

El recurso informático en el procesamiento de datos cualitativos en ciencias sociales. Su aplicación en un estudio de semiótica cognitiva sobre el concepto de pobreza.

Pablo Gustavo Rodriguez.

Cita:

Pablo Gustavo Rodriguez (1997). *El recurso informático en el procesamiento de datos cualitativos en ciencias sociales. Su aplicación en un estudio de semiótica cognitiva sobre el concepto de pobreza.* *Escenarios*, 2 (3), 7-15.

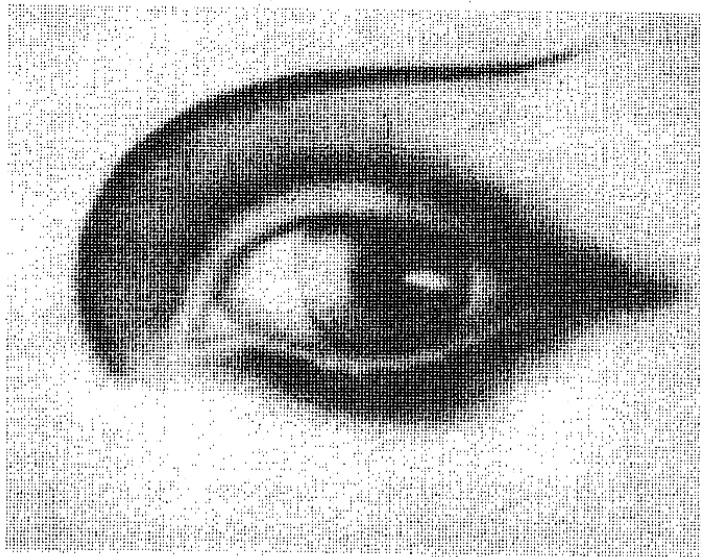
Dirección estable: <https://www.aacademica.org/pablo.gustavo.rodriguez/5>

ARK: <https://n2t.net/ark:/13683/pymh/vmf>



Esta obra está bajo una licencia de Creative Commons.
Para ver una copia de esta licencia, visite
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>.

Acta Académica es un proyecto académico sin fines de lucro enmarcado en la iniciativa de acceso abierto. Acta Académica fue creado para facilitar a investigadores de todo el mundo el compartir su producción académica. Para crear un perfil gratuitamente o acceder a otros trabajos visite: <https://www.aacademica.org>.



EL RECURSO INFORMÁTICO EN EL PROCESAMIENTO DE DATOS CUALITATIVOS EN CIENCIAS SOCIALES

SU APLICACIÓN EN UN ESTUDIO DE SEMIÓTICA COGNITIVA SOBRE EL CONCEPTO DE POBREZA

Lic. P. GUSTAVO RODRIGUEZ¹

Contexto y objetivos de este trabajo

En un trabajo anterior (Rodríguez, 1997) sosteníamos la tesis que las técnicas cualitativas y cuantitativas representan diferentes estrategias de abordaje de la realidad, y que su existencia no se fundamenta en una supuesta diferencia radical en la naturaleza del objeto de estudio de las ciencias sociales y de las ciencias naturales. Que la elección de una u otra estrategia o la combinación de ambas en di-

ferentes grados y de diversas maneras, depende de los objetivos de la investigación y de las preferencias teóricas y paradigmáticas del investigador (Rodríguez., P.G., 1997). Abogábamos por una mayor explotación de las capacidades que ofrecen las computadoras para procesar grandes cantidades de datos cualitativos, liberando así un tiempo precioso de otro modo utilizado en tareas tediosas y que insumen mucho tiempo o personal.

A diferencia de lo que ocurría sólo hace pocos años, es crecien-

te la proporción de hogares con computadoras personales, al mismo tiempo que crecen las tareas a las que éstas pueden aplicarse, tanto en el ámbito doméstico, como comercial, administrativo y científico. Muchos colegas disponen de una PC en su hogar o en su lugar de trabajo, pero a menudo las capacidades que ofrece no son aprovechadas al máximo. El propósito de este trabajo es, primeramente, exponer con cierto detalle, aunque a un nivel elemental, algunas técnicas sen-

cillas que nos permiten utilizar los programas incluidos en el software "integrado" tipo "office" (fundamentalmente procesadores de texto y bases de datos, ya que las planillas de cálculo están diseñadas para realizar operaciones matemáticas y estadísticas) en el procesamiento de nuestros datos cualitativos (básicamente textos provenientes de documentos, entrevistas o encuestas). Y en segundo término, ofrecer una reseña de los tipos disponibles de software diseñado especialmente para este propósito.

Procesadores de texto

Los programas más conocidos y utilizados por los investigadores tal vez sean los procesadores de texto. Los usamos para escribir cartas, ponencias, artículos para las revistas especializadas, notas, memorandums, informes, etc. Sus ventajas por sobre las de la máquina de escribir, en esta tarea, son ampliamente reconocidas:

- La posibilidad de efectuar correcciones, borrando texto ya tipeado sin tener que tachar o volver a escribir toda la página
- La posibilidad de insertar texto sin tener que reservar un interlineado amplio para ello, ni hacer llamadas o dibujar flechas que agregan confusión a la hora de la lectura
- La posibilidad de copiar, cortar y pegar texto cuantas veces sea necesario, sin recurrir a las tijeras, la goma de pegar o la fotocopiadora, y que el resultado tenga siempre la apariencia de un trabajo de calidad final
- El ahorro de papel que todo esto representa, puesto que todas estas correcciones pueden hacerse antes de la impresión
- La posibilidad de dar, en pocos segundos, formatos de pá-

rafo y de letras, definiendo diferentes estilos y tamaños a todo un documento, cualquiera sea su extensión

La calidad de los impresos, que se aproxima a la de una publicación, pudiendo tener tantos originales como se deseen.

Pero otras propiedades de los procesadores de texto son menos conocidas o explotadas incluso por aquellos que los utilizan habitualmente. Por ejemplo, la facilidad para realizar búsquedas textuales: indicando una cadena de caracteres cualquiera, estos programas la encuentran rápidamente en un documento extenso.

Las opciones de búsquedas frecuentemente nos permiten indicar si debe diferenciarse entre mayúsculas y minúsculas y si la cadena indicada debe corresponder exactamente con una palabra completa o puede hallarse como parte de una cadena mayor.

En el primer caso, si solicitamos que halle la cadena de caracteres "POBRE", diferenciando entre mayúsculas y minúsculas y sólo cuando la cadena indicada corresponda a una palabra completa, el programa nos mostrará todos los casos en los que aparece la palabra "POBRE" escrita en mayúsculas y no mostrará los casos de la palabra escritos como "Pobre" o "pobre", como tampoco mostrará las palabras "pobres", "pobreza", "pobrecito", "pobremente", etc., en ninguna combinación de mayúsculas y minúsculas.

También podemos desear reemplazar alguna o todas las apariciones de una palabra o carácter por otra cadena de caracteres o símbolo. En tal caso, también podemos encargar al procesador esta tarea. Podremos indicar si queremos autorizar personalmente cada reemplazo o si permitimos al programa efectuar todos los reempla-

zos sin solicitar nuestra confirmación.

Estas facilidades de búsqueda nos permiten estructurar con nuestro procesador de textos, bases de datos rudimentarias.

Una base de datos consiste en una colección de registros conteniendo información de distinto tipo sobre el mismo tema. Un catálogo es una base de datos, al igual que el fichero de una biblioteca, una agenda telefónica, o el archivo de legajos de una oficina de personal. Tomando el ejemplo de la biblioteca, cada ficha conteniendo los datos de un libro es un *registro*. Cada registro contiene diversos campos. Un *campo* es la pieza más pequeña de información en una base de datos, pues contiene un único dato. En nuestro ejemplo los campos serían: autor, año de publicación, título del artículo o del libro, nombre de la revista donde apareció publicado, lugar de publicación, temas que trata la obra y código de clasificación.

Autor:

Año de publicación:

Título:

En (revista):

Lugar de publicación:

Temas

Código:

Registro de un archivo bibliográfico

También podemos disponer la misma información en forma de tabla, como en los catálogos, con los registros en las filas y los campos en las columnas. La base de datos es la misma. Lo que varía es su forma de presentación.

Autor/Año/Título/En (revista)/Lugar/Tema/Código

Base de datos en forma de tabla

Existen programas especialmente diseñados para administrar estas bases de datos, a los cuales nos referiremos enseguida. Pero una base de datos rudimentaria, como una agenda telefónica o un fichero bibliográfico, puede funcionar bien con un procesador de textos. Desde luego, aquí no podemos definir campos² pero sí registros. Cada carácter de "salto de carro" (CR o "carriage return"), signo que se inserta al pulsar la tecla "enter" o "intro", indica el fin de un registro y el comienzo de otro. Así, para el procesador, cada párrafo sería un registro. Lo más adecuado es ingresar la información correspondiente a los diferentes campos siempre en el mismo orden, y para aquellos datos que puedan ser ambiguos (por ejemplo un apellido puede aparecer como el nombre de una persona y como el nombre de una calle), es conveniente ingresarlos de acuerdo al campo de que se trate, en mayúsculas o precedido de algún carácter ASCII no alfabético para diferenciarlos. Por ejemplo, ingresaremos los nombres en mayúsculas y los nombres de las calles en minúsculas, o bien precedidos por una barra inclinada "/". De esta manera, al buscar GONZALEZ el programa sólo nos mostrará a las personas de apellido González y no a las que viven en la calle González. Algún artificio semejante deberíamos usar para no confundir los números telefónicos con los correspondientes a los domicilios.

En nuestro caso disponemos de un archivo bibliográfico realizado con Word desde el que importamos los registros necesarios hacia un documento en elaboración, como a este mismo artículo. Ingresamos los apellidos de los

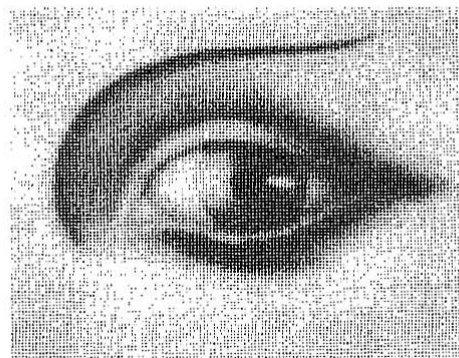
autores siempre en mayúsculas, luego el año de publicación entre paréntesis, el título del libro, los títulos de publicaciones entre comillas dobles, el lugar de edición y algunas palabras en mayúsculas que clasifican el registro por los temas que trata.

Por ejemplo: ECO, Humberto (1992): Los límites de la interpretación. Ed. Lumen, Barcelona. SEMIOL ESTRUCTURA LINGUIS

Así es posible realizar búsquedas textuales como si fueran campos. Por ejemplo, podemos buscar todos los libros publicados en el año 1996 buscando la cadena "(1996)", o todos los libros de Eco buscando la cadena "ECO" como palabra completa y discriminando mayúsculas de minúsculas, o pedir todas las obras sobre el tema "semiología" realizando una búsqueda de la cadena "SEMIOL". Si también queremos hallar las que llevan en su título la palabra semiología o semiótica realizamos una búsqueda de la cadena "semio" sin indicar palabra completa ni coincidencia de mayúsculas y minúsculas. Una vez hallados los registros buscados, se pueden copiar y pegar en otro documento.

Si trabajamos con el viejo pero todavía vigente DOS y guardamos la base como un archivo tipo ASCII, es decir, sin formatos, o sólo texto, estas búsquedas podremos realizarlas sin abrir el documento con el comando "find", y enviar los resultados de la búsqueda a un archivo que indiquemos expresamente, a la pantalla o a la impresora.

Por ejemplo, el comando: find "GONZALEZ" biblio.txt busca en el archivo "biblio.txt" todos los registros (bibliográficos, en nuestro caso) de obras en las que el autor es de apellido González y nos los muestra completos en la



pantalla. El comando: find "GONZALEZ" biblio.txt > PRN realiza la misma búsqueda pero enviando el output a la impresora en línea. Y el comando: find "GONZALEZ" biblio.txt > salida.txt realiza lo propio, pero graba el resultado de la búsqueda en el archivo de texto "salida.txt", al que posteriormente podremos editar, dándole los formatos que deseemos.

Los registros pueden ser ingresados a la base en cualquier orden y posteriormente ser ordenados alfabéticamente en forma automática por el procesador. Mantener la base ordenada alfabéticamente hace las búsquedas más veloces. Desde luego un procesador sólo nos permite un orden para todos los registros: por la primera palabra de cada registro. El orden puede ser alfabético o numérico y ascendente o descendente, pero el "campo" ordenador será el primero que aparece en el registro. Esto no afecta las búsquedas pero sí la presentación de los resultados: cualquiera sea el dato buscado (nombre, año, tema, etc.) se nos presentarán los registros ordenados por el nombre del autor si éste es como en nuestro ejemplo el primer dato de cada registro.

Las bases de datos

Ya hemos explicado en qué consiste una base de datos (ver

pág. 1). Los programas administradores de bases de datos, tales como Dbase, Excel, entre otros, en sus diferentes versiones para distintos sistemas operativos y entornos de trabajo, operan bajo los mismos principios básicos. Son instrumentos que permiten manejar la información almacenada en la base de datos. El manejo de la información incluye:

- Agregar, eliminar y modificar la información de la base
- Obtener la información ordenada en forma compleja de acuerdo a determinados parámetros suministrados por el usuario
- Efectuar cálculos con los datos y obtener estadísticas de los mismos (cantidad de registros totales o de los que satisfacen ciertas condiciones)
- Importar información desde otras bases y exportarla hacia otras bases, programas o archivos, o imprimirla.

El usuario debe definir previamente una estructura para su base de datos, decidiendo de cuántos y cuáles campos consistirá, y definiendo las propiedades de cada uno (básicamente, tipo de campo y longitud). Los tipos de campo considerados son habitualmente cinco:

* *Campos alfanuméricos*: almacenan una cantidad grande pero limitada de caracteres numéricos o alfabéticos, algunos símbolos especiales o espacios en blanco. Al definir un campo alfanumérico debemos otorgarle una extensión y el programa administrador reservará ese espacio aún cuando sólo se use parcialmente o no se use en absoluto. Si le damos una extensión muy grande estaremos desperdiciando espacio en el disco y si le damos una extensión reducida puede ser que, en un futuro, algún dato de longitud mayor que la prevista no pueda

ser ingresado. Algunos datos pueden abreviarse, pero otros no. Aún cuando siempre es posible modificar la estructura de una base de datos con un administrador, una vez que ya hemos ingresado datos no es recomendable hacerlo, puesto que entraña el riesgo de perder irrecuperablemente toda la información almacenada en el o los campos modificados. Por tal motivo, es muy importante diseñar cuidadosamente la estructura de la base haciendo todas las previsiones posibles, a fin de no tener que efectuar retoques una vez que han sido ingresado los datos.

* *Campo de fecha*: se usan para guardar fechas exclusivamente, siempre y cuando necesitemos calcular el tiempo transcurrido entre dos fechas dadas. De lo contrario, puede utilizarse para el mismo fin un campo alfanumérico.

* *Campo numérico*: se usan cuando con las cifras allí almacenadas deben realizarse operaciones matemáticas. Entre sus propiedades debemos definir su extensión y cantidad de decimales.

* *Campo lógico*: es un campo consistente en un sólo carácter representando un valor verdadero o falso para determinada condición. Pueden usarse para variables binarias, alternativas dicotómicas excluyentes, como Hombre/Mujer, Presente/Ausente, Sí/No, etc.

* *Campo memo*: es un campo de caracteres alfanuméricos de formato libre y extensión indefinida que permite almacenar cualquier texto que uno desee. Se usan para incluir observaciones, comentarios, notas, etc. Son particularmente útiles en el procesamiento informático de entrevistas, documentos y cuestionarios con preguntas abiertas. El contenido de un campo memo queda almacenado en un archivo, pero dicho

campo no puede ser usado para clasificar ni para indexar la base de datos. En nuestra investigación basada en semiótica de enunciados habíamos diseñado una base de datos consistente, entre otros, en un campo memo para almacenar las definiciones contextuales de los lexemas que interesan a nuestro estudio. Pero posteriormente la abandonamos al comparar su eficacia con la del un programa especialmente diseñado para ADC, como el NUD•IST.

Para otros campos con respuestas predeterminadas entre un número limitado de alternativas algunos administradores de bases de datos permiten definir la lista de alternativas excluyentes para poder seleccionar las opciones mediante un simple click del ratón desde un menú con botones de radio o tildando checkboxes en el caso de opciones no excluyentes. Esto es útil cuando se quieren procesar las respuestas a un cuestionario: para cada pregunta definimos un campo; cada cuestionario constituirá un registro. A las preguntas con respuestas dicotómicas se les pueden asignar campos lógicos, las preguntas con tres o más opciones se definen con campos alfanuméricos, y si las opciones están predeterminadas (cuestionario cerrado) se construyen las listas correspondiente. Para las preguntas abiertas se establecen campos de tipo memo.

Luego deben ingresarse los datos, y a partir de entonces podemos realizar búsquedas simples o complejas en uno o varios campos, y hasta en una o varias bases de datos relacionadas, valiéndonos de los operadores booleanos and, or, not, de los operadores relacionales = (igual a), < (menor que), > (mayor que), >= (mayor o igual que), <= (menor o igual

que), <> (distinto de), "incluido" y "like", y también de los comodines "*" y "?". Es común que estos programas incorporen un lenguaje mediante el cual podamos confeccionar verdaderos programas para administrar nuestros datos. Estos lenguajes incluyen diversos comandos cuya explicación detallada trasciende el carácter elemental de este trabajo, pero todos ellos pueden ser ejecutados también a través de menús o de una interfase gráfica mediante el uso del ratón. Estos programas superan en mucho a las rudimentarias capacidades de búsqueda de los procesadores de texto, puesto que realizan búsquedas lógicas.

Los resultados pueden guardarse en una tabla permanentemente o puede archivarse la estructura de la consulta para volver a realizarla posteriormente con nuevos datos. Una vez hallados los registros que satisfacen las condiciones de búsqueda indicadas, podemos imprimir informes de esas búsquedas indicando cuántos y cuáles campos de esos registros queremos incluir en el informe.

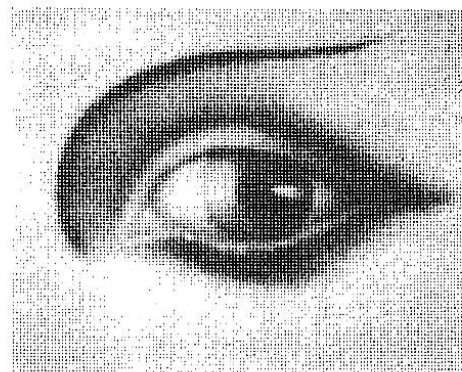
Las limitaciones más importantes que presentan las bases de datos en la investigación cualitativa son:

- Su rigidez. Una vez definida la estructura, resulta inconveniente modificarla debido al riesgo de pérdida irremediable de datos. Pero debido a las características recursivas de la investigación cualitativa, muchas veces no es posible definir a priori todos los campos que serán necesarios, o todas las alternativas posibles, o la extensión más apropiada de los campos alfanuméricos, un campo que creímos lógico puede llegar a requerir la posibilidad de una tercera alternativa. Es común en la investigación cualitativa modificar nuestro modelo inicial o bien que

el modelo se vaya construyendo durante el transcurso de la investigación (Vasilachis de Gialdino, 1992; Glaser y Strauss, 1967; ver también Rodríguez, P. G., 1997), o que el estudio intente recuperar el modelo nativo y por lo tanto ignoramos todas las categorías de análisis que utilizaremos.

- Hay que volver a tipear los textos al ingresarlos a la base. Al trabajar con entrevistas desgrabadas y fichar porciones de las mismas, debemos retipar los textos para ingresarlos a la base de datos o bien, copiarlos y pegarlos párrafo a párrafo, con lo cual si tenemos un gran número de entrevistas de gran extensión estaremos, por lo menos, duplicando el trabajo de introducir los datos.

- No existe una manera de ingresar masivamente en una misma operación, sobre un conjunto de registros, un valor dado para un campo que sea igual para todos ellos. Por ejemplo, la base de datos que elaboramos para nuestra investigación debía considerar cada utilización del término "pobreza" por una persona en una entrevista, en un registro diferente. Dado que el término es utilizado varias veces a lo largo de una entrevista, podemos tener más de un centenar de definiciones contextuales (utilizaciones efectivas) del lexema para una misma entrevista. Y si es nuestro deseo cruzar las definiciones con variables demográficas como "Religión", "sexo", "condición social", etc., a fin de comparar, por ejemplo, cómo definen la pobreza los católicos y cómo los pentecostales, deberemos ingresar en cada uno de los registros la religión de su enunciador, su sexo, condición social, etc. Esto significa que para una entrevista desgrabada en la que el lexema pobreza ocurre 173



veces. Deberemos copiar y pegar una por una las 173 definiciones en sendos registros de la base de datos y anotar manualmente 173 veces que su enunciador es católico, de sexo masculino, no pobre... Si bien es una tarea que se realiza sólo una vez, puede tomarnos horas indexar todos los enunciados relevantes de una entrevista en todas las categorías (temas o campos) en los que corresponde, insumiendo un tiempo precioso que puede ser destinado a la interpretación de los resultados.

- No es posible establecer una estructura jerárquica de los campos. Si, por ejemplo, establezco la variable "Religión" como un campo y los valores posibles son: católico, protestante, judío, islámico y ateo, no puedo subdividir a "protestante" en "pentecostal", "bautista", "luterano" y luego subdividir a "pentecostal" en "de las Asambleas de Dios", "de otras agrupaciones", etc. Existen artificios para trabajar con esta información por fuera del programa, pero el administrador considera a todas las categorías (campos) al mismo nivel jerárquico.

Los programas de Análisis de datos cualitativos (ADC)

Debido a que nuestro conocimiento personal de los programas especialmente diseñados para ADC se limita al NUD•IST, el cual comentaremos más adelante, nos

serviremos del estudio comparativo de Richards y Richards (1992) para dar un panorama más amplio, aunque no exhaustivo, de la oferta existente de software especializado y en qué medida los programas se adecuan a las necesidades de la investigación.

Richards y Richards distinguen cinco tipos de software especializado para realizar ADC, según el tipo de análisis que permiten efectuar. La mayoría de ellos han sido desarrollados en la década del ochenta:

1. *Sistemas de codificación y recuperación* (Ejemplo: *el Ethnograph*). Fue el primer tipo de software desarrollado para ADC, creado por científicos sociales que intentaban replicar las técnicas de codificación y recuperación manuales, ampliamente utilizadas. Este tipo de programas permiten al usuario ingresar y codificar segmentos de texto en una base de datos y posteriormente, recuperar y consultar todos los segmentos identificados con el mismo código o códigos, de acuerdo a condiciones de búsqueda indicadas por el usuario, que admiten filtros con utilización de operadores lógicos sobre conjuntos de documentos especificados, denominados catálogos. En realidad, todos los programas existentes incluyen esta facilidad.

"Archiva información factual acerca de cada documento bajo la forma de códigos aplicables a todo el documento (denominados variables de carátula), o acerca de individuos (en una hoja de locutor, donde se registran religión, género, grupo de edad, etc.). Estos códigos pueden utilizarse en una recuperación de códigos múltiples..." (Idem p. 454).

El inconveniente que Richards y Richards atribuyen a estos programas es que descontextualizan

los segmentos de texto codificados, sin permitir una fácil recuperación del contexto cada vez que sea necesario. Además, si bien permiten la introducción de códigos en cualquier momento, no permiten la construcción de nuevas categorías después de haber codificado muchos registros sin esas categorías.

El *Ethnograph*, en particular, permite guardar comentarios acerca de los códigos y de los datos en memos relacionados. Al combinar búsqueda textual y codificación permite la indexación automática de los resultados de las búsquedas, las cuales no se restringen al cruce de variables o intersección de códigos. Algunos programas de este tipo ofrecen también búsqueda textual basadas en patrones.

Los desarrolladores del *Ethnograph* están tratando de perfeccionarlo en sucesivas versiones. Se ha tratado de no poner límites al número de códigos disponibles o al número de veces que un pasaje particularmente fecundo puede ser codificado o removido y recodificado. Asimismo se intentan desarrollar programas en los que las categorías puedan ser consultadas, redefinidas, amalgamadas, borradas y duplicadas sin riesgo de pérdidas de información.

2. *Sistemas basados en reglas para la construcción de teorías* (Ejemplo: *el HyperResearch*). Los más conocidos entre este tipo de software pertenecen a una familia de programas desarrolladas para plataformas Macintosh, por lo cual explotan las capacidades especiales de estos sistemas para incluir imágenes, audio y video entre los documentos que pueden ser indexados. Incluye a *HyperCard*, *HyperMedia* e *HyperResearch*. Estrictamente hablando, son administradores de bases de datos

relacionales multimedia. Lo más semejante para computadoras personales es el *Toolbook*, comercializado como herramienta de autor para el desarrollo de multimedia incluido el hipertexto.

Estos programas realizan búsquedas textuales con autocodificación de los resultados. "Efectúan búsquedas booleanas de coocurrencia de códigos dentro de un documento y de intersección de códigos. Pero significativamente, permiten guardar los resultados de las búsquedas textuales. Este es un rasgo de clausura de sistema muy significativo". La clausura de sistema es la posibilidad de que los *outputs* del análisis, todos los resultados parciales de las operaciones aplicadas sobre los datos, pasen a formar parte del mismo sistema, convirtiéndose en datos primarios sobre los que se pueda volver a aplicar recursivamente las mismas operaciones en un próximo ciclo del análisis.

Pero el rasgo distintivo de esta familia de programas son las reglas de producción; que tienen la forma de un condicional: "Si un segmento de texto (denominado 'caso') está codificado como $C_1, C_2, C_3, \dots, C_n$, entonces codifíquese también como A". Posteriormente, el código A puede ser usado como condición en otra regla para producir códigos más generales y así ir edificando un sistema conceptual complejo. También puede realizarse el camino inverso: comenzando con una regla que expresa una hipótesis más amplia, tratar de encontrar las reglas cuyas conclusiones son las condiciones de la hipótesis.

Tales reglas serían equivalentes a las reglas de correspondencia que conectan las hipótesis conteniendo términos teóricos (aquí códigos conceptuales) con los da-

tos empíricos (segmentos textuales o códigos factuales). Pero el sistema tiene una debilidad que limita su utilidad. La condición C, llamada "meta", se cumplirá entonces por definición, cuando se cumplan todas sus condiciones. Vale decir que no existe en el sistema la posibilidad de codificar independientemente la presencia de C en ausencia de sus condiciones, lo cual sería necesario a los efectos de testear la hipótesis.

3. *Sistemas basados en la Lógica. (Ejemplo: el AQUAD).* También usan reglas de la forma "si... entonces", pero de mayor sofisticación y con una operatoria diferente. Son programas generalmente desarrollados en Prolog bajo el paradigma conocido como programación lógica. Su rasgo distintivo son unas sofisticadas técnicas de recuperación de texto por patrones, denominadas estructuras de hipótesis, usadas en análisis de agrupamientos (cluster analysis), además de las búsquedas booleanas. Tales técnicas incluyen la recuperación de casos positivos y negativos de un código, dentro de una cierta distancia al interior de un texto. Los informes de esas búsquedas presentan la forma de tablas numéricas conteniendo las distancias textuales entre los códigos solicitados. A partir de estos datos pueden construirse matrices de distancia, efectuar tests de hipótesis y aplicar las técnicas conocidas de taxonomía numérica. También confeccionan matrices cualitativas intersecando dos códigos o variables para sus respectivos valores y mostrando los textos que corresponden a cada intersección.

Otro rasgo distintivo sería la facultad para realizar análisis configuracional, basado en una técnica de la lógica formal, conocida como el algoritmo de Quine

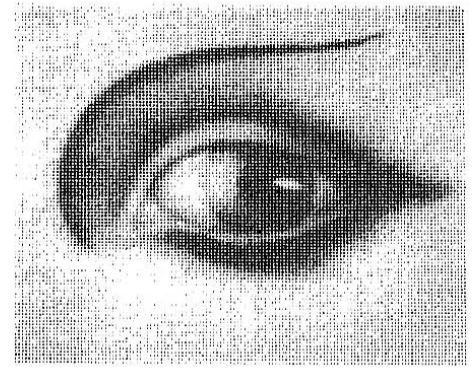
y McClusky.

"El análisis configuracional es una de varias técnicas inductivas usadas en inteligencia artificial para encontrar las condiciones necesarias o suficientes de ciertos productos o bien para hallar el conjunto más simple de reglas del tipo 'si... entonces' que predicen un cierto resultado" (idem pág. 457).

4. *Aproximación basada en índices.* Como ejemplo de la aproximación basada en índices, Richards y Richards presentan el NUD•IST, desarrollado por ellos mismos. Lo describen como un sistema híbrido, resultante de la combinación de un sistema de codificación y recuperación de texto con un sistema de indexación "[...] diseñado para permitir al usuario crear y manipular conceptos y almacenar y explorar ideas emergentes".

El programa trabaja con una sola copia de cada documento, guardando las direcciones de los segmentos indexados en "nodos" organizados en una jerarquía arborescente. No obstante, esas direcciones son usadas internamente por el sistema para hallar las unidades textuales solicitadas en las búsquedas y así poder mostrar *el texto*, no su dirección. Los documentos pueden ser ingresados directamente al programa, tipeándolos con el editor de textos que incluye, o pueden ser importados de cualquier otro procesador de textos con la sola condición de que se los ingrese en simple código ASCII, es decir libre de formatos. Los nodos representan a las clases de un sistema clasificatorio que es al mismo tiempo un esquema conceptual.

NUD•IST corre bajo Windows, aprovechando las ventajas que éste ofrece de poder trabajar con varios documentos abiertos simultá-



neamente en diferentes ventanas y tener a la vista todo el tiempo el árbol de índices. Cada nodo puede almacenar documentos completos, fragmento de uno o varios documentos, resultados de búsquedas textuales o de consultas del sistema de índices, la definición del concepto que el nodo representa o una descripción de su contenido y memos de formato libre donde se pueden registrar todo tipo de comentarios y reflexiones sobre el nodo y su contenido. También pueden adosarse memos a los documentos individualmente.

Un rasgo importante es la ya mencionada clausura del sistema que se manifiesta en que los resultados de las búsquedas y consultas de todo tipo pueden guardarse como documentos en nodos nuevos o preexistentes, y así ser a su vez indexados, comentados y consultados como todos los demás documentos.

Otro rasgo del programa, que se ajusta a las necesidades de los estudios cualitativos, es su flexibilidad: los documentos pueden moverse de un nodo a otro fácilmente, los nodos pueden ser fusionados, copiados, asignados a otro nivel jerárquico del árbol, subdivididos y eliminados sin riesgo de perder información, porque el texto no es manipulado directamente sino sólo las direcciones de las unidades textuales.

El sistema de búsqueda textual puede realizar búsquedas de ca-

denas de caracteres y también de patrones textuales en combinaciones muy complejas como para detallarlas en este trabajo.

Todas las tareas que se realicen rutinariamente pueden ser automatizadas a fin de que el sistema las efectúe tras la indicación de un solo comando, al estilo de los archivos batch del DOS o de las macros de otros programas.

NUD•IST también puede generar automáticamente matrices cualitativas similares a las obtenidas por el cruce de variables. Ello resulta muy útil si definimos nodos para cada variable demográfica y nodos "hijos" (así se llaman) para cada uno de los valores que toman estas variables. Por ejemplo, si tenemos un nodo denominado "religión" con los nodos "católico", "pentecostal" y "mormón" como hijos, bajo los cuales hemos indexado los textos completos de las entrevistas realizadas a católicos, pentecostales y mormones, respectivamente; y tenemos otro nodo denominado "pobreza" como ancestro de los nodos "pobreza material" y "pobreza espiritual", bajo los cuales hemos indexado las búsquedas textuales de dichos lexemas efectuadas sobre todas las entrevistas, entonces podemos pedir al programa que realice la intersección (cruce) de los nodos "religión" y "pobreza", obteniendo como salida una matriz de doble entrada que tendrá en las filas todos los valores de una de las variables y en las columnas todos los valores de la otra variable, y mostrará en las intersecciones las direcciones de las unidades textuales (párrafos u otra unidad que hayamos definido) en los que coocurren los valores indicados de cada variable.

Religión

Católico/Pentecostal/Mormón

Pobreza

pobreza material/pobreza espiritual

Esta matriz puede ser guardada en un nodo como un documento, indexada, consultada y editada. Pero también podemos utilizarla para cruzar con ella una tercera variable y obtener así un "vector", cuya apariencia es la de otra matriz con una sola fila y varias columnas, que presenta en las columnas cada una de las intersecciones de la primera matriz y en la fila, los valores de la tercera variable. Con esta nueva matriz podemos realizar las mismas operaciones que con la primera, salvo volver a cruzarla con otra variable.

Esta facilidad nos permite monitorear la saturación relativa de las categorías, complejizar las tipologías y contabilizar las frecuencias de casos presentes en las intersecciones como para elaborar cuadros estadísticos con los que efectuar algunas pruebas cuantitativas³. También permite hacernos algunas preguntas que orientarán no sólo nuestro análisis sino también la recolección de nuevos datos. Por ejemplo, si la intersección de "pobreza espiritual" y "Pentecostal" aparece vacía, ¿se deberá a que los pentecostales no utilizan este lexema, o será necesario realizar más entrevistas a pentecostales para poder confirmar si está vigente o no en este grupo religioso? ¿O tal vez no hemos preguntado explícitamente en las entrevistas realizadas?

Como limitaciones del programa los autores ven su incapacidad para manejar gráficos para la elaboración de esquemas conceptuales no arborescentes y su extrema flexibilidad, que podría convertirse en caos si el investigador no la sabe administrar correctamente.

5. *Sistemas de redes conceptuales.* (Ejemplo: el ATLAS). Como otros programas ya comentados, ATLAS puede codificar y recuperar texto, efectuar búsquedas de cadenas de caracteres y de patrones textuales, confeccionar memos que, a su vez, pueden ser codificados y recuperados (constituyendo así un sistema cerrado), y también, como rasgos originales, permite agrupar los códigos en "familias" y ordenarlos según diversos criterios, incluyendo la fecha de más reciente uso y el número de referencias indexadas.

Pero su particularidad reside en que posee la capacidad gráfica para diseñar los llamados diagramas conceptuales, gráficos conceptuales, redes semánticas, redes conceptuales, o mapas conceptuales⁴. Son representaciones gráficas de información conceptual. Las categorías o códigos, aquí también llamados "nodos", son visualmente representados por pequeñas cajas que podemos unir mediante flechas rotuladas denominadas "arcos orientados", que lejos de ser sólo líneas de significado indeterminado representan tipos específicos de relaciones con determinadas propiedades lógicas. El gráfico obtenido se denomina técnicamente un "gráfico orientado". Los nodos representan objetos, propiedades o conceptos y los arcos orientados son las relaciones que los unen.

Hasta aquí no se percibe la diferencia con los paquetes gráficos comerciales para diseñar diagramas de todo tipo disponibles en el mercado local, incluso en versiones shareware (Snap Graphics, Edge diagrammer, otros). Pero ATLAS otorga internamente una representación lógica a las relaciones que dibujamos, de manera que explorar el gráfico es, al mismo tiempo, razonar y deri-

var del mismo las consecuencias implícitas. De modo que el esquema conceptual no es sólo un dibujo sino una red de enunciados relacionados lógicamente.

El programa provee de algunas de las relaciones más comunes como "es un", "un tipo de", "pertenece a", "es un caso de", "(es) causa (de)", "parte de", "contradice a", entre otros. Cada una de estas relaciones posee propiedades lógicas. Por ejemplo, la relación *ako* ("es un tipo de") posee propiedad transitiva, es decir que si A es un tipo de B y B es un tipo de C, entonces A es un tipo de C. Pero el usuario también puede definir nuevas relaciones e indicar sus propiedades lógicas fácilmente.

ATLAS puede dibujar el gráfico por nosotros de acuerdo a las relaciones indicadas y puede redibujarlo automáticamente cuando se vuelve muy enmarañado.

Como limitaciones, Richards y Richards destacan que no puede representar relaciones triádicas como "el cura A casa al señor B con la señorita C", y que algunas relaciones lógicas simples como la alternitud (disyunción), negación o cuantificación lógica (todos, algunos) requieren de artificios un tanto complicados para ser representados.

Las redes conceptuales del ATLAS son mucho más fértiles, flexibles y adecuadas que los rígidos árboles jerárquicos del NUD•IST para representar nuestras ideas en desarrollo, pero desde el punto de vista de su potencial lógico para derivar conclusiones son superadas por las reglas de producción del AQUAD.

Conclusiones

Las computadoras por sí solas no pueden darle mayor seriedad

a una investigación deficiente desde su mismo diseño. Tampoco pueden hacer milagros con datos mal recogidos, parciales, sesgados, como no pueden decirnos lo que ellos significan. Pero cumplen la función de un equipo de asistentes habilidosos.

Las mismas tareas pueden ser realizadas de manera más efectiva, más precisa, más exhaustiva, con mayor cantidad de información y en menor tiempo, a medida que pasamos del procesamiento manual a la utilización de procesadores de texto y bases de datos, y de éstos al uso de programas de ADC diseñados para uso profesional. La accesibilidad y versatilidad de las computadoras personales pone estas facilidades al alcance de nuestra mano y saber aprovecharlas sólo puede traernos beneficios.

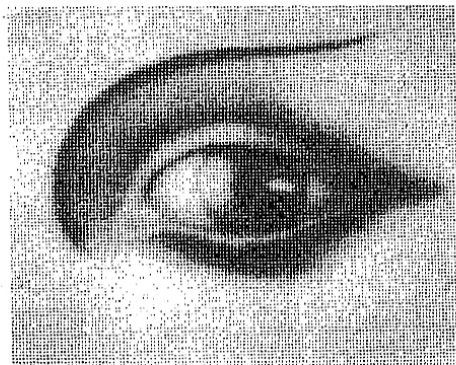
Notas

¹ Escuela Superior de Trabajo Social. U.N.L.P.

² Algunos procesadores de texto incorporan subprogramas de mailing, que permiten generar verdaderas bases de datos con campos para nombre, dirección, ciudad, código postal, etc. Pero, en este caso, tales bases no pueden ser utilizadas para realizar búsquedas independientemente del formulario al que resultan asociadas (comúnmente una carta tipo, destinada a ser enviada a diversas personas, con un texto idéntico para todas ellas, pero variando los datos personales de cada destinatario).

³ Cuya validez estará limitada, por supuesto, por la representatividad estadística de los datos.

⁴ "Un mapa conceptual es un recurso esquemático para representar un conjunto de significados conceptuales incluidos en una es-



tructura de proposiciones" (Novak y Gowin, 1988; citado en Ontoria et al. 1995:31).

Bibliografía

GLASER, Barney y Anselm STRAUSS: *The discovery of grounded theory: strategies for qualitative research*. Aldine Publishing Company. Nueva York. 1967.

RICHARDS, Thomas J. y Lyn RICHARDS: "Using computers in qualitative research". En: *Methods of collecting and analyzing empirical materials*, Cap. 28, pág. 445 - 462. 1992.

RODRIGUEZ, P. Gustavo: *Fundamentos de análisis de datos cualitativos para el desarrollo de software para ciencias sociales*. 1997.

VASILACHIS de GIALDINO, Irene: *Métodos cualitativos I. Los problemas teórico - epistemológicos*. Editorial C.E.A.L. Buenos Aires. 1992.