RENOAR2: CAPACIDADES GRAFICAS PARA ESTRAUS

HENRY MALBRAN ROJAS
Secretaria Ejecutiva Comision de Transporte Urbano
Ahumada 48 of.525 Santiago-CHILE Tlx.240767 Oplan-CL

RESUMEN

Tradicionalmente los modelos de transporte urbano han entregado el resultado de sus análisis en forma de largos y tediosos listados y archivos computacionales, difíciles de digerir aún para los especialistas. Una manera casi obvia de aliviar la complejidad de esta tarea es proveer herramientas gráficas para analizar y eventualmente editar datos y resultados. Ello tiene no sólo una importancia estética o política , como inicialmente podría pensarse, pues parece claro que la disponibilidad de capacidades gráficas debería hacer más rápido y preciso el trabajo de los tecnicos. Puesto que éste es también un objetivo de todo modelo, cada vez más la tendencia actual es incluir en los programas computacionales que los implementan, algún tipo de herramienta que permita hacer análisis gráfico de datos y resultados.

Computacionalmente sin embargo, las soluciones no son inmediatas. A pesar del fuerte desarrollo de la gráficación computacional, subsisten grandes problemas de hardware y software, que se traducen por ejemplo en ausencia de normas comunes, incompatibilidad de equipos y alto costo. Las aplicaciones en transporte urbano son especialmente sensibles a este tipo de dificultades.

El presente trabajo describe el estado de desarrollo de RENOAR2 (REpresentación de NOdos y ARcos en Ambiente Regis), un intento de implementar una herramienta gráfica para analizar información asociada a los modelos utilizados en ESTRAUS (Estudio Estrategico de Transporte Urbano de Santiago). RENOAR2, concebido para graficar datos asociados a las redes de transporte urbano, es en realidad parte de una familia de herramientas gráficas que se desarrolla con tal fin.

Acta III Congreso Chileno de Ingeniería de Transporte, 1987

1.Introduccion

Tradicionalmente los modelos de transporte urbano han entregado sus resultados en forma de largos y tediosos listados y archivos computacionales, difíciles de digerir aún para los especialistas. Si las ciudades objeto de estudio son de tamaño mediano o grande, entonces el análisis detallado de los resultados numéricos de los modelos suele adquirir tal complejidad que se hace imposible, en terminos prácticos.

Considerese por ejemplo el análisis de las cargas sobre cada uno de los arcos de una red, producidas por un modelo de asignación. O los pasajeros subiendo, bajando y transbordando en cada uno de los nodos de una Red de Transporte Publico.O por último, los resultados de la partición modal de una matriz Origen-Destino de una ciudad dividida en centenares de zonas.

Una manera casi obvia de aliviar esta tarea es la de proveer al especialista con herramientas que le permitan estudiar, y eventualmente editar, datos y resultados en forma gráfica. La ventaja inmediata de este enfoque deriva del hecho de aprovechar la mayor capacidad del analista para estudiar simultaneamente un conjunto de datos en forma visual. Ello permite focalizar rápidamente su atención en ciertas areas especificas de los datos, los que posteriormente podrán ser estudiados numéricamente con más detalles.

Otro tipo de información tiene más valor cuando puede ser desplegada gráficamente. El recorrido de una linea de buses sobre la red vial, ciertamente es más comprensible si se describe con la ayuda de un gráfico que si en cambio se presenta como una sequencia de nodos ordenados.

Parece claro entonces que la disponibilidad de capacidades gráficas debiera hacer más rápido y preciso el trabajo de los especialistas. Puesto que esto es también uno de los objetivos primarios de todo modelo, es fácil percibir porqué los programas computacionales que los convierten en herramientas prácticas, están incluyendo más y más capacidades gráficas para analizar datos y/o

resultados.

Aunque las ventajas del enfoque gráfico son evidentes, computacionalmente las soluciones no son inmediatas. Si bien la computación gráfica ha tenido un fuerte desarrollo en los últimos años, subsiste la inexistencia de normas y lenguajes gráficos estandar, obligando a que el desarrollo de software quede amarrado a la maquina y la portabilidad del mismo se ve muy limitada. Por otra parte el costo del hardware y periféricos más avanzados es todavia bastante alto y a veces prohibitivo.

Los equipos más accesibles imponen restricciones en aspectos tales como la resolución y la velocidad de despliegue de la informacion en pantalla. Las aplicaciones de transporte urbano pueden ser muy sensibles a este tipo de limitaciones, dado que sin un tiempo de respuesta razonable y un nivel de detalle mínimo , el enfoque gráfico pierde gran parte de su atractivo.

El trabajo descrito aquí ha sido desarrollado en un sistema operativo VMS utilizando FORTRAN-77. Como lenguaje gráfico se usó REGIS (Remote Graphics Intruction Set) provisto por la Digital Equipment Corporation, y que necesita un terminal del tipo VT-241. En consecuencia, los trucos de programación y la forma en que se enfrentan los problemas de representación gráfica están especificamente pensados para aprovechar las ventajas y acatar las limitaciones impuestas por este ambiente. No obstante ello, se ha hecho un esfuerzo por estructurar los programas de manera de facilitar la reescritura de las subrutinas gráficas ante la eventualidad de un cambio de ambiente.

El presente trabajo describe el estado de desarrollo de RENOAR2 (REpresentación de NOdos y ARcos en Ambiente Regis), un miembro de una familia de herramientas gráficas diseñadas para ser incorporadas al Estudio de Evaluación y Desarrollo de Transporte Urbano de Santiago. RENOAR2 fue concebido para analizar graficamente datos asociados a redes de transporte, es decir a sus nodos y arcos. La intención principal de este documento será la de explicar las lineas centrales de su implementación.

2.La Red de Transporte Urbano

La mayoría de los programas de diseño gráfico tratan de identificar los elementos básicos que conforman una figura, le asignan ciertos atributos (invariantes o temporales) y establecen una cierta conectibidad entre ellos. Las redes de transporte tienen ciertas particularidades que se pueden aprovechar o lamentar segun el caso. Una primera "ventaja" no despreciable, está dada por el hecho de que una red de transporte puede ser normalmente representada en dos dimensiones (es decir en un plano), con lo cual se elimina la necesidad de contar con ciertas funciones gráficas complicadas como rotaciones, sectores escondidos y visibles, etc. Otra característica que simplifica las cosas es que toda la red puede ser desmenuzada en sólo dos elementos básicos: nodos y arcos. Un nodo es un punto del plano que tiene un interés particular desde el punto de vista del transporte, como por ejemplo una intersección de dos calles, un terminal de buses, el centroide de una zona. Un arco es una conexión entre dos nodos contiguos (usualmente una sección de vía), y puede ser primariamente definido como un par ordenado de estos elementos.

Las dificultades del despliegue gráfico de una red de transporte dicen relación con el tamaño que esta puede alcanzar y con el volumen de atributos que es necesario graficar. Si bien es cierto que los elementos de descripción (nodos y arcos) son simples, la red de una ciudad de relativa importancia requerirá cientos, miles y hasta decenas de miles de estos elementos básicos, de manera de describirla en forma que sea útil para los especialistas de transporte. Es por tanto necesaria gran confiabilidad de la información que define las características de nodos y arcos, lo cual puede no ser una tarea fácil cuando estamos hablando de miles de estos elementos.

Por otra parte los atributos asociados a la red -que en definitiva son los que interesa gráficar- pueden ser muchos y de muy variada índole. Además, en general, estos atributos son variables y el programa graficador deberá arreglarselas para darle a cada uno de ellos en cada nodo o arco, el "peso" visual que corresponda a su importancia real. La cantidad de atributos que el Programa pueda o deba poner simultaneamente sobre la pantalla dependerá de la

Acta III Congreso Chileno de Ingeniería de Transporte, 1987

cantidad de colores y/o achurados disponibles. Pero tambien obviamente habrá una restricción en la capacidad del observador para discriminar los objetos visuales antes de que el gráfico se convierta en una mancha inentingible.

Es claro que el usuario tendrá que tener algun grado importante de decisión respecto de las caracteristicas de su gráfico. El número de atributos a ser desplegado es una tipica decision que debe quedar al arbitrio del usuario dentro de ciertos margenes, pero quizás el color o el achurado puede ser decidido por el programa sin mayores problemas, reduciendo su complejidad y dependencia.

Existen sin embargo, exigencias generales del usuario que deben ser consideradas como condiciones sine qua non para el programa. Por grande que sea la red, siempre deberá ser posible tener una visión global de la misma, lo cual significa reducir la red al tamaño de una pantalla corriente de 14 pulgadas o algo así, sin considerar las áreas reservadas para menus y otros datos no gráficos. A partir de esta visión global, el observador muy probablemente querrá tener una aproximación con más detalle de cierto sector. Este proceso puede repetirse hasta que el nivel de detalle sea aceptable para el usuario o hasta que deseche el programa por inútil. Sólo vale la pena seguir adelante con el trabajo si la habilidad del programador y las restricciones de hardware y del lenguaje gráfico permiten realizar este tipo de tareas.

3. Los Datos de la Red

Como hemos visto la red de transporte puede ser gráficamente descrita por un elemento básico (nodo) y por una entidad (arco) que relaciona dos nodos en forma única. Todos los nodos deben estar incluidos al menos en un arco para quer su existencia tenga sentido físico. Sin ánimo de ser exhaustivos podemos enumerar las caracteristicas generales de estos elementos:

NODOS - Código Numérico Sequencial
- Coordenadas X e Y
- Nombre Externo
- Dirección

ARCOS - código Numérico Sequencial
- Nodo Origen
- Nodo Destino
- Sentidos de Trafico
- Nombre de la vía
- Capacidad de la vía
- Largo de la vía

Desde el punto de vista estricto de la gráficación de la red sin embargo, es rápidamente evidente que en esta lista hay más información que la necesaria. Un nodo sólo requiere de un código y de sus coordenadas (tres datos) para quedar perfectamente definido. Igualmente un arco sólo necesita de un código, su nodo-origen y su nodo-destino (otra vez tres datos), para ser identificado inequivocamente. Todas las demás caracteristicas mencionadas no son necesarias para la graficación de la red y eventualmente pueden no existir o estar incompletas. Nótese sin embargo que ellas tienen una ventajosa propiedad: son constantes en el corto y mediano plazo. Esta propiedad es importante en casi todas las aplicaciones, porque puede aprovecharse a la hora de desplegar la información en pantalla. (Una excepción notable la constituye una aplicación especialmente importante como es la edición gráfica de redes, en donde ni los nodos ni los arcos ni sus caracteristicas son constantes. Con todo, dicha aplicación no será tratada en el contexto de este trabajo, dado que se encuentra en una etapa primaria de desarrollo).

El mayor volumen de información a ser graficada proviene por supuesto de los atributos variables. Casi siempre la fuente de estos datos son los modelos o las mediciones hechas en terreno para cuantificar alguna variable de interés. Existe un sin número de estos atributos asociados a nodos y arcos. En cada caso, el programa deberá asignar a cada atributo de cada elemento, una representación gráfica acorde con su equivalente numérico.

Conviene definir por último, un tercer tipo de información que no esta exclusivamente asociada a un nodo o a un arco. Tipicamente las rutas generadas por un modelo de asignación o las lineas de transporte publico, son ejemplos de este tipo de datos. Una ruta es un

conjunto ordenado de nodos (o de arcos), y en este caso el programa no necesita estar inquiriendo el valor del atributo y cambiando el patrón de representación nodo a nodo (o arco a arco). De nuevo, esta caracteristica facilita el despliegue gráfico en la pantalla.

La figura Nº 1 presenta un esquema general del tipo y del origen de los atributos involucrados.

Hasta ahora hemos identificado los elementos y los atributos que nos interesa graficar y hemos señalado tambien las fuentes posibles de estos datos. La velocidad de acceso a dicha información será una variable clave para mantener los tiempos de respuesta dentro de los margenes aceptables para el usuario. Veamos ahora alguna forma de ordenar los datos en archivos, tomando en cuenta esta necesidad. Podemos distinguir el siguiente grupo :

- Archivo Básico de Nodos
  - Código del nodo (Clave de Entrada)
  - Coordenada X (Clave de Entrada)
  - Coordenada Y (Clave de Entrada)

-Archivo de Atributos Constantes de Nodos
  - Código del Nodo (Clave de Entrada)
  - Atributos

- Archivo de atributos Variables de Nodos
  - Código del Nodo (Clave de Entrada)
  - Atributos

- Archivo Básico de Arcos
  - Código del arco (Clave de Entrada)
  - Nodo-origen (Clave de Entrada)
  - Nodo-destino (Clave de Entrada)

-Archivo de Atributos Constantes de Arcos
  - Código del Arco (Clave de Entrada)
  - Atributos

- Archivo de atributos Variables de Arcos
  - Código del Arco (Clave de Entrada)
  - Atributos

- Archivo de datos tipo Tallarin
  - Nombre del dato (Clave de Entrada)
  - Conjunto de Nodos (o Arcos)

Por supuesto la ordenación física en archivos computacionales reales puede no ser exactamente la anterior, pero al menos es una aproximacion. En este punto es importante tomar en cuenta algunas consideraciones practicas de implementación. Por ejemplo es posible que los arcos sean más requeridos a través de los nodos que lo conforman que a través de su código y ello deberá ser tomado en cuenta en la forma de acceso a la información. Por otra parte la existencia de varias claves de acceso en un archivo es una facilidad no siempre disponible.

Una alternativa razonable es almacenar los datos en una base de datos propiamente tal, lo cual facilita la lógica de acceso pero requiere de herramientas adicionales e impone nuevas limitaciones a la portabilidad del programa.

4. El Enfoque de RENOAR2

En esta sección discutiremos algunas de las características del programa graficador.

RENOAR2 fue originalmente diseñado para manejar la red de transporte de Santiago, la cual contiene unos mil nodos y más de 2500 arcos. Se excluyen de consideración todas las vías que conforman las redes locales de la ciudad.

La red de Santiago ha sido seccionada en tres planos que se pueden superponer como las capas de una cebolla. El primer plano contiene los ejes más importantes de la ciudad, en tanto que el segundo y tercer plano contienen el resto de las vias, seleccionadas de acuerdo a la demanda que sirven y a su continuidad física. Los tres planos son complementarios y permiten al usuario elegir el grado

de detalle del análisis. Así, si sólo está interesado en los ejes importantes, podrá "limpiar" la pantalla de las vías de los planos inferiores.

El usuario dispone además de siete niveles de ampliación ("zoom") y reducción para focalizar el análisis en un sector de la red y observar con más detalle sus características y atributos. Para tal efecto dispone de una "ventana" que puede mover a voluntad por la pantalla hasta el sector de interes. La velocidad de desplazamiento y el tamaño de esta ventana tambien son regulables por el usuario con un simple golpe de tecla.

Una vez definido el sector de interés y el grado de detalle deseados, se pueden graficar los atributos de cada elemento de la red. El programa permite poner simultaneamente en pantalla tres atributos por nodo y dos por arco, aun cuando este criterio puede ralajarse cuando se trata de atributos constantes. La forma en que efectivamente se realiza esta tarea depende del lenguaje gráfico disponible y de las características de los datos, tal como hemos dicho antes. En la sección siguiente hablaremos un poco más de esto.

Los criterios definidos anteriormente (tres planos para la red, siete niveles de zoom, dos atributos por arco y tres por nodo) son arbitrarios y pensados para Santiago. Obviamente pueden modificarse sin mayores problemas para ser aplicados en un contexto distinto.

RENOAR2 ha tratado de ser congruente con la idea generalmente aceptada ahora, de que los programas deben ser interactivos, amigables y transparentes para el usuario. Como siempre, estas son al menos las bondadosas intenciones del diseñador del programa y su veracidad sólo podrá ser juzgada por el usuario. En cualquier caso, este manejará todas sus decisiones a traves de un conjunto de menus (que tambien son gráficos) previamente definidos, y la mayoría de las veces podrá recurrir a un HELP en un momento de duda. La velocidad de despliegue de la información gráfica parece estar dentro de margenes razonables y se ha intentado definir los patrones de representación en la forma más clara posible. El usuario decidirá finalmente cuan real es la implementación de estos criterios en al práctica.

5. Consideraciones de la Implementación Gráfica

Todos los alcances que se hacen a continuación son válidos dentro del ambiente en que originalmente se ha desarrollado RENOAR2. Es muy posible que ellos se demuestren parcial o totalmente equivocados si se cambia el Hardware o el Software gráfico disponible.

Cualquier programa gráfico partirá utilizando un lenguaje de alto nivel (por ejemplo FORTRAN) para leer, ordenar, analizar y seleccionar información. Luego de seleccionados los datos, se recurre a un lenguaje gráfico (REGIS en este caso) que genera las instrucciones que son finalmente enviadas a la pantalla para producir el gráfico deseado. Todo el proceso consume tiempo, y hemos visto que este es un recurso crítico en un programa graficador. Es por tanto necesario buscar las formas de reducir este consumo aprovechando las características del ambiente.

En el corto plazo la red misma y algunos de sus atributos son constantes, luego generan siempre el mismo código gráfico. Ello sugiere la posibilidad de "enseñar" por una vez al programa las características de la red y de sus atributos invariantes, y entonces guardar el código gráfico así generado, de alguna manera accequible. La siguiente vez el programa hará la misma tarea mucho más rápidamente pues ya "sabrá" como hacerlo: sólo tendrá que accesar el código gráfico y enviarlo a la pantalla.

La segunda consideración dice relación con la forma de generar las instrucciones gráficas. Una misma instrucción repetida numerosas veces es más rápida que varias instrucciones sucediendose unas a otras. Si tenemos un arco por ejemplo, podemos hacer que el cursor de la pantalla se pocisione en el nodo-origen y en seguida trazar una linea hasta el nodo-destino. Estas dos instrucciones (pocisionar y trazar linea) deben repetirse tantas veces como arcos se quiera dibujar. Si estos son todos los de la red, o los de un conjunto de rutas, la operación puede ser excesivamente costosa en terminos de tiempo. Sabemos sin embargo, que muchas veces los arcos pueden asociarce en conjuntos ordenados (como grandes vias por ejemplo) y que en este caso el nodo-destino de un arco es el nodo-origen del siguiente. De

esta manera bastará pocisionar el cursor en el nodo-origen del primer arco, trazar una linea hasta el nodo-destino, luego otra linea hasta el nodo-destino del siguiente arco y asi sucesivamente hasta acabar con el conjunto. Este procedimiento reduce considerablemente el tiempo de despliegue,aunque por supuesto no siempre se puede usar. Es claro sin embargo, que los atributos de tipo "tallarin" (rutas y lineas de buses) pueden aprovecharlo.

La utilización de estas dos posibilidades - "enseñanza" y repetición de instrucciónes - permitió reducir el tiempo de despliegue de la red completa de Santiago (2500 arcos) desde cuatro minutos y medio hasta 35 segundos.

La información alfa-numérica en modo gráfico siempre genera instrucciones lentas. Indudablemente la posibilidad de poner nombres de nodos y vias y de cuantificar numéricamente los atributos, debe estar abierta para el usuario y poco se puede hacer al respecto. Es útil, de ser posible, reservar en la pantalla un área no gráfica para poner allí los datos alfa-numéricos cada vez que ello no atente contra la claridad.

Todo lo expuesto hasta ahora sugiere que la velocidad del despliegue gráfico es la restricción mas importante que debe enfrentar el diseñador del programa. Un dilema adicional se presenta al tratar de decidir entre la selección y la totalización de los datos a graficar. Si es necesario dibujar un sector de la red, ¿ es mejor seleccionar los datos que "caen" dentro del sector, generar el código correspondiente y graficar sólo esos datos ? ¿ o es más "eficiente" desplegar todos los datos de la red, evitando la selección y aprovechando la ya mencionada capacidad de aprender del programa, aun cuando ello signifique que trabajará inútilmente durante buena parte del despliegue ?.

Sorprendentemente quizás, la respuesta a esta disyuntiva es ambivalente. Las pruebas puramente empíricas han demostrado que (dentro de las limitaciones de RENOAR2 y del equipo disponible) si el sector de interés es pequeño respecto al total de la red, es conveniente seleccionar antes de graficar. Sin embargo, si dicho sector se refiere a un área importante como la mitad o hasta un

cuarto del tamaño total de la red, es mas rápido evitar la selección y desplegar toda la red, a pesar de que sólo una parte de ella aparecerá en pantalla.

Claramente esta última consideración es demasiado discutible y aventurada, y entonces por el bien de todos es mejor terminar aquí.

6. Conclusiones

La representación gráfica de información asociada a redes de transporte es una necesidad y una herramienta útil, entre otras cosas porque permite analizar mejor las características dela red y los resultados de ciertos modelos. RENOAR2 es un intento de implementar un programa que realice esta tarea, pero desafortunadamente existen limitaciones que impiden diseñar y escribir un programa que resuelva eficientemente el problema independientemente del equipo en que se trabaje.

Las herramientas de software gráfico del programa son relativamente simples y los elementos de descripción de la red son solamente dos (nodos y arcos). No obstante el volumen de datos involucrados y sus requerimientos de ordenamiento, acceso y confiabilidad imponen complejidad al programa. Por otra parte, para mantener los tiempos de respuesta al usuario dentro de margenes aceptables, es necesario aprovechar todas las ventajas tanto de lenguajes y equipos disponibles como de las características de las redes de transporte. Debe recordarse que en general el software siempre es lento en terminos computacionales.

El primer objetivo de RENOAR2 ha sido el de proveer el despliegue gráfico de la red y de los atributos asociados a los nodos y arcos que la conforman. Cuando este propósito es plenamente alcanzado (es decir a plena satifacción del usuario), el paso siguiente será la implementación de una aplicación especialmente importante como es la Edición Gráfica de Redes.

REFERENCIAS

HOROWITZ, A.J. and PITHAVADAIN A.R. (1987) Generalized Computer Aided Design of Transportation Networks. Transportation Quaterly Vol. XLI Nº3, 397-409.

MADSEN B.G. (1984) On the Use of Road Network Databases in the Locational and Routing Decisions. Research Report Nº8/1984 Technical University of Denmark.

COOKE F.C. (1987) Map Storage on CD-ROM BYTE VOL 12 Nº8 pags.129-137.

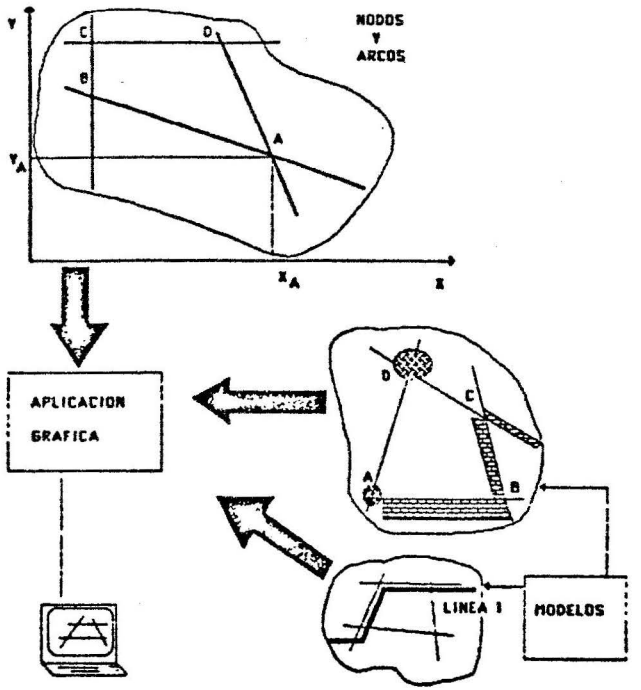

Figura Nº 1
Origen de los datos y atributos de la Red de Transporte

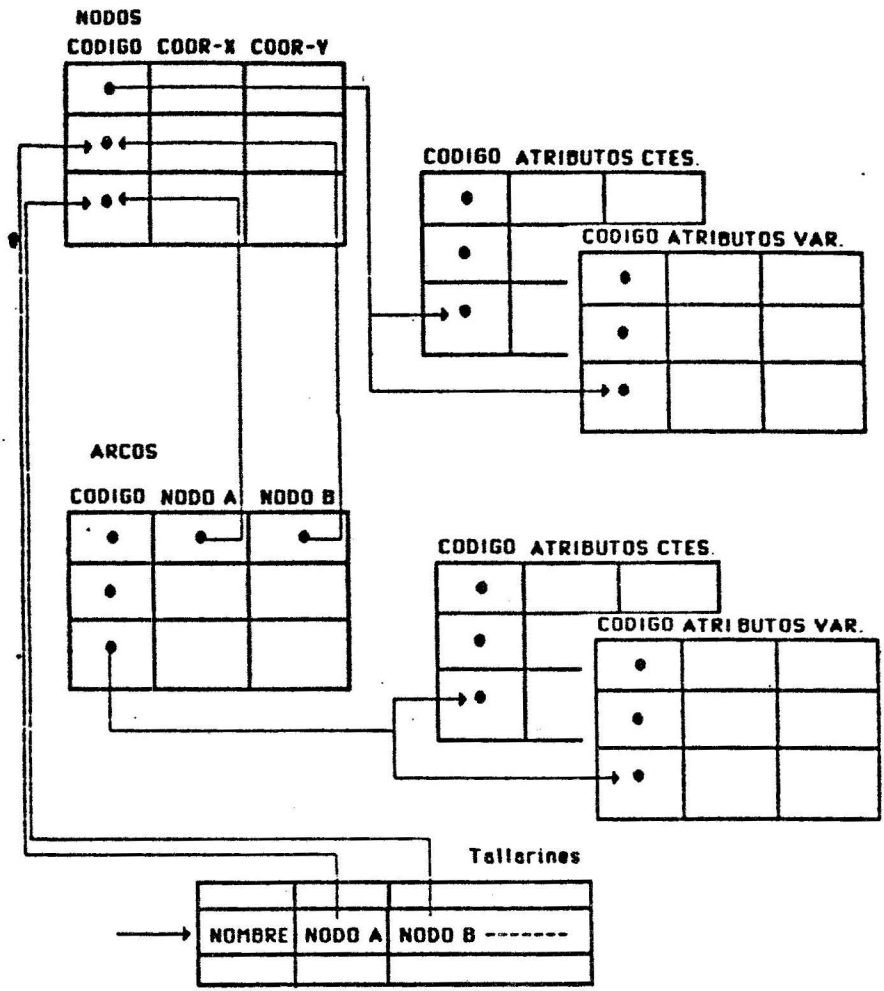

Figura Nº 2
La organización de los datos de la Red

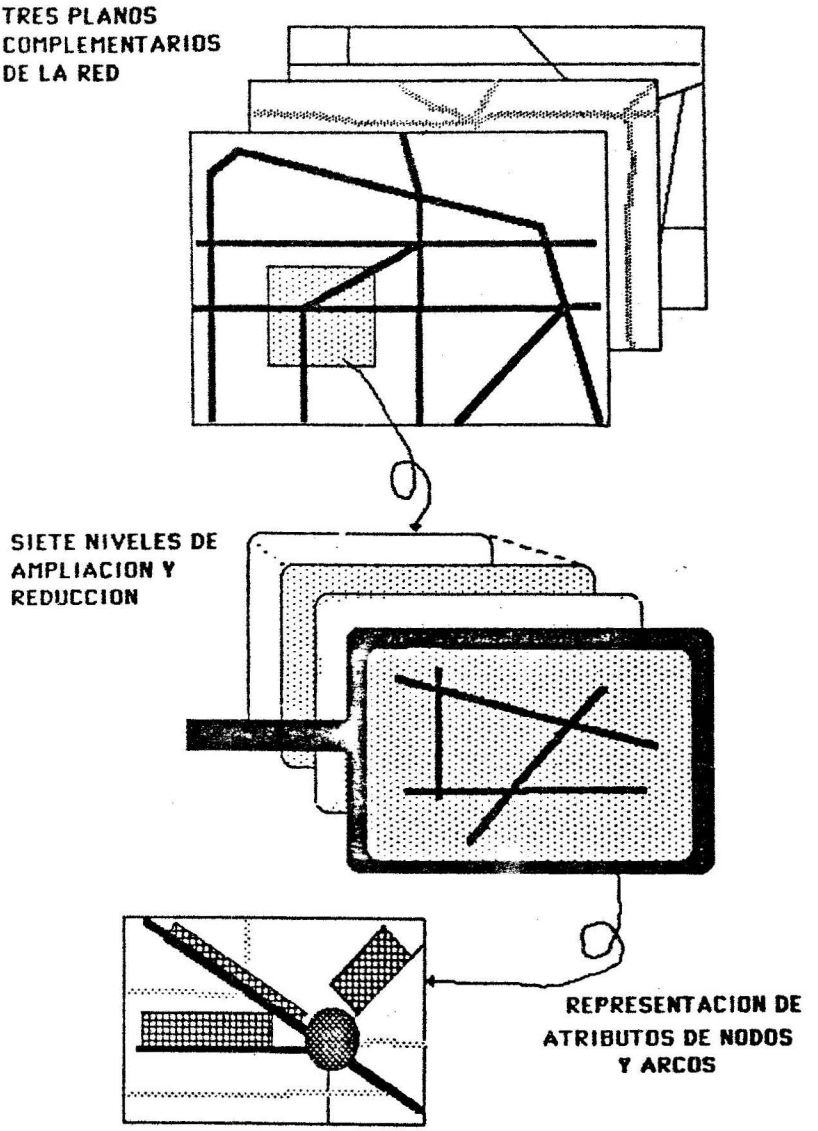

FIGURA Nº3
EL ENFOQUE RENOAR2

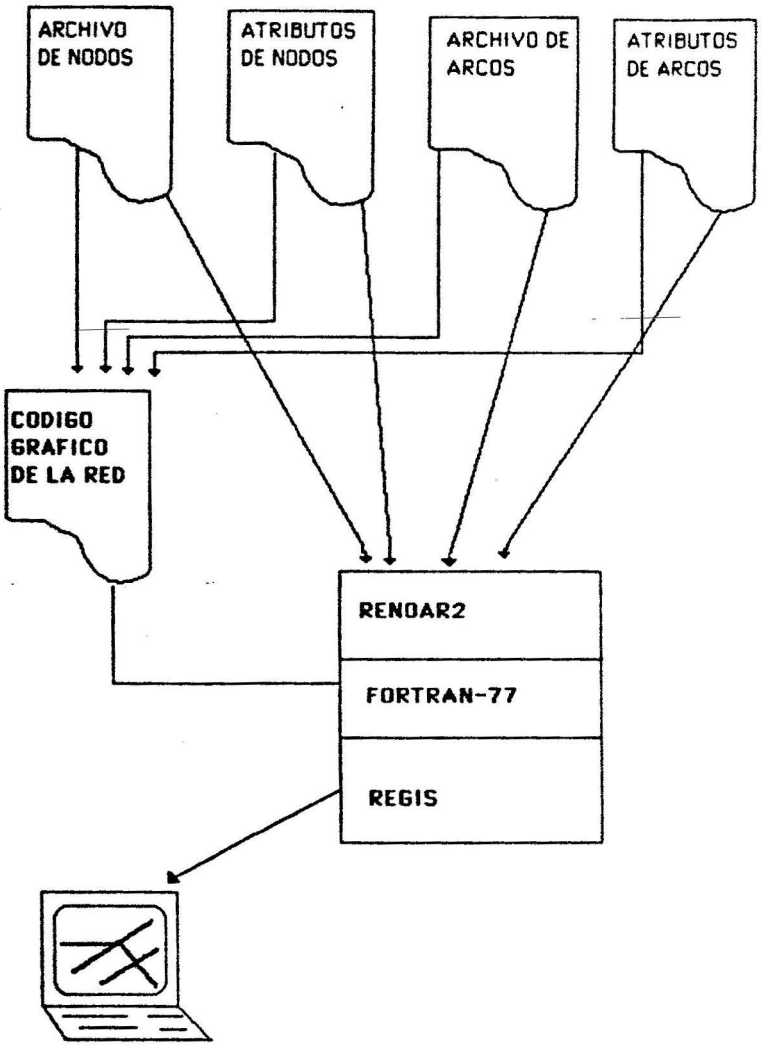

FIGURA Nº4

ESQUEMA GENERAL DE RENOAR2