815<v19915 or
       v19915<v20015) @v20015=3.

!DO !i=199 !TO 200.
!IF (!LENGTH(!i) !eq 1) !THEN.
+   !LET !iii=!CONCAT('00',!i).
!ELSE.
+   !IF (!LENGTH(!i) !eq 2) !THEN.
+   !LET !iii=!CONCAT('0',!i).
!ELSE.
+   !LET !iii=!i.
!IFEND.
!IFEND.
!LET !viii=!CONCAT('v',!iii).

!DR V=!CONCAT (!viii,'11') /LV=0110,0120,0131,0132,0133,0141,
 0142,0151,0161,0162,0210,0220,0230,
 0310,0320,0330,0340,0350,0410,0420,0430,0440,
 1100,1200,1300,
 1400,2100,2200,2300,2400,2500,2600,2700,2800,2900,3000,3100,
 3200,3300,3400,3500,3600,3700,3800,4100,4200,4300,4400,4500,
 4600,4700,4800,4900,5100,5200,5300,5400,5500,5600,5700,6100,
 6200,6300,6400,7101,7102,7103,7104,7105,7106,7107,8101,8102,
       8103,8104,8105,8201,8202,8203,8204,8301,8302,9100,9200,9900   /C=ident.

!DR V=!CONCAT (!viii,'15') /LV=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
 34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,
 59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,
 74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,99  /C=ident.
      !DR V=!CONCAT (!viii,'18') /LV=4 thru 95  /C=ident.

*Año de comienzo de curso ha de ser menor que el año de fin de curso.
COMPUTE !CONCAT ('@',!viii,'21')=$sysmis.
IF (1900+!CONCAT (!viii,'15')> 1900+!CONCAT (!viii,'21'))
 !CONCAT('@',!viii,'21')=3.
!DR V=!CONCAT (!viii,'21') /LV=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
 18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,
 34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,
 59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,
 74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,99
       /C=ident /LVL=!CONCAT (!viii,'15').

*La edad de comienzo de curso ha de ser menor que la edad de fin de convivencia.
COMPUTE !CONCAT ('@',!viii,'24')=$sysmis.
IF (!CONCAT (!viii,'18')> !CONCAT (!viii,'24')) !CONCAT('@',!viii,'24')=3.
      !DR V=!CONCAT (!viii,'24') /LV=4 thru 95  /C=ident /LVL=!CONCAT (!viii,'18').

!DR V=!CONCAT (!viii,'27') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'28') /LV=1,2,3,4,5,6,7  /C=ident.
!DR V=!CONCAT (!viii,'29') /LV=11,12,13,14,19,21,22,29,31,32,39,41,49  /C=ident.
!DR V=!CONCAT (!viii,'31') /LV=1,2,3 /C=ident /C=ident /VS=!CONCAT (!viii,'35')
 /XS=6.
!DR V=!CONCAT (!viii,'32') /LV=1,2,3,4 /C=ident /VS=!CONCAT (!viii,'35')
 /XS=6.
!DR V=!CONCAT (!viii,'33') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'35') /XS=1.
!DR V=!CONCAT (!viii,'34') /LV=1,2,3 /C=ident /VS=!CONCAT (!viii,'35') /XS=1.
!DR V=!CONCAT (!viii,'35') /LV=1,6 /C=ident.
!DR V=!CONCAT (!viii,'36') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'37') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
!DR V=!CONCAT (!viii,'38') /LV=1,2,3,4,5,6,7,8,9  /C=ident.
      !DOEND.

!INCIDEN LV=
  v00111 v00113 v00116 v00117 v00119 v00211 v00214 v00216 v00224
  v00225 v00226 v00227 v00229 v00232 v00235 v00236 v00311 v00316
  v00324 v00325 v00326 v00327 v00329 v00332 v00333 v00334 v00336
  v00411 v00416 v00424 v00425 v00426 v00427 v00429 v00432 v00433
  v00434 v00436 v00511 v00516 v00524 v00525 v00526 v00527 v00529
  v00532 v00533 v00534 v00536 v00611 v00616 v00624 v00625 v00626
  v00627 v00629 v00632 v00633 v00634 v00636 v00711 v00716 v00724
  v00725 v00726 v00727 v00729 v00732 v00733 v00734 v00736 v00811
  v00816 v00824 v00825 v00826 v00827 v00829 v00832 v00833 v00834
  v00836 v00911 v00916 v00924 v00925 v00926 v00927 v00929 v00932
```

```
  v00933 v00934 v00936 v01011 v01016 v01024 v01025 v01026 v01027
  v01029 v01032 v01033 v01034 v01036 v01111 v01116 v01124 v01125
  v01126 v01127 v01129 v01132 v01133 v01134 v01136 v01211 v01216
  v01224 v01225 v01226 v01227 v01229 v01232 v01233 v01234 v01236
  v01311 v01316 v01324 v01325 v01326 v01327 v01329 v01332 v01333
  v01334 v01336 v01411 v01416 v01424 v01425 v01426 v01427 v01429
  v01432 v01433 v01434 v01436 v01511 v01516 v01524 v01525 v01526
  v01527 v01529 v01532 v01533 v01534 v01536 v01611 v01616 v01624
  v01625 v01626 v01627 v01629 v01632 v01633 v01634 v01636 v01711
  v01716 v01724 v01725 v01726 v01727 v01729 v01732 v01733 v01734
  v01736 v01811 v01812 v01813 v01814 v01817 v01820 v01821 v01822
  v01825 v01828 v01829 v01833 v01837 v01838 v01839 v01843 v01844
  v01845 v01846 v01847 v01848 v01849 v01850 v01851 v01852 v01855
  v01856 v01911 v01912 v01913 v01914 v01917 v01920 v01921 v01922
  v01925 v01928 v01929 v01933 v01937 v01938 v01939 v01943 v01944
  v01945 v01946 v01947 v01948 v01949 v01950 v01951 v01952 v01955
  v01956 v02011 v02012 v02013 v02014 v02017 v02020 v02021 v02022
  v02025 v02028 v02029 v02033 v02037 v02038 v02039 v02043 v02044
  v02045 v02046 v02047 v02048 v02049 v02050 v02051 v02052 v02055
  v02056 v02111 v02112 v02113 v02114 v02117 v02120 v02121 v02122
  v02125 v02128 v02129 v02133 v02137 v02138 v02139 v02143 v02144
  v02145 v02146 v02147 v02148 v02149 v02150 v02151 v02152 v02155
  v02156 v02211 v02212 v02213 v02216 v02219 v02220 v02223 v02226
  v02228 v02229 v02233 v02237 v02238 v02242 v02311 v02312 v02313
  v02316 v02319 v02320 v02323 v02326 v02328 v02329 v02333 v02337
  v02338 v02342 v02411 v02412 v02413 v02416 v02419 v02420 v02423
  v02426 v02428 v02429 v02433 v02437 v02438 v02442 v02511 v02512
  v02513 v02516 v02519 v02520 v02523 v02526 v02528 v02529 v02533
  v02537 v02538 v02542 v02611 v02612 v02613 v02616 v02619 v02620
  v02623 v02626 v02628 v02629 v02633 v02637 v02638 v02642 v02711
  v02712 v02713 v02716 v02719 v02720 v02723 v02726 v02728 v02729
  v02733 v02737 v02738 v02742 v02811 v02812 v02813 v02816 v02819
  v02820 v02823 v02826 v02828 v02829 v02833 v02837 v02838 v02842
  v02911 v02912 v02913 v02916 v02919 v02920 v02923 v02926 v02928
  v02929 v02933 v02937 v02938 v02942 v03011 v03012 v03013 v03016
  v03019 v03020 v03023 v03026 v03028 v03029 v03033 v03037 v03038
  v03042 v03111 v03112 v03113 v03116 v03119 v03120 v03123 v03126
  v03128 v03129 v03133 v03137 v03138 v03142 v03211 v03212 v03213
  v03216 v03219 v03220 v03223 v03226 v03228 v03229 v03233 v03237
  v03238 v03242 v03311 v03312 v03313 v03316 v03319 v03320 v03323
  v03326 v03328 v03329 v03333 v03337 v03338 v03342 v03411 v03412
  v03413 v03416 v03419 v03420 v03423 v03426 v03428 v03429 v03433
  v03437 v03438 v03442 v03511 v03512 v03513 v03516 v03519 v03520
  v03523 v03526 v03528 v03529 v03533 v03537 v03538 v03542 v03611
  v03612 v03613 v03616 v03619 v03620 v03623 v03626 v03628 v03629
  v03633 v03637 v03638 v03642 v03711 v03712 v03713 v03716 v03719
  v03720 v03723 v03726 v03728 v03729 v03733 v03737 v03738 v03742
  v03811 v03812 v03815 v03818 v03819 v03823 v03827 v03828 v03829
  v03832 v03833 v03834 v03837 v03840 v03841 v03842 v03844
  v03845 v03846 v03847 v03848 v03849 v03852 v03853 v03854
  v03855 v03856 v03857 v03911 v03912 v03915 v03918 v03919 v03923
  v03927 v03928 v03929 v03932 v03933 v03934 v03937 v03940 v03941
  v03942 v03944 v03945 v03946 v03947 v03948 v03949 v03952 v03953
  v03954 v03955 v03956 v03957 v04011 v04012 v04015 v04018 v04019
  v04023 v04027 v04028 v04029 v04032 v04033 v04034 v04037 v04040
  v04041 v04042 v04044 v04045 v04046 v04047 v04048 v04049 v04052
  v04053 v04054 v04055 v04056 v04057 v04111 v04112 v04115 v04118
  v04119 v04123 v04127 v04128 v04129 v04132 v04133 v04134 v04137
  v04140 v04141 v04142 v04144 v04145 v04146 v04147 v04148 v04149
  v04152 v04153 v04154 v04155 v04156 v04157 v04211 v04212 v04213
  v04214 v04215 v04218 v04221 v04222 v04223 v04226 v04229 v04231
  v04232 v04235 v04238 v04239 v04242 v04243 v04311 v04312 v04313
  v04314 v04315 v04318 v04321 v04322 v04323 v04326 v04329 v04331
  v04332 v04335 v04338 v04339 v04342 v04343 v04411 v04412 v04413
  v04414 v04415 v04418 v04421 v04422 v04423 v04426 v04429 v04431
  v04432 v04435 v04438 v04439 v04442 v04443 v04511 v04512 v04513
  v04514 v04515 v04518 v04521 v04522 v04523 v04526 v04529 v04531
  v04532 v04535 v04538 v04539 v04542 v04543 v04611 v04612 v04613
  v04614 v04615 v04618 v04621 v04622 v04623 v04626 v04629 v04631
  v04632 v04635 v04638 v04639 v04642 v04643 v04711 v04712 v04713
  v04714 v04715 v04718 v04721 v04722 v04723 v04726 v04729 v04731
  v04732 v04735 v04738 v04739 v04742 v04743 v04811 v04812 v04813
  v04814 v04815 v04818 v04821 v04822 v04823 v04826 v04829 v04831
  v04832 v04835 v04838 v04839 v04842 v04843 v04911 v04912 v04913
  v04914 v04915 v04918 v04921 v04922 v04923 v04926 v04929 v04931
  v04932 v04935 v04938 v04939 v04942 v04943 v05011 v05012 v05013
  v05014 v05015 v05018 v05021 v05022 v05023 v05026 v05029 v05031
  v05032 v05035 v05038 v05039 v05042 v05043 v05111 v05112 v05113
  v05114 v05115 v05118 v05121 v05122 v05123 v05126 v05129 v05131
  v05132 v05135 v05138 v05139 v05142 v05143 v05211 v05212 v05213
  v05214 v05215 v05218 v05221 v05222 v05223 v05226 v05229 v05231
  v05232 v05235 v05238 v05239 v05242 v05243 v05311 v05312 v05313
  v05314 v05315 v05318 v05321 v05322 v05323 v05326 v05329 v05331
  v05332 v05335 v05338 v05339 v05342 v05343 v05411 v05412 v05413
  v05414 v05415 v05418 v05421 v05422 v05423 v05426 v05429 v05431
  v05432 v05435 v05438 v05439 v05442 v05443 v05511 v05512 v05513
  v05514 v05515 v05518 v05521 v05522 v05523 v05526 v05529 v05531
  v05532 v05535 v05538 v05539 v05542 v05543 v05611 v05612 v05613
  v05614 v05615 v05618 v05621 v05622 v05623 v05626 v05629 v05631
```

```
  v05632 v05635 v05638 v05639 v05642 v05643 v05711 v05712 v05713
  v05714 v05715 v05718 v05721 v05722 v05723 v05726 v05729 v05731
  v05732 v05735 v05738 v05739 v05742 v05743 v05811 v05814 v05816
  v05825 v05826 v05829 v05832 v05833 v05835 v05837 v05839 v05841
  v05843 v05845 v05847 v05849 v05850 v05851 v05852 v05853 v05854
  v05855 v05856 v05857 v05859 v05861 v05911 v05914 v05916 v05925
  v05926 v05929 v05932 v05933 v05935 v05937 v05939 v05941 v05943
  v05945 v06011 v06014 v06016 v06025 v06026 v06029 v06032 v06033
  v06035 v06037 v06039 v06041 v06111 v06114 v06116 v06125 v06126
  v06129 v06132 v06133 v06135 v06137 v06139 v06141 v06211 v06214
  v06216 v06225 v06226 v06229 v06232 v06233 v06235 v06237 v06239
  v06241 v06311 v06314 v06316 v06325 v06326 v06329 v06332 v06333
  v06335 v06337 v06339 v06341 v06411 v06414 v06416 v06425 v06426
  v06429 v06432 v06433 v06435 v06437 v06439 v06441 v06511 v06514
  v06516 v06525 v06526 v06529 v06532 v06533 v06535 v06537 v06539
  v06541 v06611 v06614 v06616 v06625 v06626 v06629 v06632 v06633
  v06635 v06637 v06639 v06641 v06711 v06714 v06716 v06725 v06726
  v06729 v06732 v06733 v06735 v06737 v06739 v06741 v06811 v06814
  v06816 v06825 v06826 v06829 v06832 v06833 v06835 v06837 v06839
  v06841 v06911 v06914 v06916 v06925 v06926 v06929 v06932 v06933
  v06935 v06937 v06939 v06941 v07011 v07014 v07016 v07025 v07026
  v07029 v07032 v07033 v07035 v07037 v07039 v07041 v07111 v07114
  v07116 v07125 v07126 v07129 v07132 v07133 v07135 v07137 v07139
  v07141 v07211 v07214 v07216 v07225 v07226 v07229 v07232 v07233
  v07235 v07237 v07239 v07241 v07311 v07314 v07316 v07325 v07326
  v07329 v07332 v07333 v07335 v07337 v07339 v07341 v07411 v07413
  v07414 v07415 v07416 v07417 v07418 v07421 v07424 v07425 v07426
  v07427 v07429 v07431 v07433 v07434 v07435 v07436 v07437 v07511
  v07513 v07514 v07515 v07516 v07517 v07518 v07521 v07524 v07525
  v07526 v07527 v07529 v07531 v07533 v07534 v07535 v07536 v07537
  v07611 v07613 v07614 v07615 v07616 v07617 v07618 v07621 v07624
  v07625 v07626 v07627 v07629 v07631 v07633 v07634 v07635 v07636
  v07637 v07711 v07713 v07714 v07715 v07716 v07717 v07718 v07721
  v07724 v07725 v07726 v07727 v07729 v07731 v07733 v07734 v07735
  v07736 v07737 v07811 v07813 v07814 v07815 v07816 v07817 v07818
  v07821 v07824 v07825 v07826 v07827 v07829 v07831 v07833 v07834
  v07835 v07836 v07837 v07911 v07913 v07914 v07915 v07916 v07917
  v07918 v07921 v07924 v07925 v07926 v07927 v07929 v07931 v07933
  v07934 v07935 v07936 v07937 v08011 v08013 v08014 v08015 v08016
  v08017 v08018 v08021 v08024 v08025 v08026 v08027 v08029 v08031
  v08033 v08034 v08035 v08036 v08037 v08111 v08113 v08114 v08115
  v08116 v08117 v08118 v08121 v08124 v08125 v08126 v08127 v08129
  v08131 v08133 v08134 v08135 v08136 v08137 v08211 v08212 v08213
  v08216 v08217 v08218 v08221 v08223 v08224 v08225 v08228 v08229
  v08230 v11811 v11816 v11824 v11825 v11826 v11827 v11829 v11832
  v11833 v11834 v11836 v11911 v11916 v11924 v11925 v11926 v11927
  v11929 v11932 v11933 v11934 v11936 v12011 v12016 v12024 v12025
  v12026 v12027 v12029 v12032 v12033 v12034 v12036 v12111 v12116
  v12124 v12125 v12126 v12127 v12129 v12132 v12133 v12134 v12136
  v12211 v12216 v12224 v12225 v12226 v12227 v12229 v12232 v12233
  v12234 v12236 v12311 v12316 v12324 v12325 v12326 v12327 v12329
  v12332 v12333 v12334 v12336 v12411 v12416 v12424 v12425 v12426
  v12427 v12429 v12432 v12433 v12434 v12436 v12511 v12516 v12524
  v12525 v12526 v12527 v12529 v12532 v12533 v12534 v12536 v12611
  v12616 v12624 v12625 v12626 v12627 v12629 v12632 v12633 v12634
  v12636 v12711 v12716 v12724 v12725 v12726 v12727 v12729 v12732
  v12733 v12734 v12736 v12811 v12816 v12824 v12825 v12826 v12827
  v12829 v12832 v12833 v12834 v12836 v12911 v12916 v12924 v12925
  v12926 v12927 v12929 v12932 v12933 v12934 v12936 v13011 v13016
  v13024 v13025 v13026 v13027 v13029 v13032 v13033 v13034 v13036
  v13811 v13812 v13813 v13816 v13819 v13820 v13823 v13826 v13828
  v13829 v13833 v13837 v13838 v13842 v13911 v13912 v13913 v13916
  v13919 v13920 v13923 v13926 v13928 v13929 v13933 v13937 v13938
  v13942 v14011 v14012 v14013 v14016 v14019 v14020 v14023 v14026
  v14028 v14029 v14033 v14037 v14038 v14042 v14111 v14112 v14113
  v14116 v14119 v14120 v14123 v14126 v14128 v14129 v14133 v14137
  v14138 v14142 v14211 v14212 v14213 v14216 v14219 v14220 v14223
  v14226 v14228 v14229 v14233 v14237 v14238 v14242 v14311 v14312
  v14313 v14316 v14319 v14320 v14323 v14326 v14328 v14329 v14333
  v14337 v14338 v14342 v14411 v14412 v14413 v14416 v14419 v14420
  v14423 v14426 v14428 v14429 v14433 v14437 v14438 v14442 v14511
  v14512 v14513 v14516 v14519 v14520 v14523 v14526 v14528 v14529
  v14533 v14537 v14538 v14542 v17526 v17411 v17414 v17416 v17425
  v17426 v17429 v17432 v17433 v17435 v17437 v17439 v17441 v17511
  v17514 v17516 v17525 v17529 v17532 v17533 v17535 v17537 v17539
  v17541 v18318 v18321 v18211 v18213 v18214 v18215 v18216 v18217
  v18218 v18221 v18224 v18225 v18226 v18227 v18229 v18231 v18233
  v18234 v18235 v18236 v18237 v18311 v18313 v18314 v18315 v18316
  v18317 v18324 v18325 v18326 v18327 v18329 v18331 v18333 v18334
  v18335 v18336 v18337 v20015 v19911 v19915 v19918 v19921 v19924
  v19927 v19928 v19929 v19931 v19932 v19933 v19934 v19935 v19936
  v19937 v19938 v20011 v20018 v20021 v20024 v20027 v20028 v20029
  v20031 v20032 v20033 v20034 v20035 v20036 v20037 v20038.

*Guardamos el fichero de datos y las variables arroba creadas.
SAVE OUTFILE ='C:\WINDOWS\Escritorio\esd\sddepur.sav' /COMPRESSED.
!ENDDEFINE.
```

Anexo 4:
SINTAXIS PARA DEPURAR EL CMDBAH

```
PRESERVE.
SET PRINTBACK NONE.
*****************************************************************************.
* DEPURACIÓ DEL CONJUNT MÍNIM BÀSIC DE DADES DE L'ALTA HOSPITALÀRIA          *.
* Creació 08.06.2000 Darrera revisió 04.07.2000                             *.
* 2000 (c) JM. Domenech & A.Bonillo                                         *.
* Email: Abonillo@metodo.uab.es                                             *.
*                                                                           *.
* Crida de la Macro:                                                        *.
*'!CmdbDep ARXIU = Llista de fitxers ASCII d'entrada                        *.
*           HOSP = Llista d'identificadors d'hospital de cada ARXIU         *.
*             DI = Data inicial de periode                                  *.
*             DF = Data final de periode                                    *.
*                                                                           *.
* Exemples de Crides:                                                       *.
*'!Cmdbah ARXIU="5JM9904.TXT" "4JM9904.TXT" /HOSP="H08000265" "H08000226"    *.
*         /DI=1,9,1999 /DF=31,12,1999.                                      *.
*****************************************************************************.
SET PRINTBACK=NONE ERRORS=NONE.
DEFINE !Cmdbah (ARXIU=!CHAREND('/') /HOSP=!CHAREND('/')
                  /DI=!CHAREND('/') /DF=!CHAREND('/') ) .
PRESERVE.
SET ERRORS=NONE.
*Comprovació de paràmetres.
!IF (!ARXIU<>!NULL !AND !HOSP<>!NULL !AND !DI<>!NULL !AND !DF<>!NULL) !THEN

*Comptadors de bucle.
!LET !CB1=!NULL.
!LET !CB2=!NULL.

!DO !I !IN (!ARXIU).
!LET !TARXIU=!CONCAT('D:\INTERNET\ENTRADA\',!UNQUOTE(!I)).
!LET !CB1=!CONCAT(!CB1,' ').

*1ª Part: Lectura de les dades i creació dels codis diagnòstics.
DATA LIST FILE !QUOTE(!TARXIU) NOTABLE
/v1 1-9 (A) v2 10-17 ·v2 10-17 (A) v3 18-22 ·v3 18-22 (A) v4 23 ·v4 23-30 (A)
 v5 31 ·v5 31 (A) v6 32-38 (A) v7 39-39 ·v7 39-39 (A) v8 40 ·v8 40-47 (A)
 v9 48-48 ·v9 48-48 (A) v10 49 ·v10 49-56 (A) v11 57 ·v11 57 (A)
 v12 58-66 (A) #v13 to #v16 67-86 (A) #v17 87-91 (A) #v18 to #v21 92-107 (A)
 v22 108-109 ·v22 108-109 (A) v23 110-113 ·v23 110-113 (A) v24 114 ·v24 114 (A)
 v25 115-118 ·v25 115-118 (A) v26 119 ·v26 119 (A).

*Construcció dels codis diagnòstics.
STRING v13 to v16 (A6).
DO REPEAT v=v13 to v16 /#v=#v13 to #v16.
+DO IF (LENGTH(RTRIM(#v))>3).
+ COMPUTE v=CONCAT( SUBSTR(#v,1,3),'.', SUBSTR(#v,4)).
+ELSE.
+ COMPUTE v=#v.
+END IF.
END REPEAT.

*Construcció dels codi E.
STRING v17 (A6).
+DO IF (LENGTH(RTRIM(#v17))>4).
+ COMPUTE v17=CONCAT( SUBSTR(#v17,1,4),'.', SUBSTR(#v17,5)).
+ELSE.
+ COMPUTE v17=#v17.
+END IF.

*Construcció dels codis de procediments.
STRING v18 to v21 (A6).
DO REPEAT v=v18 to v21 /#v=#v18 to #v21.
+DO IF (LENGTH(RTRIM(#v))>2).
+ COMPUTE v=CONCAT( SUBSTR(#v,1,2),'.', SUBSTR(#v,3)).
+ELSE.
+ COMPUTE v=#v.
+END IF.
END REPEAT.
```

```
*Creació de les variables data.
COMPUTE v4
=DATE.DMY(NUMBER(SUBSTR(·v4,1,2),F2),NUMBER(SUBSTR(·v4,3,2),F2),NUMBER(SUBSTR(·v4,5,4),F4)).
COMPUTE v8
=DATE.DMY(NUMBER(SUBSTR(·v8,1,2),F2),NUMBER(SUBSTR(·v8,3,2),F2),NUMBER(SUBSTR(·v8,5,4),F4)).
COMPUTE
v10=DATE.DMY(NUMBER(SUBSTR(·v10,1,2),F2),NUMBER(SUBSTR(·v10,3,2),F2),NUMBER(SUBSTR(·v10,5,4),F4)).
FORMATS v4 v8 v10 (EDATE).
VARIABLE WIDTH v4 v8 v10 (10).
EXECUTE.


*2ª Part: Comprovació de duplicitats d'identificadors i depuració de variables.
*Duplicitats d'identificadors.
!IDENT V=v2 v10.

*Validació del codi de l'hospital.
!DO !J !IN (!HOSP).
+ !LET !CB2=!CONCAT(!CB2,' ').
+   !IF (!CB1=!CB2) !THEN
+      !DR V=v1 /LV=!J  /C=v2 v10 /MV='000000000','999999999',' '.
+      TITLE !QUOTE(!CONCAT("Validació de les altes hospitalaries: Hospital ",!UNQUOTE(!J))).
+      !LET !CB2=!NULL.
+      !BREAK.
+   !IFEND.
 !DOEND.

*Número d'història clínica .
!DR V=v2 /LV=1 thru HI /C=v2 v10 /F=1 /MV=0,99999999.

*Número d'assistència .
!DR V=v3 /LV= 1 thru HI /C=v2 v10 /F=1 /MV=0,99999.

*Data de naixement (comprovada respecte a la data d'admissió).
!DDF V=V4 /VR=V8 /MIN=0 /MAX=365.25*105 /F=1.

*Sexe.
!DR V=v5 /LV=1,2,3 /C=v2 v10 /MV=0,9 /F=1.

*Codi de residència.
!DRKey V=v6 /PATH='D:\SCSCMBDA\residen.sav'
 /MV='0801900','0801955','0801999','0000000','0800000','1700000','2500000','4300000','5400000',' '
 /C= v2 v10.
*User-missing de codi bé a la taula original o bé en les dades.
IF (UMR=1 OR (SUBSTR(V6,1,2)='55' OR SUBSTR(V6,3,3)='555')) @V6=0.

*Règim econòmic.
!DR V=v7 /LV=1,2,3,5,6,7,8 /MV=0,9 /C=v2 v10 /F=1.

*Data d'admissió (comprovada respecte a la data d'alta, 184 dies).
!DDF V=V8 /VR=V10 /MIN=0 /MAX=184 /C=v2 v10 /F=1.

*Circumstàncies de l'admissió.
!DR V=v9 /LV=1,2,4,5 /MV=3,9 /C=v2 v10 /F=1.

*Data d'alta (correspon al peridode validat).
!DRF V=v10 /FI=!DI /FS=!DF /C=v2 v10 /F=1 .

*Circumstàncies de l'alta.
!DR V=v11 /LV=1,2,3,4,5,6,7 /C=v2 v10 /F=1.

*Codi centre de trasllat.
!DRKey V=v12 /path='D:\SCSCMBDA\HOSPITAL.SAV' /C=v2 v10 /VS=v11 /XS=1,4,5,6,7
 /MV='000000000','999999999',' '.

*Diagnòstic principal i altres diagnòstics.
!DRKey V=v13 v14 v15 v16 /path='D:\SCSCMBDA\CIE9D.sav'
 /C=v2 v10 /RENAME = sexD einfD esupD perD dsecd inspd codD /MV='999.99'.

*User missing de la variable diagnòstic principal.
IF (ANY(v13,'799.9',' ')) @V13=0.
DO REPEAT V=v13 v14 v15 v16 /@V=@v13 @v14 @v15 @v16
         /CompSexe= sexd1 sexd2 sexd3 sexd4 /Inespec= inspd1 inspd2 inspd3 inspd4
         /Edati= einfd1 einfd2 einfd3 einfd4    /Edats= esupd1 esupd2 esupd3 esupd4.
*   Diagnòstic inespecífic.
+   IF (Inespec=1) @v=7.
*   Diagnòstic incongruent amb edat.
+   IF NOT(RANGE((TRUNC(CTIME.DAYS(V8-V4)/365.25)),edati,edats)) @v=5.
*   Diagnòstic incongruent amb sexe.
+   IF (V5=1 AND CompSexe=2 OR V5=2 AND CompSexe=1) @v=6.
END REPEAT.
*Diagnòstic no principal.
IF (DSECD1=1 and not(ANY(@v13,6,5))) @V13=4.

*Codi E .
!DRKey V=v17 /path='D:\SCSCMBDA\CIE9E.sav' /C=v2 v10  /RENAME = inspE .
```

```
*Codi obligatori.
IF (InspE1=1) @v17=7.
*El Codi-E en blanc es vàlid.
IF (@v17=0)  @v17=$SYSMIS.
*Missing als casos amb Codi-E obligatori.
IF ( (codd1=1 or codd2=1 or codd3=1 or codd4=1) AND (V17=' ') ) @V17=0.

*Procediment principal i altres procediments.
!DRKey V=v18 v19 v20 V21 /path='D:\SCSCMBDA\CIE9P.sav' /C=v2 v10
 /RENAME = sexP einfp esupp perP inspP .

DO REPEAT V=v18 v19 v20 V21 /@V=@v18 @v19 @v20 @V21
         /CompSexe= sexP1 sexP2 sexP3 sexP4 /Inespec= inspP1 inspP2 inspP3 inspP4
         /Edati= einfP1 einfP2 einfP3 einfP4 /Edats= esupP1 esupP2 esupP3 esupP4.
*   Passar els blancs a valors vàlids.
+   IF (@V=0) @v=$sysmis.
*   Diagnòstic inespecífic.
+   IF (Inespec=1) @v=7.
*   Diagnòstic incongruent amb edat.
+   IF NOT(RANGE((TRUNC(CTIME.DAYS(V8-V4)/365.25)),edati,edats)) @v=5.
*   Diagnòstic incongruent amb sexe.
+   IF (V5=1 AND CompSexe=2 OR V5=2 AND CompSexe=1) @v=6.
END REPEAT.

*Creació de la variable haver parit: 0=no part, 1=part simple i 2=part múltiple.
COMPUTE PARTS=0.
IF (PERD1=1 OR PERD2=1 OR PERD3=1 OR PERD4=1 OR PERP1=1 OR PERP2=1 OR PERP3=1 OR PERP4=1) PARTS=1.
IF (PERD1=2 OR PERD2=2 OR PERD3=2 OR PERD4=2 OR PERP1=2 OR PERP2=2 OR PERP3=2 OR PERP4=2) PARTS=2.
FORMATS PARTS (F1).

*Temps de gestació.
!DR V=v22 /LV= 23 thru 44 /vs=PARTS /MV=0 /C=v2 v10 /F=1.

*Pes del primer nadó.
!DR V=v23 /LV= 500 thru 5999 /vs=PARTS /MV=0 /C=v2 v10 /F=1.

*Sexe del primer nadó.
!DR V=v24 /LV=1,2,3 /vs=PARTS /vs=PARTS /MV=0 /C=v2 v10 /F=1.

*Pes del segon nadó.
!DR V=v25 /LV= 500 thru 5999 /vs=PARTS /XS=0,1 /MV=0 /C=v2 v10 /F=1.

*Sexe del segon nadó.
!DR V=v26 /LV=1,2,3 /vs=PARTS /XS=0,1 /MV=0 /C=v2 v10 /F=1.

*3ª Part: Estadística per variable i llistat de casos amb errors.
SORT CASES BY V10 V2.
TEMPORARY.
TITLE "Duplicats".
SELECT IF (@ident<>1).
LIST V2 V3 @ident v8 v9 v10 V13 V18.

* Estadística per variable.
TEMPORARY.
RECODE @v1 @v2 @v3 @v4 @v5 @v6 @v7 @v8 @v9 @v10 @V11 @V13 @v14 @v15 @v16 @v18 @v19 @v20 @v21
(SYSMIS=-99).
IF (NOT(ANY(v11,1,4,5,6,7)) AND MISSING(@v12)) @v12=-99.
IF ((codd1=1 or codd2=1 or codd3=1 or codd4=1) AND MISSING(@V17) ) @V17=-99.
DO IF (PARTS=1 OR PARTS=2).
+  IF MISSING(@v22) @v22=-99.
+  IF MISSING(@v23) @v23=-99.
+  IF MISSING(@v24) @v24=-99.
END IF.
DO IF (PARTS=2).
+  IF MISSING(@v25) @v25=-99.
+  IF MISSING(@v26) @v26=-99.
END IF.

Var Lab @v1 "Codi de l'hospital" @v2 "Numero d'història clínica" @v3 "Número d'assistència"
@v4  "Data  de  naixement"  @v5  "Sexe"  @v6  "Codi  de  residència"  @v7  "Règim  econòmic"  @v8  "Data
d'admissió"
@v9 "Circumstàncies admissió" @v10 "Data d'alta" @v11 "Circumstàncies d'alta"
@v12 "Codi del centre trasllat" @v13 "Diagnòstic principal" @v14 "Altres diagnòstics-1"
@v15 "Altres diagnòstics-2" @v16 "Altres diagnòstics-3" @v17 "Codi E"
@v18 "Procediment principal" @v19 "Altres procediments-1"  @v20 "Altres procediments-2"
@v21 "Altres procediments-3"
@v22 "Temps de gestació" @v23 "Pes del primer nadó" @v24 "Sexe del primer nadó"
@v25 "Pes del segon nadó" @v26 "Sexe del segon nadó".

Value labels  @v1 to  @v26 -99 'Correcte' -9 'Error Format' 0 'Missing' 1 'Incorrecte' 2 'Incong.
amb salt'  3 'Error Lògic'
 4 'No principal' 5 'Incong. amb edat' 6 'Incong. amb sexe'  7 'Inespecífic'.
```

```
TABLES
 /FORMAT BLANK MISSING('.')
 /TABLES
 ( @v1 + @v2 + @v3) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT4.1) '%' )  /TITLE  'Identificadors'  /TABLES  ( @v5 + @v4 + @v6) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT7.1) '%' )  /TITLE  'Sociodemogràfiques'  /TABLES  ( @v7 + @v8 + @v10) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT7.1) '%' )  /TITLE  'Administratives'
 /TABLES
 ( @v9 + @v11 + @v12) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT7.1) '%' )   /TITLE   'Dades pacient' /CAPTION "Incong. amb salt: manca el codi de
traslla t a les circumstàncies d'alta"
 /TABLES
 ( @v13+ @v14+@v15+@v16) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT  ((PCT7.1)  '%'  )   /TITLE   'Diagnòstics'  /CAPTION  "No  principal:  relatiu  al  diagnòstic
principal"
 /TABLES
 ( @v18+@v19+@v20+@v21) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT7.1) '%' )  /TITLE  'Procediments'
 /TABLES
 ( @v17) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT7.1) '%' )  /TITLE  'Causes Externes'
 /TABLES
 ( @v22+@v23+@v24+@v25+@v26) BY
 (LABELS) > (STATISTICS)
 /STATISTICS COUNT ((F5.0) 'N' )
 CPCT ((PCT7.1) '%' )   /TITLE   'Perinatals' /CAPTION "Incong. amb salt: manca el diagnòstic o
procediment d'embaraç".

TEMPORARY.
TITLE "Llistat de casos amb errors i/o missing".
SELECT IF (NVALID(@v1 ,@v2,@v3,@v4,@v5,@v7,@v8,@v9,@v10,@v11,@v12,@v13,@v14,@v15,@v16,@v17,
                @v18,@v19,@v20,@v21,@v22,@v23,@v24,@v25,@v26)<>0).
LIST V2 V10 @v1 @v2 @v3 @v4 @v5 @v6 @v7 @v8 @v9 @v10 @v11 @v12 @v13 @v14
           @v15 @v16 @v17 @v18 @v19 @v20 @v21 @v22 @v23 @v24 @v25 @v26.

*4ªpart: grabació de les variables originals corregides i de les variables auxiliars.
*Substitució de la extensió TXT per SAV.
!LET !SAVEFIL=!CONCAT(!SUBSTR(!UNQUOTE(!I),1,!INDEX(!UNQUOTE(!I),.')),'SAV').

SAVE OUTFILE=!QUOTE(!CONCAT("D:\INTERNET\DEPURATS\",'@',!SAVEFIL))
/KEEP=v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 v16 v17 v18 v19 v20 v21 v22 v23 v24 v25
v26
     @v1 @v2 @v3 @v4 @v5 @v6 @v7 @v8 @v9 @v10 @v11 @v12 @v13 @v14 @v15 @v16 @v17
     @v18 @v19 @v20 @v21 @v22 @v23 @v24 @v25 @v26.

*5ª part: correcció dels errors.
TEMPORARY.
DO REPEAT @VN = @v2 @v3 @v4 @v5 @v7 @v9 @v10 @v11 @v22 @v23 @v24 @v25 @v26
          /VN = v2   v3  v4  v5  v7  v9  v10  v11  v22  v23  v24  v25  v26.
IF (@VN<>0) VN=$SYSMIS.
END REPEAT.
DO REPEAT @VC = @v1 @v6 @v12 @v13 @v14 @v15 @v16 @v17 @v18 @v19 @v20 @v21
          /VC =  v1  v6  v12  v13  v14  v15  v16  v17  v18  v19  v20  v21.
IF (@VC<>7 AND @VC<>0) VC=' '.
END REPEAT.
COUNT NI= @v1 @v2 @v3 @v4 @v5 @v6 @v7 @v8 @v9 @v10 @v11 @v12 @v13 @v14
         @v15 @v16 @v17 @v18 @v19 @v20 @v21 @v22 @v23 @v24 @v25 @v26 (-9 thru -1,1 thru HI).
COUNT NM= @v1 @v2 @v3 @v4 @v5 @v6 @v7 @v8 @v9 @v10 @v11 @v12 @v13 @v14
         @v15 @v16 @v17 @v18 @v19 @v20 @v21 @v22 @v23 @v24 @v25 @v26 (0).
FORMATS NI NM (F2).
VARIABLE LABELS NI 'Número d'incidències per cas' NM 'Número de valors missing per cas'.

SAVE OUTFILE=!QUOTE(!CONCAT("D:\INTERNET\DEPURATS\",!SAVEFIL))
/KEEP= v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 v16 v17 v18 v19 v20 v21 v22 v23 v24 v25
v26 ni nm.

!DOEND.
!ELSE.
TITLE "Manquen paràmetres o són incorrectes".
!IFEND.
RESTORE.
```

```
!ENDDEFINE.
RESTORE.



PRESERVE.
SET PRINTBACK NONE.
*************************************************************************.
* AFEGIT DELS FITXERS TRIMESTRALS DEL CMBDAH                            *.
* Creació 10.07.2000 Darrera revisió 26.08.2000                         *.
* 2000 (c) JM. Domenech & A.Bonillo                                     *.
* Email: Abonillo@metodo.uab.es                                         *.
*                                                                       *.
* Crida de la Macro:                                                    *.
*'!Afegir TRIM = Identificador del trimestre                (01,02,03,04) *.
*          ANY = Any depurat                                (2 caracters) *.
*                                                                       *.
* Exemples de Crides:                                                   *.
*'!Afegir TRIM=03 /ANY=00.                                              *.
*'!Afegir TRIM=03 /ANY=99.                                              *.
*************************************************************************.
DEFINE !Afegir ( TRIM=!CHAREND('/') /ANY=!CHAREND('/') ) .
*Llista d'identificadors dels hospitals catalans.
!LET !HOSP ='@1AV @1SM @1QL @1SU @1TR @1VA @2MQ @2SJ @2PT @2JX @2MO @2ME @2VA'
+ ' @3QT @3VC @4SI @4CA @4FI @4JT @4CG @4QG @4SC @4OL @4PA @4SE @5JD'
+ ' @5CR @5BE @5JM @5SB @5CA @5AA @5VI @5DR @5IG @5AP @6MB @6CA @6ES'
+ ' @6TP @6MA @7BE @7GG @7PV @7CM @7JM @7MO @7SC @7MT @7QV @7GC @7TE'
+ ' @7PE @7PT @7GV @7GM @7PU @8QS @8TT @8SP @8DF @8FI @8MA @8PT @8AD'
+ ' @8DX @8SJ @8CL @8CR @8ES @8MF @7AS @8PV @8SR @8MT @8TK @8BA @8GU'
+ ' @8GV @8TV @8HB @8MV'.

*Afegit dels fitxers que NO contenen les variables auxiliars (dades correctes).
GET FILE =!QUOTE(!CONCAT('O:\Internet\Depurats\',!SUBSTR(!HOSP,2,3),!ANY,!TRIM,'.SAV')).
!DO !I !IN (!TAIL(!HOSP)).
+   !LET !FILE=!QUOTE(!CONCAT('O:\Internet\Depurats\',!SUBSTR(!I,2,3),!ANY,!TRIM,'.SAV')).
+   ADD FILES /FILE=* /FILE=!FILE.
!DOEND.

*Creació de l'estada hospitalaria, edat cumpleanys i edat estadística amb decimals.
COMPUTE estada=CTIME.DAYS(V10-V8).
COMPUTE edat_c=CTIME.DAYS(V8-V4)/365.25.
COMPUTE edat = XDATE.YEAR(V8) - XDATE.YEAR(V4).
IF ((XDATE.MONTH(V4)=XDATE.MONTH(V8) AND XDATE.MDAY(V4)> XDATE.MDAY(V8))
 OR  XDATE.MONTH(V4) > XDATE.MONTH(V8) ) edat=edat-1.
FORMATS estada edat (F3) edat_c (F4.1).
VARIABLE LABELS estada 'Estada (dies)' edat 'Edat en anys' edat_c 'Edat decimal'.

*Grabació del fitxer trimestral.
SAVE OUTFILE=!QUOTE(!CONCAT('O:\Internet\Depurats\Dep',!ANY,!TRIM,'.SAV')).

*Afegit dels fitxers que contenen les variables auxiliars (dades errònes).
GET FILE =!QUOTE(!CONCAT('O:\Internet\Depurats\@',!SUBSTR(!HOSP,2,3),!ANY,!TRIM,'.SAV')).
!DO !I !IN (!TAIL(!HOSP)).
+   !LET !FILE=!QUOTE(!CONCAT('O:\Internet\Depurats\',!SUBSTR(!I,2,3),!ANY,!TRIM,'.SAV')).
+   ADD FILES /FILE=* /FILE=!FILE.
!DOEND.

*Creació de l'estada hospitalaria, edat cumpleanys i edat estadística amb decimals.
COMPUTE estada=CTIME.DAYS(V10-V8).
COMPUTE edat_c=CTIME.DAYS(V8-V4)/365.25.
COMPUTE edat = XDATE.YEAR(V8) - XDATE.YEAR(V4).
IF ((XDATE.MONTH(V4)=XDATE.MONTH(V8) AND XDATE.MDAY(V4)> XDATE.MDAY(V8))
 OR  XDATE.MONTH(V4) > XDATE.MONTH(V8) ) edat=edat-1.
FORMATS estada edat (F3) edat_c (F4.1).
VARIABLE LABELS estada 'Estada (dies)' edat 'Edat (anys)' edat_c 'Edat decimal'.

*Grabació del fitxer trimestral.
SAVE OUTFILE=!QUOTE(!CONCAT('O:\Internet\Depurats\@Dep',!ANY,!TRIM,'.SAV')).
!ENDDEFINE.
RESTORE.
```

Anexo 5:

SINTAXIS PARA DEPURAR LA HISTORIA CLÍNICA ELECTRÓNICA

```
!DR V=A8991 /LV= 1,2,3,4,5 /MV=99 /VAR=0 /C=USUACIP VU_DATA /L=0.
!DR V=APCT ASET /LV= 0 THRU 2000 /MV=99 /VAR=0 /C=USUACIP VU_DATA /L=0.
!DR V=A C /LV= 0 THRU 32 /ND=0 /MV=99 /VAR=0 /C=USUACIP VU_DATA /L=0.
!DR V=COLTOT /LV= 50 THRU 2000 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=D /LV= 1 THRU 20 /VAR=0 /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=E /LV= 1,2,3 /MV=99 /VAR=HI /C=USUACIP VU_DATA /L=0.
!DR V=FC /LV= 30 THRU 200 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=GLU /LV= 0 THRU 9000 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=H /LV= 1,2,3  /VAR=0 /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=IMC /LV= 10 THRU 80 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=N P /LV= 1 THRU 20 /VAR=25% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=PES /LV= 15 THRU 160 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=TAD /LV= 20 THRU 300 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=TAS /LV= 50 THRU 400 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=TALLA /LV= 35 THRU 220 /VAR=0 /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=TG /LV= 10 THRU 4000 /VAR=50% /MV=99 /C=USUACIP VU_DATA /L=0.
!DR V=TAD /LV= 1 THRU 20 /VAR=0 /MV=99 /C=USUACIP VU_DATA /L=0.
!QUALITY V=a8991 apct aset a c coltot d e fc glu h imc
          n p pes tad tas talla tg
    /FILE='C:\HClinElecOr.SAV'.
!CORREC V= @a8991 @apct @aset @a @c @coltot @d @e @fc @glu @h
          @imc @n @p @pes @tad @tas @talla @tg
      /C= USUACIP VU_CODV /INDX=VU_DATA .
```