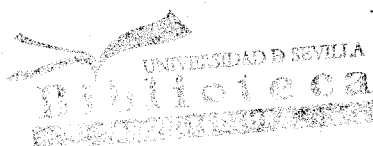


ESCUELA TÉCNICA SUPERIOR DE ARQUITECTURA DE SEVILLA



GENERACIÓN Y OPTIMIZACIÓN AUTOMÁTICA DE ESQUEMAS
DE DISTRIBUCIÓN DE EDIFICIOS EN PLANTA

Tesis Doctoral
presentada por:

Juan José Sendra Salas

Dirigida por:

D. Alberto Donaire Rodríguez

Dr. Arquitecto

Catedrático de Elementos de
Composición, E.T.S.A. de
Sevilla

Sevilla, 1984

	Pág.
<u>I.</u> <u>INTRODUCCIÓN</u>	1
<u>II.</u> <u>GENERACIÓN AUTOMÁTICA DE ESQUEMAS DIS-</u> <u>TRIBUTIVOS: DESCRIPCIÓN DEL PROBLEMA</u> ..	6
II.1. ESTADO ACTUAL DE LA CUESTIÓN	6
II.1.1. Generación de esquemas adimen- sionales	6
II.1.2. Dimensionamiento	10
II.2. ALTERNATIVAS E HIPÓTESIS DE TRABAJO ...	13
II.2.1. Generación de esquemas adimen- sionales	14
II.2.2. Dimensionamiento	16
<u>III.</u> <u>ALGORITMOS PARA LA GENERACIÓN AUTOMÁTICA</u> <u>DE ESQUEMAS DISTRIBUTIVOS</u>	18
III.1. GENERACIÓN DE ESQUEMAS ADIMENSIONALES .	19

	Pág.
III.1.1. Fijación de condiciones iniciales	19
III.1.2. Comprobación de conectividad .	20
III.1.3. Primera comprobación de planaridad	21
III.1.4. Análisis de la triconectividad	27
III.1.5. Segunda comprobación de planaridad	36
III.1.6. Comprobación de rectangularidad	37
III.1.7. Triangulación del grafo	38
III.1.8. Transformación de coordenadas de vértices	39
III.1.9. Obtención del grafo dual	46
III.2. DIMENSIONAMIENTO	49
III.2.1. Sustitución por dos grafos dirigidos	49
III.2.2. Definición de las matrices de adyacencia	51
III.2.3. Restricciones métrico-geométricas	51
III.2.4. Expresiones que traducen las restricciones	52
III.2.5. Naturaleza y costes de la función-objetivo	58
III.2.6. Resolución del problema por los métodos de programación ..	59
<u>IV. CONCLUSIONES</u>	62
<u>ANEXO I ALGUNOS CONCEPTOS DE LA TEORÍA DE GRAFOS</u>	62

	Pág.
<u>ANEXO II</u> <u>PROGRAMAS DE ORDENADOR</u>	71
A.II.1. GENERACIÓN DE ESQUEMAS ADIMENSIONALES .	72
A.II.1.1. Manual de usuario	72
A.II.1.2. Organigrama general	76
A.II.1.3. Listado del programa	77
A.II.1.4. Ejemplos	143
A.II.2. DIMENSIONAMIENTO	159
A.II.2.1. Manual de usuario	159
A.II.2.2. Organigrama general	164
A.II.2.3. Listado del programa	165
A.II.2.4. Ejemplos	191
BIBLIOGRAFÍA	200

Los teóricos de la arquitectura que, en los años sesenta y primeros setenta, auguraban una transformación radical de los modos de proyectar gracias a la utilización de los ordenadores, no han visto cumplidas sus previsiones. En contraste con ello, asistimos a una rápida adopción de técnicas informáticas como instrumento de trabajo en las ciencias físicas y también en las ciencias humanas, que han visto así eficazmente aumentadas sus posibilidades de desarrollo.

La diferencia que en este sentido existe entre el campo de las ciencias y el de la arquitectura estriba quizá en que, en el primer caso, los ordenadores han podido aportar directamente su capacidad instrumental, mientras que, en lo que al diseño arquitectónico se refiere, era necesario previamente revisar y establecer las formulaciones teóricas de partida; era imprescindible elucidar primero qué operaciones y decisiones se podrían encomendar al ordenador, y cuales otras sería inadecuado o imposible hacerlo. O incluso, llegando más lejos, averiguar si es factible, estudiando a fondo la naturaleza del proceso de diseño, sustituir al hombre por la máquina, en qué medida y con qué modelos operativos.

Negroponete (1970, p.11) afirmaba hace una década que el ordenador puede emplearse en el proyecto de tres modos posibles:

1. Automatizzare i procedimenti attuali rendendoli più rapidi e meno costosi.
2. Modificare i metodi attuali per adattarli alle esigenze e alla struttura della macchina, il che implica di considerare soltanto i problemi che si ritengono compatibili con la macchina.
3. Presentare il processo della progettazione considerato come evolutivo, a una macchina pure considerata come evolutiva, in modo che si sviluppino un apprendimento, un adattamento e una crescita reciproci.

Pero hoy, que la evolución tecnológica y los métodos de producción de componentes nos ofrecen ordenadores de gran fiabilidad a bajo precio, vemos sin embargo que las aplicaciones informáticas más extendidas en el campo de la arquitectura se limitan al primer grupo de los señalados por Negroponete: la automatización de los procedimientos conocidos, como cálculos estáticos o estimaciones de costo, en los que sólo se aprovecha la rapidez operativa o la capacidad de almacenamiento de información, aspectos en los que el ordenador supera indudablemente a la mente humana, quedando la toma de decisiones reservada enteramente al diseñador.

Además de las posibilidades señaladas, existe otro campo de empleo del ordenador en el proyecto, no tan inmediato pero de gran valor. Scarano (1979, p. 38) lo describe así:

L'uso di un calcolatore programmato per produrre immagini può aumentare in modo considerevole la possibilità di controllo del prodotto finale, sia da parte del progettista sia da parte dell'utenza; permette, infatti, di effettuare in laboratorio prove sperimentali che altrimenti sarebbe impossibile compiere, mostrando su video il prodotto architettonico in diversi ambienti e sotto differenti angolazioni prima ancora della sua reale trasformazione in prodotto fisico compiuto.

No tan extendidas como las anteriores, se ofrecen hoy aplicaciones en esta dirección. Se trata de complejos programas, - mediante los cuales el diseñador, en diálogo con la máquina, va tomando decisiones en el terreno formal, ensayando y cambiando los espacios y formas que ha hecho aparecer en pantalla. El ordenador actúa aquí como un rápido y sofisticado - instrumento de dibujo, pero no toma decisiones por sí mismo. En este sentido y aunque más espectaculares, los programas - de dibujo ayudados por ordenador (CAD) no dejan de pertenecer al primer grupo de Negroponte.

Entender el diseño de arquitectura como un proceso automatizable en su totalidad es, hoy en día, una mera elucubración. Y así, autores que propugnan el empleo del ordenador en arquitectura, reconocen sus límites al llegar a este punto. Alexander (1966 pp. 76-77) cuyo pensamiento posterior se ha alejado de estas cuestiones, decía:

Todos aquellos problemas de creación de formas que tradicionalmente se designan como "problemas de diseño" requieren inventiva |...| , no es posible, por lo tanto, reemplazar las acciones de un diseñador diestro, por decisiones computadas mecánicamente.

O Broadbent (1974, pp. 14-15) con parecido criterio escribía:

El diseño en arquitectura no puede ser nunca un hecho de - decisión automática o automatizable |...| hay cosas en el diseño arquitectónico que no se pueden cuantificar. Cosas que son cuestión de la imaginación, valores, identidad, - sentido del lugar, etc.

Otros autores, en cambio, confían en que las investigaciones sobre la naturaleza del proceso de diseño pudieran hacernos - avanzar hacia su automatización. Así, Sevilla (1981, p. 8), - afirma:

Los esfuerzos por informatizar el proyecto de arquitectura han estado vinculados a investigaciones teóricas sobre el diseño. Estas se orientan a la racionalización de procesos y búsqueda de modelos matemáticos que permitan organizaciones complejas.

Seguí (1) está también en esta línea, tratando de distinguir - categorías en el proceso de diseño, su separación en partes, - que permitan elegir modelos matemáticos adecuados.

Con motivo del Congreso de la Unión Internacional de Arquitectos que tuvo lugar en Madrid en 1975, se realizaron encuestas para recoger opiniones de diversos países sobre esta cuestión. Las respuestas fueron diversas; desde los que, como la República Democrática Alemana, afirmaban la unidad del proceso de diseño y la imposibilidad de su descomposición para un tratamiento sistemático, hasta los que, como Estados Unidos o la Unión Soviética, confiaban en la posibilidad de emplear modelos analíticos matemáticos o gráficos como instrumentos innovadores en el proceso de diseño. En las conclusiones del Congreso (2) se lee:

De todo lo anteriormente expuesto, puede deducirse que el empleo de computadoras es idóneo para el diseño de algunos aspectos técnicos de la edificación, pero no se encuentra desarrollado en forma suficiente para resolver el proceso de diseño en el campo de la ideación arquitectónica; a este respecto, nos encontramos sin duda en el comienzo de toda una búsqueda investigadora que quizás en el futuro arroje resultados más complejos.

Nuestra posición se encuentra en un punto intermedio en este debate. Nos declaramos escépticos ante una posible automatización global del diseño. Pero ello no obsta para que no pueda -

(1) Citado por Sevilla (1981, p. 17)

(2) Congreso de la U.I.A., Actas, varios autores (1975)

aceptarse la posibilidad de crear instrumentos que ayuden a la toma de decisiones que el diseño comporta.

La reflexión sobre muchos de los problemas no cuantitativos - que el diseño encierra nos lleva a observar que su estructura interna es traducible a modelos lógico-matemáticos; arquitectos como Alexander, Xenakis o Friedman han ofrecido métodos - que se basan en esta idea.

La matemática moderna ofrece, efectivamente, instrumentos adecuados para afrontar problemas también cualitativos, no sólo - cuantitativos. El desarrollo de las nociones de teoría de conjuntos, grupos y grafos permite modelar situaciones contempladas no ya por la geometría métrica, esencialmente cuantitativa, sino también por la geometría proyectiva o la topológica.

Estos modelos matemáticos son susceptibles de programación y - tratamiento en un ordenador, utilizando ampliamente su capacidad lógica.

Entre los problemas que pueden ser abordados de este modo se - encuentran los de optimización. Y nuestro trabajo pretende ha - cer una contribución a este campo, centrándonos en la genera - ción automática de esquemas de distribución del espacio en las dos dimensiones de la planta de un edificio, a partir de condi - ciones de diversa índole impuestas por el diseñador. Es un pro - blema inicialmente topológico y finalmente métrico. Y de su re - solución depende, en cierta medida, la posibilidad de contar - con un instrumento válido para una primera aproximación a la - organización del espacio al proyectar.

II. GENERACIÓN AUTOMÁTICA DE ESQUEMAS DISTRIBUTIVOS. DESCRIPCIÓN DEL PROBLEMA.

II.1. ESTADO ACTUAL DE LA CUESTIÓN

El modo de abordar el problema de la generación automática de esquemas adimensionales de organización espacial ha sido diferente según los objetivos propuestos y las condiciones fijadas. Sin embargo, los investigadores que perseguían como resultado final esquemas dimensionados, han seguido en general un proceso dividido en dos fases sucesivas:

- Generación de esquemas adimensionales.
- Dimensionamiento.

Así han tratado el problema, entre otros, Krejcirik (1969), Cousin (1970), Shaviv y Gali (1974), Mitchell, Steadman y Ligett (1976), Mitchell (1977), Korf (1977), March y Earl (1977 y 1979) y Scarano (1979). Será, pues, en función de estas dos etapas como analizaremos los distintos métodos de solución propuestos a partir de los años sesenta.

II.1.1. Generación de esquemas adimensionales

Existen dos planteamientos iniciales de este problema, que luego se traducen en una amplia gama de métodos de resolución: el

primero lo enfoca como un problema de optimización, buscando - la mejor disposición posible de los espacios que van a componer la planta; el segundo planteamiento entiende el problema - como combinatorial, y busca todas las disposiciones posibles - que cumplan unas condiciones dadas.

Son relativamente muchos los investigadores que se han inclinado por resolver la generación de esquemas adimensionales como un problema de optimización (1). Ya en los primeros trabajos - en este campo -Levin (1964), Eastman (1970), Shaviv y Gali - (1974)- el criterio de "óptimo" iba fundamentalmente dirigido a la disposición y distancias entre los distintos ambientes, y se definía por medio de una "función objetivo" en la que se hacían intervenir factores tales como la minimización de recorridos entre dos actividades -Levin (1964)- o la condición de que dos espacios estuviesen uno junto a otro -Ritzman (1972)- o - una combinación de ambos -Shaviv y Gali (1974)-. Las distintas unidades espaciales que componen el conjunto están además, sujetas a unas limitaciones que, fundamentalmente, son de adyacencia, orientación, acceso, visibilidad y distancia -Eastman (1972)-.

El problema es planteado como combinatorial en los años setenta por el LUBFS de la Escuela de Arquitectura de Cambridge (Inglaterra) y por la Escuela de Arquitectura y Planeamiento Urbano de la Universidad de California (USA). El número de soluciones posibles viene restringido por una serie de requisitos -

(1) "Il problema di localizzare in modo ottimale ambienti è nato nella cultura anglosassone all'interno di un discorso settoriale di programmazione e pianificazione della progettazione, al fine di produrre un sistema di organizzazione spaziale che, in termini di costo, minimizzi la distanza, intesta questa come tempo di spostamento e - quindi come lavoro, anch'esso da minimizzare". Scarano (1979, prefacio).

que han de cumplir los elementos espaciales, limitaciones que cuando existen en elevado número pueden llegar a alterar incluso la índole combinatorial del problema.

Han sido numerosos los intentos y variados los algoritmos propuestos en la última década para generar, las más de las veces, esquemas rectangulares formados por espacios también rectangulares. Casi todos ellos se han basado en la teoría de grafos: en efecto, desde los primeros trabajos -Cousin (1970), March y Steadman (1971)- se parte de la idea de que las condiciones de adyacencia de los espacios a ordenar pueden expresarse mediante un grafo, cuyo grafo dual será precisamente un esquema que cumpla las condiciones fijadas.

Dados los requisitos exigidos a los espacios componentes, es preciso comprobar si el grafo resultante es conexo y planar, - es decir si los distintos elementos forman un solo conjunto de espacios interconectados, y si las condiciones impuestas tienen solución en dos dimensiones. Existen procedimientos que comprueban si el grafo tiene esas características; en este sentido cabe destacar los trabajos de Kuratowski (1930), Tutte (1962), Fisher y Wing (1966); Bruno, Steiglitz y Weinberg (1969); Hopcroft y Tarjan (1973, a y 1974), Read (1979), etc.

Si, además, se impone la limitación de rectangularidad del conjunto, a los ya citados "tests" de conectividad y planaridad - habrá que añadir un test de rectangularidad -Mitchell, Steadman y Ligett (1976), Scarano (1979)-. Las operaciones de reflexión y rotación no alteran la naturaleza de las soluciones, - salvo si las condiciones de orientación de los espacios se toman como requisitos, como se hace con frecuencia.

Ha habido algunos intentos de superar la limitación de la rectangularidad de los espacios: Shaviv y Gali (1974) introduciendo rectángulos ficticios, o Korf (1977) con mayores pretensiones teóricas.

En cuanto a los métodos de solución, Nugent (2) distingue entre técnicas de "construcción" y técnicas de "mejora". Las primeras parten del estado nulo y, mediante introducción de razones de conveniencia de localización, van llegando a una solución. En las técnicas de mejora se parte de una solución inicial que va optimizándose en sucesivas evaluaciones.

Mitchell (1977) clasifica los métodos de solución, de más "débiles" o generales a más "fuertes" o específicos, del siguiente modo:

- Procedimientos de generación y comprobación: considerados como poco eficientes, consisten en una generación secuencial de elementos, hasta que aparece un elemento del conjunto "objetivo", generación que puede ser aleatoria o exhaustiva.
- Procedimientos de mejora: recomendables cuando hay una buena solución de partida, coinciden con la clasificación de Nugent. Establecida una "función objetivo" a optimizar, se introducen en la solución inicial sucesivos cambios, que son adoptados si aumenta el valor de la función. La estrategia de cambio puede ser aleatoria o de búsqueda de dirección de mayor mejora.
- Procedimientos heurísticos: se trabaja sobre un cierto conocimiento previo de la estructura del problema, a los que se incorporan los conocimientos obtenidos tras cada operación de cambio, para seleccionar secuencias de operaciones a realizar. Existen diferentes técnicas de evaluación de las operaciones efectuadas y determinación de las operaciones a realizar o planificación de los procesos de búsqueda y "feed back".

(2) Citado en Portlock y Whitehead (1974).

II.1.2. Dimensionamiento

La mayoría de los investigadores tratan la fase de dimensionamiento de esquemas como un problema de optimización: Krejcirik (1969), Cousin (1970), Mitchell, Steadman y Ligget (1976), Mitchell (1977), Gero (1977), Earl y March (1979), o Scarano (1979) están entre ellos.

Para Gero (1977) el diseño es un problema de optimización, que necesita determinar los valores actuales de las variables de una función objetivo en función de los valores a alcanzar en el futuro, a diferencia de los problemas de simulación, que interfieren los valores futuros de los actuales.

La formulación general de un problema de optimización viene dada por el establecimiento de una función objetivo y unas restricciones a cumplir por las variables que en ella intervienen. En los casos que nos ocupan, estas variables son esencialmente la longitud y anchura de cada uno de los espacios, con límites mínimos o máximos, y el contorno del conjunto. Casi todos los investigadores trabajan con esquemas rectangulares, y contorno asimismo rectangular, a lo sumo, introducen el concepto de "rectángulo ficticio" para poder emplear también espacios con planta en T, en L, en U, etc., y contorno general de similares características.

Para el dimensionamiento de esquemas se han establecido diferentes tipos de funciones-objetivo: los más usuales maximizan o minimizan el contorno general, su longitud o su anchura, su proporción o su superficie. Unas funciones objetivo serán, pues, lineales, otras no-lineales.

En cuanto a las restricciones de las variables, un primer grupo es el ya citado, referido a la longitud o anchura de los espacios, su superficie, etc., que se materializan en acotación

de estas variables. Un segundo tipo engloba los requisitos de adyacencia o de comunicación entre espacios, y existe un tercer grupo, por último, que incluye restricciones derivadas de la propia naturaleza del esquema adimensional, y de las relaciones entre sus espacios. Todas estas limitaciones pueden formularse, unas en forma lineal, otras en forma no-lineal.

Para expresar estas restricciones, muchos investigadores han reducido el modelo adimensional a dos grafos orientados, uno "vertical" y otro "horizontal", a los que se aplican las leyes de Kirchhoff para redes eléctricas.

La naturaleza, lineal o no-lineal, de las funciones-objetivo y los valores, reales o enteros, que pueden adoptar las variables, determinan el método de resolución a emplear en el dimensionamiento. A continuación se enumeran y describen los más comunes:

- Programación lineal.

Es una técnica muy poderosa, que garantiza el alcanzar una solución óptima con gran eficiencia, utilizando con gran frecuencia el algoritmo "simplex". Lógicamente, tanto las restricciones de las variables como la función-objetivo han de ser traducibles a expresiones lineales. Cuando se emplea una trama modular, las variables solo asumen valores enteros.

- Programación cuadrática.

Técnica usada con menor frecuencia, permite que, siendo lineales las restricciones de las variables, la función-objetivo pueda adoptar forma cuadrática.

- Programación no-lineal.

Se emplea esta técnica cuando tanto las restricciones de las

variables como la función-objetivo pueden adoptar forma no-lineal. Se han desarrollado varios métodos con esta técnica: algunos son rudimentarios y aplicables solo en ciertos casos -por ejemplo, los métodos de linealización-; otros utilizan algoritmos más sofisticados y de mayor aplicabilidad.

- Programación dinámica.

Se llega a la solución final a través de una secuencia de decisiones. Con ella pueden resolverse problemas tanto lineales como no-lineales.

II.2. ALTERNATIVAS E HIPÓTESIS DE TRABAJO

El punto de arranque de nuestra investigación para desarrollar un procedimiento automático de generación de esquemas distributivos bidimensionales fue la propuesta de Mitchell (1977), que utiliza la "representación mediante el grafo dual". Este autor subdivide el proceso en las siguientes etapas:

- 1.- Definición, en forma matricial, de las condiciones de adyacencia entre los distintos espacios componentes.
- 2.- Comprobación de la planaridad del grafo correspondiente a tales condiciones de adyacencia.
- 3.- Trazado del grafo planar, sin cruce de aristas.
- 4.- Construcción del correspondiente grafo dual, una representación en planta que cumple los requisitos iniciales.
- 5.- Introducción de nuevas limitaciones de formas, dimensiones y localización de los espacios que, junto con las de adyacencia, determinarán el esquema final.

Mitchell considera sin embargo su propuesta de difícil implementación en un ordenador. Siguiendo la indicación del mismo y otros autores, hemos estimado conveniente en primer lugar subdividir el procedimiento en las dos fases señaladas en II.1: en la primera se obtienen esquemas adimensionales estableciendo las condiciones de adyacencia y orientación de los espacios componentes, y limitando además la forma de los esquemas de modo que tanto los componentes como el conjunto sean rectangulares; como vimos en II.1.1, este criterio simplificador se sigue en la mayor parte de los trabajos realizados hasta la fecha. En la segunda fase se hacen intervenir restricciones métrico-geométricas de los espacios para conseguir un esquema dimensionado.

La totalidad del procedimiento se ha programado en lenguaje Fortran IV para un ordenador Hewlett Packard serie 1000; los programas operan de forma interactiva con arreglo a los algoritmos que se desarrollan en el capítulo siguiente.

II.2.1. Generación de los esquemas adimensionales

Las cuatro primeras etapas del procedimiento de Mitchell obedecen a una formulación de tipo combinatorial: dada una serie de espacios, se van a generar todas las localizaciones - "posibles" de los mismos -test de planaridad- que cumplan con los requisitos de adyacencia inicialmente impuestos. El número de soluciones dependerá lógicamente de la cantidad de restricciones introducidas.

La orientación de nuestro trabajo es distinta. Pretendemos llegar a definir las condiciones necesarias para que la solución sea única; por ello, el problema no podrá tener un planteamiento combinatorial.

Pero imponer indiscriminadamente condiciones para llegar a una solución única haría cerrado e inflexible el proceso, porque es difícil, sobre todo con un número elevado de espacios componentes, saber "a priori" que las condiciones establecidas conducen a una única solución posible. Aumentando las condiciones disminuyen las posibilidades de existencia de solución. De modo que se convertiría en un hecho casual -de probabilidad mínima- el llegar a una solución a partir de condiciones establecidas en bloque.

Por ello hemos optado, aprovechando la interactividad del programa, por una introducción gradual de requisitos, comenzando con los que se estiman imprescindibles o más importantes, y continuando con otros que, al final, conduzcan a la solución.

Nuestro procedimiento de generación de esquemas adimensionales se divide en las siguientes etapas:

1. Condiciones iniciales: definición de las adyacencias y orientaciones de los espacios componentes. Construcción del grafo inicial.
2. Test de conectividad: si el grafo inicial no es conexo, introducción de nuevas condiciones de adyacencia para lograr que lo sea.
3. Primer test de planaridad: si el grafo no es planar, sustitución o eliminación de las condiciones que lo impiden.
4. Test de triconectividad: si el grafo no es triconexo, trazado de sus componentes triconexas e introducción de condiciones que las enlacen suficientemente.
5. Segundo test de planaridad, para comprobar si las condiciones introducidas siguen permitiendo una solución planar.
6. Test de rectangularidad.
7. Triangulación del grafo.
8. Transformación de coordenadas de los vértices del grafo - como paso previo a la construcción del grafo dual.
9. Construcción del grafo dual, que es el esquema de distribución buscado, cumpliendo todas las condiciones impuestas.

Este proceso que se detalla ampliamente en el capítulo siguiente, no es lineal: permite un "feed back" si el resultado obtenido no es satisfactorio. Cuando las modificaciones - que ahora introdujéramos, que pueden incluso afectar a las condiciones iniciales, no fuesen tales que variara la estructu

tura del grafo de modo esencial, bastaría repetir las etapas 7, 8 y 9. En caso contrario se trataría de un nuevo problema, cuya solución comenzaría desde el principio.

Como puede verse, el proceso es altamente interactivo: el usuario decide en todas las etapas, añadiendo o alterando las condiciones que juzga oportunas, y determina cuándo el proceso ha terminado, al aceptar la solución generada.

II.2.2. Dimensionamiento

La quinta etapa en el proceso de Mitchell supone la introducción de requisitos dimensionales y formales en el esquema adimensional, sin atender a la formulación del mismo. Pero la mayoría de los investigadores han entendido el dimensionamiento de esquemas como un problema de "optimización" y, por consiguiente, han aplicado como métodos de resolución los derivados de la programación. Nosotros consideramos igualmente la "optimización" como técnica idónea para resolver el problema de dimensionamiento, y emplearemos la programación lineal y no lineal como métodos de resolución adecuados a esa formulación.

No aplicaremos, sin embargo, la programación dinámica, porque su utilidad depende en gran medida de la habilidad del usuario para descomponer la estructura del problema.

El procedimiento que se propone para el dimensionamiento de esquemas puede subdividirse en las siguientes etapas:

1. Sustitución del esquema adimensional por dos grafos dirigidos, uno "horizontal" y otro "vertical"
2. Definición de las condiciones de accesibilidad entre locales.

-
3. Condiciones métrico-geométricas que deben cumplir los espacios componentes y el contorno.
 4. Expresiones, lineales o no lineales, correspondientes a las condiciones anteriores.
 5. Determinación de la función objetivo.
 6. Resolución del problema por programación lineal o no lineal, según su naturaleza.

El proceso termina con el trazado del esquema dimensionado. El contorno y todos los espacios componentes tendrán forma rectangular.

III. ALGORITMOS PARA LA GENERACIÓN DE ESQUEMAS DISTRIBUTIVOS

En el capítulo anterior quedan someramente descritas las etapas a seguir en la generación de esquemas adimensionales y su posterior dimensionamiento. Como ya se dijo, estos dos procesos, independientes pero complementarios, resuelven el problema del trazado de distribuciones en planta. En el presente capítulo entraremos a detallar las operaciones y algoritmos que desarrolla cada etapa.

III.1. GENERACIÓN DE ESQUEMAS ADIMENSIONALES

III.1.1. Fijación de condiciones iniciales

Las condiciones iniciales del problema son el conjunto de espacios o locales intervinientes en el esquema y sus relaciones de adyacencia y orientación. Como se irá viendo, es conveniente en este punto fijar sólo las relaciones más importantes, dejando las demás para decisiones posteriores.

Estos datos constituyen un grafo simple (sin bucles ni aristas múltiples): los locales serán sus vértices y las relaciones entre ellos sus aristas.

Este grafo abstracto, G_1 , queda definido por una matriz de incidencia, $B (b_{ij})$, cuyas dimensiones son el número de vértices $|V|$ y el de aristas $|A|$. Sus términos serán: $b_{ij} = 0$ cuando la arista i no incide en el vértice j , y $b_{ij} = 1$ en caso contrario.

Los cuatro primeros vértices representan siempre a los puntos cardinales (vértices exteriores), en sentido antihorario: 1, norte; 2, oeste; 3, sur, y 4, este. Los diferentes locales se enumeran del 5 en adelante.

Igualmente, las cuatro primeras aristas representan las relaciones naturales de adyacencia entre puntos cardinales: 1, norte-oeste; 2, oeste-sur; 3, sur-este y, 4, este-norte. Las relaciones entre locales se enumeran del 5 en adelante.

De la matriz de incidencia se obtiene inmediatamente la de adyacencia, $A (a_{ij})$, de dimensión $(|V|, |V|)$, cuyos términos son: $a_{ij} = 0$ cuando los vértices i y j no están ligados por arista, y $a_{ij} = 1$ en caso contrario.

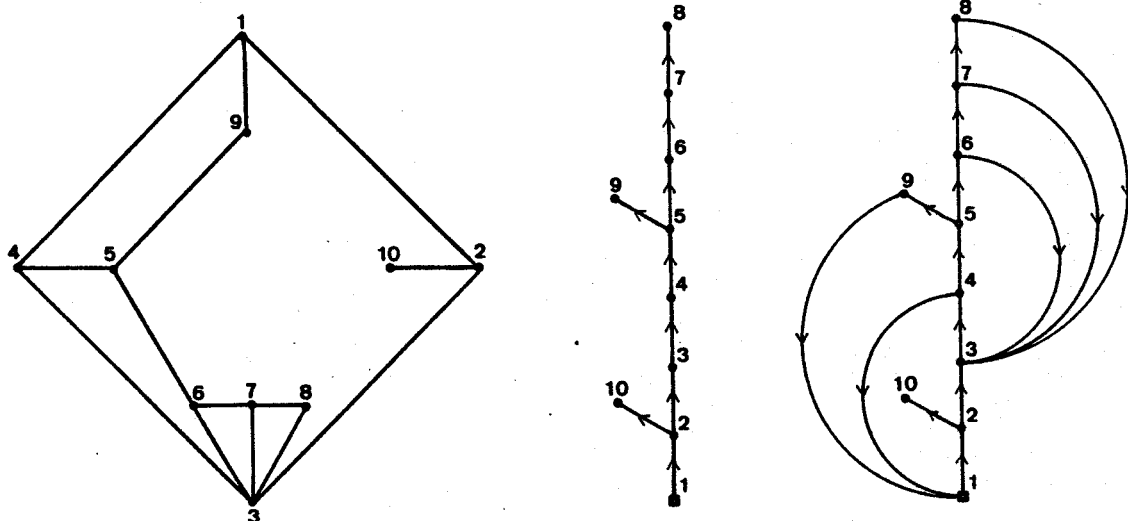


Figura 1

III.1.2. Comprobación de conectividad

Condición previa a la ulterior obtención del dual de un grafo es la comprobación de que éste es conexo y planar.

El algoritmo elaborado como "test de conectividad" se basa - en los estudios de Hopcroft y Tarjan (1973,a). Se utiliza la técnica denominada "depth first search" que, partiendo de la matriz de incidencia, transforma el grafo en una "palmera" - (véase apéndice).

La técnica consiste, en primer lugar, en construir un árbol dirigido a partir de un vértice inicial, cuyos arcos se corresponden con aristas del grafo de partida. Las aristas restantes, "ramas", se añaden al árbol dirigidas en sentido contrario a los arcos, es decir, su origen será el vértice más alejado del vértice inicial. Figura 1.

Cuando la palmera así obtenida no contiene a todos los vértices del grafo, éste será inconexo, y tendrá por lo tanto más de una componente conexa.

El algoritmo procede a "marcar" los vértices accesibles desde el que tomó como inicial. Si el grafo es inconexo, irá construyendo nuevas palmeras tomando como vértice inicial alguno de los no marcados, y determinando así todas las componentes conexas que existan.

Llegado a este punto, el programa informa del estado de conectividad del grafo, y describe las componentes en caso de no conectividad. El operador debe decidir qué nuevas relaciones introduce para conectar entre sí las componentes.

Con esta intervención, el grafo de partida G_1 , es ahora un grafo conexo, G_2 . Si G_1 era conexo, G_2 es idéntico a él; si no lo era, G_2 tiene el mismo número de vértices, pero más aristas. Las matrices de incidencia y adyacencia han quedado modificadas según esta nueva estructura conexa.

III.1.3. Primera comprobación de planaridad

La comprobación de planaridad se realiza mediante algoritmos desarrollados a partir de los trabajos de Hopcroft y Tarjan (1974) (1), que son una versión de un método originalmente propuesto por Auslander y Parter y formulados posteriormente por Goldstein.

(1) El análisis de una serie de propiedades del grafo: conectividad, bi-conectividad, triconectividad y planaridad puede abordarse desde una misma óptica. Así lo han tratado estos autores. Su eficiencia frente a otros métodos con idénticos fines y, sobre todo, su adaptabilidad al método que desarrollamos, ha resuelto que sea recogido por este trabajo.

Como en el test de conectividad, se construye ahora una palmera a partir del grafo G_2 . A continuación se aplica el procedimiento de Auslander, Parter y Goldstein, consistente en identificar un circuito que comience y termine en el vértice inicial. Al suprimir este circuito, el grafo queda reducido a unos "segmentos" conexos que inciden en ese circuito. El test de planaridad consistirá en comprobar si es posible una representación plana del circuito base y de los segmentos sin cruce de aristas.

No obstante, una aplicación eficiente de la serie de algoritmos que se necesitan desarrollar para comprobar la planaridad exige un procedimiento más completo. Comprendería: una primera comprobación de la planaridad por la aplicación de la fórmula de Euler, una descomposición del grafo en sus componentes biconexas y, por último, la comprobación de la planaridad de cada una de las componentes biconexas.

Describamos a continuación cada una de estas operaciones:

1.- Aplicación de la fórmula de Euler.

Para que un grafo sea planar, es condición necesaria que se cumpla la siguiente expresión (2): $|A| \leq 3 \cdot |V| - 6$

2.- Descomposición del grafo en sus componentes biconexas.

'Un grafo es planar si y sólo si sus componentes biconexas son planares', Berge (1970).

Las componentes biconexas se identifican con comodidad en la palmera obtenida de G_2 . Los vértices de separación de componentes serán comunes a dos subpalmeras no unidas entre sí por ramas. Figura 2.

(2) Para $|V| \geq 3$

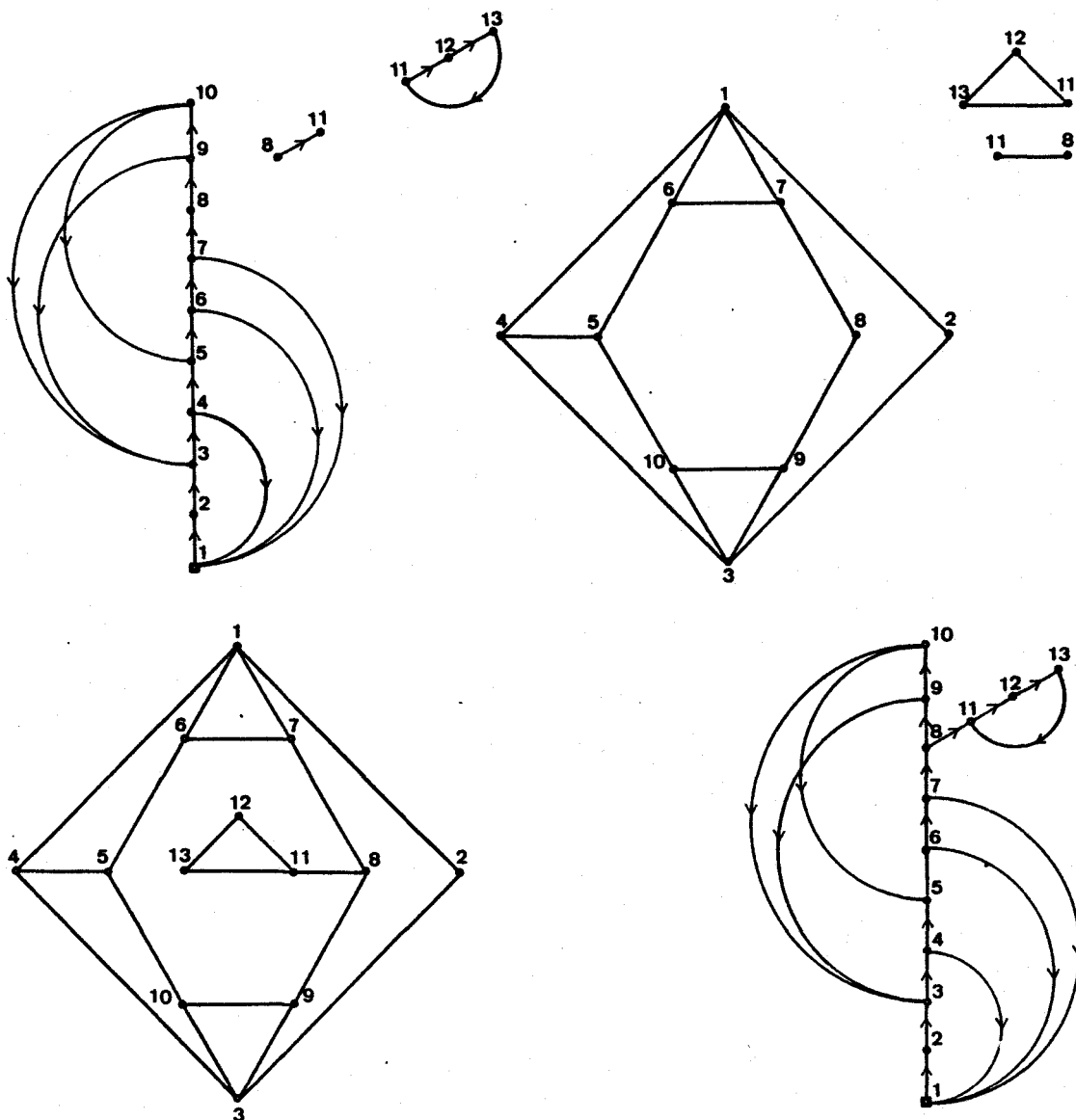


Figura 2

Si el grafo G_2 (conexo) no contiene ningún vértice de separación será además biconexo. Si G_2 no es biconexo lo subdividiremos en sus componentes biconexas. Para ello efectuaremos un nuevo recorrido sobre la palmera con arreglo a la técnica citada. Cada vez que alcanzamos un vértice de separación comienza una nueva componente biconexa. El vértice de separación para dos componentes biconexas estará incluido en ambas. Figura 2.

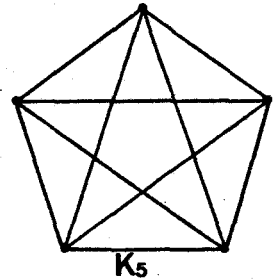
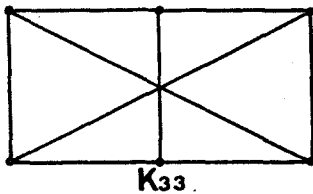


Figura 3

3.- Comprobación de la planaridad de cada una de las componentes biconexas.

Antes de efectuar la comprobación conforme a los procedimientos descritos, se puede realizar un control, rápido y cómodo, a la componente biconexa que posibilita el teorema de Kuratowski -Kuratowski (1930)- : 'Un grafo es planar si no contiene a los subgrafos $K_{3,3}$ y K_5 '. Figura 3.

El grafo nominado $K_{3,3}$ contiene seis vértices y nueve aristas y el grafo K_5 posee cinco vértices y diez aristas. Cualquier componente biconexa que tenga menor número de vértices o menor número de aristas será por tanto planar y no habrá que aplicar sobre ella ninguna otra comprobación de planaridad.

A la componente biconexa que no cumpla estas características se le aplicará el procedimiento general.

Podríamos describirlo con mayor detalle del siguiente modo:

- Aplicamos una "depth first search" sobre la componente biconexa que se analiza transformándola en una palmera.

- Volvemos a aplicar esta misma técnica para obtener todos los caminos disjuntos sobre esta palmera. La construcción de caminos se caracteriza porque cuando se alcanza una rama de la palmera, ésta constituye la última -y a veces la única-

arista orientada del presenta camino. Empezaría entonces un nuevo camino, con un nuevo vértice inicial en el vértice origen de la rama.

El primer camino que se genera comienza y termina en el vértice inicial de la componente biconexa. Sería un circuito; - lo llamaremos circuito original.

- Suprimimos el circuito original. Cada uno de los segmentos resultantes constará, o bien de una sólo rama de la palmera, (W,V) , o de un arco del árbol, (V,W) , más un subárbol de origen W y todas las ramas que salen de él.

- Analizamos cada uno de los segmentos en el orden inverso a como han sido generados. Por el teorema de la Curva de Jordan, un segmento puede ser representado en el plano bien por el exterior, o por el interior del circuito original, sin - que se produzcan cruces de sus aristas.

La localización en el plano de un nuevo segmento puede llevar consigo que algunos segmentos, ya situados, deban ser movidos del interior al exterior del circuito original, o viceversa.

- Si se ha podido ubicar todos y cada uno de los segmentos, la componente biconexa es planar. Basta que un segmento no - pueda ser situado para que dicha componente biconexa no sea planar. En éste último caso, tendríamos que sustituir un arco o rama de este segmento por otro distinto para poder continuar el proceso.

El proceso completo se recoge en la Figura 4.

La operación de generación de caminos disjuntos y representación o "empotramiento" de los segmentos pueden ser realizau

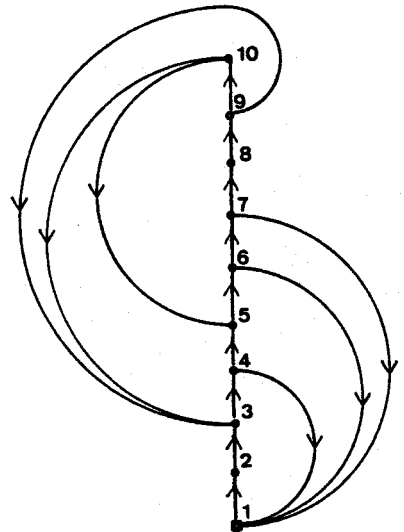
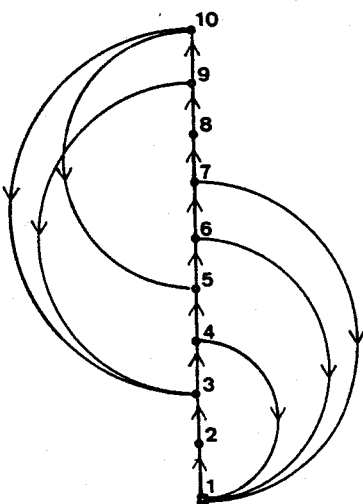
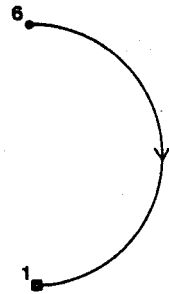
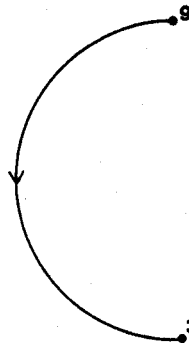
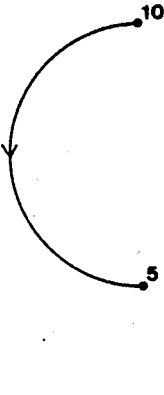
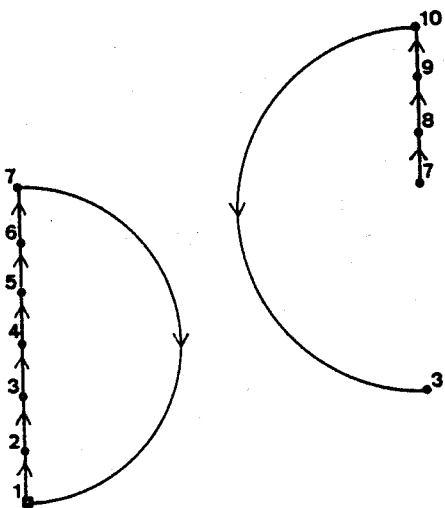
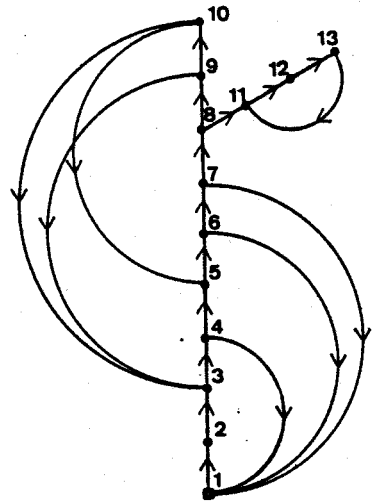
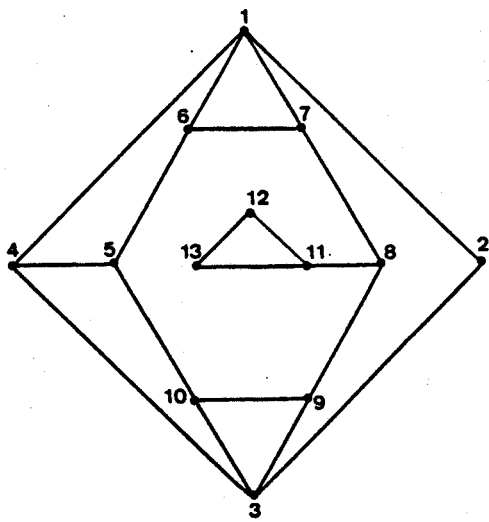


Figura 4

das simultáneamente. Así ha sido propuesto por Hopcroft y - Tarjan (1974) y así ha sido recogida en este trabajo.

Después de esta etapa el grafo, G_3 , será geométrico conexo y planar.

III.1.4. Análisis de la triconectividad

El procedimiento que estamos desarrollando limitará progresivamente el número de soluciones posibles por la introducción de nuevos requisitos de adyacencia y orientación. Pero la - adición de nuevas restricciones ha de realizarse a partir de trazados geométricos planos del grafo o de sus subgrafos. De este modo, la inclusión de nuevas relaciones puede hacerse - de forma racional para no producir alteración de la planaridad.

El teorema de Whithney nos dice: 'Un grafo ha de ser necesariamente triconexo para que tenga una representación única - en el plano para cada una de sus caras o regiones'.

Así pues, si el grafo no es triconexo, añadiremos una serie de aristas al grafo para que tenga esta propiedad.

Un trazado en el plano claro y sencillo del grafo, ya triconexo, facilitaría la introducción de nuevos requisitos de adyacencia y orientación, sin necesidad de tener que efectuar la comprobación de la planaridad. Esto nos permitiría restringir paulatinamente el problema y llegar a una solución: una representación en planta.

Pero la inclusión de nuevas aristas para hacer el grafo triconexo, requiere a su vez de trazados geométricos planos con las características citadas. La primera operación que debemos realizar, por tanto, será la determinación de las componentes triconexas del grafo. Esto se hará distinguiendo las

componentes triconexas de cada una de las componentes biconexas que se han obtenido en la etapa anterior, previo al estudio de la planaridad (3).

El procedimiento que se ha seguido se basa, igualmente, en las investigaciones de Hopcroft y Tarjan (1973,b). Comprendería las siguientes operaciones:

- Aplicación de una "depth first search" sobre la componente biconexa para transformarla en una palmera.
- Utilización de nuevo de esta técnica para obtener todos los caminos disjuntos en dicha componente.
- Determinación del circuito original y de cada uno de los segmentos.
- Exploración de los segmentos en orden decreciente a su generación para la obtención de los pares de separación.
- Clasificación de los arcos y ramas de la palmera en clases de separación correspondientes a cada uno de los pares de separación.
- Elaboración de las componentes escindidas.
- Determinación de las componentes triconexas a partir de las escindidas.

Hay muchas formas de escindir un grafo y determinar sus componentes escindidas. Éstas, por tanto, no tienen porqué ser necesariamente únicas. Responderán a unos de estos tres tipos. Figura 5.

(3) Sólo tendrá sentido esta descomposición cuando la componente biconexa tenga más de tres vértices.

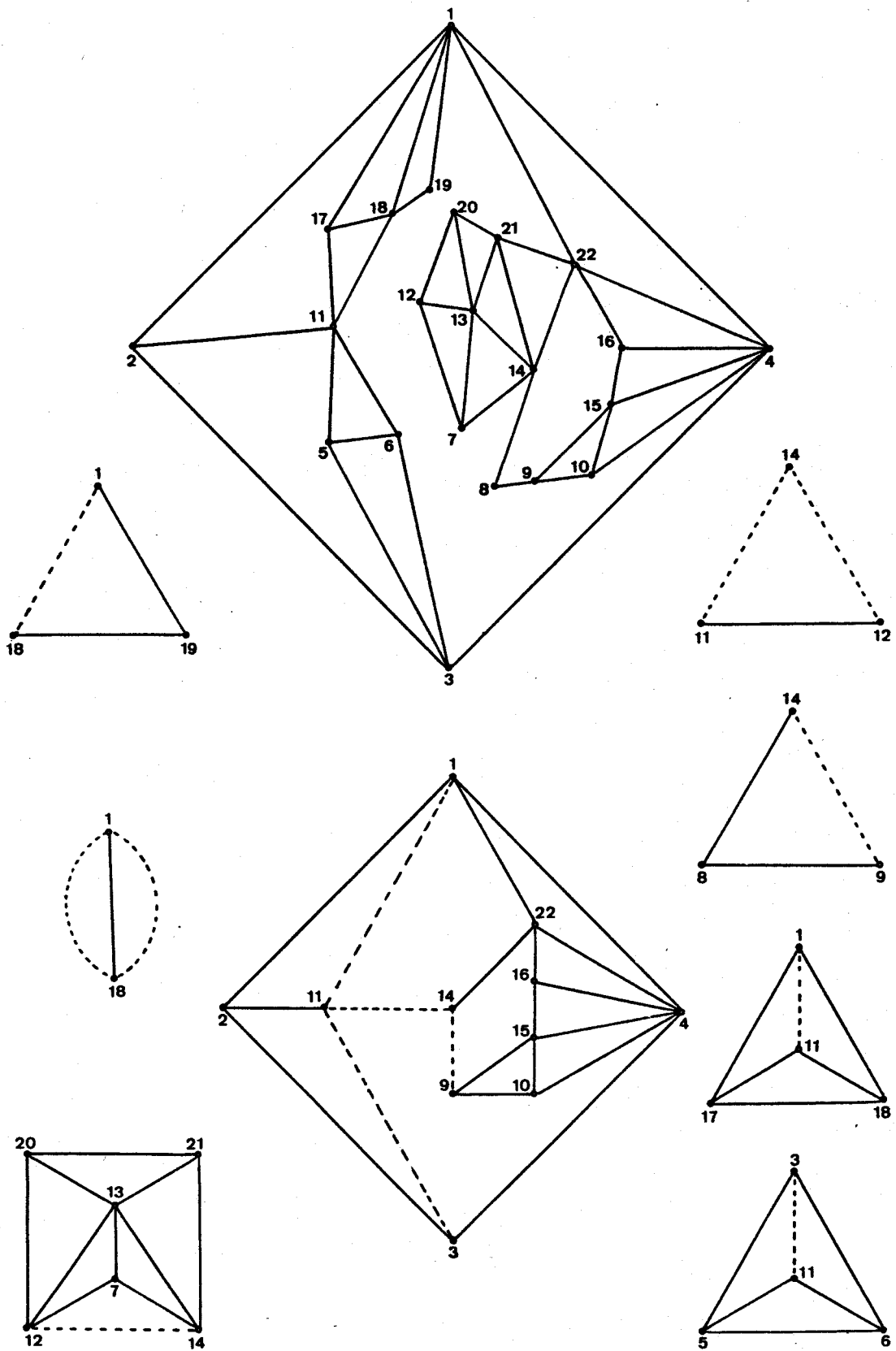


Figura 5

-
- Bucles triples: B_3 ($\{a,b\}, \{(a,b), (a,b), (a,b)\}$)
 - Triángulos: T ($\{a,b,c\}, \{(a,b), (b,c), (c,a)\}$)
 - Componentes triconexas.

Una vez que se han obtenido las componentes triconexas para cada componente, podemos fijar un método de adición progresiva de restricciones en dos estados sucesivos.

Primer estado: Adición de las aristas necesarias para convertir cada componente biconexa en un subgrafo triconexo.

Segundo estado: Inclusión de aristas para pasar de varios subgrafos triconexos articulados a un sólo grafo triconexo.

Para la representación en el plano usamos un algoritmo formulado a partir de otro que propone Tutte (1963) para grafos triconexos. Se basa en el teorema de Fary: 'Un grafo planar puede ser trazado en el plano de forma que sus aristas sean segmentos de línea recta y no se crucen'.

El algoritmo completo realiza las siguientes operaciones sobre una componente (grafo o subgrafo) triconexa:

- Genera un circuito de forma que dos vértices del mismo, no contiguos, no estén enlazados por ninguna arista. Si en la componente analizada figuran algunos vértices exteriores, éstos pertenecerán a dicho circuito. En el caso particular de que incluya a los cuatro puntos cardinales, el circuito estará formado sólo por estos cuatro vértices y las aristas que los unen.
- Se disponen los vértices de dicho circuito según los vértices de un polígono regular con el mismo número de lados que aristas tiene el circuito.

- Se elaboran las ecuaciones que permiten obtener las coordenadas de los restantes vértices. Para ello, se impone que cada vértice ha de localizarse en el centroide de sus vértices adyacentes.

Para un vértice V_i :

$$\rho_i x_i = \sum_{j=1}^{|V|} a_{ij} x_j$$

$$\rho_i y_i = \sum_{j=1}^{|V|} a_{ij} y_j$$

siendo:

$A (a_{ij})$: matriz de adyacencia de la componente

ρ_i : valencia del vértice V_i

- Se resuelve, por último, el sistema de ecuaciones lineales correspondiente por el método de eliminación de Gauss. Se obtiene, pues, una representación plana de la componente con la propiedad adicional de que todas las regiones finitas, en las que queda subdividida, tienen forma convexa. A partir de los distintos trazados planos podemos efectuar la introducción de las aristas en los dos estados citados:

Primer estado:

Si alguna componente triconexa posee una arista virtual, la

primera operación que hay que realizar es transformarla en real (4). Las aristas de cada componente biconexa podemos entonces clasificarlas en uno de estos tres tipos: aristas pertenecientes a una componente triconexa, aristas que enlazan dos componentes triconexas, y aristas no enmarcadas en los tipos anteriores. Figura 5.

Para fusionar dos componentes triconexas en una sola hemos de introducir aristas que las enlacen. Su número estará en función del grado de relación que haya entre ambas. Pueden presentarse los siguientes casos.

- 1.- Hay sólo una arista que une a las dos componentes triconexas.
- 2.- Existen dos aristas de enlace y no tienen vértices comunes.
- 3.- Dos o más aristas ligan a las dos componentes pero todas tienen un vértice en común.
- 4.- Tres o más aristas relacionan a dos componentes y todas menos una tienen un vértice en común.

La adición de aristas sería distinta en cada una de ellas. - Se haría así (5):

- Caso 1: Se incluirán dos aristas más que unan a las dos componentes. No podrán tener vértices comunes entre ellas, ni con la arista de enlace precedente. Figura 6.

(4) Esta operación no supone limitación alguna a la generalidad, pues, como veremos posteriormente, se presenta a lo largo del proceso la oportunidad de sustituir una relación introducida por otra distinta.

(5) Su justificación está en las definiciones de vértices de separación y componentes triconexas.

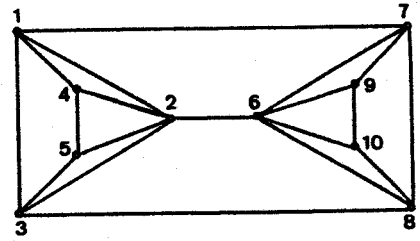
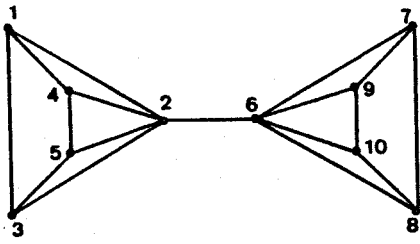


Figura 6

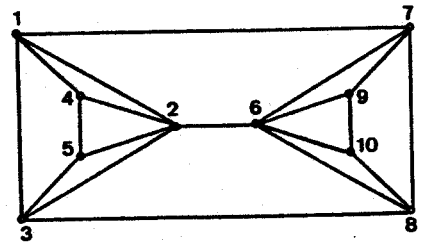
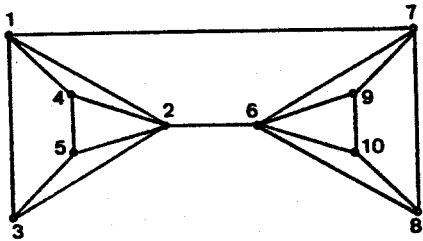


Figura 7

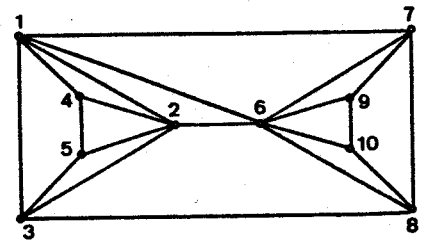
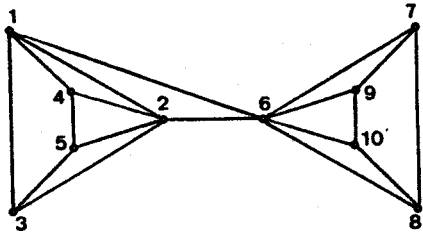


Figura 8

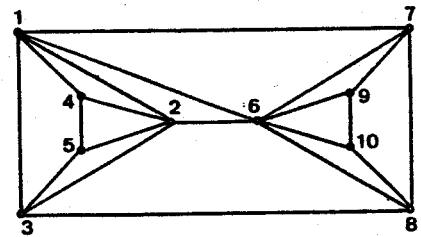
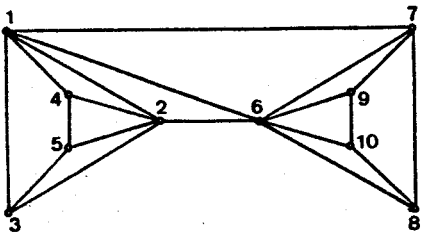


Figura 9

- Caso 2: Se añadirá una nueva arista entre las dos componentes. No podrá poseer por extremos ninguno de los vértices sobre los que inciden las restantes aristas de enlace. Figura 7.

- Caso 3: Se introducirán dos nuevas aristas que relacionen a las dos componentes. Sus vértices extremos serán distintos, y distintos también al vértice común de las dos aristas originales. Figura 8.

- Caso 4: Se tendrá que añadir una nueva arista entre componentes. No podrán incidir sobre el vértice común de las aristas en enlace ni a ninguno de los extremos de la única arista de enlace que no incide sobre el vértice común. Figura 9.

Después de ser fusionadas dos componentes triconexas, obtenemos, una nueva componente triconexa. Aplicándole las operaciones descritas podría integrarse con otra componente triconexa. La repetición sucesiva llevaría a una componente triconexa única. No obstante, puede suceder que no incluya a todos los vértices y aristas de la componente biconexa. En este caso subdividiremos las aristas restantes en dos clases: aristas aisladas y triángulos. A éstas las podemos considerar componentes triconexas "contraídas" y aplicar operaciones similares a las descritas con anterioridad.

La adición de aristas, en todo caso, se hará conforme a las posibilidades que ofrece la aplicación del teorema de la curva de Jordan. Figuras 5 y 10.

Segundo estado:

Una vez convertidas cada una de las componentes biconexas en subgrafo triconexo, se trazarían en el plano conforme al procedimiento descrito. A partir de estas representaciones se -

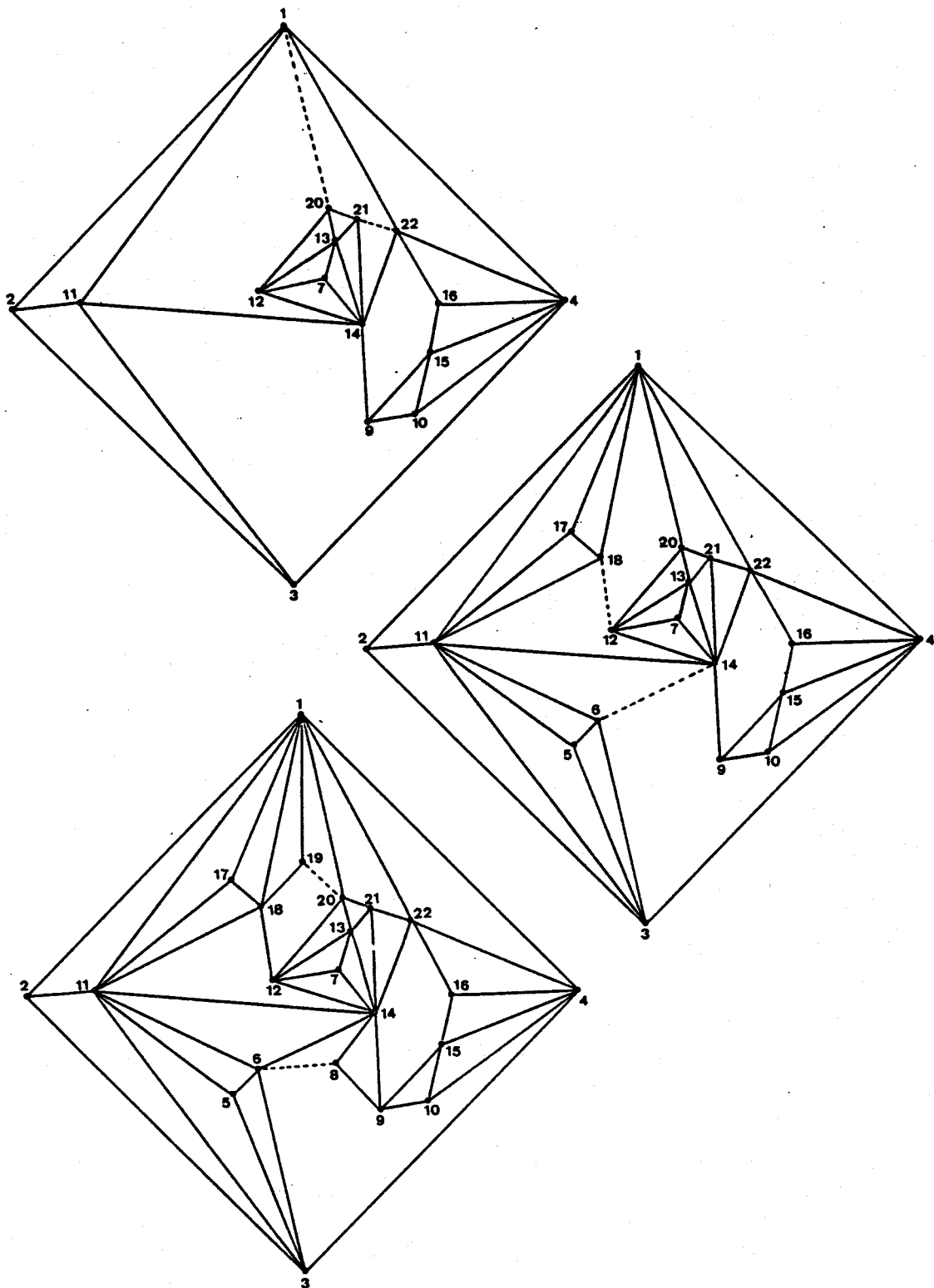


Figura 10

introducen otras nuevas limitaciones para transformar el grafo en triconexo.

Si queremos englobar dos subgrafos que tienen en común un vértice de separación, hemos de añadir dos aristas cualesquiera que los enlacen y no alteren la planaridad. Estas aristas no podrán tener un vértice extremo común ni incidir en el vértice de separación.

Las componentes biconexas arista aislada o triángulo pueden ser consideradas como subgrafos triconexos "contraídos" y tratarse de igual modo.

La característica especial de los grafos que manejamos, de tener siempre un circuito exterior formado por los cuatro puntos cardinales y las adyacencias entre ellos, hace que sea apropiado realizar el ensamble de las componentes biconexas, convertidas en subgrafos triconexos, a partir precisamente de ese circuito exterior.

El grafo, G_4 , después de esta etapa será pues triconexo.

III.1.5. Segunda comprobación de la planaridad

Como para hacer triconexo el grafo ha sido necesario añadirle aristas, hay que volverlo a someter al test de planaridad (cfr. III.1.3) por si se ha errado en la introducción de alguna de ellas. En el caso de que así sea, tendremos que sustituir aristas pertenecientes a segmentos no "empotrados" por otros, observando, de nuevo, los trazados planos conseguidos en la etapa anterior.

El grafo geométrico G_5 , triconexo y planar, podrá ser entonces representado en su totalidad usando el algoritmo gráfico.

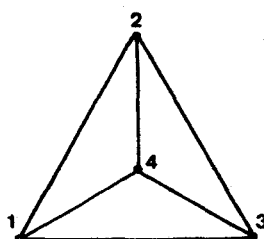


Figura 11

III.1.6. Comprobación de rectangularidad

Puesto que se impone la limitación de que tanto el contorno del esquema como cada uno de los locales sea rectangular, se hace necesario comprobar que el grafo lo permite, tras las limitaciones que sucesivamente se han ido imponiendo.

Como quedó dicho en II.1.1, diversos autores han estudiado este problema, dejando establecido que el grafo cumplirá la condición de rectangularidad si no contiene ningún "grafo completo de cuatro vértices". Figura 11.

El algoritmo que permite hacer esta comprobación utiliza la matriz de adyacencia, $A(a_{ij})$. Se trata de buscar un vértice V_i en cuya fila existan tres términos, a_{ij} , a_{ik} y a_{il} , distintos de cero; V_i es entonces adyacente a V_j , V_k y V_l . Si en las filas j , k y l ocurre que los términos a_{ji} , a_{jk} , a_{jl} ; a_{ki} , a_{kj} , a_{kl} y a_{li} , a_{lj} , a_{lk} son todos distintos de cero, entre los vértices citados, V_i , V_j , V_k y V_l existe un grafo completo.

La comprobación se simplifica por la naturaleza simétrica de la matriz de adyacencia. Basta, pues, que para toda cuaterna i, j, k, l , se cumpla que a_{ij} , a_{ik} , a_{il} ; a_{ji} , a_{jk} , a_{jl} sean todos distintos de cero. En caso contrario, el grafo G_6 cumple el test de rectangularidad.

Cuando el grafo no pasa el test, el programa avisa qué vértices son los causantes para que el operador revise consecuentemente las relaciones entre ellos.

III.1.7. Triangulación del grafo

Tras las transformaciones sufridas hasta ahora, el grafo G_6 tiene ya una representación única pero puede corresponderle un número indefinido de duales. Para definir una solución única es preciso transformarlo aún en un "grafo planar máximo".

Este grafo cumple al límite la condición de Euler, $|A| = 3|V| - 6$, y todas sus caras tienen valencia tres. Es lo que se llama una "triangulación", según los teoremas de Baglivo y Graver (1983):

- 1.- Un grafo planar que es una triangulación para una de sus representaciones en el plano es un grafo planar máximo.
- 2.- Todas las representaciones en el plano de un grafo planar máximo con más de dos vértices son triangulaciones.
- 3.- Un grafo planar con más de dos vértices es un grafo planar máximo si y sólo si $|A| = 3|V| - 6$.
- 4.- A una triangulación le corresponde una disposición arquitectónica fundamental.

La característica especial del grafo que manejamos, con un circuito exterior formado por cuatro aristas que expresan las relaciones entre puntos cardinales, impide poder triangular todas las caras del grafo, pues no podemos incluir una arista que ligue a dos puntos cardinales no contiguos. La región exterior del grafo será, por tanto, la única que no podrá tener valencia tres, y la fórmula de Euler quedaría corregida así: $|A| = 3|V| - 7$.

El procedimiento para lograr una triangulación es el siguiente:

1.- Introducción de nuevas relaciones de orientación. Aumentará así el número de locales definidos como "exteriores" y se indicarán, si no se hizo antes, cuáles van a ocupar las esquinas en el esquema de planta (orientaciones compuestas).

2.- Introducción de nuevas adyacencias entre locales.

3.- Comprobación de la condición de triangulación citada.

Si es $|A| > 3|V| - 7$, hemos añadido demasiadas aristas y, en consecuencia, el grafo ya no es planar. Si $|A| < 3|V| - 7$, no se han introducido relaciones suficientes, deben seguirse añadiendo aristas.

Hecho ésto, se puede volver a trazar el grafo G_7 , ya triangulado, sin necesidad de modificar las coordenadas de los vértices puesto que su número no ha variado.

Las aristas añadidas no hacen irreversible el proceso: se pueden hacer modificaciones siempre que no cambie el número de aristas.

III.1.8. Transformación de coordenadas de vértices

Antes de la obtención del grafo dual de G_7 , y para que la disposición arquitectónica fundamental se ajuste a las condiciones señaladas para la forma del contorno y de los locales, conviene transformar las coordenadas de los vértices que representan a los distintos espacios (los del grafo g_7 , subgrafo de G_7). El grafo transformado será g_8 y su dual, g_8^* lo será también de G_7 (considerando la región exterior), siendo por tanto el esquema adimensional buscado.

La transformación de coordenadas se efectúa según estos cri-

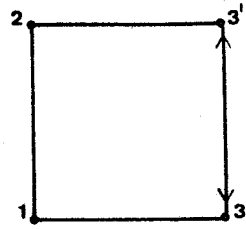
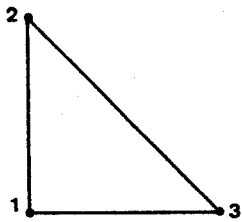


Figura 12

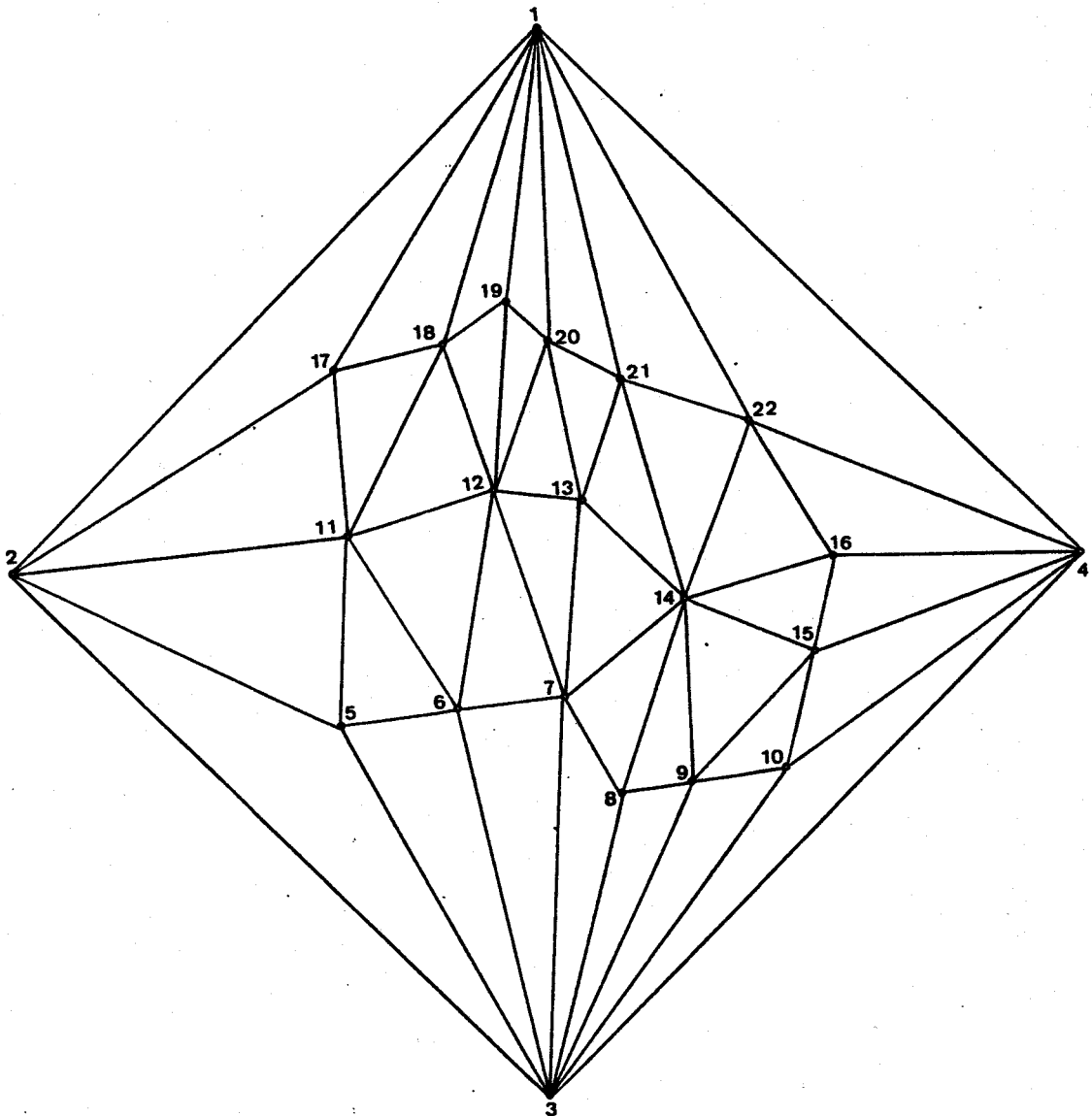


Figura 13

terios:

- Todas las aristas del grafo transformado serán ortogonales. Para ello, a un vértice de g_7 puede corresponderle más de uno en g_8 .
- El contorno será rectangular, correspondiendo una orientación a cada uno de sus lados. Los vértices sobre un mismo lado representan locales de igual orientación, y los de las esquinas espacios con doble orientación.
- Los vértices que representan los puntos cardinales, pertenecientes al grafo G_7 , pero no al g_7 , servirán para "orientar" la transformación.

Para hacer que todas las aristas sean ortogonales, lo que simplifica enormemente la obtención del grafo dual, se opera un desdoblamiento de vértices cuando es necesario, como se ve en la figura 12.

El procedimiento completo se describe a continuación y se ilustra con una aplicación sobre el grafo de la figura 13.

1. Obtención del circuito exterior del grafo g_7 , a partir del vértice nordeste y en sentido horario. Se opera sobre la matriz de incidencia.
2. El circuito exterior contiene cuatro tramos o caminos, comprendidos entre los cuatro vértices de doble orientación. Se toma como "lado inicial" aquél de estos tramos que contenga mayor número de vértices.
3. Los vértices del lado inicial se colocan uniformemente espaciados sobre un segmento de recta y se les asignan coordenadas. Figura 14.
4. Los vértices adyacentes a los del lado inicial se sitúan



Figura 14

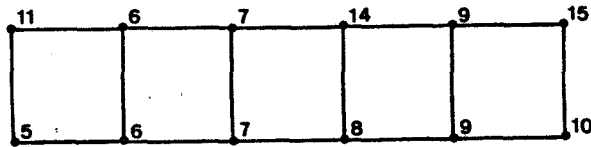


Figura 15

en una recta paralela, formando un nuevo lado enlazado al inicial mediante aristas.

Para que un vértice pueda ser transformado ha de ser el único adyacente a un determinado vértice del lado inicial. El resto del lado se completará con la repetición de los vértices que aparecen en el lado inicial y que no son adyacentes a ningún vértice con esas características. Así pues, como ya se ha dicho, a un vértice en el grafo g_7 , le puede corresponder más de uno en el transformado g_8 con distintas representaciones. Figura 15.

5. Test 1. Una vez generado este nuevo lado hay que efectuar una primera comprobación por si hay que corregirlo.

Sucedará cuando dos vértices, adyacentes en el grafo g_7 , no aparecen como tales en el presente lado. Este caso sólo se puede dar cuando ya han sido transformado -y están en lados anteriores- los vértices que aparecen entre los dos afectados y que impiden esa adyacencia. Procederemos entonces a la supresión de éstos. Esto no supone alteración alguna de las condiciones de adyacencia por la forma de extracción del dual que se describe en la etapa siguiente. Figura 16.

6. Test 2. Con este test se comprueba si se ha transformado ya algún vértice que pertenezca al tramo de orientación opuesta al lado inicial del grafo g_7 . Se señalarán los que cumplan estas condiciones.

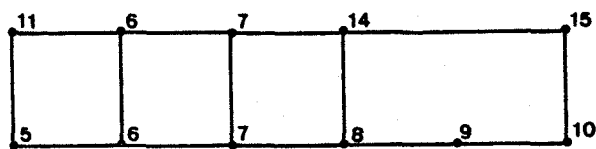


Figura 16

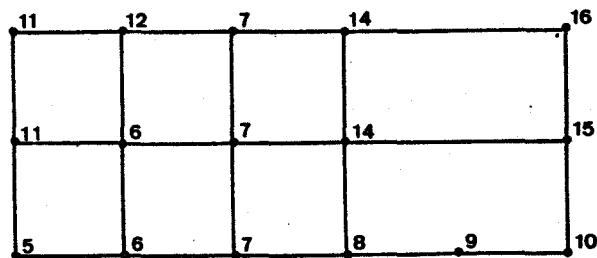


Figura 17

7. Transformación de coordenadas de vértices adyacentes a los del lado generado en último lugar y que no se han transformado aún.

Si un vértice es adyacente a un solo vértice del último lado, y éste no ha sido marcado por el test 2, aparece transformado en el nuevo lado sobre una recta paralela al último. Se completa este nuevo lado con la repetición de los vértices correspondientes del lado anterior. Figura 17.

8. Test 3. Corrección si aparecen dos lados sucesivos con los mismos vértices.

Si no se ha podido generar ningún nuevo lado a partir del último, querrá decir que no hay ningún vértice -no transformado aún- que sea el único adyacente a uno de los vértices del lado en cuestión.

Tendremos que proceder entonces al desdoblamiento de uno de los vértices del último lado. De esta forma puede haber ya - dos vértices que sean adyacentes a ese vértice duplicado. Si, a pesar de esta operación, sigue sin poderse generar un nuevo lado, habrá que efectuar un nuevo desdoblamiento del mismo vértice, siendo entonces tres los vértices que pueden - ser adyacentes al vértice elegido. Se procedería así hasta - la consecución de un nuevo lado.

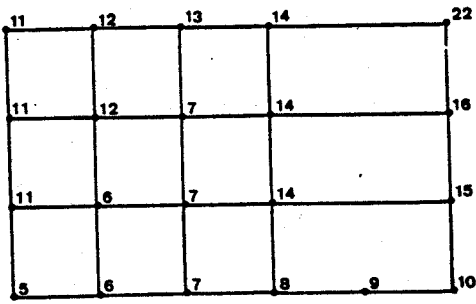
El desdoblamiento de un vértice de un determinado lado obliga también a duplicar los vértices afectados situados en lados paralelos y que han sido generados con anterioridad. Sólo de esta forma se conserva la ortogonalidad de las aristas.

Para el desdoblamiento de los vértices se elegirán primero - los vértices extremos del último lado. Si éstos pertenecen - al lado opuesto al inicial (6) se elegirán entonces uno cualquiera de los interiores que no pertenezcan a ese lado y que sea adyacente a algún vértice no transformado. La razón de - elegir en primer lugar los vértices extremos obedece a la necesidad de obtener como vértices extremos del lado opuesto - al inicial los vértices "doblemente orientados".

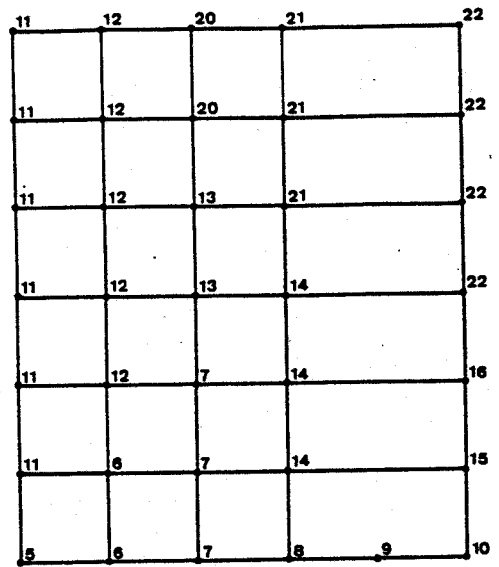
9. Se repetirán cíclicamente las operaciones 5,6,7 y 8 hasta haber transformado todos los vértices del grafo g_7 . Se obtiene entonces el grafo g_8 . Figura 18.

El grafo g_8 resultante tendrá mayor número de vértices y - aristas que el grafo g_7 , pues, como ya se ha dicho, un vértice del grafo g_7 puede dar lugar a más de uno en el transformado g_8 (expansión del vértice); todos ellos estarán unidos mediante aristas.

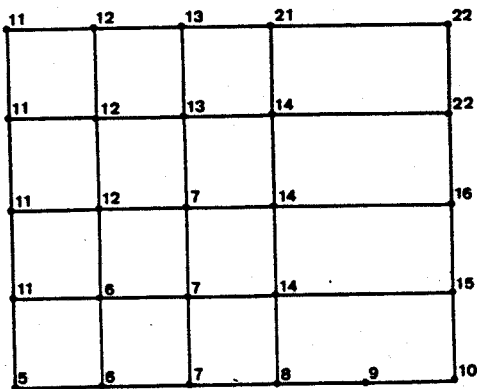
(6) Esto querrá decir que a partir de ellos no puede generarse ningún - otro.



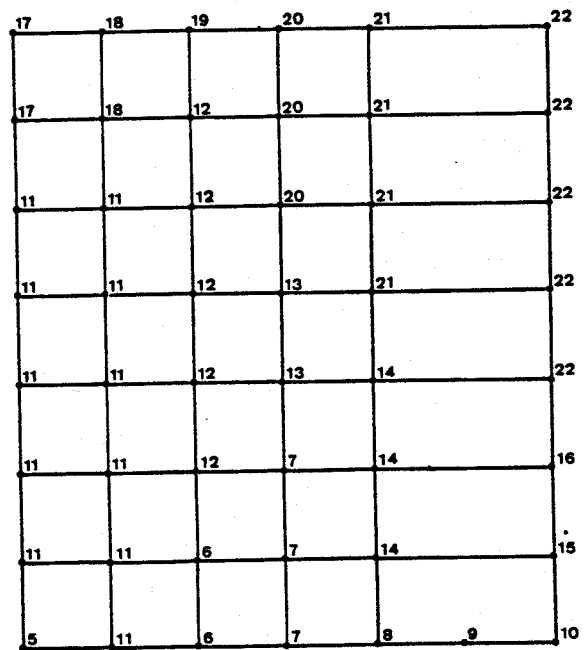
18.a



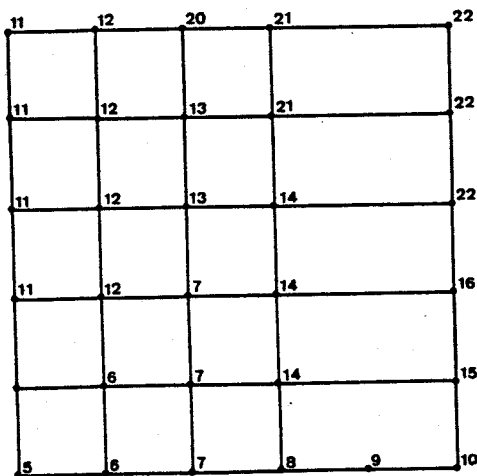
18.d



18.b



18.e



18.c

Figura 18

h.o. - opelo tues/mos 6

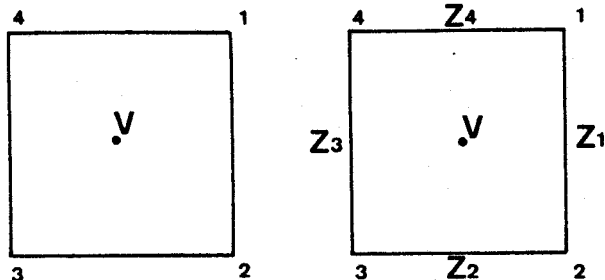


Figura 19

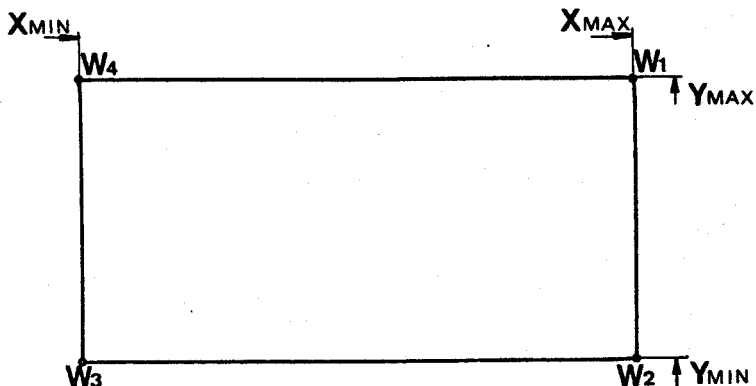


Figura 20

III.1.9. Obtención del grafo dual

La obtención del grafo dual a partir del grafo transformado g_8 es inmediata. Se extrae el grafo dual g_8^* asignando una región rectangular a cada uno de los vértices del grafo g_8 .

Cada región rectangular se definirá por las coordenadas de sus cuatro vértices extremos. A cada uno de ellos le corresponden dos coordenadas pero al estar relacionadas son sólo cuatro las magnitudes que hay que determinar para cada vértice en g_8 . Figura 19.

Para obtenerlas realizamos las siguientes operaciones:

1. Determinamos las coordenadas mínimas y máximas, X_{\min} , X_{\max} , Y_{\min} , Y_{\max} del grafo g_8 que, lógicamente, son la de los vértices de las esquinas. Figura 20.

2. Obtenemos los cuatro parámetros correspondientes a los vértices interiores del grafo g_8 . Aquellos vendrán definidos por las coordenadas del vértice analizado y de sus vértices adyacentes. Si dos vértices adyacentes tienen la misma abcisa, el parámetro correspondiente se logra por la semisuma de la abcisa de ambos. Si tienen igual ordenada, la semisuma de las ordenadas nos dará entonces aquél. Figura 21.

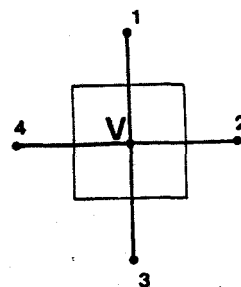


Figura 21

Si un vértice interior no es adyacente a otros cuatro vértices, el parámetro correspondiente que falte -a lo sumo uno- se obtiene sumándole a su abcisa u ordenada, según el caso, la mitad de la distancia entre el lado al cual pertenece y el lado al que éste dio lugar. Figura 22.

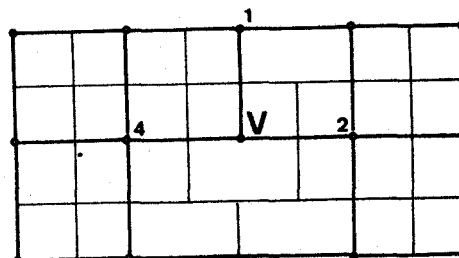


Figura 22

3. Calculamos los parámetros correspondientes a los vértices exteriores del grafo g_8 . El proceso será el mismo que en el caso anterior salvo para el parámetro correspondiente a la orientación -u orientaciones en el caso de vértice en esquina-. Este vendrá determinado -

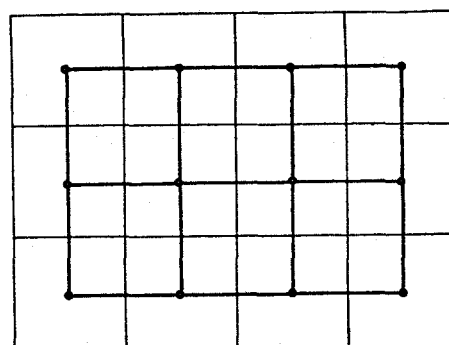


Figura 23

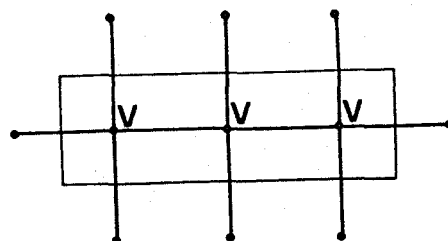


Figura 24

por las coordenadas X_{\max} , X_{\min} , Y_{\max} , Y_{\min} , obtenidas anteriormente, sumándoles o restándoles, según el caso, una determinada magnitud fija, para lograr que el contorno sea rectangular.

4. La unión de todas las regiones, correspondientes a cada uno de los vértices del grafo g_8 , nos daría el dual de éste g_8^* . Al ser la distancia entre dos lados paralelos, y sucesivos en su generación, una magnitud constante, la forma de consecución del dual nos garantiza la no existencia de regiones que no correspondan a ninguno de los vértices del grafo geométrico. Además, la intersección de dos regiones, correspondientes a vértices adyacentes en dicho grafo, nos dará siempre la arista común. Figura 23.

Si suprimimos en dicho grafo dual g_8^* la arista intersección de dos regiones que responden a un mismo vértice en el grafo g_8 , obtendremos un grafo g_9 , que podemos llamar G_9 , si tenemos en cuenta la región exterior. Este grafo geométrico G_9 será el esquema en planta adimensional buscado. Figura 24.

Este esquema, que es adimensional, cumple todas las relaciones de adyacencia y orientación impuestas (y es la única solución que las cumple), es rectangular en su contorno y en los espacios componentes, y no tiene espacios vacíos.

El esquema trazado puede ser el punto final del proceso. Pero podemos decidir que la solución no es satisfactoria, y desear introducir modificaciones. Esto puede hacerse cuantas veces se quiera, anulando una o varias relaciones y sustituyéndolas por un número igual de relaciones nuevas. Si ésto se hace a la vista del esquema obtenido, que es suficientemente expresivo, no se alterará la planaridad ni la triconectividad del grafo. Cuando el esquema sea satisfactorio podrá darse por terminado el proceso.

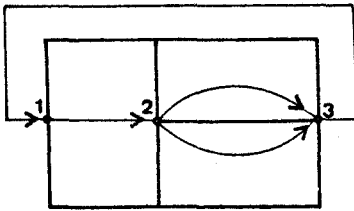


Figura 25.a

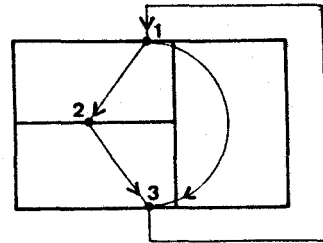


Figura 25.b

III.2. DIMENSIONAMIENTO

El procedimiento para dimensionar esquemas de contorno y componentes rectangulares comprende las etapas siguientes:

III.2.1. Sustitución por dos grafos dirigidos

El esquema se sustituye por dos grafos dirigidos, uno "horizontal" y otro "vertical".

Los vértices de grafo "vertical" representan a dos segmentos verticales del esquema, que son cerramientos y particiones paralelos entre sí. Figura 25.a.

La región exterior y los locales se sustituyen por arcos que unen los vértices correspondientes, orientados de izquierda a derecha en los locales y en sentido contrario en la región exterior.

Procedimiento análogo se seguirá para el grafo "horizontal" a partir de los segmentos horizontales del esquema y orientando de arriba abajo los arcos que representan locales, y de abajo arriba el de la región exterior. Figura 25.b.

Cuando se desea que los límites de dos espacios adyacentes estén en prolongación, ambos límites se definen por un solo vértice en el grafo que proceda. Figura 26. Esto es especialmen-

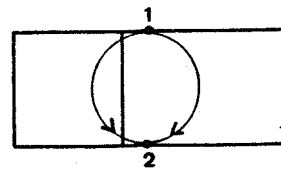
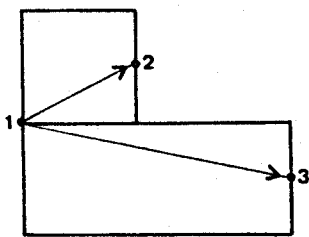


Figura 26

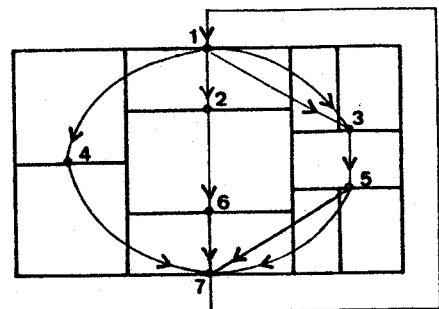
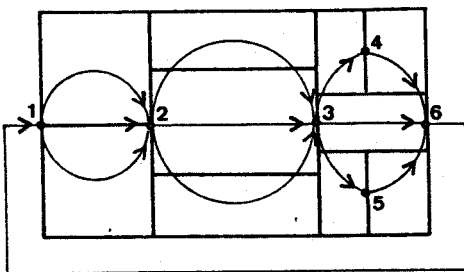


Figura 27

te importante en el contorno del esquema. Figura 27.

Como puede verse, estos grafos dirigidos pueden no ser simples, al contener arcos múltiples o paralelos. Serán en general fuertemente conexos.

El número de vértices de cada grafo igualará al de segmentos de contorno y particiones en la dirección correspondiente, y el de arcos se corresponderá con el número de locales, más el que representa la región exterior.

Ambos grafos vendrán definidos por su matriz de incidencia $B(b_{ij})$ de dimensión $(|A|, |V|)$, cuyos términos serán $b_{ij}=0$ si el arco i no incide en el vértice j ; $b_{ij} > 0$ si el vértice j es origen del arco i , y $b_{ij} < 0$, cuando el vértice j sea el extremo del arco i . Estos valores distintos de cero se definirán más adelante.

Cualquier criterio de numeración es válido para señalar los -

vértices. La numeración de los arcos estarán en correspondencia con la utilizada para designar los vértices en el problema adimensional. El arco que representa a la región exterior se numerará en último lugar.

III.2.2. Definición de las matrices de adyacencia

Elaborados los dos grafos dirigidos, han de fijarse las adyacencias entre locales, especialmente en cuanto sean relaciones de acceso. Se ha considerado que dos estancias pueden comunicarse si tienen al menos 1 m. de contorno común (hueco de paso).

Este concepto de adyacencia, entendido como accesibilidad, viene definido en la correspondiente matriz $A (a_{ij})$ de dimensión $(|A|, |A|)$, cuyos términos serán $a_{ij} = 0$ si no se establece accesibilidad entre los locales i y j , y $a_{ij} = 1$ en caso positivo. Como el acceso es simétrico, será $a_{ij} = a_{ji}$, y $A \{a_{ij}\}$ será simétrica.

III.2.3. Restricciones métrico-geométricas

Las limitaciones más usuales que en la práctica se imponen a los locales de una distribución suelen ser las dimensiones lineales o superficies, mínimas o máximas, que cada espacio deba tener.

Las dimensiones mínimas y/o máximas de los locales o del contorno, que son restricciones lineales, se reflejan en la matriz de incidencia del grafo dirigido que corresponde a la dirección en que opera cada limitación.

Si se introducen limitaciones de superficie, que son no lineales, el problema deberá ser resuelto por programación no lineal; estas limitaciones pueden introducirse indistintamente en una u otra de las dos matrices de incidencia.

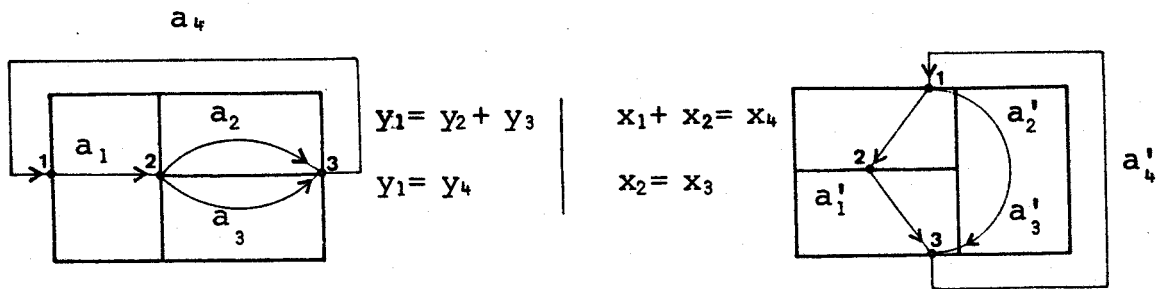


Figura 28

III.2.4. Expresiones que traducen las restricciones

Las limitaciones dimensionales dan lugar a ecuaciones o inecuaciones lineales, lo mismo que las condiciones de accesibilidad, como en seguida veremos. Las limitaciones de superficie dan ecuaciones o inecuaciones no lineales.

Las limitaciones dimensionales se traducen en un sistema de ecuaciones y/o inecuaciones lineales, aplicando la primera ley de Kirchhoff para redes eléctricas. Como es sabido, esta ley dice que, en cada vértice de la red, la suma algebraica de las intensidades de corriente de las ramas concurrentes es nula. Traducida a nuestro grafos dirigidos, la primera ley de Kirchhoff diría que la suma de los valores dimensionales (longitud o anchura de un local) atribuidos a los arcos que concurren en un vértice es igual a la de los que parten de él. Figura 28.

De este modo, en el grafo horizontal, se haría: $B_1 \cdot X = 0$. En esta expresión $B_1 = \{b'_{ij}\}$ es una matriz obtenida trasponiendo la de incidencia, a la que previamente se ha suprimido la última columna (que es combinación lineal de las demás), y dividido cada término por su valor absoluto (con lo que B_1 sólo contendrá valores 0, 1 y -1). X es una matriz columna, cuyos términos $x_1, x_2 \dots x_m$ representan las longitudes de cada local y del contorno.

Del mismo modo, en el grafo vertical se hará $B_2 \cdot Y = 0$. Donde -

B_2 es una matriz obtenida de la de incidencia del grafo vertical por el mismo procedimiento que se siguió para B_1 . Y donde Y es una matriz columna cuyos términos son los anchos de locales y contorno.

Para elaborar las inecuaciones que expresan las condiciones de accesibilidad (valores no nulos en la matriz de adyacencia de cada grafo) se procede del modo siguiente en cada uno de los - dos grafos dirigidos:

- Como cada local se define por un arco, se busca el vértice - común a los dos espacios adyacentes (término positivo y negativo, respectivamente, en la matriz de incidencia).
- Se buscan, a continuación, todos los arcos que concurren en ese vértice.
- Se analiza el centro geométrico de cada local en el esquema adimensional. (7)
- En el grafo horizontal, \underline{i} representará el local cuyo centro geométrico tiene mayor ordenada, y convendremos en que el local \underline{j} es accesible al \underline{i} .

Las inecuaciones que expresan la accesibilidad entre los locales \underline{i} y \underline{j} , serán:

$$X_i + \sum X_{SUPi} - \sum X_{SUPj} \geq 1$$

$$X_j - \sum X_{SUPi} + \sum X_{SUPj} \geq 1$$

$$X_i, X_j \text{ longitud de los locales } \underline{i} \text{ y } \underline{j}$$

(7) Esto no supone un contrasentido, como veremos después, sólo servirá para situar 'relativamente' unas estancias respecto de otras.

X_{SUPi} , X_{SUPj} : longitud del local cuyo centro geométrico tiene su abcisa \underline{X} mayor que el de \underline{i} , \underline{j} .

El término independiente $\underline{1}$ indica la anchura del hueco de paso mínimo.

En el grafo dirigido vertical, llamamos local \underline{i} a aquel cuyo centro geométrico tiene mayor abcisa \underline{X} . Convendremos que el local \underline{j} es accesible al \underline{i} .

Las inecuaciones que expresan la accesibilidad entre dichos locales serán:

$$Y_i + \sum Y_{SUPi} - \sum Y_{SUPj} \geq 1$$

$$Y_j - \sum Y_{SUPi} + \sum Y_{SUPj} \geq 1$$

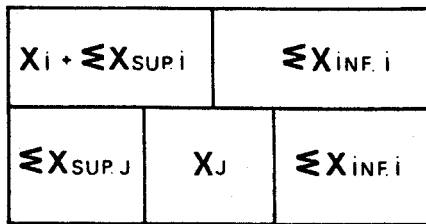
$$Y_i, Y_j \text{ anchura de los locales } \underline{i} \text{ y } \underline{j}$$

Y_{SUPi} , Y_{SUPj} : anchura del local cuyo centro geométrico tiene su ordenada \underline{Y} mayor que el de \underline{i} , \underline{j} .

Si todos los términos de la ecuación son positivos se anula esta restricción por ser obvia. (8)

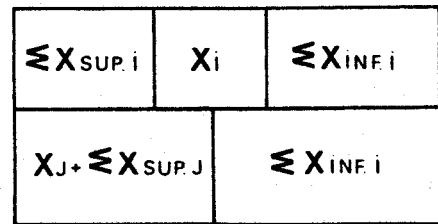
En un esquema de espacios y contornos rectangulares, estas inecuaciones expresan siempre las condiciones de accesibilidad entre dos locales cualesquiera. En efecto, podemos entender estas expresiones, para el grafo horizontal, considerando sólo dos grandes espacios: uno, el que comprende al local \underline{i} y a todos los que tienen centro de abcisa mayor que él; y otro, el -

(8) En las restricciones métrico-geométricas, se impondrá una dimensión para cada estancia que, lógicamente, será mayor que la longitud de un hueco de paso.



≥ 1

Figura 29.a



≥ 1

Figura 29.b

que contiene todos los espacios con centro de abscisa mayor que la de j. Figura 29.a.

Para la segunda inecuación se contempla la otra posibilidad: un gran espacio que comprende a todos los locales con centros de abscisa superior a la de i, y otro que englobe al j y a todos los locales con centros de abscisa superior a j. Figura - 29.b.

La operación realizada puede hacerse análogamente en el grafo vertical.

Estas inecuaciones pueden expresarse de otra forma.

En efecto: Figura 29

$$X_i + \sum X_{SUP.i} = L - \sum X_{INF.i} \Rightarrow \sum X_{SUP.i} = L - X_i - \sum X_{INF.i}$$

$$X_j + \sum X_{SUP.j} = L - \sum X_{INF.j} \Rightarrow \sum X_{SUP.j} = L - X_j - \sum X_{INF.j}$$

Las inecuaciones quedarían:

$$X_i + L - X_i - \sum X_{INF.i} - L + X_j + \sum X_{INF.j} = X_j - \sum X_{INF.i} + \sum X_{INF.j} \geq 1$$

$$X_j + L - X_j - \sum X_{INF.j} - L + X_i + \sum X_{INF.i} = X_i - \sum X_{INF.j} + \sum X_{INF.i} \geq 1$$

L: Longitud del contorno.

$X_{INF i}$, $X_{INF j}$: longitud de un gran espacio que comprende a aquellos cuyos centros tienen abcisa menor que la del local i o j.

Igualmente podríamos expresar estas inecuaciones para el otro grafo dirigido. Unas y otras, como puede observarse, son similares.

El tercer y último tipo de requisitos corresponde a las restricciones de superficie impuestas a los locales y al contorno. Podemos clasificarlas así:

a) Restricciones para el problema lineal:

Acotación superior e inferior de las variables longitud y anchura de locales y del contorno:

$$a_i \geq x_i \geq a'_i$$

$$b_i \geq y_i \geq b'_i$$

Si se quiere dar a una variable un valor fijo, bastará con hacer iguales sus límites superior e inferior.

Los valores límites a_i , a'_i , b_i , b'_i están recogidos en la matriz de incidencia de los grafos dirigidos.

Si alguna variable no está acotada inferiormente se le asignará un valor mínimo igual a 1 m.

b) Restricciones para el problema no lineal:

Acotación inferior de la longitud y anchura de cada local y del contorno:

$$x_i \geq l_i$$

$$y_i \geq l'_i$$

Los valores l_i , l'_i vienen reflejados en las matrices de incidencia de los grafos dirigidos.

Al igual que en el caso anterior, si una variable no está limitada inferiormente, se le asignaría un valor mínimo.

En realidad estas inecuaciones pretenden fijar las dimensiones mínimas de locales y del contorno, sólo que dicho valor, si se quiere, puede ser distinto en una dirección que en otra.

Acotación inferior de la superficie de cada una de las estancias y del contorno.

$$x_i \cdot y_i \geq s_i$$

Los valores s_i estarán reflejados en la matriz de incidencia de los grafos dirigidos.

Todas las restricciones impuestas, lineales y no lineales, figurarán en una matriz: la matriz de restricciones $R=(r_{ij})$ de dimensión $(|R|, 2 \cdot |A|)$, siendo:

$|R|$: número de restricciones.

$|A|$: número de locales más uno (correspondiente al contorno).

Los términos independientes de cada una de las restricciones se localizarán en una matriz-columna; el tipo (9) de cada una de ellas se expresará en una matriz-fila.

(9) El tipo de una restricción se refiere a su forma de expresión, según se trate de ecuación o inecuación, y dentro de esta última, si es del género mayor o igual, o menor o igual.

III.2.5. Naturaleza y costes de la función-objetivo

Las funciones-objetivo previstas son las siguientes:

- Funciones-objetivo lineales:

- 1) Minimizar la longitud o anchura del contorno. Su expresión sería: Minimizar x_m .
- 2) Con una longitud o anchura del contorno determinada, obtener estas mismas variables para cada uno de los locales, - con arreglo a su importancia, referida ésta a sus dimensiones relativas dentro del esquema. Su expresión sería:

$$\text{Minimizar } \sum_{i=1}^{m-1} c_i \cdot x_i$$

c_i : coste que indica la importancia relativa del local i .
Por tratarse de un problema de minimización se le asignará un valor negativo.

x_i : longitud o anchura de locales exclusivamente.

Para utilizar este último tipo de problema lineal, habrá - que definir una nueva restricción métrico-geométrica muy - particular: se fijará la dimensión del contorno.

- Funciones-objetivo no lineales:

- 1) Minimizar la superficie del contorno. Su expresión sería:
Minimizar $x_m \cdot y_m$.
- 2) Con una superficie del contorno determinada, obtener la superficie de cada local, de acuerdo con su entidad dimensional relativa. Su expresión sería:

$$\text{Minimizar } \sum_{i=1}^{m-1} c_i \cdot x_i \cdot y_i$$

Para emplear este segundo tipo de problema no lineal, habrá que definir previamente la superficie del contorno.

III. 2.6. Resolución del problema por los métodos de programación

El dimensionamiento es un problema de programación, donde se busca un resultado óptimo y cuya estructura general consta de una función-objetivo, que se quiere maximizar o minimizar, sujeta a una serie de restricciones.

Un problema de programación lineal es aquél que consta de - una función-objetivo lineal sujeta a una serie de restricciones, también lineales. Cuando, o bien la función-objetivo o alguna de las restricciones -o ambas cosas a la vez- sean no lineales, la programación será no lineal. En el caso particular de que la función objetivo sea no lineal y todas las restricciones sí lo sean, la programación no lineal recibe el nombre de cuadrática.

Al ser siempre expresiones lineales las derivadas de aplicar la primera ley de Kirchhoff a los dos grafos dirigidos, y - las que reflejan la accesibilidad entre espacios, la linealidad o no linealidad del problema vendrá dada: por un lado, por la función objetivo; por otro, por las restricciones métrico-geométricas.

Pasemos a describir los dos métodos de resolución para los - problemas de programación.

a) Programación lineal:

Su estructura general será:

Min. $C \cdot X$

s.a. $A \cdot X = 0$ (1ª ley de Kirchhoff)

$B \cdot X - 1 \geq 0$ (accesibilidad)

$X^T \geq D$ (métrico-geométricas)

$X^T \leq E$

Para la resolución del problema de programación lineal se ha utilizado un algoritmo clásico conocido con el nombre de "simplex". (Larrañeta, 1977).

Aplicando el método de resolución indicado, se obtendrá la longitud variable en una dirección independientemente de la anchura variable en la dirección ortogonal de cada local y del contorno.

b) Programación no lineal:

Su estructura general será:

Min. $C \cdot X \cdot Y$

s.a. $A \cdot X = 0$ (1ª ley de Kirchhoff)

$A' \cdot Y = 0$

$B \cdot X - 1 \geq 0$ (accesibilidad)

$B' \cdot Y - 1 \geq 0$

$X^T \geq D$

$Y^T \geq D'$ (métrico-geométricas)

$X \cdot Y \geq E$

Para la resolución del problema no lineal se ha utilizado el algoritmo de Polak-Ribière (1971), que utiliza el método de

los gradientes conjugados para minimizar funciones no necesariamente convexas. Se ha demostrado asimismo (Rao, 1978), - que cualquier método de minimización que usa las direcciones conjugadas es convergente de forma cuadrática, lo que disminuye de forma considerable el número de iteraciones hasta - llegar al óptimo.

Se ha utilizado el algoritmo de Polak-Ribière y no el tradicional de Fletcher-Reeves porque aquél supone una mejora de este último. En efecto, Polak (1971) -citando un estudio de Lootsme- indica que los métodos de penalización de funciones, exterior e interiormente, convergen más rápidamente que aquéllos que sólo penalizan exterior o interiormente. En consecuencia, su método es una versión de aquél que usa los gradientes conjugados con penalización de la función objetivo, tanto exterior como interiormente.

Aplicando ese método de resolución se obtienen las dos variables -longitud y anchura-, para cada local y el contorno, - conjuntamente.

Para ajustar aún más los resultados obtenidos por este último método, una vez obtenidos los valores para las distintas variables se eliminan las restricciones no lineales. Se aplica entonces el algoritmo "simplex" al ser ya un problema lineal. Los valores de las cotas inferiores serán ahora los resultados obtenidos para las distintas variables. Las restricciones no lineales serán satisfechas entonces y, por tanto, pueden ser eliminadas. Al ser el algoritmo "simplex" más potente los resultados obtenidos se ajustarán mejor.

Una vez resuelto el problema de dimensionamiento por los métodos de programación adecuados a cada caso, podemos dibujar el esquema dimensionado, concluyendo entonces el proceso completo de su generación.

La posibilidad teórica de generar y optimizar esquemas de distribución de espacios había sido el punto de partida de nuestra tesis; posibilidad formulada con anterioridad por diversos autores.

Tras el análisis riguroso de los procesos sugeridos por las investigaciones sobre el tema y su crítica en función de los objetivos, el trabajo que presentamos consiste, esencialmente, en la propuesta de un nuevo método de generación y optimización completo y coherente.

La elaboración de este método lleva consigo el desarrollo de algunos algoritmos completamente originales, así como la adaptación de otros tradicionalmente aplicados.

El procedimiento que proponemos utiliza la técnica clásica de representación mediante el grafo dual.

El conjunto de algoritmos desarrollado para la generación de esquemas adimensionales ofrece una serie de ventajas: no limita el número de espacios componentes; es un proceso dinámico en el que las condiciones pueden introducirse de modo progresivo; es, además, un método flexible que permite en todo

momento la revisión de las condiciones impuestas; y por último, es aplicable de un modo general -cualquiera que sea la estructura del grafo de partida-.

La comprobación de la planaridad y K-conectividad del grafo se basa en los estudios de Hopcroft y Tarjan. Los algoritmos mencionados permiten analizar de modo progresivo cada una de estas propiedades. Se ha utilizado el algoritmo que desarrolla Tutte para el trazado de grafos triconexos, con las adaptaciones pertinentes para una adecuada acción integrada usuario-ordenador.

El procedimiento empleado para la obtención del grado dual es una aportación original que consideramos de interés.

El problema de dimensionamiento de esquemas se aborda siguiendo una concepción clásica, recogida al menos teóricamente por la mayor parte de los investigadores: la de considerarlo como problema de optimización, empleando técnicas de programación lineal y no lineal.

Las condiciones de accesibilidad entre locales vienen expresadas por inecuaciones para cuya elaboración se propone un algoritmo nuevo. Los restantes requisitos o bien se formulan directamente o se establecen por algoritmos ya conocidos.

Se ha formulado también un procedimiento iterativo, basado en los estudios de Polak, para la resolución de problemas mediante programación no lineal, que dimensiona eficazmente los esquemas de distribución, con ajuste y trazado del mismo.

El trabajo desarrollado en esta tesis pretende, en suma, servir de aportación a las formulaciones teóricas necesarias para la revisión de la utilidad del ordenador en el diseño arquitectónico. Con ello contribuimos, tal vez, a la incorporación de este poderoso instrumento a una disciplina que hasta ahora se ha mantenido apartada de su inexorable influencia.

1. GRAFO

El concepto intuitivo de grafo es el de unos puntos unidos - mediante líneas. Lo que un grafo expresa no son relaciones - métricas de posición de los puntos o forma de las líneas, si no topológicas: qué pares de puntos son los que están en relación.

Un grafo G , pues, se define por una pareja de conjuntos. El de los puntos o *vértices*, conjunto finito y no vacío $V(G)$, y el de relaciones o *aristas* $A(G)$, que puede ser vacío, y cuyos elementos son pares desordenados de vértices.

Si los vértices tienen asignadas coordenadas tenemos un *grafo geométrico*. Si no, un *grafo abstracto*.

2. SUBGRAFO

Un subgrafo G' de un grafo G es otro grafo cuyos vértices - son $V(G') \subseteq V(G)$ y cuyas aristas son $A(G') \subseteq A(G)$ y unen sólo vértices del conjunto $V(G')$.

3. VÉRTICES Y ARISTAS

Dos vértices son *adyacentes* cuando están unidos por una arista.

Una arista es *incidente* a un vértice si lo tiene como extremo.

Si más de una arista incide en la misma pareja de extremos, se trata de *aristas paralelas o múltiples*. Si una arista une un vértice a sí mismo, es un *bucle o lazo*.

Un grafo sin aristas múltiples ni bucles es un *grafo simple*. Un grafo simple en el que todos los vértices son adyacentes es un *grafo completo*.

4. CAMINOS Y CIRCUITOS

Un *camino elemental* es una secuencia de aristas de un grafo, de forma que cada una de ellas tenga un vértice común con la precedente, salvo la primera, y otro con la siguiente, salvo la última. Un camino elemental cerrado, porque los vértices inicial y final de la secuencia coincidan, reciben el nombre de *circuito elemental*.

Por *longitud* de un camino o de un circuito entendemos el número de aristas que lo forman.

5. CONECTIVIDAD

Un grafo será *conexo*, si entre todo par de vértices existe un camino. En caso contrario será *inconexo*.

Componente conexas de un grafo es el subgrafo que contiene un vértice dado y todos los vértices que pueden enlazarse a él mediante un camino.

Grafo por supresión de vértice $G-v$ es el que se obtiene de G eliminando al vértice v y las aristas incidentes a él.

Grafo por supresión de arista $G-a$ es el que se obtiene de G eliminando la arista a .

Si G es un grafo conexo y $G-v$ no lo es, el vértice v es un *vértice de corte o separación*.

Un grafo es *biconexo* si no tiene vértices de corte.

Si G es un grafo conexo, una *componente biconexa* será el sub grafo biconexo que contenga el mayor número posible de vérti ces y aristas.

Si G es un grafo biconexo y contiene cuatro vértices distintos, de modo que cualquier camino que se pueda establecer en tre ellos pasa necesariamente por dos vértices determinados, estos dos últimos constituyen un *par de separación*.

Si el grafo es biconexo y no contiene ningún par de separación, es además *triconexo*.

Una *componente triconexa* de un grafo biconexo G , será un sub grafo triconexo de éste con el máximo número posible de vérti ces y aristas.

Si (a,b) es un par de separación del grafo biconexo G y divi dimos las aristas en clases de equivalencia: $A_1, A_2, A_3, \dots, A_n$, de manera que dos aristas sólo podrán estar en una misma clase; si existe un camino que las contenga y que no pase - por a o b -salvo como vértices extremos-, estas clases serán, así mismo, *de separación* respecto del par (a,b) .

Si (a,b) es un par de separación del grafo biconexo G y A_1, A_2, \dots, A_n las clases de separación del mismo, los *grafos* -

escindidos $G_1 (V_1, A_1)$ y $G_2 (V_2, A_2)$ de G respecto del par (a,b) serán aquellos que cumplan:

$$V_1 = V(A') \quad , \quad A_1 = A' \cup (a,b)$$

$$V_2 = V(A'') \quad , \quad A_2 = A'' \cup (a,b)$$

siendo:

$$A' = \bigcup_{i=1}^k A_i \quad A'' = \bigcup_{i=k+1}^n A_i \quad (a,b): \text{ arista que incide sobre los v\u00e9rtices } \underline{a} \text{ y } \underline{b}.$$

Las nuevas aristas a\u00f1adidas a G_1 y G_2 : (a,b) se denominan *aristas virtuales*.

En general, podemos decir que un grafo con al menos $K+1$ v\u00e9rtices es *K-conexo* si no puede ser hecho inconexo por la supresi\u00f3n de $K-1$, o menos v\u00e9rtices, y las aristas que inciden en ellos.

La conectividad de G , expresada por $K(G)$ es el mayor valor de K para el cual G es K -conexo.

6. PLANARIDAD

Un grafo es *planar* cuando puede ser representado en el plano de forma que no se intersecten geom\u00e9tricamente dos aristas - en puntos distintos de los v\u00e9rtices.

Si un grafo es planar y la inclusi\u00f3n de una arista entre dos v\u00e9rtices cualesquiera -que no sea un bucle o una arista para lela- lo convierte en no planar, se tratar\u00e1 de un *grafo planar m\u00e1ximo*.

Por *valencia de un v\u00e9rtice* entendemos el n\u00famero de aristas - que inciden en \u00e9l. La *valencia de una cara o regi\u00f3n* en un -

grafo geométrico, será el número de aristas que la limitan.

Un grafo geométrico conexo y planar donde cada una de sus regiones tiene valencia tres, es una *triangulación*.

7. DUALIDAD

Si G es un grafo geométrico conexo y planar, puede extraerse a partir de él su grafo dual G^* . Cada región del primero vendrá caracterizada por un vértice en el segundo y viceversa. Una arista que relaciona dos vértices en el grafo G indicará que las dos regiones correspondientes, en el grafo dual G^* , serán adyacentes.

8. GRAFO DIRIGIDO

Grafo dirigido $D\{V(D), A(D)\}$ es aquel cuyas aristas $A(D)$, ahora llamadas *arcos*, son pares ordenados de vértices.

Un arco tiene, pues, un *vértice origen*, V , y un *vértice extre*mo, W . Ambos vértices son *adyacentes* y el arco será *inciden*te desde V e *incidente* a W .

Los arcos que, partiendo de un mismo vértice origen, inciden a un mismo extremo se denominan *arcos paralelos* y el arco - que une un vértice a sí mismo es un *bucle*. Cuando el grafo - no contiene arcos paralelos ni bucles, es un *grafo dirigido simple*.

El *camino* en un grafo dirigido es una secuencia de arcos tales que el extremo de cada uno es origen del siguiente.

Cuando el extremo del último arco es a la vez origen del primero estamos ante un *circuito*.

Un grafo dirigido D será *fuertemente conexo* si para todo par

de vértices, V y W , del mismo hay un camino en el grafo D desde el primero al segundo.

El grafo subyacente del D es el G que tiene sus mismos vértices y cuyas aristas coinciden con los arcos de aquél.

9. ÁRBOLES

Un *árbol* es un grafo conexo que tiene al menos dos vértices y ningún circuito.

Si un grafo A' es subgrafo del árbol A , será también un *subárbol* del mismo.

Si G es un grafo y A es un árbol y a la vez un subgrafo G que contiene a todos sus vértices, diremos que A es un *árbol extendido sobre el grafo G* .

Un *árbol dirigido* será aquél grafo dirigido cuya estructura subyacente es un árbol. Idénticas definiciones podemos dar para *subárbol dirigido* y *árbol extendido sobre un grafo dirigido*.

Un *árbol dirigido con origen* se caracteriza por haberse distinguido de los demás uno de los vértices, llamado origen. Cualquier vértice de este árbol puede ser alcanzado por un camino que parte del vértice origen; ningún arco será incidente al vértice origen y sólo habrá un arco que incida al resto de los vértices del grafo.

Dado un grafo dirigido D , podemos establecer una partición del mismo en dos clases distintas: un conjunto de arcos del árbol extendido sobre D , y un conjunto de *ramas* que completan el número de arcos del grafo dirigido D . Si llamamos V al vértice origen de una rama y W a su vértice extremo, y

existe un camino en el árbol extendido sobre D que va desde W hasta V , y esto ocurre para cada una de las ramas, a este grafo dirigido se le denomina *palmera*.

En este anexo se presentan los dos programas que se han realizado y puesto a punto para cumplimentar los procesos descritos: generación de esquemas distributivos y dimensionamiento de los mismos.

Los dos programas se han realizado en lenguaje FORTRAN IV para un ordenador Hewlett Packard de la serie 1000-M, con un sistema operativo RTE-IV.B.

Para cada uno de los dos procedimientos se ofrece en este anexo un organigrama general en el que se destaca sobre todo la interacción usuario-ordenador, un listado y algunos ejemplos que se han implementado utilizando estos programas.

A.II.1. GENERACIÓN DE ESQUEMAS ADIMENSIONALES

A.II.1.1. Manual de usuario

Con carácter general han de tenerse en cuenta los siguientes criterios:

1. Los datos pueden escribirse en cualquier posición de una tarjeta o en pantalla, separados por caracteres "blanco" (), o "coma" (,). En el caso de utilizar la "coma" como separador, sólo podrá emplearse una para separar dos datos consecutivos.
2. Entre los datos, pueden incluirse tantos comentarios como se deseen, con la condición de que figure en la primera -columna el carácter barra (/).

Sin embargo, este carácter no debe ser utilizado en la expresión que se emplea para titular el problema.

3. Aquellos datos que puedan tomar un valor real pueden escribirse con o sin punto decimal. En último caso, la conversión la efectúa el programa en el momento de la lectura.
4. Aquellos datos que vengan referidos por una lista de variables necesitarán tantas tarjetas o registros como variables haya. Al final de la lista se introducirá en otro registro un "0" para indicar que se ha terminado con esa orden de lectura.
5. La entrada de datos puede realizarse por tarjetas, pantalla o cinta magnética.

La secuencia de tarjetas o registros para la entrada de datos del programa PLP es constante a lo largo del mismo por su

interactividad. Comprendería una lectura previa y una lectura a la vista de los mensajes que el ordenador envía.

La lectura previa se realiza introduciendo los siguientes datos:

1. Titulación del problema.

FORMATO : (un máximo de 80 caracteres alfanuméricos)

2.a Identificación de los espacios. (a = 1, 2, ... n; n = n° de espacios)

FORMATO : (un máximo de 80 caracteres alfanuméricos)

3.a Matriz de incidencia (a = 1, 2, ... n; n = n° de espacios)

FORMATO: LA, LO, LF.

LA : n° de la arista (a partir del 5)

LO : n° de un vértice extremo (a partir del 5)

LF : n° del otro vértice extremo (a partir del 5)

La lectura a lo largo del programa responde a los mensajes - que el ordenador envía. Fundamentalmente es de dos tipos según se trate de añadir nuevas aristas o de sustituir aristas ya introducidas:

1.a Introducción de nuevas aristas. (a = 1, 2, ... m; m = n° de aristas introducidas)

FORMATO: LA, LO, LF.

LA : n° de la arista (a partir de la última que se había numerado)

LO : n° de un vértice extremo.

LF : n° del otro vértice extremo.

2.a Sustitución de aristas ($a=1,2, \dots h$; $h=n^\circ$ de aristas
sustituídas)

FORMATO: LA, LO, LF, MO, MF.

LA : n° de la arista que se sustituye.

LO : n° de un vértice extremo de la arista, antes de la
sustitución .

LF : n° del otro vértice extremo.

MO : n° del vértice extremo de la arista después de la -
sustitución.

MF : n° del otro vértice extremo.

Una vez que se ha obtenido una solución el ordenador pregunta si se desea terminar o, por el contrario, se quiere continuar. En el primer caso se introduce un: "0", en el segundo - un: "1".

Además de la entrada de datos propiamente dicha el usuario - presenta una serie de opciones para la salida de resultados que conviene destacar. Estas operaciones se les ofrece al - pulsar distintos "siwtchs" del ordenador. Son las siguientes:

SIWTCH 1 : Se desea imprimir las coordenadas de los grafos o subgrafos geométricos trazados.

SIWTCH 2 : No se quiere trazar en el "plotter" el grafo o subgrafo geométrico analizado.

SIWTCH 3 : No se desea trazar el grafo ya triangulado.

SIWTCH 4 : Se indica que se trace el grafo triangulado - después de transformadas las coordenadas de - sus vértices.

SIWTCH 5 : Se desea imprimir las coordenadas de los vértices del grafo dual.

A lo largo de la ejecución del programa el ordenador puede - enviar mensajes de errores. Obedecerán a las siguientes causas:

ERROR 1 : El número de vértices o aristas es mayor que - el permitido.

ERROR 2 : La valencia de un vértice es mayor de lo que - se ha previsto.

ERROR 3 : La componente biconexa analizada no cumple la relación de Euler y, por tanto, no es planar.

ERROR 4 : No existe un circuito de vértices orientados - al exterior.

El programa que se presenta está elaborado para un número máximo de 150 aristas y 50 vértices.

A.II.1.3. Listado del Programa

FTN4		PLP-000
#ENAC(IJK,3)		PLP-001
C		PLP-002
C		PLP-003
C	*****	PLP-004
C	PRDGRAM PLP	PLP-005
C	*****	PLP-006
C		PLP-007
C	-----	PLP-008
C	PROGRAMA PARA GENERACION DE ESQUEMAS DISTRIBUTIVOS EN PLANTA	PLP-009
C	-----	PLP-010
C		PLP-011
C	PROGRAMA PRINCIPAL LECTURA DE DATOS	PLP-012
C		PLP-013
C	TABLAS UTILIZADAS:	PLP-014
C		PLP-015
C	IB(150,50) ... MATRIZ DE INCIDENCIA	PLP-016
C	IA(100,50) ... MATRIZ DE ADYACENCIA	PLP-017
C	IAA(50,10) ... MATRIZ DE ADYACENCIA ORDENADA	PLP-018
C		PLP-019
C		PLP-020
C	LISTADO DE ERRORES:	PLP-021
C		PLP-022
C	ERROR 1 EL NUMERO DE VERTICES O ARISTAS ES MAYOR RUE EL PERMI-	PLP-023
C		PLP-024
C	ERROR 2 LA VALENCIA DE UN VERTICE ES MAYOR DE LO QUE SE HA PRE-	PLP-025
C		PLP-026
C	ERROR 3 LA COMPONENTE BICONEXA ANALIZADA NO CUMPLE LA RELACION	PLP-027
C		PLP-028
C	DE EULER Y, POR TANTO, NO ES PLANAR.	

C	ERROR 4	NO EXISTE UN CIRCUITO DE VERTICES ORIENTADOS AL EXTERIOR.	PLP-029
C				PLP-030
C				PLP-031
C				PLP-032
C	ESCRITURAS OPCIONALES:			PLP-033
C				PLP-034
C	SWITCH 1	SE DESEA IMPRIMIR LAS COORDENADAS DE LOS GRAFOS O SUB-	PLP-035
C			GRAFOS GEOMETRICOS TRAZADOS.	PLP-036
C	SWITCH 2	NO SE DESEA TRAZAR EL GRAFO O SUBGRAFO GEOMETRICO ANA-	PLP-037
C			LIZADO.	PLP-038
C	SWITCH 3	NO SE DESEA DIBUJAR EL GRAFO YA TRIANGULADO.	PLP-039
C	SWITCH 4	SE DESEA TRAZAR EL GRAFO TRIANGULADO TRANSFORMADO.	PLP-040
C	SWITCH 5	SE DESEA IMPRIMIR LAS COORDENADAS DE LOS VERTICES DEL	PLP-041
C			GRAFO DUAL.	PLP-042
C				PLP-043
C				PLP-044
C			COMMON IPAR(4),NAME(3)	PLP-045
C			COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKT	PLP-046
C			COMMON NON(40),JAUX(2200)	PLP-047
C			COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IAA(50,10)	PLP-048
C				PLP-049
C	PARAMETROS DE ENTRADAS Y SALIDAS			PLP-050
C				PLP-051
C			CALL RMPAR(IPAR)	PLP-052
C			IENT=IPAR(1)	PLP-053
C			ISAL=IPAR(2)	PLP-054
C			IEAUX=IPAR(3)	PLP-055
C			ISAUX=IPAR(4)	PLP-056
C			IF(IENT.EQ.0) IENT=1	PLP-057
C			IF(ISAL.EQ.0) ISAL=6	PLP-058
C			IF(IEAUX.EQ.0) IEAUX=1	PLP-059
C			IF(ISAUX.EQ.0) ISAUX=6	PLP-060
C				PLP-061
C	** LLAMADA A LA SUBROUTINA DE LECTURA **			PLP-062
C				PLP-063
C			CALL LEP(IENT,ISAL,ISAUX,NV,NA,NON,IB)	PLP-064
C				PLP-065
C	** LLAMADA AL SEGMENTO PLI QUE APLICA AL GRAFO EL TEST DE CONECTIVIDAD **			PLP-066
C				PLP-067
C				PLP-068
C			NAME(1)=2HPL	PLP-069
C			NAME(2)=2H1	PLP-070
C			NAME(3)=2H	PLP-071
C			CALL EXEC(3,NAME)	PLP-072
C			STOP	PLP-073
C			END	PLP-074
C				LEP-075
C				LEP-076
C			SUBROUTINE LEP(IENT,ISAL,ISAUX,N,NA,NON,IB)	LEP-077
C			*****	LEP-078
C				LEP-079
C	SUBROUTINA PARA LECTURA DE DATOS			LEP-080
C				LEP-081
C			EMA IB(150,50)	LEP-082
C			DIMENSION NON(40)	LEP-083
C			WRITE(ISAL,1001)	LEP-084
C			WRITE(ISAL,2001)	LEP-085
C			READ(IENT,2002) NON	LEP-086
C			WRITE(ISAL,2003) NON	LEP-087
C				LEP-088


```

4001 FORMAT(/1X'MENSAJE (ED): ASIGNE UN NOMBRE A CADA UNO DE LOS ESPA LEP-150
&CIDS//16X'AL FINAL DE LA LISTA INTRODUCE:0//1X'ESP NOMBRE'/1X'- LEP-151
&-- -----') LEP-152
4002 FORMAT(I4,2X,40A2) LEP-153
5001 FORMAT(I4,2I3) LEP-154
END LEP-155
C LEP-156
C LEP-157
SUBROUTINE LE1(IEAUX,ISAL,ISAUX,K,IA,IB) LEP-158
***** LEP-159
C LEP-160
C SUBROUTINA QUE LEE LAS ARISTAS QUE SUSTITUYEN A OTRAS LEP-161
C LEP-162
EN A IB(150,50),IA(100,50) LEP-163
WRITE(ISAL,3001) LEP-164
10 READ(IEAUX,*) LA,MO,MF,LO,LF LEP-165
IF(LA.EQ.0) GO TO 50 LEP-166
WRITE(ISAUX,5001) LA,MO,MF,LO,LF LEP-167
IB(LA,MO)=0 LEP-168
IB(LA,MF)=0 LEP-169
IA(MO,MF)=0 LEP-170
IA(MF,MO)=0 LEP-171
IB(LA,LO)=(-1)**K LEP-172
IB(LA,LF)=(-1)**K LEP-173
IA(LO,LF)=1 LEP-174
IA(LF,LO)=1 LEP-175
GO TO 10 LEP-176
50 RETURN LEP-177
C LEP-178
C FORMATS LEP-179
C LEP-180
3001 FORMAT(/16X'LOS DATOS SE DARAN EN EL SIGUIENTE ORDEN:*/16X'- NUMER LEP-181
&O DE LA ARISTA*/16X'- VERTICE PRIMITIVO*/16X'- VERTICE PRIMITIVO*/ LEP-182
&16X'- VERTICE ACTUAL*/16X'- VERTICE ACTUAL*/16X'AL FINAL DE LA LIS LEP-183
&TA INTRODUCE: 0//1X' A VP VP VA VA*/1X' - - - - -') LEP-184
5001 FORMAT(5I4) LEP-185
END LEP-186
C LEP-187
C LEP-188
SUBROUTINE LE2(IEAUX,ISAL,ISAUX,K,NA,NV,IA,IB) LEP-189
***** LEP-190
C LEP-191
C SUBROUTINA QUE LEE LAS NUEVAS ARISTAS INTRODUCIDAS LEP-192
C LEP-193
EN A IB(150,50),IA(100,50) LEP-194
WRITE(ISAL,3001) NA LEP-195
WRITE(ISAL,3002) LEP-196
10 READ(IEAUX,*) LA,LO,LF LEP-197
IF(LA.EQ.0) GO TO 50 LEP-198
WRITE(ISAL,5001) LA,LO,LF LEP-199
NA=NA+1 LEP-200
DO 20 J=1,NV LEP-201
IB(NA,J)=0 LEP-202
20 CONTINUE LEP-203
IB(NA,LO)=(-1)**K LEP-204
IB(NA,LF)=(-1)**K LEP-205
IA(LO,LF)=1 LEP-206
IA(LF,LO)=1 LEP-207
GO TO 10 LEP-208
50 RETURN LEP-209

```

C		LE2-210
C	FORMATOS	LE2-211
C		LE2-212
	3001 FORMAT(/1X"MENSAJE :ISR: EL NUMERO DE ARISTAS ACTUALES ES:"14)	LE2-213
	3002 FORMAT(/1X"MENSAJE :IED: INTRODUCE LAS NUEVAS ARISTAS"/16X"AL F	LE2-214
	&INAL DE LA LISTA INTRODUCE: 0"/1X" A V V"/1X" - - -")	LE2-215
	5001 FORMAT(14,213)	LE2-216
	END	LE2-217
C		AD1-218
C		AD1-219
	SUBROUTINE AD1(EV,A,B,FIC,PTR,V,E,IB)	AD1-220
	*****	AD1-221
C		AD1-222
C	SUBROUTINA QUE ALMACENA UN ELEMENTO NUEVO EN UN CONJUNTOAS	AD1-223
C		AD1-224
	EMA IB(150,50)	AD1-225
	INTEGER A,B,PTR,V,E,EV	AD1-226
	INTEGER FIC(EV)	AD1-227
	DO 50 I=1,E	AD1-228
	IF(IB(I,A).EQ.0.OR.IB(I,B).EQ.0) GO TO 50	AD1-229
	PTR=PTR+1	AD1-230
	FIC(PTR)=I	AD1-231
	GO TO 60	AD1-232
	50 CONTINUE	AD1-233
	60 RETURN	AD1-234
	END	AD1-235
C		AD2-236
C		AD2-237
	SUBROUTINE AD2(EA,A,B,FIC,PTR)	AD2-238
	*****	AD2-239
C		AD2-240
C	SUBROUTINA QUE ALMACENA DOS ELEMENTOS NUEVOS EN UN CONJUNTO	AD2-241
C		AD2-242
	INTEGER EA,A,B,PTR	AD2-243
	INTEGER FIC(EA)	AD2-244
	PTR=PTR+2	AD2-245
	FIC(PTR-1)=A	AD2-246
	FIC(PTR)=B	AD2-247
	RETURN	AD2-248
	END	AD2-249
C		AD3-250
C		AD3-251
	SUBROUTINE AD3(FF,II,JJ,KK,B,K)	AD3-252
	*****	AD3-253
C		AD3-254
C	SUBROUTINA QUE ALMACENA TRES ELEMENTOS EN UN NUEVO CONJUNTO	AD3-255
C		AD3-256
	INTEGER FF	AD3-257
	INTEGER B(K)	AD3-258
	B(FF+1)=II	AD3-259
	B(FF+2)=JJ	AD3-260
	B(FF+3)=KK	AD3-261
	FF=FF+3	AD3-262
	RETURN	AD3-263
	END	AD3-264
C		SU2-265
C		SU2-266
	SUBROUTINE SD2(I,FF,B,K)	SU2-267
	*****	SU2-268
C		SU2-269
C	SUBROUTINA QUE SUPRIME DOS ELEMENTOS CONSECUTIVOS DE UN CONJUNTO	SU2-270

C	INTEGER FF	SU2-271
	INTEGER B(K)	SU2-272
	IF(FF.LE.2) GO TO 20	SU2-273
	IF(I.EQ.FF.OR.I.EQ.0) GO TO 15	SU2-274
	DO 10 J=I-1,FF-2	SU2-275
	B(J)=B(J+2)	SU2-276
10	CONTINUE.	SU2-277
15	B(FF)=0	SU2-278
	B(FF-1)=0	SU2-279
	FF=FF-2	SU2-280
	GO TO 30	SU2-281
20	FF=0	SU2-282
	B(1)=0	SU2-283
	B(2)=0	SU2-284
30	RETURN	SU2-285
	END	SU2-286
C		SU2-287
C		SU2-288
	SUBROUTINE SU3(I,FF,B,K)	SU3-289
	*****	SU3-290
C		SU3-291
C		SU3-292
C	SUBROUTINA QUE SUPRIME TRES ELEMENTOS CONSECUTIVOS DE UN CONJUNTO	SU3-293
C		SU3-294
	INTEGER FF	SU3-295
	INTEGER B(K)	SU3-296
	IF(FF.LE.3) GO TO 20	SU3-297
	IF(I.EQ.FF.OR.I.EQ.0) GO TO 15	SU3-298
	DO 10 J=I-2,FF-3	SU3-299
	B(J)=B(J+3)	SU3-300
10	CONTINUE	SU3-301
15	B(FF)=0	SU3-302
	B(FF-1)=0	SU3-303
	B(FF-2)=0	SU3-304
	FF=FF-3	SU3-305
	GO TO 30	SU3-306
20	FF=0	SU3-307
	B(1)=0	SU3-308
	B(2)=0	SU3-309
	B(3)=0	SU3-310
30	RETURN	SU3-311
	END	SU3-312
C		VA1-313
C		VA1-314
	SUBROUTINE VA1(SAVE,KKK)	VA1-315
	*****	VA1-316
C		VA1-317
C		VA1-318
C	SUBROUTINA DE OPERACION AUXILIAR	VA1-319
C		VA1-320
	INTEGER SAVE	VA1-321
	SAVE=KKK	VA1-322
	RETURN	VA1-323
	END	VA1-324
C		VA2-325
C		VA2-326
	SUBROUTINE VA2(MMM,NNN)	VA2-327
	*****	VA2-328
C		VA2-329
C	SUBROUTINA DE OPERACION AUXILIAR	VA2-330
C		

	NNN=NNN	VA2-331
	RETURN	VA2-332
	END	VA2-333
C		VA3-334
C		VA3-335
	SUBROUTINE VA3(NNN,NNN,SAVE)	VA3-336
	*****	VA3-337
C		VA3-338
C	SUBROUTINA DE OPERACION AUXILIAR	VA3-339
C		VA3-340
	INTEGER SAVE	VA3-341
	NNN=NNN	VA3-342
	NNN=SAVE	VA3-343
	RETURN	VA3-344
	END	VA3-345
C		CA1-346
C		CA1-347
	SUBROUTINE CA1(NV,PTBJ1,PTBJ2,ESTA,IAA)	CA1-348
	*****	CA1-349
C		CA1-350
C	SUBROUTINA QUE MODIFICA LA ORDENACION DE LA PLANARIDAD PARA ADAPTARLA A	CA1-351
C	LA NECESARIA PARA EL ANALISIS DE TRICONECTIVIDAD	CA1-352
C		CA1-353
	ENR IAA(50,10)	CA1-354
	INTEGER J,W	CA1-355
	INTEGER PTBJ1(NV),PTBJ2(NV),ESTA(NV)	CA1-356
	DO 10 I=1,NV	CA1-357
	ESTA(I)=0	CA1-358
10	CONTINUE	CA1-359
	DO 200 I=1,NV	CA1-360
	DO 100 J=1,9	CA1-361
	IF(IAA(I,J).EQ.0) GO TO 110	CA1-362
	W=IAA(I,J)	CA1-363
	IF(I.LT.W) GO TO 60	CA1-364
	ESTA(W)=2*W+1	CA1-365
	GO TO 100	CA1-366
60	IF(PTBJ2(W).LT.I) ESTA(W)=2*PTBJ1(W)	CA1-367
	IF(PTBJ2(W).GE.I) ESTA(W)=2*PTBJ1(W)+1	CA1-368
100	CONTINUE	CA1-369
110	DO 150 J=2,9	CA1-370
	IF(IAA(I,J).EQ.0) GO TO 200	CA1-371
	W=IAA(I,J)	CA1-372
	U=IAA(I,J-1)	CA1-373
	IF(ESTA(W).GE.ESTA(U)) GO TO 150	CA1-374
	IAVE=IAA(I,J)	CA1-375
	IAA(I,J)=IAA(I,J-1)	CA1-376
	IAA(I,J-1)=IAVE	CA1-377
	GO TO 110	CA1-378
150	CONTINUE	CA1-379
200	CONTINUE	CA1-380
	RETURN	CA1-381
	END	CA1-382
C		CA2-383
C		CA2-384
	SUBROUTINE CA2(NV,II,NUMB,VECT,VAUX)	CA2-385
	*****	CA2-386
C		CA2-387
C	SUBROUTINA QUE CAMBIA EL ORDEN DE ALMACENAMIENTO DE UN VECTOR PARA HA-	CA2-388
C	CERLA SEGUN LA ANTIGUA NUMERACION	CA2-389
C		CA2-390
	ENR VAUX(NV),VECT(NV)	CA2-391

	DIMENSION NUMB(NV)	CA2-392
	DO 10 I=1,NV	CA2-393
10	VAUX(I)=0	CA2-394
	DO 100 I=1,II	CA2-395
	DO 50 J=1,NV	CA2-396
	IF(NUMB(J).NE.I) GO TO 50	CA2-397
	VAUX(J)=VECT(I)	CA2-398
	GO TO 100	CA2-399
50	CONTINUE	CA2-400
100	CONTINUE	CA2-401
	DO 150 I=1,NV	CA2-402
	VECT(I)=VAUX(I)	CA2-403
150	CONTINUE	CA2-404
	RETURN	CA2-405
	END	CA2-406
C		CA3-407
C		CA3-408
	SUBROUTINE CA3(NV,VECT,NUENU,VAUX)	CA3-409
	*****	CA3-410
C		CA3-411
C	SUBROUTINA PARA CAMBIAR VECTORES DE ACUERDO A UNA NUMERACION	CA3-412
C		CA3-413
	INTEGER VECT(NV),VAUX(NV),NUENU(NV)	CA3-414
	DO 10 I=1,NV	CA3-415
10	VAUX(I)=0	CA3-416
	DO 50 I=1,NV	CA3-417
	IF(VECT(I).LE.0) GO TO 50	CA3-418
	VAUX(NUENU(I))=VECT(I)	CA3-419
50	CONTINUE	CA3-420
	DO 60 I=1,NV	CA3-421
	VECT(I)=VAUX(I)	CA3-422
60	CONTINUE	CA3-423
	RETURN	CA3-424
	END	CA3-425
C		CA4-426
C		CA4-427
	SUBROUTINE CA4(NV,IAA,NUENU,VAUX,MN)	CA4-428
	*****	CA4-429
C		CA4-430
C	SUBROUTINA PARA CAMBIAR EL ORDEN DE LAS FILAS DE LA MATRIZ DE ORDENA -	CA4-431
C	CION	CA4-432
C		CA4-433
	EMA IAA(50,10)	CA4-434
	INTEGER VAUX(10),NUENU(NV)	CA4-435
	DO 1 I=1,MN	CA4-436
1	IAA(I,10)=0	CA4-437
	DO 5 I=1,NV	CA4-438
	DO 3 J=1,10	CA4-439
	IF(IAA(I,J).EQ.0) GO TO 5	CA4-440
	IF(IAA(I,J).LT.0) IAA(I,J)=-IAA(I,J)	CA4-441
3	CONTINUE	CA4-442
5	CONTINUE	CA4-443
	DO 10 I=1,10	CA4-444
10	VAUX(I)=0	CA4-445
	K=0	CA4-446
20	K=K+1	CA4-447
	IF(K.GT.MN) GO TO 70	CA4-448
	DO 30 I=1,9	CA4-449
	VAUX(I)=IAA(K,I)	CA4-450
30	CONTINUE	CA4-451

	DO 40 I=1,NV	CA4-452
	IF(K.EB.NUENU(I)) GO TO 50	CA4-453
	40 CONTINUE	CA4-454
	50 J=I	CA4-455
	51 IF(IAA(J,10).EQ.0) GO TO 55	CA4-456
	J=IAA(J,10)	CA4-457
	GO TO 51	CA4-458
	55 DO 60 I=1,9	CA4-459
	IAA(K,I)=IAA(J,I)	CA4-460
	IAA(J,I)=VAUX(I)	CA4-461
	60 CONTINUE	CA4-462
	IAA(K,10)=J	CA4-463
	GO TO 20	CA4-464
	70 RETURN	CA4-465
	END	CA4-466
C		PRO-467
C		PRO-468
	SUBROUTINE PRO(V,EE,EEV,PUNTO,VAL,FRENE,SIG,CABE)	PRO-469
	*****	PRO-470
C		PRO-471
C	SUBROUTINA QUE ELABORA EL CONJUNTO CABE PARA ALMACENAR LAS INCIDENCIAS	PRO-472
C		PRO-473
	INTEGER V,EE,EEV,FRENE,PUNTO,VAL	PRO-474
	INTEGER SIG(EEV),CABE(EE)	PRO-475
	FRENE=FRENE+1	PRO-476
	SIG(FRENE)=SIG(PUNTO)	PRO-477
	SIG(PUNTO)=FRENE	PRO-478
	CABE(FRENE-V)=VAL	PRO-479
	RETURN	PRO-480
	END	PRO-481
C		TRA-482
C		TRA-483
	SUBROUTINE TRA(V,E,EE,LISA,IB)	TRA-484
	*****	TRA-485
C		TRA-486
C	SUBROUTINA QUE ELABORA EL LISTADO DE ARISTAS POR SUS VERTICES EXTREMOS	TRA-487
C		TRA-488
	ENA IB(150,50)	TRA-489
	INTEGER V,E,EE	TRA-490
	INTEGER LISA(EE)	TRA-491
	K=0	TRA-492
	DO 100 I=1,E	TRA-493
	DO 50 J=1,V	TRA-494
	IF(IB(I,J).EQ.0) GO TO 50	TRA-495
	K=K+1	TRA-496
	LISA(K)=J	TRA-497
	50 CONTINUE	TRA-498
	100 CONTINUE	TRA-499
	RETURN	TRA-500
	END	TRA-501
C		DRD-502
C		DRD-503
	SUBROUTINE DRD(1SAL,NV,NA,K2,FROND,NUNB,PTBJ1,PTBJ2,FI,IB,IAA)	DRD-504
	*****	DRD-505
C		DRD-506
C	SUBROUTINA QUE ORDENA LOS TERMINOS DE LA NUEVA MATRIZ DE ADYACENCIA	DRD-507
C	USANDO EL D.F.S.	DRD-508
C		DRD-509
	ENA IB(150,50),IAA(50,10)	DRD-510
	INTEGER V,N	DRD-511
	INTEGER PTBJ1(NV),PTBJ2(NV),NUNB(NV),FI(NA),FROND(NA)	DRD-512

DD 80 I=1,NA	DRD-513
V=0	DRD-514
W=0	DRD-515
DD 30 J=1,NV	DRD-516
IF(IB(I,J).LT.0.AND.V.EQ.0) GO TO 20	DRD-517
IF(IB(I,J).LT.0.AND.V.NE.0) GO TO 40	DRD-518
GO TO 30	DRD-519
20 V=J	DRD-520
30 CONTINUE	DRD-521
40 W=J	DRD-522
DD 50 J=1,K2	DRD-523
IF(I.EQ.FROND(J)) GO TO 60	DRD-524
50 CONTINUE	DRD-525
IF(NUMB(V).LT.NUMB(W)) GO TO 55	DRD-526
LL=V	DRD-527
V=W	DRD-528
W=LL	DRD-529
55 IF(PTBJ2(W).GE.NUMB(V)) FI(I)=2*PTBJ1(W)	DRD-530
IF(PTBJ2(W).LT.NUMB(V)) FI(I)=2*PTBJ1(W)+1	DRD-531
GO TO 80	DRD-532
60 IF(NUMB(V).GT.NUMB(W)) GO TO 65	DRD-533
LL=V	DRD-534
V=W	DRD-535
W=LL	DRD-536
65 FI(I)=2*NUMB(W)	DRD-537
80 CONTINUE	DRD-538
DD 200 J=1,2*NV+1	DRD-539
DD 190 I=1,NA	DRD-540
IF(FI(I).NE.J) GO TO 190	DRD-541
VW=I	DRD-542
V=0	DRD-543
W=0	DRD-544
DD 120 JJ=1,NV	DRD-545
IF(IB(I,JJ).LT.0.AND.V.EQ.0) GO TO 110	DRD-546
IF(IB(I,JJ).LT.0.AND.V.NE.0) GO TO 130	DRD-547
GO TO 120	DRD-548
110 V=JJ	DRD-549
120 CONTINUE	DRD-550
130 W=JJ	DRD-551
DD 140 JJ=1,K2	DRD-552
IF(VW.EQ.FROND(JJ)) GO TO 150	DRD-553
140 CONTINUE	DRD-554
IF(NUMB(V).LT.NUMB(W)) GO TO 160	DRD-555
LL=V	DRD-556
V=W	DRD-557
W=LL	DRD-558
GO TO 160	DRD-559
150 IF(NUMB(V).GT.NUMB(W)) GO TO 160	DRD-560
LL=V	DRD-561
V=W	DRD-562
W=LL	DRD-563
160 DD 170 JJ=1,9	DRD-564
IF(IAA(V,JJ).EQ.0) GO TO 180	DRD-565
170 CONTINUE	DRD-566
C	DRD-567
C SALIDA DE ERROR SI LA VALENCIA ES MAYOR DE LO QUE PERMITE LA CAPACIDAD	DRD-568
C DE ALMACENAMIENTO	DRD-569
C	DRD-570
IERR=NER(ISAL,2)	DRD-571
STOP 0002	DRD-572

180	TAA(V, JJ)=M	BRD-573
190	CONTINUE	BRD-574
200	CONTINUE	BRD-575
	RETURN	BRD-576
	END	BRD-577
C		PAT-578
C		PAT-579
	SUBROUTINE PAT(UU, K1, T, CAN, EEV)	PAT-580
	*****	PAT-581
C		PAT-582
C	SUBROUTINA QUE ESTABLECE UNA OPERACION DETERMINADA SEGUN DOS VERTICES	PAT-583
C	ESTEN O NO SOBRE UN MISMO CANINO	PAT-584
C		PAT-585
	INTEGER UU, T, EEV	PAT-586
	INTEGER CAN(EEV)	PAT-587
	DO 100 I=1, K1-2	PAT-588
	J1=0	PAT-589
	J2=0	PAT-590
	IF(CAN(I).GE.0) GO TO 100	PAT-591
	DO 50 J=I+1, K1-1	PAT-592
	IF(CAN(J).LT.0) GO TO 60	PAT-593
	IF(CAN(J).EQ.UU.AND.J2.EQ.0) J1=1	PAT-594
	IF(CAN(J).EQ.T.AND.J1.NE.0) J2=1	PAT-595
50	CONTINUE	PAT-596
60	IF(J1.NE.0.AND.J2.NE.0) GO TO 110	PAT-597
100	CONTINUE	PAT-598
	GO TO 120	PAT-599
110	JU=T	PAT-600
120	RETURN	PAT-601
	END	PAT-602
C		PA1-603
C		PA1-604
	SUBROUTINE PA1(FLAG, JJ, SAV1, SAV2, HH, T, U, COMPO, GRADO, NV, EEE)	PA1-605
	*****	PA1-606
C		PA1-607
C	SUBROUTINA QUE REALIZA OPERACIONES AUXILIARES	PA1-608
C		PA1-609
	INTEGER FLAG, SAV1, SAV2, HH, T, U, EEE	PA1-610
	INTEGER GRADO(NV), COMPO(EEE)	PA1-611
	FLAG=0	PA1-612
	JJ=JJ+1	PA1-613
C		PA1-614
C	** LLAMADAS A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO COMPO	PA1-615
C	**	PA1-616
C		PA1-617
	CALL AC2(EEE, SAV1, SAV2, COMPO, HH)	PA1-618
	CALL AD2(EEE, T, U, COMPO, HH)	PA1-619
	CALL AD2(EEE, T, U, COMPO, HH)	PA1-620
	HH=HH+1	PA1-621
	COMPO(HH)=-JJ	PA1-622
	GRADO(T)=GRADO(T)-1	PA1-623
	GRADO(U)=GRADO(U)-1	PA1-624
	RETURN	PA1-625
	END	PA1-626
C		PA2-627
C		PA2-628
	SUBROUTINE PA2(CG, U, T, JJ, ESTA, GRADO, PAD, UU, K1, CAN, V, EEV, NV, EE)	PA2-629
	*****	PA2-630
C		PA2-631
C	SUBROUTINA QUE REALIZA OPERACIONES AUXILIARES	PA2-632
C		PA2-633

	INTEGER GG,U,T,UU,V,EEV,EE	PA2-634
	INTEGER PAD(NV),GRADO(NV),CAN(EEV),ESTA(EE)	PA2-635
C		PA2-636
C	** LLAMADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO ESTA **	PA2-637
C		PA2-638
	CALL AD2(EE,U,T,ESTA,GG)	PA2-639
	GRADO(T)=GRADO(T)+1	PA2-640
	GRADO(U)=GRADO(U)+1	PA2-641
	PAD(T)=U	PA2-642
C		PA2-643
C	** LLAMADA A LA SUBROUTINA QUE REALIZA UNA SERIE DE OPERACIONES SEGUN	PA2-644
C	DOS VERTICES ESTEN O NO SOBRE UN MISMO CAMINO GENERADO **	PA2-645
C		PA2-646
	CALL PAT(UU,K1,T,CAN,EEV)	PA2-647
	V=T	PA2-648
	RETURN	PA2-649
	END	PA2-650
C		NUL-651
C		NUL-652
	SUBROUTINE MUL(EE,EEE,NV,GG,V,U,HH,JJ,ESTA,GRADO,COMP,IOF)	NUL-653
	*****	NUL-654
C		NUL-655
C	SUBROUTINA PARA ELIMINAR LAS ARISTAS MULTIPLES DE LAS COMPONENTES TRI-	NUL-656
C	CONEXAS	NUL-657
C		NUL-658
	INTEGER V,W,GG,HH,XX,YY,EE,EEE	NUL-659
	INTEGER ESTA(EE),COMP(EEE),GRADO(NV)	NUL-660
	DO 100 I=GG,1,-2	NUL-661
	XX=ESTA(I-1)	NUL-662
	YY=ESTA(I)	NUL-663
	IF((XX.HE.V.DR.YY.HE.W).AND.(XX.HE.U.DR.YY.HE.V)) GO TO 100	NUL-664
	JJ=JJ+1	NUL-665
C		NUL-666
C	** LLAMADAS A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO COMP	NUL-667
C	**	NUL-668
C		NUL-669
	CALL AD2(EEE,XX,YY,COMP,HH)	NUL-670
	CALL AD2(EEE,V,W,COMP,HH)	NUL-671
	CALL AD2(EEE,V,W,COMP,HH)	NUL-672
	HH=HH+1	NUL-673
	COMP(HH)=-JJ	NUL-674
	IF(IOF.EQ.2) GO TO 50	NUL-675
		NUL-676
C		NUL-677
C	** LLAMADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA	NUL-678
C	**	NUL-679
C		NUL-680
	CALL SU2(I,GG,ESTA,EE)	NUL-681
	50 GRADO(V)=GRADO(V)-1	NUL-682
	GRADO(W)=GRADO(W)-1	NUL-683
	100 CONTINUE	NUL-684
	RETURN	NUL-685
	END	TUT-686
C		TUT-687
C		TUT-688
	SUBROUTINE TUT(MA,NV,R,NUME,X,Y,IB,LM,ISAL)	TUT-689
	*****	TUT-690
C		TUT-691
C	DIBUJA EL GRAFO D SUBGRAFOS POR EL ALGORITMO DE TUTTE	TUT-692
C		TUT-693
	EMA IB(150,50),X(NV),Y(NV)	

	DIMENSION NUNE(NV)	TUT-694
	CALL PLTLU(10)	TUT-695
	WRITE(15AL,3001)	TUT-696
	CALL PLDT(R,R+0.40,-3)	TUT-697
C		TUT-698
C	NUMERACION DE LOS VERTICES	TUT-699
C		TUT-700
	IF(LN.LT.5) GO TO 5	TUT-701
	CALL PLDT(10.0,-44.0,-3)	TUT-702
	LN=0	TUT-703
	GO TO 7	TUT-704
	5 IF(LN.NE.0) CALL PLDT(0.,11.2,-3)	TUT-705
	LN=LN+1	TUT-706
	7 DO 10 J=1,NV	TUT-707
	IF(NUNE(J).EQ.0) GO TO 10	TUT-708
	RJ=J	TUT-709
	CALL NUNE(K(J)-0.25,Y(J)-0.25,0.15,RJ,0.,-1)	TUT-710
	10 CONTINUE	TUT-711
C		TUT-712
C	TRAZADO DE LAS ARISTAS	TUT-713
C		TUT-714
	DO 100 I=1,NA	TUT-715
	J1=0	TUT-716
	J2=0	TUT-717
	DO 50 J=1,NV	TUT-718
	IF(1B(I,J).EQ.0) GO TO 50	TUT-719
	IF(1B(I,J).NE.-1)GO TO 100	TUT-720
	IF(J1.NE.0) GO TO 60	TUT-721
	J1=J	TUT-722
	50 CONTINUE	TUT-723
	60 J2=J	TUT-724
	X1=X(J1)	TUT-725
	Y1=Y(J1)	TUT-726
	X2=X(J2)	TUT-727
	Y2=Y(J2)	TUT-728
	CALL PLDT(X1,Y1,3)	TUT-729
	CALL PLDT(X2,Y2,2)	TUT-730
	100 CONTINUE	TUT-731
	CALL PLDT(-R,-R-0.40,-3)	TUT-732
	RETURN	TUT-733
C		TUT-734
C	FORMATOS	TUT-735
C		TUT-736
	3001 FORMAT(/IX*MENSAJE 11SR11 SE DIBUJA EN EL PLOTTER*)	TUT-737
	END	TUT-738
C		CIR-739
C		CIR-740
	SUBROUTINE CIR(NV,NA,V1,NUMB,PILA,PAD,CIRCO,K.CODE,IA)	CIR-741
	*****	CIR-742
C		CIR-743
C	SUBROUTINA QUE DETERMINA UN CIRCUITO PARA EL TRAZADO DEL GRAFO POR TU-	CIR-744
C	ITE	CIR-745
C		CIR-746
	ENA IA(100,50)	CIR-747
	INTEGER V1,V,W,CODE,S	CIR-748
	INTEGER NUMB(V1),PILA(V1),PAD(NV),CIRCO(NV)	CIR-749
C		CIR-750
C	PUESTA A CERO	CIR-751
C		CIR-752
	DO 5 I=1,NV	CIR-753
	CIRCO(I)=0	CIR-754

NUMB(I)=0	CIR-755
PAD(I)=0	CIR-756
PILA(I)=0	CIR-757
5 CONTINUE	CIR-758
NUMB(V1)=0	CIR-759
PILA(V1)=0	CIR-760
C	CIR-761
C DETERMINACION DEL VERTICE INICIAL DEL CIRCUITO. OBTENCION DE ESTE UL-	CIR-762
C TING	CIR-763
C	CIR-764
I1=IA(1,2)	CIR-765
I2=IA(3,4)	CIR-766
I3=IA(1,3)	CIR-767
I4=IA(2,4)	CIR-768
DO 20 I=1,NV	CIR-769
DO 20 J=1,NV	CIR-770
IF(IA(I,J).EQ.1) GO TO 30	CIR-771
20 CONTINUE	CIR-772
30 S=I	CIR-773
LN=0	CIR-774
V=S	CIR-775
CODE=1	CIR-776
KONT=1	CIR-777
NUMB(V)=CODE	CIR-778
PILA(KONT)=V	CIR-779
40 KT=0	CIR-780
45 DO 50 J=1,NV	CIR-781
IF(IA(V,J).NE.1) GO TO 50	CIR-782
IF(NUMB(J).NE.0.AND.KT.EQ.0) GO TO 50	CIR-783
IA(V,J)=0	CIR-784
IA(J,V)=0	CIR-785
M=J	CIR-786
GO TO 60	CIR-787
50 CONTINUE	CIR-788
IF(KT.NE.0) GO TO 55	CIR-789
KT=KT+1	CIR-790
GO TO 45	CIR-791
55 IF(KONT.EQ.1) GO TO 80	CIR-792
PILA(KONT)=0	CIR-793
KONT=KONT-1	CIR-794
V=PILA(KONT)	CIR-795
GO TO 40	CIR-796
60 IF(NUMB(M).NE.0) GO TO 70	CIR-797
CODE=CODE+1	CIR-798
KONT=KONT+1	CIR-799
NUMB(M)=CODE	CIR-800
PILA(KONT)=M	CIR-801
PAD(M)=V	CIR-802
V=M	CIR-803
GO TO 40	CIR-804
70 IF(M.NE.LN) GO TO 75	CIR-805
IF(NUMB(M).LT.NUMB(JS)) GO TO 45	CIR-806
75 LN=V	CIR-807
JS=M	CIR-808
GO TO 45	CIR-809
80 IF(I1.NE.1.OR.I2.NE.1) GO TO 85	CIR-810
IF(I3.EQ.1.OR.I4.EQ.1) GO TO 85	CIR-811
CIRCO(1)=1	CIR-812
CIRCO(2)=4	CIR-813
CIRCO(3)=3	CIR-814

CIRCO(4)=2	CIR-015
K=4	CIR-016
GO TO 110	CIR-017
85 K=1	CIR-018
CIRCO(K)=JS	CIR-019
K=K+1	CIR-020
CIRCO(K)=LM	CIR-021
IS=LM	CIR-022
90 K=K+1	CIR-023
IF(PAD(IS).EQ.JS) GO TO 100	CIR-024
CIRCO(K)=PAD(IS)	CIR-025
IS=PAD(IS)	CIR-026
GO TO 90	CIR-027
100 K=K-1	CIR-028
C	CIR-029
C REMUNERACION DE LOS VERTICES	CIR-030
C	CIR-031
110 KT=0	CIR-032
DO 180 J=1, CODE	CIR-033
DO 150 I=1, NV	CIR-034
IF(HUMB(I).NE.J) GO TO 150	CIR-035
DO 130 IT=1, K	CIR-036
IF(I.NE.CIRCO(IT)) GO TO 130	CIR-037
KT=KT+1	CIR-038
GO TO 180	CIR-039
130 CONTINUE	CIR-040
HUMB(I)=HUMB(I)-KT	CIR-041
GO TO 130	CIR-042
150 CONTINUE	CIR-043
180 CONTINUE	CIR-044
DO 200 I=1, K	CIR-045
HUMB(CIRCO(I))=CODE-K+I	CIR-046
200 CONTINUE	CIR-047
RETURN	CIR-048
END	CIR-049
C	DBC-050
C	DBC-051
SUBROUTINE OBC(NV, NA, K, CODE, HUMB, ICIRC, XC, YC, BX, BY, VAUX, A, IB, NH,	DBC-052
AISAL, ISAUX)	DBC-053
*****	DBC-054
C	DBC-055
C SUBROUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES SEGUN EL AL-	DBC-056
C GORITMO DE TBTTE	DBC-057
C	DBC-058
ENA IB(150,59), A(50,59), XC(NV), YC(NV), BX(NV), BY(NV)	DBC-059
INTEGER CODE, VAUX(NV)	DBC-060
DIMENSION HUMB(NV), ICIRC(K)	DBC-061
C	DBC-062
C PUESTA A CERO	DBC-063
C	DBC-064
DO 10 I=1, NV	DBC-065
BX(I)=0	DBC-066
BY(I)=0	DBC-067
XC(I)=0	DBC-068
YC(I)=0	DBC-069
DO 10 J=1, NV	DBC-070
10 A(I, J)=0	DBC-071
C	DBC-072
C COORDENADAS DE LOS VERTICES DEL CIRCUITO	DBC-073
C	DBC-074
ANG=6.2831853/K	DBC-075

R=3.	OBC-876
DO 50 II=1,K	OBC-877
I=ICIRC(II)	OBC-878
ANG1=-ANG*(II-1)+3.1415927/2.	OBC-879
XC(NUMB(I))=R*COB(ANG1)	OBC-880
YC(NUMB(I))=R*SIN(ANG1)	OBC-881
50 CONTINUE	OBC-882
C	OBC-883
C ELABORACION DE LA SUBMATRIZ DE ADYACENCIA DE ACUERDO A LA NUEVA NUNE-	OBC-884
C RACION DE LOS VERTICES	OBC-885
C	OBC-886
DO 200 I=1,NA	OBC-887
KT=0	OBC-888
JJ=0	OBC-889
DO 150 J=1,NV	OBC-890
IF(I8(I,J).NE.-1) GO TO 150	OBC-891
DO 100 L=1,K	OBC-892
IF(J.EQ.ICIRC(L)) GO TO 110	OBC-893
100 CONTINUE	OBC-894
GO TO 120	OBC-895
110 JJ=NUMB(J)	OBC-896
GO TO 130	OBC-897
120 IF(KT.NE.0) GO TO 160	OBC-898
KT=KT+1	OBC-899
J1=NUMB(J)	OBC-900
150 CONTINUE	OBC-901
IF(KT.EB.0) GO TO 200	OBC-902
BX(J1)=BX(J1)+XC(JJ)	OBC-903
BY(J1)=BY(J1)+YC(JJ)	OBC-904
A(J1,J1)=A(J1,J1)+1	OBC-905
GO TO 200	OBC-906
160 J2=NUMB(J)	OBC-907
A(J1,J2)=-1.	OBC-908
A(J2,J1)=-1.	OBC-909
A(J1,J1)=A(J1,J1)+1	OBC-910
A(J2,J2)=A(J2,J2)+1	OBC-911
200 CONTINUE	OBC-912
C	OBC-913
C PIVOTAMIENTO DE LA MATRIZ Y DE LOS VECTORES	OBC-914
C	OBC-915
IF(CODE-K.EB.1) GO TO 360	OBC-916
DO 350 J=1, CODE-K	OBC-917
I=J+1	OBC-918
310 RH=A(I,J)/A(J,J)	OBC-919
A(I,J)=0	OBC-920
L=J+1	OBC-921
320 A(I,L)=A(I,L)-RH*A(J,L)	OBC-922
IF(L.EB.CODE-K) GO TO 330	OBC-923
L=L+1	OBC-924
GO TO 320	OBC-925
330 BX(I)=BX(I)-RH*BX(J)	OBC-926
BY(I)=BY(I)-RH*BY(J)	OBC-927
IF(I.EB.CODE-K) GO TO 340	OBC-928
I=I+1	OBC-929
GO TO 310	OBC-930
340 IF(J.EB.CODE-K-1) GO TO 360	OBC-931
350 CONTINUE	OBC-932
C	OBC-933
C ** LLAMADAS A LAS SUBROUTINAS QUE DETERMINAN LAS COORDENADAS DE LOS -	OBC-934
C VERTICES SOLUCIONANDO EL SISTEMA POR EL METODO DE ELIMINACION DE	OBC-935

C	GAUSS **	OBC-936
C		OBC-937
	360 CALL SOL(CODE-K,A,BX,XC)	OBC-938
	CALL SOL(CODE-K,A,BY,YC)	OBC-939
C	** LLAMADA A LA SUBROUTINA QUE ASIGNA LAS COORDENADAS A LOS VERTICES	OBC-940
C	DE ACUERDO A LA PRIMITIVA NUMERACION **	OBC-941
C		OBC-942
	CALL CA2(NV, CODE, HUMB, XC, BX)	OBC-943
	CALL CA2(NV, CODE, HUMB, YC, BY)	OBC-944
	NAA=0	OBC-945
		OBC-946
C	** LLAMADA A LA SUBROUTINA QUE DIBUJA EL GRAFO O SUBGRAFO POR EL ALGO-	OBC-947
C	RITMO DE TUTTE **	OBC-948
C		OBC-949
	IF(ISSU(1).LT.0) 370,400	OBC-950
	370 WRITE(ISAUK,5001)	OBC-951
	DO 380 I=1,NV	OBC-952
	WRITE(ISAUK,5002) I,XC(I),I,YC(I)	OBC-953
	380 CONTINUE	OBC-954
	400 IF(ISSU(2).LT.0) 50 TO 450	OBC-955
	CALL TUT(NV,NV,R,HUMB,XC,YC,IB,NN,ISAL)	OBC-956
	450 RETURN	OBC-957
		OBC-958
C		OBC-959
C	FORMATS	OBC-960
C		OBC-961
	3001 FORMAT(/1X"COORDENADAS DE LOS VERTICES: ")	OBC-962
	302 FORMAT(1X"X("I2)"=F6.2,3X"Y("I2)"=F6.2)	OBC-963
	3002 FORMAT(1X"X("I2)"=F6.2,3X"Y("I2)"=F6.2)	OBC-964
	END	OBC-965
C		SOL-966
C		SOL-967
	SUBROUTINE SOL(N,A,B,Z)	SOL-968
	*****	SOL-969
C		SOL-970
C	SUBROUTINA QUE SOLUCIONA, POR EL METODO DE ELIMINACION DE GAUS, LAS	SOL-971
C	ECUACIONES DE TUTTE EN X E Y, OBTENIENDO POR TANTO LAS COORDENADAS	SOL-972
C	DE LOS VERTICES INTERIORES.	SOL-973
C		SOL-974
	EMA A(50,50),B(N),Z(N)	SOL-975
	DO 50 I=1,N	SOL-976
	50 Z(I)=0.	SOL-977
	Z(N)=B(N)/A(N,N)	SOL-978
	IF(N.EQ.1) GO TO 110	SOL-979
	DO 100 I=N-1,1,-1	SOL-980
	J=I+1	SOL-981
	S=0	SOL-982
	80 S=S+A(I,J)*Z(J)	SOL-983
	IF(J.EQ.N) GO TO 90	SOL-984
	J=J+1	SOL-985
	SO TO 80	SOL-986
	90 Z(I)=(B(I)-S)/A(I,I)	SOL-987
	IF(I.EQ.1) RETURN	SOL-988
	100 CONTINUE	SOL-989
	110 RETURN	SOL-990
	END	HER-991
C		HER-992
C		HER-993
	FUNCTION HER(ISAL,IERR)	HER-994
	*****	HER-995
C		

```
C FUNCION QUE DETERMINA EL TIPO DE ERROR
C
  WRITE(15AL,3001) IERR
  HER=IERR
  RETURN
C
C FORMATS
C
3001 FORMAT(//"ERROR ..... "I3)
  END
*
```

```
HER-996
HER-997
HER-998
HER-999
HER-000
HER-001
HER-002
HER-003
HER-004
HER-005
HER-006
```

FTN4		PL1-000
SEMA(IJK,3)		PL1-001
C		PL1-002
C		PL1-003
C	*****	PL1-004
C	PROGRAM PL1(5)	PL1-005
C	*****	PL1-006
C		PL1-007
C		PL1-008
C	SEGMENTO 1..... APLICACION DEL TEST DE CONECTIVIDAD	PL1-009
C		PL1-010
C		PL1-011
C	COMMON IPAR(4),NAME(3)	PL1-012
C	COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKT	PL1-013
C	COMMON NOM(40),JAUX(2200)	PL1-014
C	COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IAA(50,10)	PL1-015
C		PL1-016
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE LA MATRIZ DE ADYACENCIA **	PL1-017
C		PL1-018
C	CALL ADY(NV,NA,IA,IB)	PL1-019
C		PL1-020
C	CONTADOR PARA EL DIBUJO DEL GRAFO	PL1-021
C		PL1-022
C	KKT=0	PL1-023
C		PL1-024
C	** LLAMADA A LA SUBROUTINA QUE COMPROBA LA CONECTIVIDAD DEL GRAFO **	PL1-025
C		PL1-026
C	170 CALL CON(NV,NA,2*NA,2*NA+NV,NV+1,NA+NV,K1,JAUX(1),JAUX(NA+2*NV+1),	PL1-027
C	2JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(3*NA+5*NV+3),JAUX(5*NA+5*	PL1-028
C	3NV+3),IB)	PL1-029
C		PL1-030
C	SI SOLO HAY UNA COMPONENTE CONEXA EL GRAFO ES CONEXO	PL1-031
C	EN CASO CONTRARIO, HAY QUE INTRODUCIR NUEVAS ARISTAS	PL1-032
C		PL1-033
C	IF(JAUX(1).EQ.NA) GO TO 200	PL1-034
C	WRITE(ISAL,3001)	PL1-035
C	K2=1	PL1-036
C	KONT=1	PL1-037
C	180 WRITE(ISAL,3002) KONT	PL1-038
C	WRITE(ISAL,3003) (JAUX(J),J=K2+1,K2+JAUX(K2))	PL1-039
C	IF(K2+JAUX(K2).GE.K1) GO TO 190	PL1-040
C	KONT=KONT+1	PL1-041
C	K2=K2+JAUX(K2)+1	PL1-042
C	GO TO 180	PL1-043
C	190 WRITE(ISAL,3004)	PL1-044
C		PL1-045
C	** LLAMADA A LA SUBROUTINA QUE LEE LAS NUEVAS ARISTAS INTRODUCIDAS **	PL1-046
C		PL1-047
C	CALL LEZ(IEAUX,ISAL,ISAUX,2,NA,NV,IA,IB)	PL1-048
C		PL1-049
C	UNA VEZ INTRODUCIDAS LAS ARISTAS SE VOLVERIA A COMPROBAR LA CONECTIVI-	PL1-050
C	DAD.	PL1-051
C		PL1-052
C	GO TO 170	PL1-053
C	200 WRITE(ISAL,3005)	PL1-054
C		PL1-055
C	** LLAMADA AL SEGMENTO PL2 QUE APLICA LOS TEST DE BICONECTIVIDAD,	PL1-056
C	PLANARIDAD Y TRICONECTIVIDAD Y OBTIENE EL GRAFO GEOMETRICO TRICO-	PL1-057
C	NEO **	PL1-058

C	NAME(2)=2N2	PL1-059
	NAME(3)=2H	PL1-060
	CALL EXEC(3,NAME)	PL1-061
	BTDP	PL1-062
C		PL1-063
C	FORMATDS	PL1-064
C		PL1-065
		PL1-066
	3001 FORMAT(/1X"MENSAJE :1SR: EL GRAFO NO ES CONEXO",/1X"LAS COMPONE	PL1-067
	NTES CONEXAS SON:"/)	PL1-068
	3002 FDRMAT(1X"COMPONENTE:"13,/1X,14"-"/)	PL1-069
	3003 FDRMAT(1X"ARISTAS:"10(14"-12),20(/9X,10(14"-12)))	PL1-070
	3004 FDRMAT(/1X"MENSAJE :1SR: MAS DE INTRODUCIR NUEVAS ARISTAS PARA L	PL1-071
	IGAR TODAS LAS COMPONENTES")	PL1-072
	3005 FDRMAT(/1X"MENSAJE :1SR: EL GRAFO ES CONEXO"/)	PL1-073
	END	PL1-074
C		ADY-075
C		ADY-076
C	SUBROUTINE ADY(NV,NA,IA,IB)	ADY-077
C	*****	ADY-078
C		ADY-079
C	SUBROUTINA QUE OBTIENE LA MATRIZ DE ADYACENCIA DEL GRAFO	ADY-080
C		ADY-081
	ENA IB(150,50),IA(100,50)	ADY-082
	DO 50 I=1,2*NV	ADY-083
	DO 50 J=1,NV	ADY-084
	50 IA(I,J)=0	ADY-085
	DO 150 I=1,NA	ADY-086
	KT=0	ADY-087
	DO 100 J=1,NV	ADY-088
	IF(1B(I,J).EQ.0) GO TO 100	ADY-089
	IF(KT.NE.0) GO TO 110	ADY-090
	KT=1	ADY-091
	JJ=J	ADY-092
	100 CONTINUE	ADY-093
	110 JJJ=J	ADY-094
	IA(JJ,JJJ)=1	ADY-095
	IA(JJJ,JJ)=1	ADY-096
	150 CONTINUE	ADY-097
	RETURN	ADY-098
	END	ADY-099
C		CON-100
C		CON-101
C	SUBROUTINE CON(V,E,EE,EEV,VI,EV,CPTR,CON,SIG,NUMB,PILA,CABE,LISA,	CON-102
C	IB)	CON-103
C	*****	CON-104
C		CON-105
C	SUBROUTINA QUE OBTIENE LAS COMPONENTES CONEXAS DEL GRAFO	CON-106
C		CON-107
	ENA IB(150,50)	CON-108
	INTEGER V,E,EE,EEV,VI,EV,CPTR,PUNTO,CODE,KONT,OLDTR,PT,OLDPT,V2	CON-109
	INTEGER FRENE	CON-110
	INTEGER LISA(EE),SIG(EEV),NUMB(VI),PILA(V1),CABE(EE),CON(EV)	CON-111
C		CON-112
C	PDESTA A CERO	CON-113
C		CON-114
	DO 5 I=1,VI	CON-115
	NUMB(I)=0	CON-116
	PILA(I)=0	CON-117
	5 CONTINUE	CON-118

	DO 10 I=1,EEV	CON-119
	SIG(I)=0	CON-120
10	CONTINUE	CON-121
	DO 15 I=1,EE	CON-122
	LISA(I)=0	CON-123
	CABE(I)=0	CON-124
15	CONTINUE	CON-125
	DO 18 I=1,EV	CON-126
	CON(I)=0	CON-127
18	CONTINUE	CON-128
C		CON-129
C	** LLAMADA A LA SUBROUTINA QUE ELABORA LA LISTA DE ARISTAS POR SUS -	CON-130
C	VERTICES EXTREMOS **	CON-131
C		CON-132
	CALL TRA(V,E,EE,LISA,18)	CON-133
	FRENE=V	CON-134
C		CON-135
C	** LLAMADA A LA SUBROUTINA QUE ELABORA EL CONJUNTO CABE **	CON-136
C		CON-137
	DO 20 I=1,E	CON-138
	CALL PRD(V,EE,EEV,LISA(2*I-1),LISA(2*I),FRENE,SIG,CABE)	CON-139
	CALL PRD(V,EE,EEV,LISA(2*I),LISA(2*I-1),FRENE,SIG,CABE)	CON-140
20	CONTINUE	CON-141
	CPTR=0	CON-142
	PUNTO=1	CON-143
35	IF(PUNTO.GT.V) GO TO 110	CON-144
	CODE=1	CON-145
	KONT=1	CON-146
	NUMB(PUNTO)=CODE	CON-147
	PILA(KONT)=PUNTO	CON-148
	CPTR=CPTR+1	CON-149
	OLDTR=CPTR	CON-150
	PT=PUNTO	CON-151
	OLDPT=0	CON-152
50	IF(SIG(PT).EQ.0) GO TO 60	CON-153
	V2=CABE(SIG(PT)-V)	CON-154
	SIG(PT)=SIG(SIG(PT))	CON-155
	IF(NUMB(V2).GE.NUMB(PT).OR.V2.EQ.OLDPT) GO TO 50	CON-156
C		CON-157
C	** LLAMADA A LA SUBROUTINA QUE ALMACENA EL CONJUNTO DE COMPONENTES CO-	CON-158
C	NEXAS **	CON-159
C		CON-160
	CALL ADI(EV,PT,V2,CON,CPTR,V,E,18)	CON-161
	IF(NUMB(V2).NE.0) GO TO 50	CON-162
	CODE=CODE+1	CON-163
	KONT=KONT+1	CON-164
	NUMB(V2)=CODE	CON-165
	PILA(KONT)=V2	CON-166
	OLDPT=PT	CON-167
	PT=V2	CON-168
	GO TO 50	CON-169
60	IF(KONT.EQ.1) GO TO 90	CON-170
	PILA(KONT)=0	CON-171
	KONT=KONT-1	CON-172
	IF(KONT.EQ.1)70,80	CON-173
70	PT=1	CON-174
	OLDPT=0	CON-175
	GO TO 50	CON-176
80	PT=PILA(KONT)	CON-177
	OLDPT=PILA(KONT-1)	CON-178
	GO TO 50	CON-179

```
90 COM(OLDTR)=CPTR-OLDTR
100 IF(NUMB(PUNTD).EQ.0) GO TO 35
    PUNTD=PUNTD+1
    GO TO 100
110 RETURN
END
```

```
COM-180
COM-181
COM-182
COM-183
COM-184
COM-185
COM-186
```

FTN4		PL2-000
SENA(IJK,3)		PL2-001
C		PL2-002
C		PL2-003
C	*****	PL2-004
	PRDGRAM PL2(5)	PL2-005
	*****	PL2-006
C		PL2-007
C		PL2-008
C	SEGMENTO 2	APLICACION DE LOS TEST DE:
C		- PLANARIDAD
C		- BICONECTIVIDAD
C		- TRICONECTIVIDAD
C		PL2-010
		PL2-011
		PL2-012
		PL2-013
		PL2-014
	COMMON IPAR(4),HANE(3)	PL2-015
	COMMON IENT,ISAL,IEAJK,ISAUK,NV,NA,KKTT	PL2-016
	COMMON NOM(40),JAJK(2200)	PL2-017
	COMMON /IJK/ IB(150,50),IA(100,50),K(300),Y(300),IAB(50,10)	PL2-018
C		PL2-019
C	** LLANADA A LA SUBROUTINA QUE OBTIENE LAS COMPONENTES BICOHERAS **	PL2-020
C		PL2-021
210	CALL BIC(NV,NA,2*NA,2*NA+NV,NV+1,NA+NV,K1,JOBT,IAUK(1),IAUK(NA+NV+1),IAUK(NA+2*NV+1),IAUK(3*NA+3*NV+1),IAUK(3*NA+4*NV+2),IAUK(3*NA+5*NV+3),IAUK(5*NA+5*NV+3),IAUK(7*NA+5*NV+3),IAUK(9*NA+5*NV+3),IB)	PL2-022
	IF(IAUK(1).EQ.NA) GO TO 310	PL2-023
	WRITE(ISAL,3001)	PL2-024
	K2=1	PL2-025
	KOBT=1	PL2-026
260	WRITE(ISAL,3002) KOBT	PL2-027
	WRITE(ISAL,3003) (IAUK(J),J=K2+1,K2+IAUK(K2))	PL2-028
	IF(K2+IAUK(K2).GE.K1) GO TO 270	PL2-029
	KOBT=KOBT+1	PL2-030
	K2=K2+IAUK(K2)+1	PL2-031
	GO TO 260	PL2-032
270	WRITE(ISAL,3004)	PL2-033
	DO 200 I=1,JOBT	PL2-034
	II=IAUK(NA+NV+1)	PL2-035
	K2=1	PL2-036
	KOBT=1	PL2-037
80	IF(K2.GE.K1) GO TO 200	PL2-038
	DO 100 J=K2+1,K2+IAUK(K2)	PL2-039
	JJ=IAUK(J)	PL2-040
	IF(IB(JJ,II).NE.0) GO TO 110	PL2-041
190	CONTINUE	PL2-042
105	KOBT=KOBT+1	PL2-043
	K2=K2+IAUK(K2)+1	PL2-044
	GO TO 80	PL2-045
110	K3=K2+IAUK(K2)+1	PL2-046
	IOBT=KOBT+1	PL2-047
120	IF(K3.GE.K1) GO TO 195	PL2-048
	DO 150 K=K3+1,K3+IAUK(K3)	PL2-049
	KK=IAUK(K)	PL2-050
	IF(IB(KK,II).NE.0) GO TO 160	PL2-051
150	CONTINUE	PL2-052
	K3=K3+IAUK(K3)+1	PL2-053
	IOBT=IOBT+1	PL2-054
	GO TO 120	PL2-055
160	WRITE(ISAL,3005) KOBT,IOBT,II	PL2-056
		PL2-057
		PL2-058

200 CONTINUE	PL2-059
GO TO 320	PL2-060
310 WRITE(ISAL,3006)	PL2-061
C	PL2-062
C EN CADA COMPONENTE BICONEXA VAMOS A REALIZAR LAS SIGUIENTES OPERACIONES:	PL2-063
C	PL2-064
C - ESTUDIO DE LA PLANARIDAD	PL2-065
C - DESCOMPOSICION EN COMPONENTES TRICONEXAS	PL2-066
C - INTRODUCCION DE ARISTAS PARA HACER TRICONEXA LA COMPONENTE BICONEXA	PL2-067
C	PL2-068
320 K2=1	PL2-069
NCB=0	PL2-070
C	PL2-071
C PUESTA A CERO DE LA MATRIZ DE ADYACENCIA	PL2-072
C	PL2-073
325 DO 330 I=1,2*NV	PL2-074
DO 330 J=1,NV	PL2-075
IA(I,J)=0	PL2-076
330 CONTINUE	PL2-077
DO 335 I=1,NA	PL2-078
DO 335 J=1,NV	PL2-079
IF(IB(I,J).NE.0) IB(I,J)=1	PL2-080
335 CONTINUE	PL2-081
C	PL2-082
C HACEMOS NEGATIVOS LOS TERMINOS DE LA MATRIZ DE INCIDENCIA CORRESPONDIENTES A LA COMPONENTE BICONEXA. LOS DE LA MATRIZ DE ADYACENCIA LOS HACEMOS POSITIVOS	PL2-083
C	PL2-084
IF(K2.EE.K1) GO TO 930	PL2-085
NCB=NCB+1	PL2-086
IF(JAUX(1).NE.NA) WRITE(ISAL,3007) NCB	PL2-087
IF(JAUX(1).NE.NA) WRITE(ISAL,3003) (JAUX(J),J=K2+1,K2+JAUX(K2))	PL2-088
DO 400 I=K2+1,K2+JAUX(K2)	PL2-089
K=0	PL2-090
J1=0	PL2-091
J2=0	PL2-092
DO 350 J=1,NV	PL2-093
IF(IB(JAUX(1),J).EQ.0) GO TO 350	PL2-094
IF(IB(JAUX(1),J).NE.0) IB(JAUX(1),J)=-1	PL2-095
IF(K.EQ.0) 340,345	PL2-096
340 J1=J	PL2-097
K=K+1	PL2-098
GO TO 350	PL2-099
345 J2=J	PL2-100
GO TO 360	PL2-101
350 CONTINUE	PL2-102
IF(J1.EQ.0.OR.J2.EQ.0) GO TO 400	PL2-103
360 IA(J1,J2)=1	PL2-104
IA(J2,J1)=1	PL2-105
400 CONTINUE	PL2-106
C	PL2-107
C SALTO SI LA COMPONENTE BICONEXA ES UNA ARISTA O UN TRIANGULO	PL2-108
C	PL2-109
IF(JAUX(K2).LE.3) GO TO 920	PL2-110
II=0	PL2-111
IT=0	PL2-112
KKK=JAUX(K2)	PL2-113
C	PL2-114
C ** LLAMADA A LA SUBROUTINA QUE APLICA LA FORMULA DE EULER A LA COMPONENTE BICONEXA **	PL2-115
C	PL2-116
	PL2-117
	PL2-118

C		PL2-119
C	410 CALL EUL(KKK, NV, II, IA)	PL2-120
C		PL2-121
C	SALIDA DE ERROR SI NO ES PLANAR	PL2-122
C		PL2-123
C	IF(II.EQ.0) GO TO 420	PL2-124
C	IERR=NER(ISAL, 3)	PL2-125
C	STOP 0003	PL2-126
C		PL2-127
C	** LLAMADA A LA SUBROUTINA QUE COMPROBEA LA PLANARIDAD DE LA COMPONENTE BICONEXA**	PL2-128
C		PL2-129
C		PL2-130
C	420 CALL PLA(IS, NV, NA, 2*NA, 2*NA+NV, NV+1, N1, N2, II, IT, JAUX(NA+2*NV+1),	PL2-131
C	2*JAUX(3*NA+3*NV+1), JAUX(3*NA+4*NV+2), JAUX(3*NA+5*NV+3), JAUX(4*NA+5*	PL2-132
C	8*NV+3), JAUX(5*NA+5*NV+3), JAUX(6*NA+5*NV+3), JAUX(7*NA+5*NV+3), JAUX(9	PL2-133
C	8*NA+7*NV+3), JAUX(9*NA+8*NV+3), JAUX(9*NA+9*NV+3), JAUX(9*NA+10*NV+3)	PL2-134
C	8, JAUX(10*NA+10*NV+3), IAA, IA, IB, ISAUX)	PL2-135
C		PL2-136
C	SALIDA DE ERROR SI NO ES PLANAR	PL2-137
C		PL2-138
C	IF(II.EQ.0.AND.IT.EQ.0) GO TO 430	PL2-139
C	II=NA+2*NV+II	PL2-140
C	IT=NA+2*NV+IT	PL2-141
C	WRITE(ISAL, 3008) (JAUX(I), I=II, IT)	PL2-142
C	WRITE(ISAL, 3009)	PL2-143
C		PL2-144
C	** LLAMADA A LA SUBROUTINA QUE SUSTITUYE UNAS RELACIONES POR OTRAS **	PL2-145
C		PL2-146
C	CALL LEI(IEAUX, ISAL, ISAUX, I, IA, IB)	PL2-147
C	GO TO 420	PL2-148
C	430 IF(JAUX(I).EQ.NA) WRITE(ISAL, 3010)	PL2-149
C	IF(JAUX(I).NE.NA) WRITE(ISAL, 3011)	PL2-150
C		PL2-151
C	** LLAMADA A LA SUBROUTINA QUE DESCOMPONE UNA COMPONENTE BICONEXA EN SUS COMPONENTES TRICONEXAS **	PL2-152
C		PL2-153
C		PL2-154
C	CALL TRI(IS, NV, NA, 2*NA, 2*NA+NV, NV+1, 6*NA, N1, N2, IOHT, JAUX(NA+2*NV+1	PL2-155
C	8), JAUX(3*NA+3*NV+1), JAUX(3*NA+4*NV+2), JAUX(3*NA+5*NV+3), JAUX(9*NA+	PL2-156
C	85*NV+3), JAUX(9*NA+6*NV+3), JAUX(9*NA+7*NV+3), JAUX(9*NA+8*NV+3),	PL2-157
C	8*JAUX(9*NA+9*NV+3), JAUX(9*NA+10*NV+3), JAUX(10*NA+10*NV+3), JAUX(11*NA	PL2-158
C	82+10*NV+3), JAUX(13*NA+10*NV+3), JAUX(13*NA+11*NV+3), JAUX(13*NA+12*NA	PL2-159
C	8*NV+3), IAA, IB)	PL2-160
C		PL2-161
C	SALTO SI EL GRAFO ES TRICONEXO	PL2-162
C		PL2-163
C	IF(IOHT-1.EQ.2*NA) GO TO 938	PL2-164
C	KT1=3*NA+3*NV+2	PL2-165
C	KF=3*NA+5*NV+2+IOHT	PL2-166
C	KT2=0	PL2-167
C	KT3=0	PL2-168
C	KT5=0	PL2-169
C	HAA=NA	PL2-170
C	DO 435 JJ=KT1+1, KF	PL2-171
C	IF(JAUX(JJ).SE.0) GO TO 435	PL2-172
C	KT5=KT5+1	PL2-173
C	435 CONTINUE	PL2-174
C		PL2-175
C	SI KT5=1 LA COMPONENTE BICONEXA ES TRICONEXA	PL2-176
C		PL2-177
C		PL2-178
C	IF(KT5.EQ.1) GO TO 910	PL2-179
C	IF(JAUX(I).EQ.NA) WRITE(ISAL, 3027)	

IF(JAUX(1).NE.NA) WRITE(ISAL,3012)	PL2-180
DO 460 JJ=KT1+1,KF	PL2-181
IF(JAUX(JJ).GE.0) GO TO 460	PL2-182
KT2=JJ	PL2-183
C	PL2-184
C SI SON MENOS DE CUATRO EL NUMERO DE ARISTAS NO ES UNA COMPONENTE TRI-	PL2-185
C CDNEXA EN UN PRINCIPIO	PL2-186
C	PL2-187
IF(KT2-KT1.LT.7) GO TO 454	PL2-188
IF(KT2-KT1.EQ.7) GO TO 455	PL2-189
KT3=KT3+1	PL2-190
WRITE(ISAL,3013) KT3,(JAUX(J),J=KT1+1,KT2-1)	PL2-191
DO 450 K=KT1+1,KT2-1,2	PL2-192
J1=JAUX(K)	PL2-193
J2=JAUX(K+1)	PL2-194
IF(IA(J1,J2).GT.0) GO TO 450	PL2-195
WRITE(ISAL,3014) KT3,J1,J2	PL2-196
C	PL2-197
C SE INTRODUCE LA ARISTA VIRTUAL QUE TIENE COMO EXTREMO J1,J2	PL2-198
C	PL2-199
IA(J1,J2)=1	PL2-200
IA(J2,J1)=1	PL2-201
NA=NA+1	PL2-202
DO 440 L=1,NV	PL2-203
440 IB(NA,L)=0	PL2-204
IB(NA,J1)=-1	PL2-205
IB(NA,J2)=-1	PL2-206
450 CONTINUE	PL2-207
JAUX(KT2)=-KT3	PL2-208
GO TO 456	PL2-209
454 JAUX(KT2)=JAUX(KT2)-200	PL2-210
GO TO 456	PL2-211
455 JAUX(KT2)=JAUX(KT2)-100	PL2-212
456 KT1=KT2	PL2-213
IF(KT1.GE.KF) GO TO 465	PL2-214
460 CONTINUE	PL2-215
C	PL2-216
C SI HAY UNA COMPONENTE ESCIADIDA QUE ES UN TRIANGULO DONDE TODAS LAS	PL2-217
C ARISTAS SON REALES Y NO INCLUIDAS EN NINGUNA COMPONENTE TRICONEXA LA	PL2-218
C SEÑALANDOS TAMBIEN	PL2-219
C	PL2-220
465 KT1=3*NA+5*NV+2	PL2-221
DO 490 K=KT1+1,KF	PL2-222
IF(JAUX(K).GT.-100.OR.JAUX(K).LT.-200) GO TO 490	PL2-223
IF(JAUX(K-1).EQ.JAUX(K-5).AND.JAUX(K-2).EQ.JAUX(K-6)) GO TO 490	PL2-224
IF(JAUX(K-1).EQ.JAUX(K-6).AND.JAUX(K-2).EQ.JAUX(K-5)) GO TO 490	PL2-225
DO 480 JJ=K-6,K-1,2	PL2-226
IX=JAUX(JJ)	PL2-227
IY=JAUX(JJ+1)	PL2-228
IF(IA(IX,IY).LE.0) GO TO 490	PL2-229
KK1=KT1+1	PL2-230
DO 475 II=KT1+1,KF	PL2-231
IF(JAUX(II).GT.0) GO TO 475	PL2-232
KK2=II-1	PL2-233
IF(JAUX(II).LT.-100) GO TO 474	PL2-234
DO 470 KK=KK1,KK2,2	PL2-235
IF(IX.EQ.JAUX(KK).AND.IY.EQ.JAUX(KK+1)) GO TO 490	PL2-236
IF(IY.EQ.JAUX(KK).AND.IX.EQ.JAUX(KK+1)) GO TO 490	PL2-237
470 CONTINUE	PL2-238
474 KK1=KK2+2	PL2-239

	IF(KK1,SE,KF) GO TO 480	PL2-240
475	CONTINUE	PL2-241
480	CONTINUE	PL2-242
	KT3=KT3+1	PL2-243
	JAX(K)=-KT3	PL2-244
	WRITE(ISAL,3013) KT3,(JAX(J),J=K-6,K-1)	PL2-245
490	CONTINUE	PL2-246
	IF(NAA,HE,NA) WRITE(ISAL,3015)	PL2-247
C		PL2-248
C	ARISTAS QUE LIGAN UNAS COMPONENTES CON OTRAS	PL2-249
C	ARISTAS QUE NO LIGAN A NINGUNA COMPONENTE	PL2-250
C		PL2-251
	L1=9*NAA+10*WV+2	PL2-252
	DO 501 LL=L1+1,L1+2+4AA	PL2-253
	JAX(LL)=0	PL2-254
501	CONTINUE	PL2-255
	KT5=0	PL2-256
	KT1=3*NAA+5*WV+2	PL2-257
	KT2=0	PL2-258
	KT6=0	PL2-259
	KT9=0	PL2-260
	L2=0	PL2-261
	DO 517 JJ=KT1+1,KF	PL2-262
	IF(JAX(JJ).GT.-100.OR.JAX(JJ).LT.-200) GO TO 517	PL2-263
	DO 515 II=JJ-6,JJ-1,2	PL2-264
	IX=JAX(II)	PL2-265
	IY=JAX(II+1)	PL2-266
	IF(IA(IX,IY).LE.0) GO TO 515	PL2-267
	IF(L2.EQ.0) GO TO 505	PL2-268
	DO 504 LL=1,L2,2	PL2-269
	IF(IX.EQ.JAX(LL+L1).AND.IY.EQ.JAX(LL+L1+1)) GO TO 515	PL2-270
	IF(IY.EQ.JAX(LL+L1).AND.IX.EQ.JAX(LL+L1+1)) GO TO 515	PL2-271
504	CONTINUE	PL2-272
505	KT3=0	PL2-273
	KT4=0	PL2-274
	KT7=3*NAA+5*WV+2	PL2-275
	KT8=0	PL2-276
	DO 510 KK=KT7+1,KF	PL2-277
	IF(JAX(KK).GT.0) GO TO 510	PL2-278
	KT8=KK	PL2-279
	IF(JAX(KK).LT.-100) GO TO 509	PL2-280
	DO 506 NN=KT7+1,KT8-1	PL2-281
	IF(IX.EQ.JAX(NN)) KT3=-JAX(KT8)	PL2-282
	IF(IY.EQ.JAX(NN)) KT4=-JAX(KT8)	PL2-283
506	CONTINUE	PL2-284
	IF(KT3.EQ.0.OR.KT4.EQ.0) GO TO 509	PL2-285
	IF(KT3.NE.KT4) GO TO 514	PL2-286
	GO TO 515	PL2-287
509	KT7=KT8	PL2-288
	IF(KT7,SE,KF) GO TO 511	PL2-289
510	CONTINUE	PL2-290
511	IF(L2.EQ.0) GO TO 513	PL2-291
	DO 512 LL=1,L2,2	PL2-292
	IF(IX.EQ.JAX(LL+L1).AND.IY.EQ.JAX(LL+L1+1)) GO TO 515	PL2-293
	IF(IY.EQ.JAX(LL+L1).AND.IX.EQ.JAX(LL+L1+1)) GO TO 515	PL2-294
512	CONTINUE	PL2-295
513	IF(KT9.EQ.0) WRITE(ISAL,3016)	PL2-296
	KT9=1	PL2-297
	WRITE(ISAL,3017) IX,IY	PL2-298
	JAX(LL+L2+1)=IX	PL2-299
	JAX(LL+L2+2)=IY	PL2-300

L2=L2+2	PL2-301
GO TO 515	PL2-302
514 JAUX(L1+L2+1)=IX	PL2-303
JAUX(L1+L2+2)=IY	PL2-304
L2=L2+2	PL2-305
IF(KT9.EQ.1) WRITE(ISAL,3016)	PL2-306
WRITE(ISAL,3018) IX,IY,KT3,KT4	PL2-307
KT9=1	PL2-308
515 CONTINUE	PL2-309
517 CONTINUE	PL2-310
C	PL2-311
C DIBUJO DE CADA COMPONENTE TRICONEXA DE LA BICONEXA	PL2-312
C	PL2-313
NCT=0	PL2-314
KT1=3*NAA+5*NV+2	PL2-315
DD 800 JJ=KT1+1,KF	PL2-316
IF(JAUX(JJ).GT.0) GO TO 800	PL2-317
KT2=JJ	PL2-318
IF(JAUX(JJ).LT.-100) GO TO 760	PL2-319
DD 650 I=1,NA	PL2-320
J1=0	PL2-321
J2=0	PL2-322
DD 550 J=1,NV	PL2-323
IF(IB(I,J).GT.0) GO TO 650	PL2-324
IF(IB(I,J).EQ.0) GO TO 550	PL2-325
IF(J1.EQ.0) 519,520	PL2-326
519 J1=J	PL2-327
GO TO 550	PL2-328
520 J2=J	PL2-329
GO TO 560	PL2-330
550 CONTINUE	PL2-331
560 KT3=0	PL2-332
DD 600 K=KT1+1,KT2-1,2	PL2-333
IF(JAUX(K).EQ.J1.AND.JAUX(K+1).EQ.J2) KT3=1	PL2-334
IF(JAUX(K).EQ.J2.AND.JAUX(K+1).EQ.J1) KT3=1	PL2-335
IF(KT3.EQ.1) GO TO 640	PL2-336
600 CONTINUE	PL2-337
IB(I,J1)=-2	PL2-338
IB(I,J2)=-2	PL2-339
IA(J1,J2)=0	PL2-340
IA(J2,J1)=0	PL2-341
GO TO 650	PL2-342
640 IA(J1,J2)=1	PL2-343
IA(J2,J1)=1	PL2-344
650 CONTINUE	PL2-345
NCT=NCT+1	PL2-346
C	PL2-347
IF(KT2-KT1.NE.7) GO TO 690	PL2-348
WRITE(ISAL,3019) NCT,(JAUX(IJ),IJ=KT1+1,KT2-1)	PL2-349
WRITE(ISAL,3020)	PL2-350
GO TO 690	PL2-351
690 WRITE(ISAL,3021) NCT,(JAUX(IJ),IJ=KT1+1,KT2-1)	PL2-352
C	PL2-353
C	PL2-354
C ** LLAMADA A LA SUBROUTINA QUE OBTIENE UN CIRCUITO EN LA COMPONENTE	PL2-355
C TRICONEXA **	PL2-356
C	PL2-357
CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+	PL2-358
26*NV+3),JAUX(9*NA+7*NV+3),KK,LDNT,IA)	PL2-359
C	PL2-360

C	** LLAMADA A LA SUBROUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES	PL2-361
C	APLICANDO EL ALGORITMO DE TUTTE.**	PL2-362
C		PL2-363
	CALL DDC(NV,NA,KN,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),	PL2-364
	Y(1),K(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKTT,ISAL,ISAUX)	PL2-365
698	DO 699 I=1,2*NV	PL2-366
	DO 699 J=1,NV	PL2-367
699	IA(I,J)=0	PL2-368
	DO 750 I=1,NA	PL2-369
	J1=0	PL2-370
	J2=0	PL2-371
	DO 700 J=1,NV	PL2-372
	IF(IB(I,J).GE.-1) GO TO 700	PL2-373
	IB(I,J)=-1	PL2-374
	IF(J1.NE.0) GO TO 710	PL2-375
	J1=J	PL2-376
700	CONTINUE	PL2-377
	GO TO 750	PL2-378
710	J2=J	PL2-379
	IA(J1,J2)=1	PL2-380
	IA(J2,J1)=1	PL2-381
750	CONTINUE	PL2-382
750	KT1=KT2	PL2-383
	IF(KT1.GE.KF) GO TO 810	PL2-384
800	CONTINUE	PL2-385
810	WRITE(ISAL,3022)	PL2-386
	DO 815 I=1,NA	PL2-387
	J1=0	PL2-388
	J2=0	PL2-389
	DO 812 J=1,NV	PL2-390
	IF(IB(I,J).EQ.0) GO TO 812	PL2-391
	IF(IB(I,J).GT.0) GO TO 815	PL2-392
	IF(J1.NE.0) GO TO 813	PL2-393
	J1=J	PL2-394
812	CONTINUE	PL2-395
813	J2=J	PL2-396
	IA(J1,J2)=1	PL2-397
	IA(J2,J1)=1	PL2-398
815	CONTINUE	PL2-399
C		PL2-400
C	** LLAMADA A LA SUBROUTINA QUE LEE LAS NUEVAS RELACIONES INTRODUCIDAS	PL2-401
C	QUE LIGAN A LAS COMPONENTES TRICONEXAS **	PL2-402
C		PL2-403
	CALL LE2(IEAUX,ISAL,ISAUX,1,NA,NV,IA,IB)	PL2-404
	WRITE(ISAL,3023)	PL2-405
C		PL2-406
C	** LLAMADA A LA SUBROUTINA QUE SUSTITUYE ALGUNA DE LAS RELACIONES IN-	PL2-407
C	TRDUCIDAS POR OTRAS, POR EJEMPLO, LAS ARISTAS VIRTUALES HECHAS	PL2-408
C	REALES **	PL2-409
C		PL2-410
	CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)	PL2-411
	KKK=0	PL2-412
	DO 900 I=1,NA	PL2-413
	KKI=0	PL2-414
	DO 850 J=1,NV	PL2-415
	IF(IB(I,J).GT.0) GO TO 900	PL2-416
	IF(IB(I,J).EQ.0) GO TO 850	PL2-417
	IF(KKI.EQ.0) KKK=KKK+1	PL2-418
	KKI=1	PL2-419
	IF(IB(I,J).LT.0) IB(I,J)=-1	PL2-420
850	CONTINUE	PL2-421

900	CONTINUE	PL2-422
	II=0	PL2-423
	GO TO 410	PL2-424
C		PL2-425
C	DIBUJO DE LA COMPONENTE BICONEXA QUE ES ADEMAS TRICONEXA	PL2-426
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE UN CIRCUITO EN LA COMPONENTE	PL2-427
C	TRICONEXA **	PL2-428
C		PL2-429
C		PL2-430
910	WRITE(ISAL,3024)	PL2-431
	CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+	PL2-432
	8*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)	PL2-433
C		PL2-434
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES	PL2-435
C	APLICANDO EL ALGORITMO DE TUTTE **	PL2-436
C		PL2-437
	CALL DBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),	PL2-438
	Y(1),K(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKT,ISAL,ISAUX)	PL2-439
	K2=K2+JAUX(K2)+1	PL2-440
	GO TO 325	PL2-441
920	IF(JAUX(K2).EQ.1) GO TO 925	PL2-442
	L1=JAUX(K2+1)	PL2-443
	L2=JAUX(K2+2)	PL2-444
	L3=JAUX(K2+3)	PL2-445
	WRITE(ISAL,3020)	PL2-446
	DO 923 J=1,NV	PL2-447
	IF(IB(L1,J).NE.0) IB(L1,J)=1	PL2-448
	IF(IB(L2,J).NE.0) IB(L2,J)=1	PL2-449
	IF(IB(L3,J).NE.0) IB(L3,J)=1	PL2-450
923	CONTINUE	PL2-451
	GO TO 929	PL2-452
925	L1=JAUX(K2+1)	PL2-453
	WRITE(ISAL,3020)	PL2-454
	DO 927 J=1,NV	PL2-455
	IF(IB(L1,J).NE.0) IB(L1,J)=1	PL2-456
927	CONTINUE	PL2-457
929	K2=K2+JAUX(K2)+1	PL2-458
	GO TO 325	PL2-459
930	DO 935 I=1,NA	PL2-460
	DO 935 J=1,NV	PL2-461
	IF(IB(I,J).NE.0) IB(I,J)=1	PL2-462
935	CONTINUE	PL2-463
	WRITE(ISAL,3025)	PL2-464
C		PL2-465
C	** LLAMADA A LA SUBROUTINA QUE INTRODUCE NUEVAS ARISTAS PARA HACER	PL2-466
C	TRICONEXO AL GRAFO **	PL2-467
C		PL2-468
	CALL LE2(IEAUX,ISAL,ISAUX,2,NA,NV,IA,IB)	PL2-469
	WRITE(ISAL,3023)	PL2-470
C		PL2-471
C	** LLAMADA A LA SUBROUTINA QUE SUSTITUYE ALGUNA DE LAS RELACIONES IN-	PL2-472
C	TRODUCIDAS POR OTRAS **	PL2-473
C		PL2-474
	CALL LE1(IEAUX,ISAL,ISAUX,2,IA,IB)	PL2-475
	GO TO 210	PL2-476
C		PL2-477
C	DIBUJO DEL GRAFO TRICONEXO	PL2-478
C		PL2-479
C		PL2-480
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE UN CIRCUITO EN EL GRAFO TRICO-	PL2-481

```

C      NEXO **
C
938 WRITE(15AL,3026)
940 CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+
      6*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)
C
C      ** LLAMADA A LA SUBROUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES
C      APLICANDO EL ALGORITMO DE TOTTE **
C
      CALL DBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),
      Y(1),K(NV+1),Y(NV+1),JAUX(9*NA+5*NV+3),IA,IB,KKTY,ISAL,ISAUX)
      DO 941 I=1,2*NV
      DO 941 J=1,NV
941  IA(I,J)=0
      DO 944 I=1,NA
      J1=0
      J2=0
      DO 942 J=1,NV
      IF(18(I,J).EQ.0) GO TO 942
      IF(J1.NE.0) GO TO 943
      J1=J
942  CONTINUE
943  J2=J
      IA(J1,J2)=1
      IA(J2,J1)=1
944  CONTINUE
C
C      ** LLAMADA AL SEGMENTO PL3 QUE OBTIENE EL DUAL DEL GRAFO INICIAL **
C
990 NAME(2)=2N3
      NAME(3)=2N
      CALL EXEC(8,NAME)
      STOP
C
C      FORMATS
C
3001 FORMAT(//1X'MENSAJE :SR: EL GRAFO NO ES BICONEXO',//1X'LAS COMPON
      ENENTES BICONEXAS SON:')
3002 FORMAT(//1X'COMPONENTE'13,/,1X,14'-')
3003 FORMAT(1X'ARISTAS:'1714,15(/9X,1714))
3004 FORMAT(//1X'LOS VERTICES DE SEPARACION DE ESAS COMPONENTES SON:')
3005 FORMAT(1X'COMPONENTES:'12'-12' ... VERT. DE SEPAR:'13)
3006 FORMAT(//1X'MENSAJE :SR: EL GRAFO ES BICONEXO')
3007 FORMAT(//57X'COMPONENTE BICONEXO:'13,/,1X,79'-')
3008 FORMAT(//1X'MENSAJE :SR: EL GRAFO NO ES PLANAR',//1X'EL CAMINO QU
      SE LE PUEDE SER REPRESENTADO EN EL PLANO (DADO POR LOS VERTICES) ES
      1:'/,/(26I3))
3009 FORMAT(//1X'MENSAJE :SR: HA DE SUSTITUIRSE UNA DE LAS ARISTAS DE
      EL CAMINO POR OTRA. '//16X'ESTA NUEVA ARISTA TIENE QUE INCIDIR EN DO
      OS VERTICES DE LA MISMA'//16X'COMPONENTE BICONEXO.')
3010 FORMAT(//1X'MENSAJE :SR: EL GRAFO ES PLANAR')
3011 FORMAT(//1X'MENSAJE :SR: LA COMPONENTE BICONEXA ES PLANAR')
3012 FORMAT(//1X'MENSAJE :SR: LA COMPONENTE BICONEXA NO ES TRICONEXA'
      //1X'LAS COMPONENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VI
      SEN EXPRESADAS'//1X'POR SUS VERTICES EXTREMOS) SON:')
3013 FORMAT(//1X'COMPONENTE'12,/,1X'ARISTAS:'19(14'-12),20(/9X,10(14'-
      12)))
3014 FORMAT(//1X'MENSAJE :SR: ARISTA VIRTUAL EN LA COMPONENTE'12'114
      1'-12)
3015 FORMAT(//1X'MENSAJE :SR: LAS ARISTAS VIRTUALES CITADAS HAN PASAD
      O A SER REALES. '//16X'POSTERIORMENTE, SI SE DESEA, PUEDEN SER CAMBIA

```

```

PL2-482
PL2-483
PL2-484
PL2-485
PL2-486
PL2-487
PL2-488
PL2-489
PL2-490
PL2-491
PL2-492
PL2-493
PL2-494
PL2-495
PL2-496
PL2-497
PL2-498
PL2-499
PL2-500
PL2-501
PL2-502
PL2-503
PL2-504
PL2-505
PL2-506
PL2-507
PL2-508
PL2-509
PL2-510
PL2-511
PL2-512
PL2-513
PL2-514
PL2-515
PL2-516
PL2-517
PL2-518
PL2-519
PL2-520
PL2-521
PL2-522
PL2-523
PL2-524
PL2-525
PL2-526
PL2-527
PL2-528
PL2-529
PL2-530
PL2-531
PL2-532
PL2-533
PL2-534
PL2-535
PL2-536
PL2-537
PL2-538
PL2-539
PL2-540
PL2-541
PL2-542

```

```

3005 PDR OTRAS.")
3016 FORMAT(/IX*MENSAJE :ISR: ADEMAS DE LAS ARISTAS PERTENECIENTES A PL2-543
ALGUNA COMPONENTE",/16X*HAY OTRAS QUE NO PERTENECEN A NINGUNA DE E PL2-544
LLAS.",/IX*LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS PL2-545
Y LAS QUE NO ENLAZAN A/IX*DOS COMPONENTES CUALESQUIERA SON:*/) PL2-547
3017 FORMAT(IX,*ARISTA:"I4"-*I2,2X*(NO LIGA A DOS COMPONENTES*)) PL2-548
3018 FORMAT(IX,*ARISTA:"I4"-*I2,2X*COMPONENTE:"I2" Y*I2) PL2-549
3019 FORMAT(/IX,25"-*/IX*COMPONENTE:"I2" (TRIANGULO)*/IX,25"-*/IX*ARI PL2-550
STAS:"I0(I4"-*I2),20(/3X,10(I4"-*I2))) PL2-551
3020 FORMAT(/IX*MENSAJE :ISR: NO SE DIBUJA POR SER ELEMENTAL") PL2-552
3021 FORMAT(/IX,25"-*/IX*COMPONENTE:"I2" (TRICONEXA)*/IX,25"-*/IX*ARI PL2-553
STAS:"I0(I4"-*I2),20(/3X,10(I4"-*I2))) PL2-554
3022 FORMAT(/IX*MENSAJE :ISR: SE HAN DE INTRODUCIR NUEVAS ARISTAS PAR PL2-555
A CONVERTIR EN TRICONEXA*/16X*LA COMPONENTE BICONEXA.",/16X*LA ADI PL2-556
CION SE HARA A LA VISTA DEL SUBGRAFO TRAZADO.",/16X*SE PUEDE REALI PL2-557
ZAR ESCALONADAMENTE, INTRODUCIENDO SOLO PARTE DE*/16X*ELLAS. ANADIE PL2-558
ENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.") PL2-559
3023 FORMAT(/IX*MENSAJE :ISR: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS" PL2-560
A) PL2-561
3024 FORMAT(/IX*MENSAJE :ISR: LA COMPONENTE BICONEXA ES ADEMAS TRICON PL2-562
EXA") PL2-563
3025 FORMAT(/IX*MENSAJE :ISR: SE HAN DE ANADIR NUEVAS ARISTAS PARA QU PL2-564
E EL GRAFO SEA TRICONEJO*/16X*ESTAS ARISTAS ENLAZARAN A LAS DISTI PL2-565
NTAS COMPONENTES BICONEJAS*/16X*ENTRE SI.") PL2-566
3026 FORMAT(/IX*MENSAJE :ISR: EL GRAFO ES TRICONEJO") PL2-567
3027 FORMAT(/IX*MENSAJE :ISR: EL GRAFO NO ES TRICONEJO*/IX*LAS COMPO PL2-568
NENTES TRICONEJAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADA PL2-569
S POR SUS VERTICES EXTREMOS) SON:*/) PL2-570
END PL2-571
C BIC-572
C BIC-573
SUBROUTINE BIC(V,E,EE,EEV,V1,EV,EPTR,JOINT,BICOM,ARTIC,SIG,NUMB,
PILA,CABE,LISA,PIAR,PTBJ,IB)
*****
C BIC-574
C BIC-575
C BIC-576
C BIC-577
C SUBROUTINA QUE OBTIENE LAS COMPONENTES BICONEJAS DE UN GRAFO BIC-578
C BIC-579
C BIC-580
ENA IB(150,50)
INTEGER V,E,EPTR,EPTR,PUNTO,OLDPT,CDDE,KONT,V2,JOINT,OLDTR
INTEGER EE,EEV,V1,EV,FRENE
INTEGER LISA(EE),SIG(2EV),NUMB(V1),PILA(V1),PTBJ(V),CABE(EE)
INTEGER PIAR(EE),ARTIC(V),BICOM(EV)
C BIC-581
C BIC-582
C BIC-583
C BIC-584
C BIC-585
C PUESTA A CERO C PUESTA A CERO BIC-586
C BIC-587
DO 5 I=1,V BIC-588
NUMB(I)=0 BIC-589
PILA(I)=0 BIC-590
PTBJ(I)=0 BIC-591
ARTIC(I)=0 BIC-592
5 CONTINUE BIC-593
PILA(V1)=0 BIC-594
NUMB(V1)=0 BIC-595
DO 10 I=1,EEV BIC-596
SIG(I)=0 BIC-597
10 CONTINUE BIC-598
DO 15 I=1,EE BIC-599
CABE(I)=0 BIC-600
PIAR(I)=0 BIC-601
LISA(I)=0 BIC-602

```


	15 CONTINUE	BIC-603
	DO 18 I=1,EV	BIC-604
	BICON(I)=0	BIC-605
	18 CONTINUE	BIC-606
C		BIC-607
C	** LLAMADA A LA SUBROUTINA QUE ELABORA LA LISTA DE ARISTAS POR SUS VER-	BIC-608
C	TICES EXTREMOS **	BIC-609
C		BIC-610
	CALL TRA(V,E,EE,LISA,IS)	BIC-611
	FRENE=V	BIC-612
C		BIC-613
C	** LLAMADA A LA SUBROUTINA QUE ELABORA EL CONJUNTO CABE **	BIC-614
C		BIC-615
	DO 20 I=1,E	BIC-616
	CALL PRD(V,EE,EEV,LISA(2*I-1),LISA(2*I),FRENE,SIG,CABE)	BIC-617
	CALL PRD(V,EE,EEV,LISA(2*I),LISA(2*I-1),FRENE,SIG,CABE)	BIC-618
	20 CONTINUE	BIC-619
	JONT=0	BIC-620
	EPTA=0	BIC-621
	SPTR=0	BIC-622
	PUNTO=1	BIC-623
	OLDPT=0	BIC-624
	CODE=1	BIC-625
	KONT=1	BIC-626
	NUMB(PUNTO)=CODE	BIC-627
	PILA(KONT)=PUNTO	BIC-628
	PTBJ(PUNTO)=1	BIC-629
	40 IF(SIG(PUNTO).EQ.0) GO TO 70	BIC-630
	V2=CABE(SIG(PUNTO)-V)	BIC-631
	SIG(PUNTO)=SIG(SIG(PUNTO))	BIC-632
	IF(NUMB(V2).SE.NUMB(PUNTO).DR.V2.EQ.OLDPT) GO TO 40	BIC-633
C		BIC-634
C	** LLAMADA A LA SUBROUTINA QUE ALMACENA ARISTAS POR SUS VERTICES EX-	BIC-635
C	TREMOS **	BIC-636
C		BIC-637
	CALL RD2(EE,PUNTO,V2,PIAR,EPTA)	BIC-638
	IF(NUMB(V2).EQ.0) 50,60	BIC-639
	50 CODE=CODE+1	BIC-640
	KONT=KONT+1	BIC-641
	NUMB(V2)=CODE	BIC-642
	PILA(KONT)=V2	BIC-643
	OLDPT=PUNTO	BIC-644
	PUNTO=V2	BIC-645
	PTBJ(PUNTO)=NUMB(OLDPT)	BIC-646
	GO TO 40	BIC-647
	60 IF(NUMB(V2).LT.PTBJ(PUNTO)) PTBJ(PUNTO)=NUMB(V2)	BIC-648
	GO TO 40	BIC-649
	70 IF(KONT.EQ.1) GO TO 150	BIC-650
	IF(PTBJ(PUNTO).NE.NUMB(OLDPT)) 80,110	BIC-651
	80 IF(PTBJ(PUNTO).LT.PTBJ(OLDPT)) PTBJ(OLDPT)=PTBJ(PUNTO)	BIC-652
	PILA(KONT)=0	BIC-653
	KONT=KONT-1	BIC-654
	PUNTO=PILA(KONT)	BIC-655
	OLDPT=PILA(KONT-1)	BIC-656
	GO TO 40	BIC-657
	110 DO 115 I=1,JONT	BIC-658
	IF(OLDPT.EQ.ARTIC(I)) GO TO 116	BIC-659
	115 CONTINUE	BIC-660
	JONT=JONT+1	BIC-661
C		BIC-662
C	SE OBTIENEN LOS PUNTOS DE ARTICULACION	BIC-663

C	ARTIC(JJNT)=OLDPT	BIC-664
116	BPTR=BPTR+1	BIC-665
	OLDTR=BPTR	BIC-666
120	IF(NUMB(PIAR(EPTR-1)).GT.NUMB(OLDPT)) 130,140	BIC-667
C		BIC-668
C	** LLAMADA A LA SUBROUTINA QUE ALMACENA EL CONJUNTO DE COMPONENTES BI-	BIC-669
C	CONEXAS **	BIC-670
C		BIC-671
130	CALL ADI(EV,PIAR(EPTR-1),PIAR(EPTR),BICON,BPTR,V,E,IB)	BIC-672
	EPTR=EPTR-2	BIC-673
	GO TO 120	BIC-674
140	CALL ADI(EV,OLDPT,PUNTO,BICON,BPTR,V,E,IB)	BIC-675
	EPTR=EPTR-2	BIC-676
	BICON(OLDTR)=BPTR-OLDTR	BIC-677
	GO TO 80	BIC-678
150	RETURN	BIC-679
	END	BIC-680
C		BIC-681
C		EUL-682
C	SUBROUTINE EUL(KA,NV,II,IA)	EUL-683
C	*****	EUL-684
C		EUL-685
C	SUBROUTINA QUE APLICA LA FORMULA DE EULER CONO PRIMER TEST DE PLANARI-	EUL-686
C	DAD	EUL-687
C		EUL-688
	EMA IA(100,50)	EUL-689
	KV=0	EUL-690
	DO 100 I=1,NV	EUL-691
	DO 50 J=1,NV	EUL-692
	IF(IA(I,J).LE.0) GO TO 50	EUL-693
	KV=KV+1	EUL-694
	GO TO 100	EUL-695
50	CONTINUE	EUL-696
100	CONTINUE	EUL-697
	IF(KA.GT.3*KV-6) II=1	EUL-698
	RETURN	EUL-699
	END	EUL-700
C		EUL-701
C		PLA-702
C		PLA-703
	SUBROUTINE PLA(S,NV,NA,EE,EEV,V1,K1,M2,JF,KF,CAN,NUMB,PILA,FI,	PLA-704
	&BUC,FROND,ARAR,ALNAC,ND,PTBJ1,PTBJ2,CADE,F,IAA,IA,IB,ISAUX)	PLA-705
C	*****	PLA-706
C		PLA-707
C	SUBROUTINA QUE CHEQUEA LA PLANARIDAD PARA UN GRAFO BICONEXO	PLA-708
C		PLA-709
	EMA IB(150,50),IA(100,50),IAA(50,10)	PLA-710
	INTEGER CODE,V,S,U,W,VW,EE,EEV,V1	PLA-711
	INTEGER ARAR(NA),FROND(NA),NUMB(V1),FI(NA)	PLA-712
	INTEGER PILA(V1),PTBJ1(NV),PTBJ2(NV),BUC(NA)	PLA-713
	INTEGER CAN(EEV),ND(NV),ALNAC(EE),CADE(NA),F(NA)	PLA-714
C		PLA-715
C	PUESTA A CERD	PLA-716
C		PLA-717
	S=0	PLA-718
	DO 5 I=1,EEV	PLA-719
	CAN(I)=0	PLA-720
5	CONTINUE	PLA-721
	DO 10 I=1,EE	PLA-722
	ALNAC(I)=0	PLA-723

10	CONTINUE	PLA-724
	DO 15 I=1,50	PLA-725
	DO 15 J=1,10	PLA-726
	IAA(I,J)=0	PLA-727
15	CONTINUE	PLA-728
	DO 20 I=1,NA	PLA-729
	ARAR(I)=0	PLA-730
	FROND(I)=0	PLA-731
	FI(I)=0	PLA-732
	SUC(I)=0	PLA-733
20	CONTINUE	PLA-734
	DO 30 I=1,NV	PLA-735
	PTBJ1(I)=0	PLA-736
	PTBJ2(I)=0	PLA-737
	ND(I)=1	PLA-738
	NUMB(I)=0	PLA-739
	PILA(I)=0	PLA-740
	CODE(I)=0	PLA-741
	F(I)=0	PLA-742
30	CONTINUE	PLA-743
	PILA(V1)=0	PLA-744
	NUMB(V1)=0	PLA-745
C		PLA-746
C	DETERMINACION DEL VERTICE INICIAL	PLA-747
C		PLA-748
	DO 32 I=1,NV	PLA-749
	DO 31 J=1,NV	PLA-750
	IF(IAA(I,J).LE.0) GO TO 31	PLA-751
	S=I	PLA-752
	GO TO 33	PLA-753
31	CONTINUE	PLA-754
32	CONTINUE	PLA-755
33	CODE=1	PLA-756
	KONT=1	PLA-757
	V=S	PLA-758
	U=0	PLA-759
	N1=0	PLA-760
	N2=0	PLA-761
	JF=0	PLA-762
	KF=0	PLA-763
	NUMB(V)=CODE	PLA-764
	PILA(KONT)=V	PLA-765
	PTBJ1(V)=NUMB(V)	PLA-766
	PTBJ2(V)=NUMB(V)	PLA-767
C		PLA-768
C	D.F.S.	PLA-769
C		PLA-770
35	DO 100 J=1,NV	PLA-771
	IF(IA(V,J).LE.0) GO TO 100	PLA-772
	U=J	PLA-773
	IA(V,J)=-IA(V,J)	PLA-774
	IA(J,V)=-IA(J,V)	PLA-775
	DO 40 I=1,NA	PLA-776
	IF(IB(I,V).LT.0.AND.IB(I,U).LT.0) GO TO 50	PLA-777
40	CONTINUE	PLA-778
50	VU=I	PLA-779
	IF(NUMB(U).EQ.0) GO TO 70	PLA-780
60	N1=N1+1	PLA-781
	ARAR(N1)=VU	PLA-782
	CODE=CODE+1	PLA-783
	KONT=KONT+1	PLA-784

NUMB(U)=CODE	PLA-785
PILA(KONT)=U	PLA-786
PTBJ1(U)=NUMB(U)	PLA-787
PTBJ2(U)=NUMB(U)	PLA-788
U=V	PLA-789
V=U	PLA-790
GO TO 35	PLA-791
70 N2=N2+1	PLA-792
FROND(N2)=VU	PLA-793
IF(NUMB(U).LT.PTBJ1(V))GO TO 80	PLA-794
IF(NUMB(U).GT.PTBJ1(V)) PTBJ2(V)=MIN0(PTBJ2(V),NUMB(U))	PLA-795
GO TO 100	PLA-796
80 PTBJ2(V)=PTBJ1(V)	PLA-797
PTBJ1(V)=NUMB(U)	PLA-798
100 CONTINUE	PLA-799
IF(KONT.EQ.1) GO TO 140	PLA-800
PILA(KONT)=0	PLA-801
KONT=KONT-1	PLA-802
ND(U)=ND(U)+ND(V)	PLA-803
IF(PTBJ1(V).LT.PTBJ1(U)) GO TO 110	PLA-804
IF(PTBJ1(V).EQ.PTBJ1(U)) PTBJ2(U)=MIN0(PTBJ2(U),PTBJ2(V))	PLA-805
IF(PTBJ1(V).GT.PTBJ1(U)) PTBJ2(U)=MIN0(PTBJ2(U),PTBJ1(V))	PLA-806
GO TO 120	PLA-807
110 PTBJ2(U)=MIN0(PTBJ1(U),PTBJ2(V))	PLA-808
PTBJ1(U)=PTBJ1(V)	PLA-809
120 IF(KONT.EQ.1)GO TO 130	PLA-810
V=PILA(KONT)	PLA-811
U=PILA(KONT-1)	PLA-812
GO TO 35	PLA-813
130 V=1	PLA-814
U=0	PLA-815
GO TO 35	PLA-816
140 DO 150 I=1,NV	PLA-817
DO 150 J=1,NV	PLA-818
IF(IA(I,J).NE.0) IA(I,J)=1	PLA-819
150 CONTINUE	PLA-820
C	PLA-821
C ** LLANADA A LA SUBROUTINA QUE ORDENA LOS TERMINOS DE LA NUEVA MATRIZ	PLA-822
C DE ADYACENCIA USANDO EL D.F.S. **	PLA-823
C	PLA-824
C CALL ORD(ISAL,NV,NA,N2,FROND,NUMB,PTBJ1,PTBJ2,FI,IB,IAA)	PLA-825
C	PLA-826
C ** LLANADA A LA SUBROUTINA QUE OBTIENE EL CIRCUITO ORIGINAL Y LOS CA-	PLA-827
C NINOS DE LOS DISTINTOS SEGMENTOS Y LOS REPRESENTA EN EL PLANO	PLA-828
C CORROBANDO, POR TANTO, LA PLANARIDAD **	PLA-829
C	PLA-830
C CALL CAD(S,NA,NV,EE,EEV,V1,N2,IAA,NUMB,CAN,PILA,FI,BUC,ALBAC,	PLA-831
C &F,CADE,K1,JF,KF)	PLA-832
C RETURN	PLA-833
C END	PLA-834
C	CAD-835
C	CAD-836
C SUBROUTINE CAD(S,NA,NV,EE,EEV,V1,NF,IAA,NUMB,CAN,PILA,SIG,B,	CAD-837
C &ALBAC,F,CADE,K1,JF,KF)	CAD-838
C *****	CAD-839
C	CAD-840
C SUBROUTINA QUE OBTIENE EL CIRCUITO ORIGINAL Y LOS CANINOS DE LOS DIS-	CAD-841
C TOS SEGMENTOS Y LOS REPRESENTA EN EL PLANO, CORROBANDO, POR TANTO,	CAD-842
C LA PLANARIDAD	CAD-843
C	CAD-844

ENH IAA(50,10)	CAD-843
INTEGER S,U,V,W,VW,FF,FREE,K,Y,NE1,NE9,SAVE,RR,SS,SSTT,EE,EEV,V1	CAD-846
INTEGER PILA(NV),CAN(EEV),NUMB(V1)	CAD-847
INTEGER B(MA),ALNAC(EE),SIG(MA),F(MA),CADE(MA)	CAD-848
C	CAD-849
C RENUNERACION	CAD-850
C	CAD-851
N=1	CAD-852
DO 1 I=1,NV	CAD-853
IF(NUMB(I).LE.0) GO TO 1	CAD-854
IF(NUMB(I).GT.N) N=NUMB(I)	CAD-855
1 CONTINUE	CAD-856
NN=N	CAD-857
C	CAD-858
C ** LLAMADA A LA SUBROUTINA QUE MODIFICA LA MATRIZ DE ORDENACION **	CAD-859
C	CAD-860
CALL CA(NV,IAA,NUMB,ALNAC,NN)	CAD-861
DO 3 I=1,NV	CAD-862
DO 2 J=1,9	CAD-863
IF(IAA(I,J).EQ.0) GO TO 3	CAD-864
IAA(I,J)=NUMB(IAA(I,J))	CAD-865
2 CONTINUE	CAD-866
3 CONTINUE	CAD-867
C	CAD-868
C PUESTA A CERO	CAD-869
C	CAD-870
DO 5 I=1,V1	CAD-871
PILA(I)=0	CAD-872
ALNAC(I)=0	CAD-873
5 CONTINUE	CAD-874
DO 8 I=1,MA	CAD-875
B(I)=0	CAD-876
SIG(I)=9	CAD-877
8 CONTINUE	CAD-878
FF=0	CAD-879
NE9=0	CAD-880
NE1=0	CAD-881
STA0=0	CAD-882
FREE=1	CAD-883
SS=0	CAD-884
SSTT=0	CAD-885
U=0	CAD-886
V=5	CAD-887
KONT=1	CAD-888
PILA(I)=V	CAD-889
CADE(V)=1	CAD-890
K1=1	CAD-891
K2=0	CAD-892
10 K2=K2+1	CAD-893
CAN(K1)=-K2	CAD-894
K1=K1+1	CAD-895
15 CAN(K1)=V	CAD-896
20 DO 30 J=1,10	CAD-897
IF(IAA(V,J).GT.0) GO TO 280	CAD-898
IF(IAA(V,J).EQ.0) GO TO 40	CAD-899
30 CONTINUE	CAD-900
40 IF(KONT.LE.1) GO TO 500	CAD-901
C	CAD-902
C SUPRESION DE ENTRADAS A FICHERO Y BLOQUES CORRESPONDIENTES A VERTICES	CAD-903
C NO MAYORES QUE U.	CAD-904
C	CAD-905

45 DO 100 I=FF,1,-2	CAD-906
X=B(I-1)	CAD-907
Y=B(I)	CAD-908
IF(X.EQ.0) GO TO 60	CAD-909
IF(Y.EQ.0) GO TO 70	CAD-910
IF(ALMAC(X).GE.U.AND.ALMAC(Y).GE.U) GO TO 80	CAD-911
IF(ALMAC(X).LT.U) GO TO 50	CAD-912
B(I-1)=0	CAD-913
50 IF(ALMAC(Y).LT.U) GO TO 100	CAD-914
B(I)=0	CAD-915
GO TO 100	CAD-916
60 IF(Y.EQ.0) GO TO 80	CAD-917
IF(ALMAC(Y).LT.U) GO TO 100	CAD-918
GO TO 80	CAD-919
70 IF(ALMAC(X).LT.U) GO TO 100	CAD-920
GO TO 80	CAD-921
C	CAD-922
C ** LLAMADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B **	CAD-923
C	CAD-924
80 CALL SU2(I,FF,B,NA)	CAD-925
IF(FF.EQ.0) GO TO 110	CAD-926
GO TO 45	CAD-927
100 CONTINUE	CAD-928
110 IF(NE1.EQ.0) GO TO 120	CAD-929
IF(ALMAC(NE1).LT.U) GO TO 120	CAD-930
NE1=SIG(NE1)	CAD-931
GO TO 110	CAD-932
120 IF(NE0.EQ.0) GO TO 130	CAD-933
IF(ALMAC(NE0).LT.U) GO TO 130	CAD-934
NE0=SIG(NE0)	CAD-935
GO TO 120	CAD-936
130 IF(CADE(U).EQ.CADE(V)) GO TO 270	CAD-937
C	CAD-938
C TODD SEGMENTO CON LA PRIMERA ARISTA (U,V) HA SIDO EMPOTRADO.	CAD-939
C NUEVOS BLOQUES DEBEN SER MOVIDOS DE DERECHA A IZQUIERDA.	CAD-940
C	CAD-941
LL=0	CAD-942
135 DO 200 I=FF,1,-2	CAD-943
X=B(I-1)	CAD-944
Y=B(I)	CAD-945
IF(NE1.EQ.0) GO TO 146	CAD-946
IF(ALMAC(NE1).EQ.0) GO TO 200	CAD-947
141 IF(X.EQ.0) GO TO 147	CAD-948
IF(ALMAC(X).GT.F(CADE(V))) GO TO 155	CAD-949
142 IF(Y.EQ.0) GO TO 148	CAD-950
IF(ALMAC(Y).GT.F(CADE(V))) 149,200	CAD-951
146 IF(STA0.EQ.0) 200,141	CAD-952
147 IF(STA0.GT.F(CADE(V))) 155,142	CAD-953
148 IF(STA0.GT.F(CADE(V))) 149,200	CAD-954
149 IF(LL.EQ.0) GO TO 151	CAD-955
C	CAD-956
C ** LLAMADAS A SUBROUTINAS QUE REALIZAN OPERACIONES AUXILIARES **	CAD-957
C	CAD-958
CALL VA1(SAVE,NE0)	CAD-959
CALL VA2(NE1,NE0)	CAD-960
151 IF(LL.EQ.-1) CALL VA1(SAVE,NE1)	CAD-961
IF(LL.LE.0) GO TO 152	CAD-962
CALL VA1(SAVE,SIG(LL))	CAD-963
CALL VA2(NE1,SIG(LL))	CAD-964
152 IF(Y.EQ.0) CALL VA3(NE1,NE0,SAVE)	CAD-965

	IF(Y.EQ.-1) CALL VA3(NE1,NE1,SAVE)	CAD-966
	IF(Y.GT.0) CALL VA3(NE1,SIG(Y),SAVE)	CAD-967
	LL=Y	CAD-968
	GO TO 160	CAD-969
155	IF(Y.EQ.0) GO TO 156	CAD-970
	IF(ALMAC(Y).GT.F(CADE(Y))) GO TO 490	CAD-971
	GO TO 158	CAD-972
156	IF(STAO.GT.F(CADE(Y))) GO TO 490	CAD-973
158	LL=X	CAD-974
C		CAD-975
C	** LLANADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B **	CAD-976
C		CAD-977
160	CALL SU2(I,FF,B,NA)	CAD-978
	IF(FF.EQ.0) GO TO 260	CAD-979
	GO TO 135	CAD-980
200	CONTINUE	CAD-981
C		CAD-982
C	EL BLOQUE SOBRE B DEBE SER COMBINADO CON OTROS BLOQUES YA SUPRIMIDOS.	CAD-983
C		CAD-984
210	DO 250 I=FF,1,-2	CAD-985
	X=B(I-1)	CAD-986
	Y=B(I)	CAD-987
	IF(X.NE.0) GO TO 250	CAD-988
	IF(LL.NE.0.OR.Y.NE.0) GO TO 220	CAD-989
C		CAD-990
C	** LLANADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B **	CAD-991
C		CAD-992
	CALL SU2(I,FF,B,NA)	CAD-993
	IF(FF.EQ.0) GO TO 260	CAD-994
	GO TO 210	CAD-995
220	B(I-1)=LL	CAD-996
250	CONTINUE	CAD-997
C		CAD-998
C	SUPRESION DE LA MARCA FIN DE FICHERO SOBRE EL FICHERO DE LA DERECHA.	CAD-999
C		CAD-000
260	IF(NE1.EQ.0) CALL VA2(NE0,NE1)	CAD-001
	IF(NE1.GT.0) CALL VA2(SIG(NE1),NE1)	CAD-002
270	PILA(KONT)=0	CAD-003
	KONT=KONT-1	CAD-004
	V=PILA(KONT)	CAD-005
	U=PILA(KONT-1)	CAD-006
	GO TO 15	CAD-007
280	M=IAA(V,J)	CAD-008
	IAA(V,J)=-IAA(V,J)	CAD-009
	K1=K1+1	CAD-010
	CAN(K1)=M	CAD-011
	IF(V.GT.U) GO TO 330	CAD-012
	U=V	CAD-013
	V=U	CAD-014
	KONT=KONT+1	CAD-015
	PILA(KONT)=M	CAD-016
	CADE(U)=K2	CAD-017
	GO TO 20	CAD-018
330	F(<2)=U	CAD-019
C		CAD-020
C	CAMBIAR LOS BLOQUES DE ENTRADA DE IZQUIERDA A DERECHA DE FORMA QUE	CAD-021
C	PUEDA SER EMPOTRADA POR LA IZQUIERDA	CAD-022
C		CAD-023
	LL=0	CAD-024
	RR=-1	CAD-025
335	IF(LL.EQ.0) GO TO 340	CAD-026

	IF(LL.EB.-1) GO TO 350	CAD-027
	IF(SIG(LL).EB.0) GO TO 360	CAD-028
	IF(ALHAC(SIG(LL)).LE.W) 360.400	CAD-029
340	IF(NE0.EB.0) GO TO 360	CAD-030
	IF(ALHAC(NE0).LE.W) 360.400	CAD-031
350	IF(NE1.EB.0) GO TO 360	CAD-032
	IF(ALHAC(NE1).LE.W) 360.400	CAD-033
360	IF(RR.EB.0) GO TO 370	CAD-034
	IF(RR.EB.-1) GO TO 380	CAD-035
	IF(SIG(RR).EB.0) GO TO 460	CAD-036
	IF(ALHAC(SIG(RR)).LE.W) 460.400	CAD-037
370	IF(NE0.EB.0) GO TO 460	CAD-038
	IF(ALHAC(NE0).LE.W) 460.400	CAD-039
380	IF(NE1.EB.0) GO TO 460	CAD-040
	IF(ALHAC(NE1).LE.W) GO TO 460	CAD-041
400	X=B(FF-1)	CAD-042
	Y=B(FF)	CAD-043
	IF(X.NE.0.AND.Y.NE.0) GO TO 420	CAD-044
	IF(X.EB.0) GO TO 410	CAD-045
	IF(LL.EB.0) GO TO 401	CAD-046
	IF(LL.EB.-1) GO TO 402	CAD-047
	IF(SIG(LL).EB.0) GO TO 460	CAD-048
	IF(ALHAC(SIG(LL)).LE.W) GO TO 460	CAD-049
	GO TO 405	CAD-050
401	IF(NE0.EB.0) GO TO 460	CAD-051
	IF(ALHAC(NE0).LE.W) GO TO 460	CAD-052
	GO TO 405	CAD-053
402	IF(NE1.EB.0) GO TO 460	CAD-054
	IF(ALHAC(NE1).LE.W) GO TO 460	CAD-055
C		CAD-056
C	** LLANADAS A UNA SERIE DE SUBROUTINAS QUE REALIZAN OPERACIONES AUXI-	CAD-057
C	LARES **	CAD-058
C		CAD-059
405	IF(X.EB.-1) CALL VA1(SAVE,NE1)	CAD-060
	IF(X.NE.-1) CALL VA1(SAVE,SIG(X))	CAD-061
	IF(RR.EB.0.AND.X.EB.-1) CALL VA2(NE0,NE1)	CAD-062
	IF(RR.GT.0.AND.X.EB.-1) CALL VA2(SIG(RR),NE1)	CAD-063
	IF(RR.EB.-1.AND.X.NE.-1) CALL VA2(NE1,SIG(X))	CAD-064
	IF(RR.GT.0.AND.X.NE.-1) CALL VA2(SIG(RR),SIG(X))	CAD-065
	IF(RR.EB.0.AND.X.NE.-1) CALL VA2(NE0,SIG(X))	CAD-066
	IF(RR.EB.-1.AND.LL.EB.0) CALL VA3(NE1,NE0,SAVE)	CAD-067
	IF(RR.EB.-1.AND.LL.EB.-1) CALL VA3(NE1,NE1,SAVE)	CAD-068
	IF(RR.GT.0.AND.LL.EB.0) CALL VA3(SIG(RR),NE0,SAVE)	CAD-069
	IF(RR.GT.0.AND.LL.GT.0) CALL VA3(SIG(RR),SIG(LL),SAVE)	CAD-070
	IF(RR.GT.0.AND.LL.EB.-1) CALL VA3(SIG(RR),NE1,SAVE)	CAD-071
	IF(RR.EB.0.AND.LL.GT.0) CALL VA3(NE0,SIG(LL),SAVE)	CAD-072
	IF(RR.EB.-1.AND.LL.GT.0) CALL VA3(NE1,SIG(LL),SAVE)	CAD-073
	RR=X	CAD-074
	GO TO 440	CAD-075
410	IF(RR.EB.0) GO TO 411	CAD-076
	IF(RR.EB.-1) GO TO 412	CAD-077
	IF(SIG(RR).EB.0) GO TO 460	CAD-078
	IF(ALHAC(SIG(RR)).LE.W) GO TO 460	CAD-079
	GO TO 415	CAD-080
411	IF(NE0.EB.0) GO TO 460	CAD-081
	IF(ALHAC(NE0).LE.W) GO TO 460	CAD-082
	GO TO 415	CAD-083
412	IF(NE1.EB.0) GO TO 460	CAD-084
	IF(ALHAC(NE1).LE.W) GO TO 460	CAD-085
415	IF(Y.NE.0) RR=Y	CAD-086

	GO TO 440	CAD-087
420	IF(LL.EQ.0) MNH=NE0	CAD-088
	IF(LL.EQ.-1) MNH=NE1	CAD-089
	IF(LL.GT.0) MNH=SIG(LL)	CAD-090
	IF(MNH.EQ.0) SSTT=STAO	CAD-091
	IF(MNH.NE.0) SSTT=ALHAC(MNH)	CAD-092
	IF(SSTT.GT.W) GO TO 430	CAD-093
	LL=K	CAD-094
	RR=Y	CAD-095
	GO TO 440	CAD-096
430	IF(RR.EQ.0) MNH=NE0	CAD-097
	IF(RR.EQ.-1) MNH=NE1	CAD-098
	IF(RR.GT.0) MNH=SIG(RR)	CAD-099
	IF(MNH.EQ.0) SSTT=STAO	CAD-100
	IF(MNH.NE.0) SSTT=ALHAC(MNH)	CAD-101
	IF(SSTT.GT.W) GO TO 550	CAD-102
C		CAD-103
C	** LLAMADAS A UNA SERIE DE SUBROUTINAS QUE REALIZAN OPERACIONES AUXILIARES **	CAD-104
C		CAD-105
C		CAD-106
	IF(RR.EQ.0) CALL VAI(SAVE,NE0)	CAD-107
	IF(RR.EQ.-1) CALL VAI(SAVE,NE1)	CAD-108
	IF(RR.GT.0) CALL VAI(SAVE,SIG(RR))	CAD-109
	IF(RR.EQ.0.AND.LL.EQ.-1) CALL VA3(NE0,NE1,SAVE)	CAD-110
	IF(RR.EQ.-1.AND.LL.EQ.0) CALL VA3(NE1,NE0,SAVE)	CAD-111
	IF(RR.GT.0.AND.LL.EQ.0) CALL VA3(SIG(RR),NE0,SAVE)	CAD-112
	IF(RR.GT.0.AND.LL.EQ.-1) CALL VA3(SIG(RR),NE1,SAVE)	CAD-113
	IF(RR.GT.0.AND.LL.GT.0) CALL VA3(SIG(RR),SIG(LL),SAVE)	CAD-114
	IF(RR.EQ.0.AND.LL.GT.0) CALL VA3(NE0,SIG(LL),SAVE)	CAD-115
	IF(RR.EQ.-1.AND.LL.GT.0) CALL VA3(NE1,SIG(LL),SAVE)	CAD-116
	SAVE=SIG(X)	CAD-117
	SIG(X)=SIG(Y)	CAD-118
	SIG(Y)=SAVE	CAD-119
	LL=Y	CAD-120
	RR=K	CAD-121
440	II=FF	CAD-122
C		CAD-123
C	** LLAMADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO B **	CAD-124
C		CAD-125
	CALL SU2(II,FF,B,NA)	CAD-126
	IF(FF.EQ.0) GO TO 460	CAD-127
	GO TO 335	CAD-128
450	DO 467 I=K1,1,-1	CAD-129
	IF(CAN(I).NE.-K2) GO TO 467	CAD-130
	SS=CAN(I+1)	CAD-131
	GO TO 460	CAD-132
467	CONTINUE	CAD-133
C		CAD-134
C	ANADIR EL CANINO AL FICHERO DE LA IZQUIERDA SI EL CANINO ES NORMAL	CAD-135
C		CAD-136
468	IF(F(CADE(SS)).GE.W.AND.K2.GT.1) GO TO 470	CAD-137
	IF(LL.EQ.0) LL=FREE	CAD-138
	ALHAC(FREE)=W	CAD-139
	SIG(FREE)=NE0	CAD-140
	NE0=FREE	CAD-141
	FREE=FREE+1	CAD-142
C		CAD-143
C	ANADIR NUEVOS BLOQUES CORRESPONDIENTES A LOS ANTIGUOS BLOQUES COMBINADOS. NUEVOS BLOQUES PUEDEN SER VACIADOS SI EL SEGMENTO QUE CONTIENE AL	CAD-144
C	CANINO ACTUAL NO ES UNA SOLA RANA.	CAD-145
C		CAD-146
C		CAD-147

470 IF(RR.EE.-1) RR=0	CAD-148
C	CAD-149
C ** LLAMADA A LA SUBROUTINA QUE ALMACENA DOS NUEVOS VALORES EN EL CON-	CAD-150
C JUNTO B **	CAD-151
C	CAD-152
IF(LL.NE.0.OR.RR.NE.0.DR.V.NE.99) CALL AD2(HA,LL,RR,B,FF)	CAD-153
IF(V.EQ.99) GO TO 480	CAD-154
C SI EL SEGMENTO QUE CONTIENE AL CAMINO ACTUAL NO ES UNA SOLA RAMA,	CAD-155
C ANADIR UNA MARCA FIN DE FICHERO AL FICHERO DE LA DERECHA.	CAD-156
ALMAC(FREE)=0	CAD-157
SIG(FREE)=NE1	CAD-158
NE1=FREE	CAD-159
FREE=FREE+1	CAD-160
480 K1=K1+1	CAD-161
GO TO 10	CAD-162
490 KF=K1-2	CAD-163
DO 495 I=K1-2,1,-1	CAD-164
IF(CAN(I).LT.0) GO TO 496	CAD-165
495 CONTINUE	CAD-166
496 JF=I+1	CAD-167
GO TO 560	CAD-168
500 K1=K1-1	CAD-169
CAN(K1+1)=0	CAD-170
GO TO 600	CAD-171
550 KF=K1	CAD-172
DO 555 I=K1,1,-1	CAD-173
IF(CAN(I).LT.0) GO TO 556	CAD-174
555 CONTINUE	CAD-175
556 JF=I+1	CAD-176
560 DO 570 I=JF,KF	CAD-177
DO 565 J=1,NV	CAD-178
IF(CAN(I).EQ.NUMB(J)) GO TO 567	CAD-179
565 CONTINUE	CAD-180
GO TO 570	CAD-181
567 CAN(I)=J	CAD-182
570 CONTINUE	CAD-183
600 RETURN	CAD-184
END	CAD-185
C	TRI-186
C	TRI-187
SUBROUTINE TRI(S,NV,HA,EE,EEV,V1,EEE,K1,N2,IOHT,CAN,NUMB,PILA,	TRI-188
ATICON,ALTO,PAD,ND,PTBJ1,PTBJ2,TSTA,INDT,ESTA,GRADO,NUENU,A1,IAA,	TRI-189
SIB)	TRI-190
*****	TRI-191
C	TRI-192
C SUBROUTINA QUE OBTIENE LAS COMPONENTES TRICONEXAS DE UN GRAFO BICONEXO	TRI-193
C	TRI-194
ENA IB(150,50),IAA(50,10)	TRI-195
INTEGER CODE,V,S,D,W,VU,EE,EEV,V1,EEE	TRI-196
INTEGER NUMB(NV),ESTA(EE),TSTA(HA),INDT(HA)	TRI-197
INTEGER PILA(V1),PTBJ1(NV),PTBJ2(NV),TICON(EEE),NUENU(NV)	TRI-198
INTEGER CAN(EEV),ND(NV),PAD(NV),ALTO(NV),A1(NV),GRADO(NV)	TRI-199
C	TRI-200
C PUESTA A CERO	TRI-201
C	TRI-202
DO 10 I=1,EE	TRI-203
ESTA(I)=0	TRI-204
10 CONTINUE	TRI-205
DO 15 I=1,EEV	TRI-206
CAN(I)=0	TRI-207

15	CONTINUE	TRI-208
	DO 20 I=1,EEE	TRI-209
	TICOM(I)=0	TRI-210
20	CONTINUE	TRI-211
	DO 30 I=1,NV	TRI-212
	AI(I)=0	TRI-213
	GRADO(I)=0	TRI-214
	HUENO(I)=0	TRI-215
	PAD(I)=0	TRI-216
	ALTO(I)=0	TRI-217
	PILA(I)=0	TRI-218
30	CONTINUE	TRI-219
	PILA(VI)=0	TRI-220
	DO 40 I=1,NA	TRI-221
	TSTA(I)=0	TRI-222
	INDT(I)=0	TRI-223
40	CONTINUE	TRI-224
	N=1	TRI-225
	DO 50 I=1,NV	TRI-226
	IF(NUMB(I).GT.N) N=NUMB(I)	TRI-227
50	CONTINUE	TRI-228
	NN=N	TRI-229
	DO 160 J=1,NV	TRI-230
	K=0	TRI-231
	DO 150 I=1,NA	TRI-232
	IF(IA(I,J).LT.0) K=K+1	TRI-233
150	CONTINUE	TRI-234
	GRADO(J)=K	TRI-235
160	CONTINUE	TRI-236
C		TRI-237
C	** LLAMADA A LA SUBROUTINA QUE CAMBIA LOS VECTORES DE ACUERDO A LA	TRI-238
C	PRIMERA NUMERACION **	TRI-239
C	NUMERACION **	TRI-240
C		TRI-241
	CALL CA3(NV,ND,NUMB,ESTA)	TRI-242
	CALL CA3(NV,PTBJ1,NUMB,ESTA)	TRI-243
	CALL CA3(NV,PTBJ2,NUMB,ESTA)	TRI-244
	CALL CA3(NV,GRADO,NUMB,ESTA)	TRI-245
	DO 170 I=1,NV	TRI-246
	DO 165 J=1,9	TRI-247
	IF(IAA(I,J).EQ.0) GO TO 170	TRI-248
	IF(IAA(I,J).LT.0) IAA(I,J)=-IAA(I,J)	TRI-249
165	CONTINUE	TRI-250
170	CONTINUE	TRI-251
C		TRI-252
C	** LLAMADA A LA SUBROUTINA QUE CAMBIA LA ORDENACION PARA EL ANALISIS	TRI-253
C	DE LA TRICONECTIVIDAD **	TRI-254
C	TRICONECTIVIDAD **	TRI-255
C		TRI-256
	CALL CA1(NV,PTBJ1,PTBJ2,ESTA,IAA)	TRI-257
C		TRI-258
C	NUEVA NUMERACION DE LOS VERTICES	TRI-259
C		TRI-260
	N2=0	TRI-261
	V=1	TRI-262
	KONT=1	TRI-263
	PILA(KONT)=1	TRI-264
	HUENO(V)=1	TRI-265
	K1=1	TRI-266
	K2=0	TRI-267
210	K2=K2+1	TRI-268

	CAN(K1)=-K2.	TRI-269
	K1=K1+1	TRI-270
215	CAN(K1)=V	TRI-271
220	DO 250 J=1,9	TRI-272
	IF(IAA(V,J).EQ.0) GO TO 260	TRI-273
	IF(IAA(V,J).GT.0) GO TO 270	TRI-274
250	CONTINUE	TRI-275
260	IF(KONT.EQ.1) GO TO 290	TRI-276
	PILA(KONT)=0	TRI-277
	KONT=KONT-1	TRI-278
	N=N-1	TRI-279
	V=PILA(KONT)	TRI-280
	GO TO 215	TRI-281
270	U=IAA(V,J)	TRI-282
	IAA(V,J)=-IAA(V,J)	TRI-283
	K1=K1+1	TRI-284
	CAN(K1)=U	TRI-285
	IF(UENU(U).NE.0) GO TO 280	TRI-286
	UENU(U)=N-ND(U)+1	TRI-287
	PAD(UENU(U))=UENU(V)	TRI-288
	KONT=KONT+1	TRI-289
	PILA(KONT)=U	TRI-290
	V=U	TRI-291
	GO TO 220	TRI-292
280	IF(ALTO(UENU(U)).EQ.0) ALTO(UENU(U))=UENU(V)	TRI-293
	K1=K1+1	TRI-294
	GO TO 210	TRI-295
290	PILA(KONT)=0	TRI-296
C		TRI-297
C	** LLAMADA A LA SUBROUTINA QUE CAMBIA LOS VECTORES , FORMADOS A PARTIR	TRI-298
C	DE LA NUMERACION PRIMITIVA, SEGUN LA ULTIMA NUMERACION.	TRI-299
C		TRI-300
	CALL CA3(NV,ND,UENU,ESTA)	TRI-301
	CALL CA3(NV,GRADO,UENU,ESTA)	TRI-302
	CALL CA3(NV,PTBJ1,UENU,ESTA)	TRI-303
	CALL CA3(NV,PTBJ2,UENU,ESTA)	TRI-304
C		TRI-305
C	CAMBIO DE LOS VECTORES PTBJ1 Y PTBJ2	TRI-306
C		TRI-307
	DO 295 I=1,NV	TRI-308
	IF(PTBJ1(I).LE.0) GO TO 295	TRI-309
	PTBJ1(I)=UENU(PTBJ1(I))	TRI-310
	PTBJ2(I)=UENU(PTBJ2(I))	TRI-311
295	CONTINUE	TRI-312
C		TRI-313
C	** LLAMADA A LA SUBROUTINA QUE CAMBIA LA MATRIZ DE ORDENACION SEGUN LA	TRI-314
C	NUEVA NUMERACION **	TRI-315
C		TRI-316
	CALL CA4(NV,IAA,UENU,ESTA,NN)	TRI-317
	DO 305 I=1,NV	TRI-318
	DO 304 J=1,9	TRI-319
	IF(IAA(I,J).EQ.0) GO TO 305	TRI-320
	IAA(I,J)=UENU(IAA(I,J))	TRI-321
304	CONTINUE	TRI-322
305	CONTINUE	TRI-323
	DO 350 I=1,K1	TRI-324
	IF(CAN(I).LE.0) GO TO 350	TRI-325
	CAN(I)=UENU(CAN(I))	TRI-326
350	CONTINUE	TRI-327
	DO 360 I=1,NN	TRI-328

	AI(I)=IAA(I,1)	TRI-329
	360 CONTINUE	TRI-320
C		TRI-321
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE LAS COMPONENTES ESCINDIDAS DE	TRI-322
C	GRAF0 DCONEXO **	TRI-323
C		TRI-324
	CALL ESC(NA,NV,EE,EEV,V1,EEE,K1,N2,INDT,CAN,PILA,TSTA,INDT,ESTA,	TRI-325
	ALTO,PAO,ND,GRADO,A1,PTBJ1,PTBJ2,TIC08,IAA)	TRI-326
	DO 400 I=1,INDT	TRI-327
	IF(TIC08(I).LT.0) GO TO 400	TRI-328
	DO 380 J=1,NV	TRI-329
	IF(HUENU(J).EQ.TIC08(I)) GO TO 390	TRI-340
	380 CONTINUE	TRI-341
	390 JJ=J	TRI-342
	DO 395 J=1,NV	TRI-343
	IF(HURB(J).EQ.JJ) GO TO 398	TRI-344
	395 CONTINUE	TRI-345
	398 TIC08(I)=J	TRI-346
	400 CONTINUE	TRI-347
	KT=0	TRI-348
	420 RETURN	TRI-349
	END	TRI-350
C		ESC-351
C		ESC-352
	SUBROUTINE ESC(NA,NV,EE,EEV,V1,EEE,K1,NF,HR,CAN,PILA,TSTA,INDT,	ESC-353
	ESTA,ALTO,PAO,ND,GRADO,A1,PTBJ1,PTBJ2,COMPO,IAA)	ESC-354
C	*****	ESC-355
	ENA IAA(50,10)	ESC-356
	INTEGER FLAG,V,U,B,VM,FF,GG,HH,W,A,B,OB,T,KX,YY,SAV1,SAV2,FFF,Y	ESC-357
	INTEGER EE,EEV,V1,EEE	ESC-358
	INTEGER TSTA(NA),ESTA(EE),PAD(NV),GRADO(NV),A1(NV)	ESC-359
	INTEGER PTBJ1(NV),PTBJ2(NV),ND(NV),ALTO(NV),PILA(V1)	ESC-360
	INTEGER CAN(EEV),COMPO(EEE),INDT(NA)	ESC-361
	DO 10 I=1,EE	ESC-362
	10 ESTA(I)=0	ESC-363
	JJ=0	ESC-364
	FF=0	ESC-365
	GG=0	ESC-366
	HH=0	ESC-367
	KONT=1	ESC-368
	SAV1=0	ESC-369
	SAV2=0	ESC-370
	FLAG=0	ESC-371
	V=1	ESC-372
	U=0	ESC-373
	T=0	ESC-374
	PILA(KONT)=V	ESC-375
	40 DO 50 I=1,10	ESC-376
	IF(IAA(V,I).GT.0) GO TO 280	ESC-377
	IF(IAA(V,I).EQ.0) GO TO 60	ESC-378
	50 CONTINUE	ESC-379
	60 IF(KONT.LE.1) GO TO 700	ESC-380
	JJJ=JJ	ESC-381
C		ESC-382
C	** LLAMADA A LA SUBROUTINA USADA PARA ELIMINAR ARISTAS MULTIPLES **	ESC-383
C	EN LAS COMPONENTES TRICONEXAS **	ESC-384
C		ESC-385
	CALL MUL(EE,EEE,NV,GG,U,V,HH,JJ,ESTA,GRADO,COMPO,2)	ESC-386
	IF(JJJ.NE.JJ) GO TO 63	ESC-387
C		ESC-388
C	** LLAMADA A LA SUBROUTINA QUE ALMACENA DOS ELEMENTOS EN EL CONJUNTO	ESC-389

C	ESTA **	ESC-390
C	CALL AD2(EE,U,V,ESTA,GG)	ESC-391
	63 IF(U,ED,1) GO TO 205	ESC-392
	65 DO 200 I=FF,1,-3	ESC-393
	IF(FF,ED,0) GO TO 75	ESC-394
	IF(INDT(FF),NE,0) GO TO 75	ESC-395
	H=TSTA(I-2)	ESC-396
	A=TSTA(I-1)	ESC-397
	B=TSTA(I)	ESC-398
	IF((GRADO(V),ED,2,AND,A1(V),GT,V),OR,U,ED,A) 70,200	ESC-399
		ESC-400
C	PAR DE SEPARACION TIPO 2	ESC-401
C		ESC-402
C	70 IF(U,NE,A,DR,PAD(B),NE,A) GO TO 75	ESC-403
		ESC-404
C	** LLAMADA A LA SUBROUTINA ENPLEADA PARA ELIMINAR TRES ELENENTOS DEL	ESC-405
C	CONJUNTO TSTA **	ESC-406
C		ESC-407
	CALL SUB(I,FF,TSTA,HA)	ESC-408
	GO TO 65	ESC-409
	75 IF(GRADO(V),ED,2,AND,A1(V),GT,V) 80,113	ESC-410
	80 JJ=JJ+1	ESC-411
	IF(ESTA(GG-3),NE,V) T=ESTA(GG-3)	ESC-412
	IF(ESTA(GG-2),NE,V) T=ESTA(GG-2)	ESC-413
		ESC-414
C	** LLAMADAS A LA SUBROUTINA QUE AHADE DOS ELENENTOS AL CONJUNTO COMPO	ESC-415
C	**	ESC-416
C		ESC-417
C	CALL AD2(EEE,U,V,COMPO,HH)	ESC-418
	CALL AD2(EEE,V,T,COMPO,HH)	ESC-419
	CALL AD2(EEE,U,T,COMPO,HH)	ESC-420
	HH=HH+1	ESC-421
	COMPO(HH)=-JJ	ESC-422
		ESC-423
C	** LLAMADAS A LA SUBROUTINA QUE SUPRINE DOS ELENENTOS DEL CONJUNTO ES-	ESC-424
C	TA **	ESC-425
C		ESC-426
C	CALL SUB(0,GG,ESTA,EE)	ESC-427
	CALL SUB(0,GG,ESTA,EE)	ESC-428
	GRADO(V)=GRADO(V)-2	ESC-429
	GRADO(U)=GRADO(U)-1	ESC-430
	GRADO(T)=GRADO(T)-1	ESC-431
	DO 100 J=GG,1,-2	ESC-432
	XX=ESTA(J-1)	ESC-433
	YY=ESTA(J)	ESC-434
	IF(XX,ED,T,AND,YY,ED,U) 90,100	ESC-435
	90 FLAG=1	ESC-436
	SAV1=XX	ESC-437
	SAV2=YY	ESC-438
		ESC-439
C	** LLAMADA A LA SUBROUTINA QUE SUPRINE DOS ELENENTOS DEL CONJUNTO ESTA	ESC-440
C	**	ESC-441
C		ESC-442
C	CALL SUB(J,GG,ESTA,EE)	ESC-443
	GO TO 160	ESC-444
	100 CONTINUE	ESC-445
	GO TO 170	ESC-446
	113 IF(FF,ED,0) GO TO 205	ESC-447
	IF(INDT(FF),NE,0) GO TO 205	ESC-448
		ESC-449

	JJ=JJ+1	ESC-450
C		ESC-451
C	** LLAMADA A LA SUBROUTINA QUE SUPRIME TRES ELEMENTOS DEL CONJUNTO	ESC-452
C	TSTA **	ESC-453
C		ESC-454
	CALL SU3(I,FF,TSTA,HA)	ESC-455
116	IF(GG.EQ.0) GO TO 135	ESC-456
	DO 150 J=GG,1,-2	ESC-457
	XX=ESTA(J-1)	ESC-458
	YY=ESTA(J)	ESC-459
	IF(A.LE.XX.AND.XX.LE.N.AND.A.LE.YY.AND.YY.LE.N) 120,150	ESC-460
120	IF((XX.EQ.A.AND.YY.EQ.B).OR.(XX.EQ.B.AND.YY.EQ.A)) GO TO 130	ESC-461
C		ESC-462
C	** LLAMADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA	ESC-463
C	**	ESC-464
C		ESC-465
	CALL SU2(J,GG,ESTA,EE)	ESC-466
C		ESC-467
C	** LLAMADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO COMPO	ESC-468
C	**	ESC-469
C		ESC-470
	CALL AD2(EEE,XX,YY,COMPO,HH)	ESC-471
	GRADO(XX)=GRADO(XX)-1	ESC-472
	GRADO(YY)=GRADO(YY)-1	ESC-473
	GO TO 116	ESC-474
130	FLAG=1	ESC-475
	SAV1=XX	ESC-476
	SAV2=YY	ESC-477
C		ESC-478
C	** LLAMADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA	ESC-479
C	**	ESC-480
C		ESC-481
	CALL SU2(J,GG,ESTA,EE)	ESC-482
	GO TO 116	ESC-483
150	CONTINUE	ESC-484
C		ESC-485
C	** LLAMADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO COMPO	ESC-486
C	**	ESC-487
C		ESC-488
155	CALL AD2(EEE,A,B,COMPO,HH)	ESC-489
	HH=HH+1	ESC-490
	COMPO(HH)=-JJ	ESC-491
	T=B	ESC-492
C		ESC-493
C	** LLAMADAS A SUBROUTINAS QUE REALIZAN OPERACIONES AUXILIARES **	ESC-494
C		ESC-495
160	IF(FLAG.EQ.1) CALL PA1(FLAG,JJ,SAV1,SAV2,HH,T,U,COMPO,GRADO,HV,	ESC-496
	&EEE)	ESC-497
170	CALL PA2(GG,U,T,JJ,ESTA,GRADO,PAD,A1(U),K1,CAN,V,EEV,HV,EE)	ESC-498
	GO TO 65	ESC-499
200	CONTINUE	ESC-500
C		ESC-501
C	PAR DE SEPARACION TIPO 1	ESC-502
C		ESC-503
205	IF(PTBJ2(V).GE.U.AND.(PTBJ1(V).NE.1.OR.PAD(U).NE.1.OR.V.GT.3)) 210	ESC-504
	&,265	ESC-505
210	JJ=JJ+1	ESC-506
215	IF(GG.EQ.0) GO TO 255	ESC-507
	DO 250 I=GG,1,-2	ESC-508
	XX=ESTA(I-1)	ESC-509
	YY=ESTA(I)	ESC-510

	IF((V.LE.XX.AND.XX.LT.V+ND(V)).OR.(V.LE.YY.AND.YY.LT.V+ND(V))) 220	ESC-511
	2, 250	ESC-512
C	** LLANADA A LA SUBROUTINA QUE SUPRIME DOS ELEMENTOS DEL CONJUNTO ESTA	ESC-513
C	**	ESC-514
C	220 CALL SU2(I,GG,ESTA,EE)	ESC-515
C	** LLANADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO CONPO	ESC-516
C	**	ESC-517
C	CALL AD2(EEE,XX,YY,CONPO,HH)	ESC-518
C	GRADO(XX)=GRADO(XX)-1	ESC-519
C	GRADO(YY)=GRADO(YY)-1	ESC-520
C	GO TO 215	ESC-521
C	250 CONTINUE	ESC-522
C	** LLANADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO CONPO	ESC-523
C	**	ESC-524
C	255 IF(U.NE.1.OR.PTBJ1(V).NE.1) CALL AD2(EEE,U,PTBJ1(V),CONPO,HH)	ESC-525
C	HH=HH+1	ESC-526
C	CONPO(HH)=-JJ	ESC-527
C	IF(A1(U).EQ.V) A1(U)=PTBJ1(V)	ESC-528
C	** LLANADA A LA SUBROUTINA EMPLEADA PARA ELIMINAR ARISTAS MULTIPLES EN	ESC-529
C	LAS COMPONENTES TRICONEXAS **	ESC-530
C	CALL MUL(EE,EEE,MY,GG,U,PTBJ1(V),HH,JJ,ESTA,GRADO,CONPO,1)	ESC-531
C	IF(PTBJ1(V).NE.PAD(U)) GO TO 260	ESC-532
C	JJ=JJ+1	ESC-533
C	** LLANADAS A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO CONPO	ESC-534
C	**	ESC-535
C	CALL AD2(EEE,U,PTBJ1(V),CONPO,HH)	ESC-536
C	CALL AD2(EEE,U,PTBJ1(V),CONPO,HH)	ESC-537
C	CALL AD2(EEE,PTBJ1(V),U,CONPO,HH)	ESC-538
C	HH=HH+1	ESC-539
C	CONPO(HH)=-JJ	ESC-540
C	GO TO 265	ESC-541
C	** LLANADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO ESTA **	ESC-542
C	260 CALL AD2(EE,U,PTBJ1(V),ESTA,GG)	ESC-543
C	GRADO(U)=GRADO(U)+1	ESC-544
C	GRADO(PTBJ1(V))=GRADO(PTBJ1(V))+1	ESC-545
C	265 DO 266 I=1,K1-1	ESC-546
C	IF(CAN(I).GE.0) GO TO 266	ESC-547
C	IF(CAN(I+1).EQ.U.AND.CAN(I+2).EQ.V) GO TO 267	ESC-548
C	266 CONTINUE	ESC-549
C	GO TO 270	ESC-550
C	LA ARISTA (U,V) ES LA PRIMERA DE UN CAMINO GENERADO	ESC-551
C	267 IF(FF.EQ.0) GO TO 279	ESC-552
C	IF(INDT(FF).NE.0) GO TO 269	ESC-553
C	DO 268 I=FF,1,-3	ESC-554
C	** LLANADA A LA SUBROUTINA QUE SUPRIME TRES ELEMENTOS AL CONJUNTO TSTA	ESC-555
C		ESC-556
C		ESC-557
C		ESC-558
C		ESC-559
C		ESC-560
C		ESC-561
C		ESC-562
C		ESC-563
C		ESC-564
C		ESC-565
C		ESC-566
C		ESC-567
C		ESC-568
C		ESC-569
C		ESC-570

C	**	ESC-571
C	CALL SUB(I,FF,TSTA,NA)	ESC-572
	GO TO 267	ESC-573
268	CONTINUE	ESC-574
269	INDT(FF)=0	ESC-575
270	IF(FF.EQ.0) GO TO 279	ESC-576
	IF(INDT(FF).NE.0) GO TO 279	ESC-577
	DO 275 I=FF,1,-3	ESC-578
	H=TSTA(I-2)	ESC-579
	A=TSTA(I-1)	ESC-580
	B=TSTA(I)	ESC-581
	IF(ALTD(U).LE.N) GO TO 275	ESC-582
C	** LLANADA A LA SUBROUTINA QUE SUPRIME TRES ELEMENTOS AL CONJUNTO TSTA	ESC-583
C	**	ESC-584
C	CALL SUB(I,FF,TSTA,NA)	ESC-585
	GO TO 270	ESC-586
275	CONTINUE	ESC-587
279	PILA(KONT)=0	ESC-588
	KONT=KONT-1	ESC-589
	V=PILA(KONT)	ESC-590
	U=PILA(KONT-1)	ESC-591
	GO TO 40	ESC-592
280	M=IAA(V,I)	ESC-593
	IAA(V,I)=-IAA(V,I)	ESC-594
	IF(V.GT.U) GO TO 310	ESC-595
	DO 400 I=1,K1-1	ESC-596
	IF(CAN(I).GE.0) GO TO 400	ESC-597
	IF(CAN(I+1).EQ.V.AND.CAN(I+2).EQ.U) GO TO 410	ESC-598
400	CONTINUE	ESC-599
	GO TO 305	ESC-600
C	LA ARISTA (V,U) ES LA PRIMERA DE UN CAMINO GENERADO	ESC-601
C	410 Y=0	ESC-602
	FFF=FF	ESC-603
420	IF(FF.EQ.0) GO TO 455	ESC-604
	IF(INDT(FF).NE.0) GO TO 455	ESC-605
	DO 450 I=FF,1,-3	ESC-606
	H=TSTA(I-2)	ESC-607
	A=TSTA(I-1)	ESC-608
	B=TSTA(I)	ESC-609
	IF(A.LE.PTBJI(W)) GO TO 450	ESC-610
	Y=MAX0(Y,H)	ESC-611
C	** LLANADA A LA SUBROUTINA QUE SUPRIME TRES ELEMENTOS DEL CONJUNTO	ESC-612
C	TSTA **	ESC-613
C	CALL SUB(I,FF,TSTA,NA)	ESC-614
	BB=8	ESC-615
	GO TO 420	ESC-616
450	CONTINUE	ESC-617
455	IF(FFF.EQ.FF) GO TO 460	ESC-618
	MN=MAX0(Y,U+RD(W)-1)	ESC-619
C	** LLANADAS A LA SUBROUTINA QUE ANADE TRES ELEMENTOS AL CONJUNTO TSTA	ESC-620
C	**	ESC-621
C	CALL AD3(FF,MN,PTBJI(U),BB,TSTA,NA)	ESC-622
		ESC-623
		ESC-624
		ESC-625
		ESC-626
		ESC-627
		ESC-628
		ESC-629
		ESC-630
		ESC-631

	GO TO 505	ESC-632
460	CALL AD3(FF,U+ND(U)-1,PTBJ1(U),V,TSTA,NA)	ESC-633
505	INDT(FF)=1	ESC-634
	KONT=KONT+1	ESC-635
	PILA(KONT)=U	ESC-636
	U=V	ESC-637
	V=U	ESC-638
	GO TO 40	ESC-639
510	DO 550 I=1,KI-1	ESC-640
	IF(CAN(I).GE.0) GO TO 550	ESC-641
	IF(CAN(I+1).EQ.V.AND.CAN(I+2).EQ.U) GO TO 560	ESC-642
550	CONTINUE	ESC-643
	GO TO 620	ESC-644
C		ESC-645
C	LA ARISTA (V,U) ES LA PRIMERA (Y ULTIMA) DE UN CAMINO GENERADO	ESC-646
C		ESC-647
560	Y=0	ESC-648
	FFF=FF	ESC-649
570	IF(FF.EQ.0) GO TO 605	ESC-650
	IF(INDT(FF).NE.0) GO TO 605	ESC-651
	DO 600 I=FF,1,-3	ESC-652
	H=TSTA(I-2)	ESC-653
	A=TSTA(I-1)	ESC-654
	B=TSTA(I)	ESC-655
	IF(A.LE.U) GO TO 600	ESC-656
	Y=MAX0(Y,H)	ESC-657
C		ESC-658
C	** LLAMADA A LA SUBROUTINA QUE SUPRIME TRES ELEMENTOS DEL CONJUNTO TSTA	ESC-659
C	**	ESC-660
C		ESC-661
	CALL SUB(I,FF,TSTA,NA)	ESC-662
	BB=B	ESC-663
	GO TO 570	ESC-664
600	CONTINUE	ESC-665
605	IF(FFF.EQ.FF) GO TO 610	ESC-666
C		ESC-667
C	** LLAMADAS A LA SUBROUTINA QUE AÑADE TRES ELEMENTOS AL CONJUNTO TSTA	ESC-668
C	**	ESC-669
C		ESC-670
	CALL AD3(FF,V,U,BB,TSTA,NA)	ESC-671
	GO TO 620	ESC-672
610	CALL AD3(FF,V,U,V,TSTA,NA)	ESC-673
620	JJJ=JJ	ESC-674
C		ESC-675
C	** LLAMADA A LA SUBROUTINA EMPLEADA PARA ELIMINAR ARISTAS MULTIPLES EN	ESC-676
C	LAS COMPONENTES TRICONEXAS **	ESC-677
C		ESC-678
	CALL NUL(EE,EEE,NV,GG,V,U,HH,JJ,ESTA,GRADO,COMPO,2)	ESC-679
	IF(JJJ.NE.JJ.OR.V.EQ.PAD(V)) GO TO 625	ESC-680
C		ESC-681
C	** LLAMADA A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO ESTA **	ESC-682
C		ESC-683
	CALL AD2(EE,V,U,ESTA,GG)	ESC-684
	GO TO 40	ESC-685
625	IF(U.NE.PAD(V)) GO TO 40	ESC-686
630	JJ=JJ+1	ESC-687
C		ESC-688
C	** LLAMADAS A LA SUBROUTINA QUE AÑADE DOS ELEMENTOS AL CONJUNTO COMPO	ESC-689
C	**	ESC-690
C		ESC-691

```
CALL AD2(EEE,V,U,COMPO,HH)
CALL AD2(EEE,V,U,COMPO,HH)
CALL AD2(EEE,U,V,COMPO,HH)
HH=HH+1
COMPO(HH)=-JJ
GRADO(V)=GRADO(V-1)
GRADO(U)=GRADO(U-1)
GO TO 40
700 RETURN
END
```

```
ESC-692
ESC-693
ESC-694
ESC-695
ESC-696
ESC-697
ESC-698
ESC-699
ESC-700
ESC-701
ESC-702
```

FTN4		PL3-000
SENA(IJK,3)		PL3-001
C		PL3-002
C		PL3-003
C	*****	PL3-004
C	PROGRAM PL3(S)	PL3-005
C	*****	PL3-006
C		PL3-007
C		PL3-008
C	SECCIONTO 3	PL3-009
C		PL3-010
C		PL3-011
C	COMMON IPAR(4),NAME(3)	PL3-012
C	COMMON IENT,ISAL,IEAUX,ISAUX,NV,NA,KKTT	PL3-013
C	COMMON NON(40),JAUX(2200)	PL3-014
C	COMMON /IJK/ IB(150,50),IA(100,50),X(300),Y(300),IAR(50,10)	PL3-015
C	IOP=0	PL3-016
C		PL3-017
C	** LLAMADA A LA SUBROUTINA QUE APLICA EL TEST DE RECTANGULARIDAD **	PL3-018
C		PL3-019
C	60 CALL REC(KI,NV,NA,JAUX(1),JAUX(101),IA)	PL3-020
C		PL3-021
C	SI NO LO SUPERA SE HAN DE SUSTITUIR LAS ARISTAS AFECTADAS POR OTRAS	PL3-022
C		PL3-023
C	IF(KI.EB.0) GO TO 110	PL3-024
C	WRITE(ISAL,3001)	PL3-025
C	DO 100 I=1,KI	PL3-026
C	II=4*(I-1)+1	PL3-027
C	IT=4*(I-1)+4	PL3-028
C	WRITE(ISAL,3002) (JAUX(J),J=II,IT)	PL3-029
C	100 CONTINUE	PL3-030
C	WRITE(ISAL,3003)	PL3-031
C		PL3-032
C	** LLAMADA A LA SUBROUTINA QUE LEE LAS ARISTAS QUE SUSTITUYEN A OTRAS	PL3-033
C	**	PL3-034
C		PL3-035
C	CALL LEI(IEAUX,ISAL,ISAUX,1,IA,IB)	PL3-036
C		PL3-037
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE UN CIRCUITO DEL GRAFO **	PL3-038
C		PL3-039
C	CALL CIR(NV,NA,NV+1,JAUX(3*NA+3*NV+1),JAUX(3*NA+4*NV+2),JAUX(9*NA+	PL3-040
C	86*NV+3),JAUX(9*NA+7*NV+3),KK,LONT,IA)	PL3-041
C		PL3-042
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE LAS COORDENADAS DE LOS VERTICES	PL3-043
C	APLICANDO EL ALGORITMO DE TOUTE **	PL3-044
C		PL3-045
C	CALL OBC(NV,NA,KK,LONT,JAUX(3*NA+3*NV+1),JAUX(9*NA+7*NV+3),X(1),	PL3-046
C	SY(1),K(4V+1),Y(NV+1),JAUX(9*NA+3*NV+3),IA,IB,KKTT,ISAL,ISAUX)	PL3-047
C		PL3-048
C	SE VUELVE A COMPROBAR LA RECTANGULARIDAD	PL3-049
C		PL3-050
C	GO TO 60	PL3-051
C	110 IF(IOP.NE.0) GO TO 120	PL3-052
C	NA=NA	PL3-053
C	111 WRITE(ISAL,3004)	PL3-054
C		PL3-055
C	** LLAMADA A LA SUBROUTINA QUE LEE LAS NUEVAS ARISTAS INTRODUCIDAS **	PL3-056
C		PL3-057

CALL LE2(IEAUX,ISAL,ISAUX,1,NA,MV,IA,IB)	PL3-058
NEE=3*MV-7	PL3-059
IF(NA.EB.NEE) GO TO 118	PL3-060
WRITE(ISAL,3005) NA,NEE	PL3-061
IF(NA.LT.NEE) GO TO 111	PL3-062
DO 116 I=NEE+1,NA	PL3-063
J1=0	PL3-064
J2=0	PL3-065
DO 114 J=1,MV	PL3-066
IF(18(1,J).EB.0) GO TO 115	PL3-067
IF(J1.NE.0) GO TO 115	PL3-068
J1=J	PL3-069
IB(1,J1)=0	PL3-070
114 CONTINUE	PL3-071
115 J2=J	PL3-072
IB(1,J2)=0	PL3-073
IA(J1,J2)=0	PL3-074
IA(J2,J1)=0	PL3-075
116 CONTINUE	PL3-076
NA=NEE	PL3-077
WRITE(ISAL,3006)	PL3-078
CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)	PL3-079
118 WRITE(ISAL,3007)	PL3-080
C	PL3-081
C ** LLANADA A LA SUBROUTINA QUE DIBUJA EL GRAFO TRIANGULADO **	PL3-082
C	PL3-083
120 IF(ISSU(3).LT.0) GO TO 130	PL3-084
R=5.	PL3-085
CALL TUT(NA,MV,R,JAUX(3*NA+3*MV+1),K,Y,IB,KKT,ISAL)	PL3-086
C	PL3-087
C ** LLANADA A LA SUBROUTINA QUE TRASFORMA LAS COORDENADAS DE LOS VERTI-	PL3-088
C CES INTERIORES **	PL3-089
C	PL3-090
130 CALL DIC(ISAL,MV,VA,KONT,JONT,KHAK,YHAK,5*MV,JAUX(1),JAUX(9),	PL3-091
BJAUX(9+5*MV),X(MV+1),Y(MV+1),IA,IB)	PL3-092
C	PL3-093
C ** LLANADA A LA SUBROUTINA QUE OBTIENE EL GRAFO DUAL **	PL3-094
C	PL3-095
CALL DUA(ISAL,ISAUX,JONT,KONT,JAUX(9),JAUX(9+5*MV),X(MV+1),Y(MV+1)	PL3-096
&,KHAK,YHAK)	PL3-097
C	PL3-098
C ELECCION DEL PARAMETRO OPCIONAL INTERACTIVO	PL3-099
C	PL3-100
WRITE(ISAL,3008)	PL3-101
READ(IEAUX,*) IOP	PL3-102
IF(IOP.EB.0) GO TO 220	PL3-103
C	PL3-104
C ** LLANADA A LA SUBROUTINA QUE LEE LAS ARISTAS QUE SUSTITUYEN A OTRAS	PL3-105
C	PL3-106
CALL LE1(IEAUX,ISAL,ISAUX,1,IA,IB)	PL3-107
GO TO 60	PL3-108
220 STOP	PL3-109
C	PL3-110
C FDRMATOS	PL3-111
C	PL3-112
3001 FORMAT(/IX*MENSAJE I:SR: EL GRAFO NO SUPERA EL TEST DE RECTANGUL	PL3-113
ARIDAD//IX*LOS GRUPOS DE VERTICES AFECTADOS SON: /2X*V1 V2 V3	PL3-114
&V4*/2X*-- -- --)	PL3-115
3002 FORMAT(4I4)	PL3-116
3003 FORMAT(/IX*MENSAJE I:SR: HAS DE SUSTITUIR UNA ARISTA DE CADA GRUP	PL3-117
LO POR OTRA DE FORMA QUE*/16X*SE SUPERE EL TEST.')	PL3-118

3004	FORMAT(/1X'MENSAJE :1SR: A LA VISTA DEL GRAFO TRICONEJO DIBUJADO	PL3-119
	&, INTRODUCE LAS ARISTAS"/16K'NECESARIAS PARA TRIANGULAR EL GRAFO."	PL3-120
)	PL3-121
3005	FORMAT(/1X'MENSAJE :1SR: EL NUMERO DE ARISTAS ES"14"Y SIN ENBARG	PL3-122
	&D TENDRIA QUE SER"14)	PL3-123
3006	FORMAT(/1X'MENSAJE :1SR: SE HAN SUPRIMIDO LAS ULTIMAS ARISTAS "/	PL3-124
	&/16K'SI DESEAS PUEDES SUSTITUIR ALGUNAS ARISTAS")	PL3-125
3007	FORMAT(/1X'MENSAJE :1SR: GRAFO TRIANGULADO")	PL3-126
3008	FORMAT(/1X'MENSAJE :1SR: SI DESEAS TERMINAR INTRODUCE: 0"/16K'EN	PL3-127
	& CASO CONTRARIO INTRODUCE: 1")	PL3-128
	END	PL3-129
C		REC-130
C		REC-131
	SUBROUTINE REC(K1,NV,NA,JAUX,IAUX,IA)	REC-132
	*****	REC-133
C		REC-134
C	SUBROUTINA QUE APLICA EL TEST DE RECTANGULARIDAD	REC-135
C		REC-136
	ENA IA(100,50)	REC-137
	DIMENSION IAUX(15),JAUX(100)	REC-138
	DO 10 I=1,15	REC-139
	IAUX(I)=0	REC-140
10	CONTINUE	REC-141
	DO 20 I=1,100	REC-142
	JAUX(I)=0	REC-143
20	CONTINUE	REC-144
	K1=0	REC-145
	DO 300 I=1,NV	REC-146
	K=0	REC-147
	DO 50 J=1,NV	REC-148
	IF(IA(I,J).EQ.0) GO TO 50	REC-149
	K=K+1	REC-150
	IAUX(K)=J	REC-151
50	CONTINUE	REC-152
	IF(K.LT.3) GO TO 210	REC-153
	DO 200 J1=1,K-2	REC-154
	DO 150 J2=J1+1,K-1	REC-155
	IF(IA(IAUX(J1),IAUX(J2)).LE.0) GO TO 150	REC-156
	DO 100 J3=J2+1,K	REC-157
	IF(IA(IAUX(J1),IAUX(J3)).LE.0) GO TO 100	REC-158
	IF(IA(IAUX(J2),IAUX(J3)).LE.0) GO TO 100	REC-159
	K1=K1+1	REC-160
	JAUX(4*(K1-1)+1)=I	REC-161
	JAUX(4*(K1-1)+2)=IAUX(J1)	REC-162
	JAUX(4*(K1-1)+3)=IAUX(J2)	REC-163
	JAUX(4*(K1-1)+4)=IAUX(J3)	REC-164
	IA(IAUX(J1),I)=-1	REC-165
	IA(IAUX(J2),I)=-1	REC-166
	IA(IAUX(J3),I)=-1	REC-167
	IA(I,IAUX(J1))=-1	REC-168
	IA(I,IAUX(J2))=-1	REC-169
	IA(I,IAUX(J3))=-1	REC-170
100	CONTINUE	REC-171
150	CONTINUE	REC-172
200	CONTINUE	REC-173
210	DO 250 J=1,K	REC-174
250	IAUX(J)=0	REC-175
300	CONTINUE	REC-176
	DO 350 I=1,NV	REC-177
	DO 350 J=1,NV	REC-178

	IF(IA(I,J).LT.0) IA(I,J)=1	REC-179
350	CONTINUE	REC-180
	RETURN	REC-181
	END	REC-182
C		DIC-183
C		DIC-184
	SUBROUTINE DIC(IAL,NV,NA,KONT,JOINT,XMAX,YMAX,V5,IVER,LCN,LP,XN,	DIC-185
	LYN,IA,IB)	DIC-186
C	*****	DIC-187
C		DIC-188
C	SUBROUTINA QUE TRANSFORMA LAS COORDENADAS DE LOS VERTICES INTERIORES DE	DIC-189
C	FORMA QUE TODAS LAS ARISTAS SEAN ORTOGONALES	DIC-190
C		DIC-191
	INTEGER V5	DIC-192
	ENA IB(150,50),IA(100,50),XN(V5),YN(V5)	DIC-193
	DIMENSION IVER(8),LCN(NV),LP(V5)	DIC-194
C		DIC-195
C	PUESTA A CERO	DIC-196
C		DIC-197
	DO 10 I=1,8	DIC-198
	IVER(I)=0	DIC-199
10	CONTINUE	DIC-200
	DO 20 I=1,NV	DIC-201
	LCN(I)=0	DIC-202
20	CONTINUE	DIC-203
	DO 30 I=1,V5	DIC-204
	XN(I)=0	DIC-205
	YN(I)=0	DIC-206
	LP(I)=0	DIC-207
	30 CONTINUE	DIC-208
C		DIC-209
C	OSTENCION DE LOS VERTICES INTERIORES DOBLEMENTE ORIENTADOS Y ORDENA-	DIC-210
C	CION DE LOS MISMOS	DIC-211
C		DIC-212
	KT=0	DIC-213
	DO 100 I=5,NV	DIC-214
	KTI=0	DIC-215
	DO 50 J=1,NV	DIC-216
	IF(IA(I,J).EQ.0) GO TO 50	DIC-217
	IF(J.LE.4) KTI=KTI+1	DIC-218
50	CONTINUE	DIC-219
	IF(KTI.EQ.2) GO TO 100	DIC-220
	KT=KT+1	DIC-221
	IVER(KT)=1	DIC-222
	IF(KT.EQ.4) GO TO 110	DIC-223
100	CONTINUE	DIC-224
110	DO 150 I=1,4	DIC-225
	II=IVER(I)	DIC-226
	IF(IA(II,1).NE.0.AND. IA(II,4).NE.0) IVER(5)=II	DIC-227
	IF(IA(II,4).NE.0.AND. IA(II,3).NE.0) IVER(6)=II	DIC-228
	IF(IA(II,3).NE.0.AND. IA(II,2).NE.0) IVER(7)=II	DIC-229
	IF(IA(II,2).NE.0.AND. IA(II,1).NE.0) IVER(8)=II	DIC-230
150	CONTINUE	DIC-231
	DO 200 I=1,4	DIC-232
	IVER(I)=IVER(4+I)	DIC-233
200	CONTINUE	DIC-234
C		DIC-235
C	OSTENCION DEL CIRCUITO INTERIOR MAS CERCAHO AL EXTERIOR QUE ESTARA	DIC-236
C	FORMADO POR TODOS LOS VERTICES QUE SON ADYACENTES A LOS EXTERIORES	DIC-237
C		DIC-238
	LCN(1)=IVER(1)	DIC-239

KONT=1	DIC-240
KT=0	DIC-241
LCA=0	DIC-242
210 KT=KT+1	DIC-243
220 LCC=LCN(KONT)	DIC-244
DO 250 I=5,NV	DIC-245
IF(I.EQ.LCC.OR.I.EQ.LCA) GO TO 250	DIC-246
IF(IA(I,LCC).NE.0.AND.IA(I,KT).NE.0) 230,250	DIC-247
230 KONT=KONT+1	DIC-248
LCN(KONT)=I	DIC-249
IF(LCN(I).EQ.I) GO TO 360	DIC-250
LCA=LCC	DIC-251
IF(KT.EQ.4) GO TO 220	DIC-252
IF(IA(I,KT+1).NE.0) 210,220	DIC-253
250 CONTINUE	DIC-254
C	DIC-255
C SALIDA DE ERROR SI NO EXISTE ESTE CIRCUITO	DIC-256
C	DIC-257
IERR=NER(ISAL,4)	DIC-258
BTJP 0004	DIC-259
360 DO 600 I=1,KONT	DIC-260
IF(LCN(I).EQ.IVER(2)) LCN(KONT+1)=I	DIC-261
IF(LCN(I).EQ.IVER(3)) LCN(KONT+2)=I	DIC-262
IF(LCN(I).EQ.IVER(4)) LCN(KONT+3)=I	DIC-263
600 CONTINUE	DIC-264
IF(LCN(KONT+1).GT.LCN(KONT+2)) 601,609	DIC-265
601 J=KONT/2	DIC-266
DO 605 I=1,J	DIC-267
LL=LCN(I)	DIC-268
LCN(I)=LCN(KONT+1-I)	DIC-269
LCN(KONT+1-I)=LL	DIC-270
605 CONTINUE	DIC-271
DO 608 I=1,KONT	DIC-272
IF(LCN(I).EQ.IVER(2)) LCN(KONT+1)=I	DIC-273
IF(LCN(I).EQ.IVER(3)) LCN(KONT+2)=I	DIC-274
IF(LCN(I).EQ.IVER(4)) LCN(KONT+3)=I	DIC-275
608 CONTINUE	DIC-276
C	DIC-277
C SUBDIVISION DEL CIRCUITO EN CUATRO LADOS CADA UNO CORRESPONDIENTE A	DIC-278
C CADA UNA DE LAS ORIENTACIONES.	DIC-279
C DETERMINACION DEL LADO (PORCION DE CIRCUITO) QUE CONTENGA MAYOR NUMERO	DIC-280
C DE VERTICES INTERIORES	DIC-281
C ESTE LADO SERA EL INICIAL EN LA TRANSFORMACION DE COORDENADAS DE LOS	DIC-282
C VERTICES INTERIORES DEL GRAFO.	DIC-283
C	DIC-284
609 INAX=LCN(KONT+1)-1	DIC-285
JJ=18	DIC-286
IF(LCN(KONT+2)-LCN(KONT+1).LT.INAX) GO TO 610	DIC-287
INAX=LCN(KONT+2)-LCN(KONT+1)	DIC-288
JJ=16	DIC-289
610 IF(LCN(KONT+3)-LCN(KONT+2).LT.INAX) GO TO 620	DIC-290
INAX=LCN(KONT+3)-LCN(KONT+2)	DIC-291
JJ=17	DIC-292
620 IF(KONT-LCN(KONT+3).GT.INAX) JJ=15	DIC-293
C	DIC-294
IF(JJ.NE.16) GO TO 630	DIC-295
INI=LCN(KONT+2)	DIC-296
INF=LCN(KONT+1)	DIC-297
IND=KONT	DIC-298
IFO=LCN(KONT+3)	DIC-299

INC=-1	DIC-300
IY=1	DIC-301
IX=0	DIC-302
630 IF(JJ.NE.15) GO TO 640	DIC-303
INI=LCM(KONT+3)	DIC-304
INF=KONT	DIC-305
IND=LCM(KONT+1)	DIC-306
IFD=LCM(KONT+2)	DIC-307
INC=1	DIC-308
IY=-1	DIC-309
IX=0	DIC-310
640 IF(JJ.NE.16) GO TO 660	DIC-311
INI=LCM(KONT+1)	DIC-312
INF=1	DIC-313
IND=LCM(KONT+3)	DIC-314
IFD=LCM(KONT+2)	DIC-315
INC=-1	DIC-316
IY=0	DIC-317
IX=-1	DIC-318
650 IF(JJ.NE.17) GO TO 670	DIC-319
INI=LCM(KONT+2)	DIC-320
INF=LCM(KONT+3)	DIC-321
IND=1	DIC-322
IFD=LCM(KONT+1)	DIC-323
INC=1	DIC-324
IY=0	DIC-325
IX=1	DIC-326
670 JT=JJ	DIC-327
C	DIC-328
C	DIC-329
C ** LLAMADA A LA SUBROUTINA QUE OBTIENE LAS NUEVAS COORDENADAS DE LOS	DIC-330
C VERTICES INTERIORES **	DIC-331
C	DIC-332
C LA LLAMADA VARIA SEGUN EL LADO INICIAL SEA NORTE O SUR (LA PRIMERA) O	DIC-333
C ESTE U DESTE (LA SEGUNDA)	DIC-334
C	DIC-335
C 800 IF(IX.EQ.0) CALL COR(MA,V5,MV,INI,INF,INC,IY,JONT,KN,YN,LCM,IVER,	DIC-336
C &LP,IB,KONT,IND,IFD,IA)	DIC-337
C IF(IX.NE.0) CALL COR(MA,V5,MV,INI,INF,INC,IX,JONT,YN,XN,LCM,IVER,	DIC-338
C &LP,IB,KONT,IND,IFD,IA)	DIC-339
C	DIC-340
C ** DIBUJO OPCIONAL DEL SUBGRAFO DE VERTICES Y ARISTAS INTERIORES	DIC-341
C TRASFORMADO CON LLAMADA A LA SUBROUTINA CORRESPONDIENTE **	DIC-342
C	DIC-343
C IF(ISSM(4).LT.0) CALL DIB(I3AL,JONT,KN,YN,LP,XMAX,YMAX)	DIC-344
C RETURN	DIC-345
C END	DIC-346
C	COR-347
C	COR-348
C SUBROUTINE COR(MA,V5,MV,INI,INF,INC,IXY,KONT,K,Y,LCM,IVER,LP,IB,	COR-349
C &KTT,IND,IFD,IA)	COR-350
C *****	COR-351
C	COR-352
C SUBROUTINA QUE OBTIENE LAS NUEVAS COORDENADAS DE LOS VERTICES INTERIO-	COR-353
C RES. ESTA BASADA EN EL DESDUBLAMIENTO DE UNO DE LOS VERTICES DE CADA	COR-354
C TRIANGULO OBTENIENDO SUCESIVOS RECTANGULOS	COR-355
C	COR-356
C INTEGER V5,V4UX(10)	COR-357
C ENA IB(150,50),IA(100,50),X(V5),Y(V5)	COR-358
C DIMENSION LCM(MV),IVER(8),LP(V5)	COR-359
C	COR-360

C INCREMENTOS EN X E Y DE LAS COORDENADAS: 2 CMS.

C

KONT=0
SUNX=-2.
SUNY=0.

C

C COORDENADAS DE LOS VERTICES DEL LADO INICIAL

C

DO 100 I=INI,INF,INC
J=LCN(I)
SUNX=SUNX+2.
KONT=KONT+1
LP(KONT)=J
K(KONT)=SUNX
Y(KONT)=SUNY
100 CONTINUE

C

C COORDENADAS DE UN NUEVO LADO

C

JONT=KONT
LONT=0
DO 800 IT=1,100
SUNY=SUNY+2*IKY
IONT=LONT+1
LONT=KONT
DO 325 JJ=IONT,LONT
J=LP(JJ)
JJJ=0
K1=0
DO 250 I=1,NA
IF(I8(I,J).EQ.0) GO TO 250
DO 200 K=1,NV
IF(K.EQ.J.OR.I8(I,K).EQ.0) GO TO 200
IF(K.LE.4) GO TO 250
DO 150 KK=1,LONT
IF(K.EQ.LP(KK)) GO TO 250
150 CONTINUE
K1=K1+1
JJJ=K
GO TO 250
200 CONTINUE
250 CONTINUE
DO 300 IH=IND,IFD,INC
IF(J.EQ.LCN(IH)) GO TO 310
300 CONTINUE
IF(K1-1) 325,305,310
305 IF(JJJ.NE.LP(KONT)) JONT=JONT+1
GO TO 320
310 JJJ=J
320 KONT=KONT+1
LP(KONT)=JJJ
K(KONT)=K(JJJ)
Y(KONT)=SUNY
325 CONTINUE
326 J1=0
J2=0
JJJ=0
K1=1

C

C ** LLAMADA A LA SUBROUTINA-TEST PARA TERMINAR LA TRANSFORMACION **

COR-361
COR-362
COR-363
COR-364
COR-365
COR-366
COR-367
COR-368
COR-369
COR-370
COR-371
COR-372
COR-373
COR-374
COR-375
COR-376
COR-377
COR-378
COR-379
COR-380
COR-381
COR-382
COR-383
COR-384
COR-385
COR-386
COR-387
COR-388
COR-389
COR-390
COR-391
COR-392
COR-393
COR-394
COR-395
COR-396
COR-397
COR-398
COR-399
COR-400
COR-401
COR-402
COR-403
COR-404
COR-405
COR-406
COR-407
COR-408
COR-409
COR-410
COR-411
COR-412
COR-413
COR-414
COR-415
COR-416
COR-417
COR-418
COR-419
COR-420

C

CALL BAL(NV,VS,LOHT,KOHT,LP,INO,IFD,INC,LCN,KRR)	COR-421
IF(KRR.EQ.0) GO TO 367	COR-422
DO 366 I=LOHT+1,KOHT-1	COR-423
IF(LP(I).EQ.LP(I+1)) GO TO 327	COR-424
IF(JJJ.EQ.0) GO TO 366	COR-425
J2=1	COR-426
GO TO 328	COR-427
327 K1=K1+1	COR-428
IF(JJJ.EQ.0) J1=1	COR-429
JJJ=LP(I)	COR-430
IF(I.NE.KOHT-1) GO TO 366	COR-431
J2=KOHT	COR-432
328 KKK=0	COR-433
DO 330 IH=INO,IFD,INC	COR-434
IF(JJJ.EQ.LCN(IH)) GO TO 365	COR-435
330 CONTINUE	COR-436
NT=0	COR-437
KT=KOHT-LOHT	COR-438
DO 332 IN=LOHT+1,KOHT	COR-439
DO 331 JH=INO,IFD,INC	COR-440
IF(K(JH).NE.K(IN)) GO TO 331	COR-441
IF(LP(IH).EQ.LP(JH)) NT=NT+1	COR-442
GO TO 332	COR-443
331 CONTINUE	COR-444
332 CONTINUE	COR-445
IF(KT.NE.NT) GO TO 365	COR-446
DO 340 K=3,NV	COR-447
DO 335 L=1,KOHT	COR-448
IF(K.EQ.LP(L)) GO TO 340	COR-449
335 CONTINUE	COR-450
IF(IA(K,JJJ).EQ.0) GO TO 340	COR-451
KKK=KKK+1	COR-452
VAUX(KKK)=K	COR-453
340 CONTINUE	COR-454
IF(KKK.EQ.0) GO TO 365	COR-455
IF(KKK.GT.K1) GO TO 363	COR-456
IF(J1.NE.LOHT+1.AND.J2.NE.KOHT) GO TO 354	COR-457
NN=0	COR-458
IF(J1.NE.LOHT+1) GO TO 352	COR-459
342 DO 350 K=1,KKK	COR-460
KK=VAUX(K)	COR-461
DO 345 L=1,4	COR-462
IF(IA(L,JJJ).NE.0.AND.IA(L,KK).NE.0) GO TO 351	COR-463
345 CONTINUE	COR-464
350 CONTINUE	COR-465
351 IF(NN.NE.0) GO TO 353	COR-466
LP(J1)=KK	COR-467
JOHT=JOHT+1	COR-468
J1=J1+1	COR-469
352 NN=NN+1	COR-470
IF(J2.NE.KOHT) 354,342	COR-471
353 LP(J2)=KK	COR-472
JOHT=JOHT+1	COR-473
J2=J2+1	COR-474
354 JJ=LP(J1-1)	COR-475
NN=0	COR-476
I1=0	COR-477
I2=0	COR-478
355 DO 356 K=1,KKK	COR-479
KK=VAUX(K)	COR-480
	COR-481

IF(IACKK, JJ).NE.0) GO TO 357	COR-482
356 CONTINUE	COR-483
IF(NH.ED.0) 358,360	COR-484
357 IF(NH.NE.0) GO TO 359	COR-485
I1=KK	COR-486
358 NH=NH+1	COR-487
IF(J1.ED.J2) GO TO 360	COR-488
JJ=LP(J2+1)	COR-489
GO TO 355	COR-490
359 I2=KK	COR-491
360 IF(I1.ED.0.AND.I2.ED.0) GO TO 362	COR-492
IF(I1.ED.0) GO TO 361	COR-493
LP(J1)=I1	COR-494
JONT=JONT+1	COR-495
J1=J1+1	COR-496
361 IF(I2.ED.0) GO TO 362	COR-497
LP(J2)=I2	COR-498
JONT=JONT+1	COR-499
J2=J2+1	COR-500
362 IF(J1.LE.J2) GO TO 354	COR-501
363 DO 364 K=1, KKK	COR-502
VAJK(K)=0	COR-503
364 CONTINUE	COR-504
365 JJJ=0	COR-505
J1=0	COR-506
J2=0	COR-507
K1=1	COR-508
366 CONTINUE	COR-509
C	COR-510
C OPERACION QUE SE REALIZA SI DOS VERTICES DEL NUEVO LADO QUE DEBIERAN	COR-511
C SER ADYACENTES NO LO SON.	COR-512
C	COR-513
367 IS=LONT+1	COR-514
368 NT=0	COR-515
NT=0	COR-516
DO 370 JJ=IS+1, KONT-1	COR-517
LT=LP(JJ)	COR-518
IF(NT.ED.0.AND.LP(JJ-1).NE.LP(JJ)) NT=LP(JJ-1)	COR-519
IF(NT.ED.0.AND.LP(JJ+1).NE.LP(JJ)) NT=LP(JJ+1)	COR-520
IF(NT.NE.0.AND.NT.NE.0) GO TO 375	COR-521
370 CONTINUE	COR-522
GO TO 376	COR-523
375 DO 380 I=1, NA	COR-524
IF(IB(I, NT).EQ.0.OR.IB(I, NT).EQ.9) 380, 385	COR-525
380 CONTINUE	COR-526
IS=JJ	COR-527
GO TO 368	COR-528
385 KT=0	COR-529
DO 390 JJ=LONT+1, KONT	COR-530
IF(LP(JJ).EQ.LT) KT=KT+1	COR-531
IF(KT.ED.0) GO TO 390	COR-532
IF(LP(JJ).NE.LT) GO TO 386	COR-533
LP(JJ)=0	COR-534
K(JJ)=0.	COR-535
Y(JJ)=0.	COR-536
GO TO 390	COR-537
386 LP(JJ-KT)=LP(JJ)	COR-538
K(JJ-KT)=K(JJ)	COR-539
Y(JJ-KT)=Y(JJ)	COR-540
390 CONTINUE	COR-541

DO 393 JJ=KDNT-KT+1,KDNT	COR-542
K(JJ)=0.	COR-543
Y(JJ)=0.	COR-544
393 LP(JJ)=0	COR-545
KDNT=KDNT-KT	COR-546
GO TO 367	COR-547
C	COR-548
C ** LLAMADA A LA SUBROUTINA-TEST PARA TERMINAR LA TRANSFORMACION **	COR-549
C	COR-550
396 CALL SAL(NV,VS,LONT,KDNT,LP,IND,IFD,INC,LCH,KRR)	COR-551
IF(KRR.EQ.0) GO TO 310	COR-552
NT=0	COR-553
KT=KDNT-LONT	COR-554
DO 405 I=LONT+1,KDNT	COR-555
DO 400 J=IONT,LONT	COR-556
IF(K(J).NE.K(I)) GO TO 400	COR-557
IF(LP(I).EQ.LP(J)) NT=NT+1	COR-558
GO TO 405	COR-559
400 CONTINUE	COR-560
405 CONTINUE	COR-561
IF(KT.NE.NT) GO TO 300	COR-562
C	COR-563
C OPERACION QUE SE REALIZA SI SE OBTIENE UN NUEVO LADO IDENTICO A AQUEL	COR-564
C DE QUIEN PROCEDE.	COR-565
C	COR-566
I=LONT+1	COR-567
GO TO 410	COR-568
407 I=KDNT	COR-569
GO TO 410	COR-570
408 DO 600 I=LONT+2,KDNT-1	COR-571
410 NT=0	COR-572
DO 415 J=LONT+1,KDNT	COR-573
IF(LP(I).EQ.LP(J)) NT=NT+1	COR-574
415 CONTINUE	COR-575
LPP=LP(I)	COR-576
DO 420 IN=IND,IFD,INC	COR-577
LCC=LCH(IN)	COR-578
IF(LCC.EQ.LPP.AND.I.EQ.LONT+1) GO TO 407	COR-579
IF(LCC.EQ.LPP.AND.I.EQ.KDNT) GO TO 403	COR-580
IF(LCC.EQ.LPP) GO TO 600	COR-581
420 CONTINUE	COR-582
NT=0	COR-583
DO 450 K=5,NV	COR-584
DO 430 J=1,KDNT	COR-585
IF(K.EQ.LP(J)) GO TO 450	COR-586
430 CONTINUE	COR-587
IF(IA(K,LPP).NE.0) NT=NT+1	COR-588
450 CONTINUE	COR-589
IF(NT.LE.NT) GO TO 326	COR-590
J1=1	COR-591
JF=KDNT	COR-592
JT=0	COR-593
460 DO 500 J=J1,JF	COR-594
IF(K(I+JT).NE.X(J)) GO TO 500	COR-595
JT=JT+1	COR-596
DO 470 K=JF+1,J+1,-1	COR-597
LP(K)=LP(K-1)	COR-598
X(K)=X(K-1)	COR-599
Y(K)=Y(K-1)	COR-600
470 CONTINUE	COR-601
DO 480 K=J+1,JF+1	COR-602

IF(Y(K).NE.Y(J)) GO TO 510	COR-603
K(K)=K(K)+2.	COR-604
480 CONTINUE	COR-605
GO TO 520	COR-606
500 CONTINUE	COR-607
GO TO 520	COR-608
510 JI=K	COR-609
JF=JF+1	COR-610
GO TO 460	COR-611
520 LONT=LONT+JT-1	COR-612
KONT=KONT+JT	COR-613
IONT=IONT+JT-2	COR-614
GO TO 410	COR-615
600 CONTINUE	COR-616
800 CONTINUE	COR-617
C	COR-618
C CORRECCION DE LAS COORDENADAS DE TODOS LOS VERTICES INTERIORES PARA	COR-619
C QUE TODAS LAS COORDENADAS X E Y SEAN MAYORES O IGUALES A CERO.	COR-620
C	COR-621
810 RINCX=0.	COR-622
RINCY=0.	COR-623
DO 850 I=1,KONT	COR-624
IF(X(I).LT.RINCX) RINCX=X(I)	COR-625
IF(Y(I).LT.RINCY) RINCY=Y(I)	COR-626
850 CONTINUE	COR-627
DO 900 I=1,KONT	COR-628
X(I)=X(I)-RINCX	COR-629
Y(I)=Y(I)-RINCY	COR-630
900 CONTINUE	COR-631
RETURN	COR-632
END	COR-633
C	SAL-634
C	SAL-635
SUBROUTINE SAL(NV,V5,LONT,KONT,LP,INO,IFO,INC,LCN,KRR)	SAL-636
*****	SAL-637
C	SAL-638
C SUBROUTINA DE TEST PARA TERMINAR LA TRANSFORMACION	SAL-639
C	SAL-640
INTEGER V5	SAL-641
DIMENSION LCN(NV),LP(V5)	SAL-642
KRR=0	SAL-643
DO 100 I=LONT+1,KONT	SAL-644
DO 50 J=INO,IFO,INC	SAL-645
IF(LP(I).EQ.LCN(J)) GO TO 100	SAL-646
50 CONTINUE	SAL-647
KRR=1	SAL-648
GO TO 110	SAL-649
100 CONTINUE	SAL-650
110 RETURN	SAL-651
END	SAL-652
C	DIB-653
C	DIB-654
SUBROUTINE DIB(ISAL,KONT,X,Y,LP,XMAX,YMAX)	DIB-655
*****	DIB-656
C	DIB-657
C DIBUJA EL NUEVO SUBGRAFO DE VERTICES INTERIORES Y ARISTAS INTERIORES	DIB-658
C DE FORMA QUE TODAS LAS ARISTAS SEAN ORTOGONALES ENTRE SI.	DIB-659
C	DIB-660
ENA X(KONT),Y(KONT)	DIB-661
DIMENSION LP(KONT)	DIB-662

CALL PTLUC(10)	DIB-663
C	DIB-664
C DETERMINACION DE LA ABCISA Y ORDENADA MAXIMA Y NUMERACION DE LOS VER-	DIB-665
C TICES INTERIORES.	DIB-666
C	DIB-667
XMAX=0.	DIB-668
YMAX=0.	DIB-669
DO 50 I=1,KONT	DIB-670
IF(X(I).GT.XMAX) XMAX=X(I)	DIB-671
IF(Y(I).GT.YMAX) YMAX=Y(I)	DIB-672
50 CONTINUE	DIB-673
WRITE(19AL,3001) XMAX,YMAX	DIB-674
PAUSE 0001	DIB-675
CALL PLOT(0.,0.,-3)	DIB-676
DO 55 I=1,KONT	DIB-677
RI=LP(I)	DIB-678
CALL HUSB(K(I)-0.3,Y(I)-0.2,0.15,RI,0.,-1)	DIB-679
55 CONTINUE	DIB-680
XX=0.	DIB-681
YY=0.	DIB-682
K1=0	DIB-683
C	DIB-684
C TRAZADO DE LAS ARISTAS: PRIMERO EN UNA DIRECCION Y LUEGO EN LA ORTOGO-	DIB-685
C NAL.	DIB-686
C	DIB-687
60 DO 100 I=1,KONT	DIB-688
IF(XX.NE.X(I).OR.YY.NE.Y(I)) 100,110	DIB-689
100 CONTINUE	DIB-690
GO TO 115	DIB-691
110 IF(K1.EQ.0) CALL PLOT(XX,YY,3)	DIB-692
IF(K1.NE.0) CALL PLOT(4X,YY,2)	DIB-693
K1=K1+1	DIB-694
115 IF(XX.EQ.XMAX) GO TO 120	DIB-695
XX=XX+2.	DIB-696
GO TO 60	DIB-697
120 IF(YY.EQ.YMAX) GO TO 125	DIB-698
YY=YY+2.	DIB-699
XX=0.	DIB-700
K1=0	DIB-701
GO TO 60	DIB-702
125 XX=0.	DIB-703
YY=0.	DIB-704
K1=0	DIB-705
130 DO 150 I=1,KONT	DIB-706
IF(XX.NE.X(I).OR.YY.NE.Y(I)) 150,160	DIB-707
150 CONTINUE	DIB-708
GO TO 165	DIB-709
160 IF(K1.EQ.0) CALL PLOT(XX,YY,3)	DIB-710
IF(K1.NE.0) CALL PLOT(XX,YY,2)	DIB-711
K1=K1+1	DIB-712
165 IF(YY.EQ.YMAX) GO TO 170	DIB-713
YY=YY+2.	DIB-714
GO TO 130	DIB-715
170 IF(XX.EQ.XMAX) GO TO 180	DIB-716
XX=XX+2.	DIB-717
YY=0.	DIB-718
K1=0	DIB-719
GO TO 130	DIB-720
180 CALL PLOT(0.,0.,3)	DIB-721
RETURN	DIB-722
C	DIB-723

C	FORMATOS	DIB-724
C		DIB-725
	3001 FORMAT(//IX*MENSAJE :ISR: PON EL ORIGEN DE LA PLUNA DONDE DESEES*	DIB-726
	&//16X*LA DIMENSION DEL CONTORNO ES*F7.2* X*F7.2)	DIB-727
	END	DIB-728
C		DUA-729
C		DUA-730
	SUBROUTINE DUA(ISAL,ISAUX,JDNT,KDNT,LCN,LP,X,Y,XMAX,YMAX)	DUA-731
	*****	DUA-732
C		DUA-733
C	OBTIENE EL GRAFO DUAL DEL INICIAL QUE SERA LA REPRESENTACION EN PLANTA	DUA-734
C	DESCADA	DUA-735
C		DUA-736
	ENR X(JDNT),Y(JDNT)	DUA-737
	DIMENSION LCN(KDNT),LP(JDNT)	DUA-738
	CALL PLTLU(10)	DUA-739
	XMAX=0.	DUA-740
	YMAX=0.	DUA-741
	DO 5 I=1,JDNT	DUA-742
	IF(X(I).GT.XMAX) XMAX=X(I)	DUA-743
	IF(Y(I).GT.YMAX) YMAX=Y(I)	DUA-744
	5 CONTINUE	DUA-745
	XX=XMAX+2.	DUA-746
	YY=YMAX+2.	DUA-747
	WRITE(ISAL,3001) XX,YY	DUA-748
	PAUSE 0002	DUA-749
	CALL PLOT(1.,1.,-3)	DUA-750
	IF(ISSU(5).LT.0) 10,20	DUA-751
	10 WRITE(ISAL,5001)	DUA-752
	WRITE(ISAL,5002)	DUA-753
C		DUA-754
C	OBTENCION DE LOS CUATRO PARAMETROS CORRESPONDIENTES A CADA VERTICE.	DUA-755
C		DUA-756
	20 DO 150 I=1,JDNT	DUA-757
	I1=LP(I)	DUA-758
	Z1=0.	DUA-759
	Z2=0.	DUA-760
	Z3=0.	DUA-761
	Z4=0.	DUA-762
	DO 55 JJ=1,KDNT	DUA-763
	J=LCN(JJ)	DUA-764
	IF(I1.NE.J) GO TO 55	DUA-765
	IF(X(I).EQ.0.) Z2=-1.	DUA-766
	IF(X(I).EQ.XMAX) Z4=XMAX+1.	DUA-767
	IF(Y(I).EQ.YMAX) Z3=YMAX+1.	DUA-768
	IF(Y(I).EQ.0.) Z1=-1.	DUA-769
	GO TO 60	DUA-770
	55 CONTINUE	DUA-771
	60 D1=100.	DUA-772
	D2=100.	DUA-773
	D3=100.	DUA-774
	D4=100.	DUA-775
	J1=0	DUA-776
	J2=0	DUA-777
	J3=0	DUA-778
	J4=0	DUA-779
	DO 100 J=1,JDNT	DUA-780
	JJ=LP(J)	DUA-781
	IF(I.EQ.J) GO TO 100	DUA-782
	IF(X(J).NE.X(I)) GO TO 80	DUA-783

IF(Y(J).GT.Y(I)) GO TO 70	DUA-784
IF(Y(I)-Y(J).GT.D1) GO TO 100	DUA-785
J1=J	DUA-786
D1=Y(I)-Y(J)	DUA-787
GO TO 100	DUA-788
70 IF(Y(J)-Y(I).GT.D3) GO TO 100	DUA-789
J3=J	DUA-790
D3=Y(J)-Y(I)	DUA-791
GO TO 100	DUA-792
80 IF(Y(J).NE.Y(I)) GO TO 100	DUA-793
IF(X(J).GT.X(I)) GO TO 90	DUA-794
IF(X(I)-X(J).GT.D2) GO TO 100	DUA-795
J2=J	DUA-796
D2=X(I)-X(J)	DUA-797
GO TO 100	DUA-798
90 IF(X(J)-X(I).GT.D4) GO TO 100	DUA-799
J4=J	DUA-800
D4=X(J)-X(I)	DUA-801
100 CONTINUE	DUA-802
IF(J1.NE.0) Z1=(Y(I)+Y(J1))/2.	DUA-803
IF(J3.NE.0) Z3=(Y(I)+Y(J3))/2.	DUA-804
IF(J2.NE.0) Z2=(X(I)+X(J2))/2.	DUA-805
IF(J4.NE.0) Z4=(X(I)+X(J4))/2.	DUA-806
IF(Z1.EQ.0) Z1=Y(I)-1.	DUA-807
IF(Z3.EQ.0) Z3=Y(I)+1.	DUA-808
IF(Z2.EQ.0) Z2=X(I)-1.	DUA-809
IF(Z4.EQ.0) Z4=X(I)+1.	DUA-810
IF(ISSU(5).LT.0) WRITE(19AL,5003) I1,Z1,Z2,Z3,Z4	DUA-811
C	DUA-812
C NUMERACION Y TRAZADO DEL GRAFO DUAL	DUA-813
C	DUA-814
RI=I1	DUA-815
X1=(Z2+Z4)/2.	DUA-816
Y1=(Z1+Z3)/2.	DUA-817
CALL HUNB(X1,Y1,0,15,RI,0.,-1)	DUA-818
CALL PLOT(Z4,Z1,3)	DUA-819
IF(LP(J1).NE.I1) CALL PLOT(Z2,Z1,2)	DUA-820
IF(LP(J1).EQ.I1) CALL PLOT(Z2,Z1,3)	DUA-821
IF(LP(J2).NE.I1) CALL PLOT(Z2,Z3,2)	DUA-822
IF(LP(J2).EQ.I1) CALL PLOT(Z2,Z3,3)	DUA-823
IF(LP(J3).NE.I1) CALL PLOT(Z4,Z3,2)	DUA-824
IF(LP(J3).EQ.I1) CALL PLOT(Z4,Z3,3)	DUA-825
IF(LP(J4).NE.I1) CALL PLOT(Z4,Z1,2)	DUA-826
IF(LP(J4).EQ.I1) CALL PLOT(Z4,Z1,3)	DUA-827
150 CONTINUE	DUA-828
RETURN	DUA-829
C	DUA-830
C FORMATS	DUA-831
C	DUA-832
3001 FORMAT(/1X'MENSAJE :ISR: PDN EL ORIGEN DE LA PLUMA DONDE DESEES'	DUA-833
8//16X'LA DIMENSION DEL CONTORNO ES'F7.2' X'F7.2)	DUA-834
5001 FORMAT(/1X'PARAMETROS CORRESPONDIENTES A CADA VERTICE DEL GRAFO D	DUA-835
DUAL:')	DUA-836
5002 FORMAT(/2X,'Y Z1 Z2 Z3 Z4'/2X'-- -- --')	DUA-837
5003 FORMAT(13,F5.2)	DUA-838
END	DUA-839
9	DUA-840

A.II.1.4. Ejemplos

MENSAJE (1ED): TITULO EL PROBLEMA

..... EJEMPLO 1 (ADIMENSIONAL)

..... MENSAJE (1ED): ASIGNE UN NOMBRE A CADA UNO DE LOS ESPACIOS

AL FINAL DE LA LISTA INTRODUCIR 0

- ESP NOMBRE
- 5 VESTIBULO
- 6 COCINA
- 7 ESTAR-COMEDOR
- 8 LAVADERO
- 9 DESPESA
- 10 PASILLO
- 11 CUARTO DE BANO
- 12 DORMITORIO DOBLE
- 13 DORMITORIO SENCILLO
- 14 DORMITORIO DOBLE
- 15 ARMARIO
- 16 ARMARIO

MENSAJE (1ED): MATRIZ DE INCIDENCIA

LA NUMERACION DE LOS VERTICES Y ARISTAS SE HARA A PARTIR DEL 5.
 LOS 4 PRIMEROS VERTICES Y ARISTAS CORRESPONDEN A LOS PUNTOS
 CARDINALES Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE.
 AL FINAL DE LA LISTA INTRODUCIR 0

4	V	V
5	S	I
6	S	4
7	S	6
8	S	7
9	6	8
10	6	9
11	7	2
12	7	10
13	8	4
14	10	11
15	10	12
16	10	13
17	10	14
18	11	12
19	12	3
20	12	15
21	13	4
22	13	14
23	14	4
24	14	16
25	15	16
26	15	3

27 15 3
28 6 7

Mensaje 1181: EL GRAFO ES CONEXO

Mensaje 1182: EL GRAFO NO ES BICONEXO

LAS COMPONENTES BICONEXAS SON:

COMPONENTE: 1

ARISTAS: 10

COMPONENTE: 2

ARISTAS: 7 4 6 3 12 15 21 22 16 17 14 18 1 2 11 19 20
25 26 27 24 23 13 9 28 8 5

LOS VERTICES DE SEPARACION DE ESTAS COMPONENTES SON:

COMPONENTES: 1- 2 VERT. DE SEPAR: 6

----- COMPONENTE BICONEXA: 1

ARISTAS: 10

Mensaje 1183: NO SE DIBUJA POR SER ELEMENTAL

----- COMPONENTE BICONEXA: 2

ARISTAS: 7 4 6 3 12 15 21 22 16 17 14 18 1 2 11 19 20
25 26 27 24 23 13 9 28 8 5

Mensaje 1184: LA COMPONENTE BICONEXA ES PLANAR

Mensaje 1185: LA COMPONENTE BICONEXA NO ES TRICONEXA

LAS COMPONENTES TRICONEXAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADAS
POR SUS VERTICES EXTREMOS) SON:

COMPONENTE: 1
 ARISTAS: 1- 2 2- 3 3- 4 4- 1 4- 5 5- 6 6- 4 6- 7 7- 5 7- 2
7- 10 10- 12 12- 15 15- 3 15- 16 16- 14 14- 10 10- 4 14- 12 12- 10
13- 4 16- 3 12- 2 5- 1

Mensaje 1186: ARISTA VIRTUAL EN LA COMPONENTE 1: 6- 4

Mensaje 1187: LAS ARISTAS VIRTUALES CITADAS HAN PASADO A SER REALES.

POSTERIORMENTE, SI SE DESEA, PUEDEN SER CAMBIADAS POR OTRAS.

Mensaje 11B1: ADEMAS DE LAS ARISTAS PERTENECIENTES A ALGUNA COMPONENTE HAY OTRAS QUE NO PERTENECEN A NINGUNA DE ELLAS.

LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS Y LAS QUE NO ENLAZAN A DOS COMPONENTES CUALEQUIERA SON:

ARISTA: 10-11 (NO LIGA A DOS COMPONENTES)
ARISTA: 11-12 (NO LIGA A DOS COMPONENTES)
ARISTA: 6-8 (NO LIGA A DOS COMPONENTES)
ARISTA: 8-4 (NO LIGA A DOS COMPONENTES)

COMPONENTE 1 (TRICOMERA)

ARISTAS: 1-2 2-3 3-4 4-1 4-5 5-6 6-4 6-7 7-9 7-2
17-10 10-12 12-15 15-2 15-16 16-14 14-10 14-4 14-13 13-10
13-4 16-3 12-2 5-1

Mensaje 11B1: SE DIBUJA EN EL PLOTTER.

Mensaje 11B1: SE HAN DE INTRODUCIR NUEVAS ARISTAS PARA CONVERTIR EN TRICOMERA LA COMPONENTE BICOMERA.

LA ADICION SE HARA A LA VISTA DEL SUBRASO Trazado.
 SE PUEDE REALIZAR ESCALONADAMENTE, INTRODUCIENDO SOLO PARTE DE ELLAS, ANDIENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.

Mensaje 11B1: EL NUMERO DE ARISTAS ACTUALES ES: 29

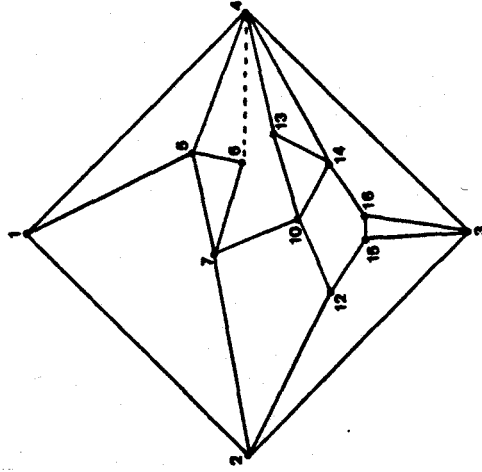
Mensaje 11B1: INTRODUCIR LAS NUEVAS ARISTAS
AL FINAL DE LA LISTA INTRODUCIR: 0

A V V
- - -
30 2 11

Mensaje 11B1: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS
LOS DATOS DE DABAN EN EL SIGUIENTE ORDEN:

- NUMERO DE LA ARISTA
 - VERTICE PRIMITIVO
 - VERTICE PRIMITIVO
 - VERTICE ACTUAL
 - VERTICE ACTUAL
- AL FINAL DE LA LISTA, INTRODUCIR: 0

A VP VP VA VA
- - - - -
29 6 4 0 13



Mensaje 11B1: La componente bicóncava es plana

Mensaje 11B2: La componente bicóncava es además triconcava

Mensaje 11B3: Se dibuja en el plotter

Mensaje 11B4: Se han de añadir nuevas aristas para que el grafo sea triconcavo

Estas aristas enlazarán a las distintas componentes bicóncavas entre sí.

Mensaje 11B5: El número de aristas actuales es: 30

Mensaje 11E1: Introduce las nuevas aristas

Al final de la lista introduce: 0

```
A V V
- - -
31 9 4
32 9 8
```

Mensaje 11B6: Si se desea se puede sustituir aristas

Los datos se darán en el siguiente orden:

- Número de la arista
- Vértice primitivo
- Vértice primitivo
- Vértice actual
- Vértice actual

Al final de la lista introduce: 0

```
A VP VP VA VA
- - - - -
```

Mensaje 11B7: El grafo es bicóncavo

Mensaje 11B8: El grafo es plano

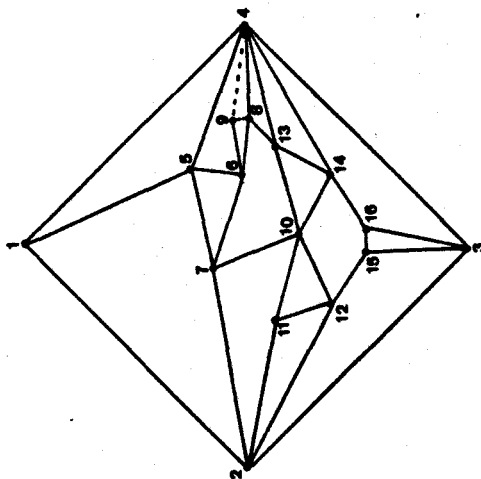
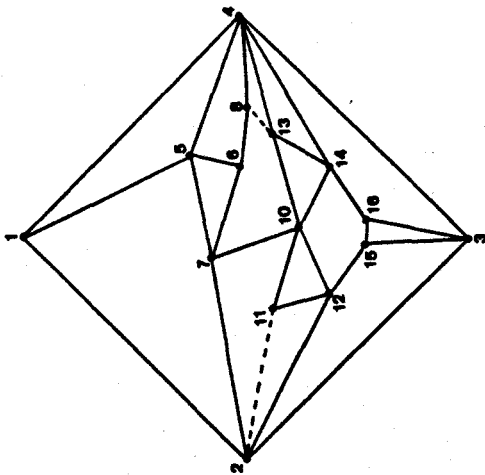
Mensaje 11B9: El grafo es triconcavo

Mensaje 11B10: Se dibuja en el plotter

Mensaje 11B11: A la vista del grafo triconcavo dibujado, introduce las aristas necesarias para triangular el grafo.

Mensaje 11B12: El número de aristas actuales es: 32

Mensaje 11E2: Introduce las nuevas aristas



AL FINAL DE LA LISTA INTRODUCIR 0

- 4 V -
- 33 5 9
- 34 6 10
- 35 6 12
- 36 7 1
- 37 7 11
- 38 10 13
- 39 10 16
- 40 12 2
- 41 14 3

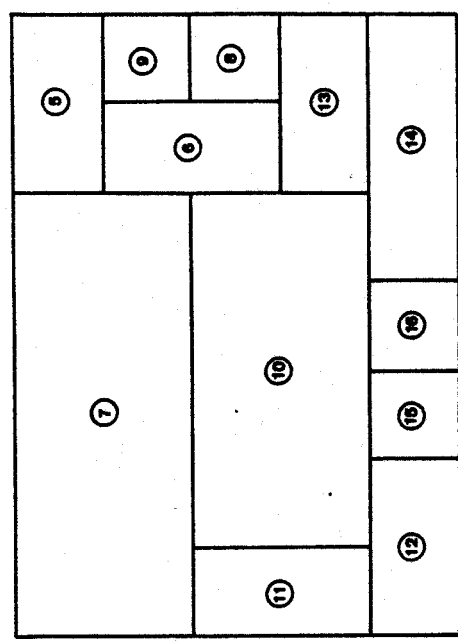
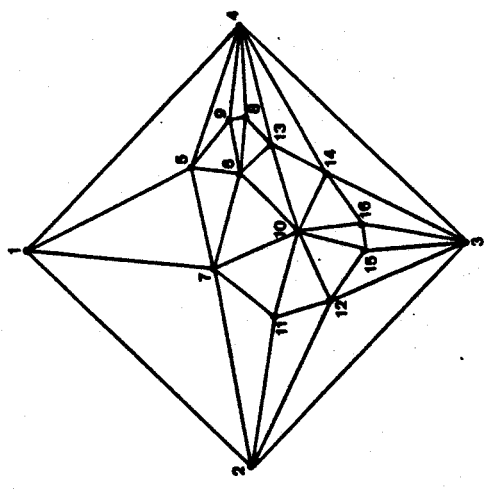
MENSAJE 11DR11 DRAP0 TRIANGULADO

MENSAJE 11DR11 DE DIBUJA EN EL PLOTTER

MENSAJE 11DR11 POR EL ORIGEN DE LA PLUMA DONDE DEBEES

LA DIMENSION DEL CONTORNO ES 14.00 X 10.00

MENSAJE 11DR11 SI DESEAS TERMINAR INTRODUCIR 0
EN CASO CONTRARIO INTRODUCIR 1



MESENSE-11ED): TITULO EL PROBLEMA

EJEMPLO 2 (ADIMENSIONAL)

.....

Mensaje 11ED): ASIGNE UN NOMBRE A CADA UNO DE LOS ESPACIOS
AL FINAL DE LA LISTA INTRODUCE 0

- ESP NOMBRE
- 5 DORNITORIO PRINCIPAL
- 6 ESTUDIO
- 7 CONEDOR
- 8 ARRARIO
- 9 CUARTO DE BANO
- 10 COCINA
- 11 PASILLO
- 12 DORNITORIO SENCILLO
- 13 ARRARIO
- 14 VESTIBULO
- 15 ESTAR
- 16 DORNITORIO DOBLE
- 17 ENTRADA

Mensaje 11ED): MATRIZ DE INCIDENCIA

LA NUMERACION DE LOS VERTICES Y ARISTAS SE HARA A PARTIR DEL 5.
LOS 4 PRIMEROS VERTICES Y ARISTAS CORRESPONDEN A LOS PUNTOS
CARDINALES Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE.
AL FINAL DE LA LISTA INTRODUCE: 0

- A V V
- - -
- 5 2 5
- 6 2 12
- 7 2 16
- 8 2 17
- 9 2 15
- 10 4 7
- 11 1 7
- 12 5 6
- 13 5 7
- 14 5 11
- 15 7 10
- 16 5 8
- 17 11 12
- 18 9 11
- 19 9 10
- 20 11 16
- 21 11 14
- 22 10 14
- 23 13 16
- 24 14 15
- 25 10 15

26 14 17
27 1 6
28 9 14
29 8 12
30 2 9

Mensaje 1: El grafo es conexo

Mensaje 1: El grafo no es biconexo

Las componentes biconeexas son:

Componente: 1

Aristas: 22

Componente: 2

Aristas: 11 22 1 4 10 3 26 8 9 2 5 12 14 16 6 17 29
20 7 20 21 18 19 28 24 25 15 13 27

Los vertices de separacion de esas componentes son:

Componentes: 1-2 Vert. de separ: 16

Componente biconexa: 1

Aristas: 22

Mensaje 1: No se dibuja por ser elemental

Componente biconexa: 2

Aristas: 11 22 1 4 10 3 26 8 9 2 5 12 14 16 6 17 29
20 7 20 21 18 19 28 24 25 15 13 27

Mensaje 1: La componente biconexa es planar

Mensaje 1: La componente biconexa no es triconexa

Las componentes triconexas y triangulos (donde las aristas vienen expresadas por sus vertices extremos) son:

Componente: 1
Aristas: 1-2 2-3 3-4 4-1 4-7 7-1 7-6 6-5 5-2 5-8
9-2 8-12 12-2 12-11 11-9 9-10 10-7 10-16 16-9
16-11 16-3 16-15 15-19 19-3 11-2 6-1

Mensaje 1: Arista virtual en la componente 1: 14-3

mensaje 11R1: ARISTA VIRTUAL EN LA COMPONENTE 1: 11- 2

mensaje 11R1: LAS ARISTAS VIRTUALES CITADAS HAN PASADO A SER REALES.
POSTERIORMENTE, SI SE DESEA, PUEDEN SER CAMBIADAS POR OTRAS.

mensaje 11R1: ADEMAS DE LAS ARISTAS PERTENECIENTES A ALGUNA COMPONENTE
HAY OTRAS QUE NO PERTENECEN A NINGUNA DE ELLAS.

LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS Y LAS QUE NO ENLAZAN A
DOS COMPONENTES CUALESQUIERA SON:

ARISTA: 11-16 (NO LIGA A DOS COMPONENTES)
ARISTA: 16- 2 (NO LIGA A DOS COMPONENTES)
ARISTA: 14-17 (NO LIGA A DOS COMPONENTES)
ARISTA: 17- 3 (NO LIGA A DOS COMPONENTES)

COMPONENTE: 1 (TRICONERA)

ARISTAS: 1- 2 2- 3 3- 4 4- 1 4- 7 7- 1 7- 6 6- 5 5- 2 5- 8
8- 2 8-12 12- 2 12-11 11- 5 11- 9 9-10 10- 7 10-16 14- 9
14-11 14- 3 14-15 15-10 15- 3 11- 2 6- 1

mensaje 11R1: SE DIBUJA EN EL PLOTTER

mensaje 11R1: SE HAN DE INTRODUCIR NUEVAS ARISTAS PARA CONVERTIR EN TRICONERA
LA COMPONENTE DICONERA.

LA ADICION DE HARA A LA VISTA DEL SUBGRUPO TRAZADO.
SE PUEDE REALIZAR ESCALONADAMENTE, INTRODUCIENDO SOLO PARTE DE
ELLAS, ANADIDIENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.

mensaje 11R1: EL NUMERO DE ARISTAS ACTUALES ES: 32

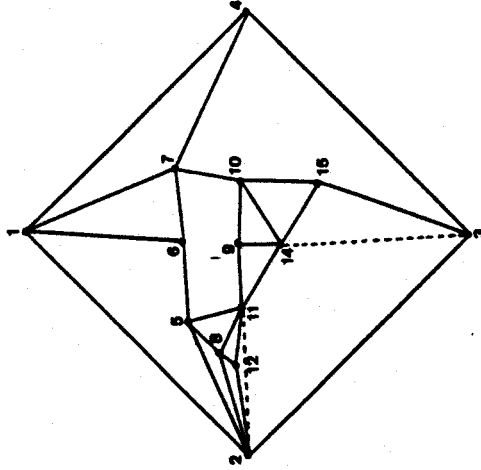
mensaje 11R2: INTRODUCIR LAS NUEVAS ARISTAS

AL FINAL DE LA LISTA INTRODUCIR: 0

A V V
- - -

mensaje 11R1: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

LOS DATOS DE PARAM EN EL SIGUIENTE ORDEN:
- NUMERO DE LA ARISTA
- VERTICE PRIMITIVO
- VERTICE PRIMITIVO
- VERTICE ACTUAL
- VERTICE ACTUAL
AL FINAL DE LA LISTA INTRODUCIR: 0



A VP VP VA VA
 - - - -
 31 14 2 16 17
 32 11 2 14 16

Mensaje 1: LA COMPONENTE DICOMERA ES PLANAR

Mensaje 2: LA COMPONENTE DICOMERA ES ADENAS TRICOMERA

Mensaje 3: SE DIBUJA EN EL PLOTTER

Mensaje 4: SE HAN DE AÑADIR NUEVAS ARISTAS PARA QUE EL GRAFO SEA TRICOMERO
 ESTAS ARISTAS ENLAZARAN A LAS DISTINTAS COMPONENTES DICOMERAS
 ENTRE SI.

Mensaje 5: EL NUMERO DE ARISTAS ACTUALES ES: 32

Mensaje 6: INTRODUCE LAS NUEVAS ARISTAS
 AL FINAL DE LA LISTA INTRODUCE: 0

A V V
 - - -
 33 2 12
 34 12 13

Mensaje 7: SI SE DESEA SE PUEDE SUSTITUIR ARISTAS

LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:
 - NUMERO DE LA ARISTA
 - VERTICE PRIMITIVO
 - VERTICE ACTUAL
 - VERTICE ACTUAL
 AL FINAL DE LA LISTA INTRODUCE: 0

A VP VP VA VA
 - - - -

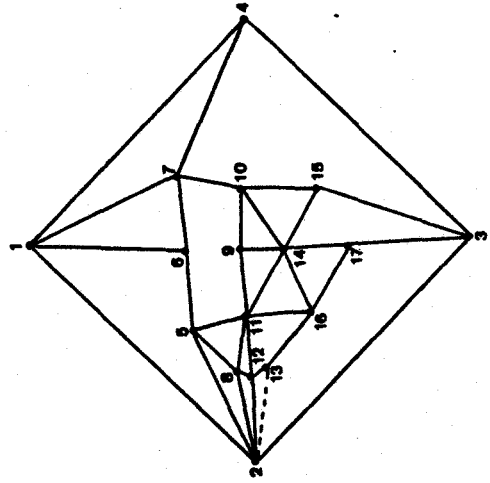
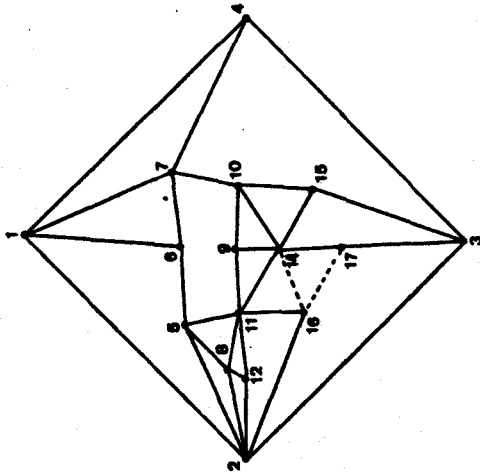
Mensaje 8: EL GRAFO ES DICOMERO

Mensaje 9: EL GRAFO ES PLANAR

Mensaje 10: EL GRAFO ES TRICOMERO

Mensaje 11: SE DIBUJA EN EL PLOTTER

Mensaje 12: A LA VISTA DEL GRAFO TRICOMERO DIBUJADO, INTRODUCIR LAS ARISTAS
 NECESARIAS PARA TRIANGULAR EL GRAFO.



MENSAJE 'IBR': EL NUMERO DE ARISTAS ACTUALES ES: 34

MENSAJE 'IED': INTRODUCIR LAS NUEVAS ARISTAS

AL FINAL DE LA LISTA INTRODUCIR: 0

A	V	V
35	1	5
36	2	16
37	4	15
38	6	11
39	6	9
40	6	10
41	8	11
42	11	12
43	15	17
44	4	10

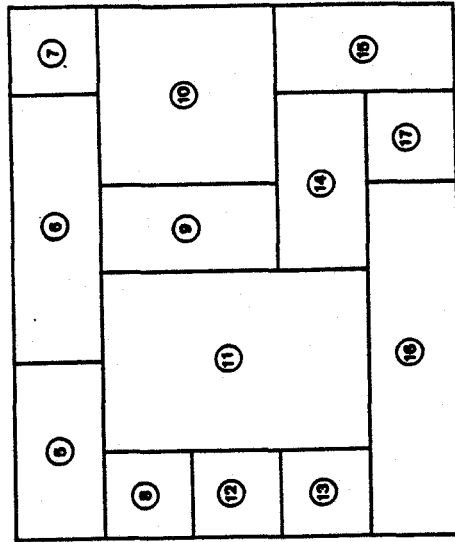
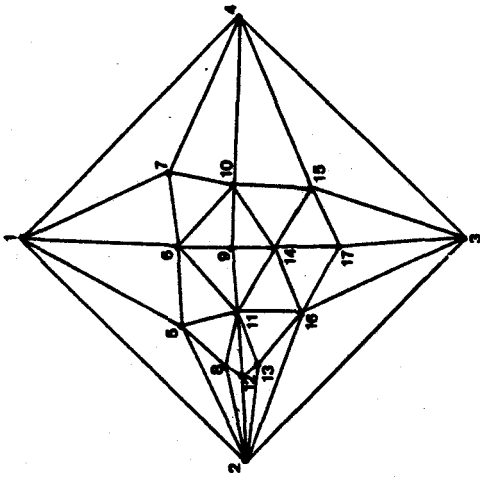
MENSAJE 'IBR': GRAFO TRIANGULADO

MENSAJE 'IBR': SE DIBUJA EN EL PLOTTER

MENSAJE 'IBR': PON EL ORIGEN DE LA PLUMA DONDE DESEES

LA DIMENSION DEL CONTORNO ES 12.00 X 10.00

MENSAJE 'IBR': SI DESEAS TERMINAR INTRODUCIR: 0
EN CASO CONTRARIO INTRODUCIR: 1



MENSAJE 11ED11 TITULO EL PROBLEMA

EJEMPLO 3 (ADIMENSIONAL)

AL FINAL DE LA LISTA INTRODUCIR: 0

- ESP HOMBRE
- 5 VESTIBULO
- 6 COCINA
- 7 COMEDOR
- 8 SALA DE ESTAR
- 9 DORMITORIO DOBLE
- 10 CUARTO DE BANO
- 11 DISTRIBUIDOR
- 12 DORMITORIO DOBLE
- 13 DORMITORIO PRINCIPAL
- 14 GUARDARROPA
- 15 ASEO
- 16 ARMARIO
- 17 ARMARIO

AL FINAL DE LA LISTA INTRODUCIR: 0

LA NUMERACION DE LOS VERTICES Y ARISTAS SE HARA A PARTIR DEL 5.
LOS 4 PRIMEROS VERTICES Y ARISTAS CORRESPONDER A LOS PUNTOS
CARDINALES Y SUS RELACIONES. ESTAN INTRODUCIDOS AUTOMATICAMENTE.

AL FINAL DE LA LISTA INTRODUCIR: 0

26 10 12
27 9 10
28 1 16

MENSAJE 1181: EL GRAFO ES CONEXO

MENSAJE 1182: EL GRAFO NO ES BICONEXO

LAS COMPONENTES BICONEXAS SON:

COMPONENTE: 1

ARISTAS: 8

COMPONENTE: 2

ARISTAS: 7

COMPONENTE: 3

ARISTAS: 4 1 13 15 14 18 19 24 9 10 11 25 17 12 2 3 6
5 29 21 22 23 27 26 16 -28

LOS VERTICES DE SEPARACION DE ESTAS COMPONENTES SON:

COMPONENTES: 1- 2 VERT. DE SEPAR: 14

COMPONENTES: 2- 3 VERT. DE SEPAR: 5

COMPONENTE BICONEXA: 1

ARISTAS: 9

MENSAJE 1183: NO SE DIBUJA POR SER ELEMENTAL

COMPONENTE BICONEXA: 2

ARISTAS: 7

MENSAJE 1184: NO SE DIBUJA POR SER ELEMENTAL

COMPONENTE BICONEXA: 3

ARISTAS: 4 1 13 15 14 18 19 24 9 10 11 25 17 12 2 3 6
5 29 21 22 23 27 26 16 28

MENSAJE 1185: LA COMPONENTE BICONEXA ES PLANAR

MENSAJE 1186: LA COMPONENTE BICONEXA NO ES TRICONEXA

LAS COMPONENTES TRICOMERAS Y TRIANGULOS (DONDE LAS ARISTAS VIENEN EXPRESADAS POR SUS VERTICES ENTRENDS) SON:

COMPONENTE: 1
 ARISTAS: 1-2 2-3 3-4 4-1 4-6 6-2 6-8 8-2 8-11 11-9
 9-4 9-10 10-11 10-12 12-4 12-16 16-1 16-13 13-11
 13-2 13-1

MENSAJE '10R': ARISTA VIRTUAL EN LA COMPONENTE '1' 6-8

MENSAJE '10R': ARISTA VIRTUAL EN LA COMPONENTE '1' 16-13

MENSAJE '10R': LAS ARISTAS VIRTUALES CITADAS HAN PASADO A SER REALES. POSTERIORMENTE, SI SE DESEA, PUEDEN SER CANTADAS, POR OTRAS.

MENSAJE '10R': ADENAS DE LAS ARISTAS PERTENECIENTES A ALGUNA COMPONENTE HAY OTRAS QUE NO PERTENECEN A NINGUNA DE ELLAS.

LAS ARISTAS QUE RELACIONAN UNAS COMPONENTES CON OTRAS Y LAS QUE NO ENLAZAN A DOS COMPONENTES CUALQUIERA SON:

ARISTA: 16-17 (NO LIGA A DOS COMPONENTES)
 ARISTA: 17-12 (NO LIGA A DOS COMPONENTES)
 ARISTA: 6-7 (NO LIGA A DOS COMPONENTES)
 ARISTA: 7-9 (NO LIGA A DOS COMPONENTES)
 ARISTA: 6-5 (NO LIGA A DOS COMPONENTES)
 ARISTA: 5-3 (NO LIGA A DOS COMPONENTES)

 COMPONENTE: 1 (TRICOMERA)

ARISTAS: 1-2 2-3 3-4 4-1 4-6 6-2 6-8 8-2 8-11 11-9
 9-4 9-10 10-11 10-12 12-4 12-16 16-1 16-13 13-11
 13-2 13-1

MENSAJE '10R': SE DIBUJA EN EL PLOTTER

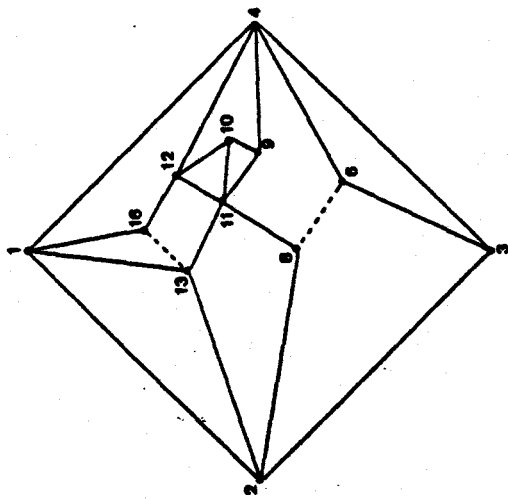
MENSAJE '10R': SE HAY DE INTRODUCIR NUEVAS ARISTAS PARA CONVERTIR EN TRICOMERA LA COMPONENTE DICOMERA.

LA ADICION SE HARA A LA VISTA DEL DIBUJADO TRAZADO. SE PUEDE REALIZAR ESCALONADAMENTE, INTRODUCIENDO SOLO PARTE DE ELLAS, ANDIENDO EL RESTO A PARTIR DE UN NUEVO TRAZADO.

MENSAJE '10R': EL NUMERO DE ARISTAS ACTUALES ES: 30

MENSAJE '10R': INTRODUCIR LAS NUEVAS ARISTAS
 AL FINAL DE LA LISTA INTRODUCIR: 0

A V V



A VP VP VA VA
- - - - -

Mensaje 1181: EL GRAFO ES BICONEXO

Mensaje 1181: EL GRAFO ES PLANAR

Mensaje 1181: EL GRAFO ES TRICONEXO

Mensaje 1181: SE DIBUJA EN EL PLOTTER

Mensaje 1181: A LA VISTA DEL GRAFO TRICONEXO DIBUJADO, INTRODUCIR LAS ARISTAS
NECESARIAS PARA TRIANGULAR EL GRAFO.

Mensaje 1181: EL NUMERO DE ARISTAS ACTUALES ES: 38

Mensaje 1181: INTRODUCIR LAS NUEVAS ARISTAS

AL FINAL DE LA LISTA INTRODUCIR: 0

A	V
39	2
40	1
41	9
42	17
43	8
44	7

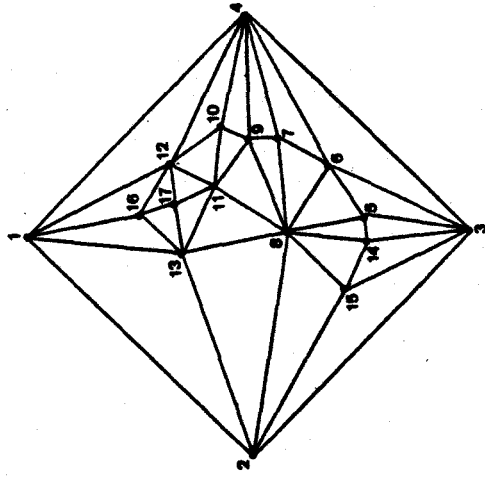
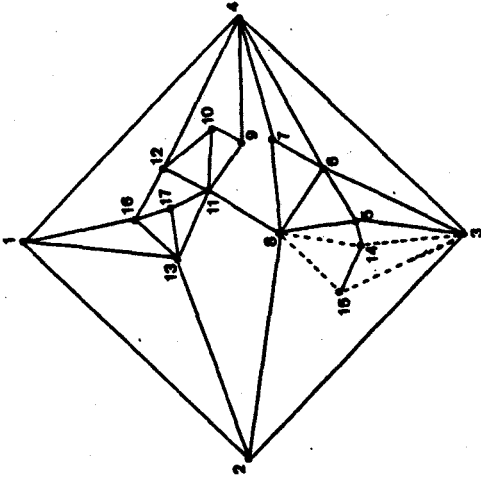
Mensaje 1181: GRAFO TRIANGULADO

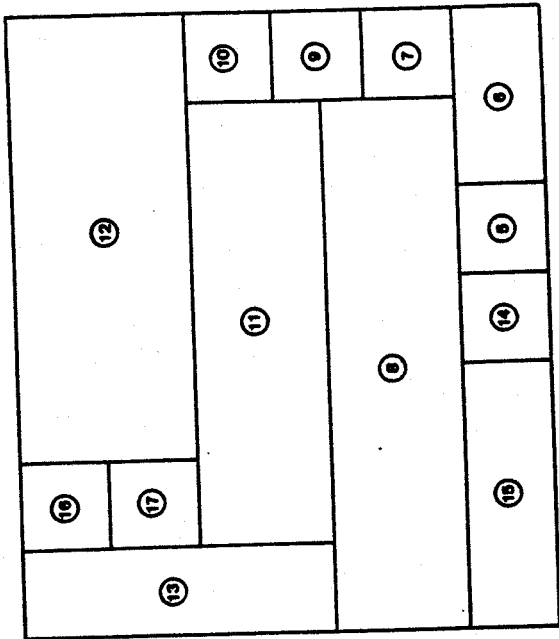
Mensaje 1181: SE DIBUJA EN EL PLOTTER

Mensaje 1181: PON EL ORIGEN DE LA PLUMA DONDE DEBEES

LA DIMENSION DEL CONTORNO ES 14.00 X 12.00

Mensaje 1181: SI DEBERAS TERMINAR INTRODUCIR: 0
EN CASO CONTRARIO INTRODUCIR: 1





A.II.2. DIMENSIONAMIENTO

A.II.2.1. Manual del usuario

Ha de tenerse en cuenta los criterios generales señalados en A.II.1.1.

La secuencia de tarjetas o registros para la entrada de datos del programa DIM es casi toda previa a la ejecución del programa:

1. Titulación del problema

FORMATO : (Un máximo de 80 caracteres alfanuméricos)

2. Número de vértices de los grafos dirigidos

FORMATO: NVV, NVH

NVV : n° de vértices del grafo vertical.

NVH : n° de vértices del grafo horizontal.

3.a Coordenadas de los centros geométricos de los distintos espacios en el esquema adimensional ($a=1,2, \dots n$; $n =$ n° de espacios)

4.a Condiciones de accesibilidad entre locales ($a=1,2, \dots r$; $r =$ n° de condiciones)

FORMATO : L, L₁, L₂, L₃, ... L_n

L : Local accesible a L₁, L₂, ... L_n

Si un local L_i es accesible a otro L_j, la comunicación es mútua. No será necesario, por tanto, señalar que L_j es accesible a L_i.

5. Tipo de problema

FORMATO : TIPO

TIPO = 1 LINEAL

TIPO = 2 NO LINEAL

6. Tipo de función objetivo

FORMATO : OBJ

OBJ = 0 y TIPO = 1 MINIMIZACION DE LA LONGITUD Y ANCHURA

OBJ = 1 y TIPO = 1 MINIMIZACION DE UNA EXPRESION LINEAL

OBJ = 0 y TIPO = 2 MINIMIZACION DE LA SUPERFICIE

OBJ = 1 y TIPO = 2 MINIMIZACION DE UNA EXPRESION NO LINEAL.

7. Costos de la función objetivo

FORMATO : $C_5, C_6, C_7, \dots, C_{n+5}$ (n = n° de espacios)

C_i : Costo correspondiente al espacio i.

8. Superficie del contorno. (Sólo cuando TIPO = 2 y OBJ = 1)

FORMATO : SC

SC : Superficie del contorno

9.a Matriz de incidencia vertical (a = 1, 2, .. n+1; n = n° de espacios)

Si TIPO = 1 (problema lineal) FORMATO: LA, LO, LF, (CI), (CS)

LA: n° de la arista que representa al espacio o contorno.
(Numerada a partir del 5. La arista que representa -
al contorno se numerará en último lugar).

LO: n° del vértice origen.

LF: n° del vértice extremo.

CI: valor de la cota inferior de la anchura.

CS: valor de la cota superior de la anchura.

Si se quiere dar un valor determinado a la anchura de -
una variable se da un valor a CI igual a CS.

Si TIPO = 2 (problema no lineal) FORMATO: LA, LO, LF, (CI),
(SI)

LA: n° de la arista que representa al espacio o contorno.
(Numerada a partir del 5)

LO: n° del vértice origen.

LF: n° del vértice extremo

CI: valor de la cota inferior de la anchura

SI: valor de la cota inferior de la superficie.

Los valores entre paréntesis en uno y otro caso indican
que son datos opcionales. Si no se reflejan como datos -
los valores correspondientes se toman iguales a cero.

10.a Matriz de incidencia horizontal ($a = 1, 2, \dots, n+1$; $n = n^\circ$
de espacios)

Si TIPO = 1 (Problema lineal) FORMATO: LA, LO, LF, (CI),
(CS).

Igual significado que en la matriz de incidencia verti-
cal pero referidos los valores de las cotas a la longi-
tud en vez de a la anchura.

Si TIPO = 2 (Problema no lineal) FORMATO: LA,LO,LF,(CI)

Igual significado que en la matriz de incidencia vertical siendo el valor de la cota inferior de la longitud el que se dará ahora, y no figurando como entrada de datos la superficie mínima.

La numeración de las aristas en uno y otro grafo dirigido ha de ser de tal forma que un determinado espacio venga representado por un mismo número.

Una vez resuelto y trazado el esquema dimensionado el ordenador envía un mensaje por si se desea terminar o se quiere - ajustar los resultados. En el primer caso se introduce un: "0", en el segundo un: "1". En este último caso hay que efectuar - una nueva entrada de datos:

1. Valoración relativa para los distintos espacios

FORMATO : $V_5, V_6, V_7, V_8 \dots V_{n+5}$ (n: n° de espacios)

V_i : Valoración relativa para el espacio i.

Hay una sola opción prevista para salida de resultados intermedios: la impresión de la matriz de restricciones, los términos independientes y los códigos de las restricciones. Si se desea se pulsará el SWITCH 1.

Los mensajes de errores que el ordenador puede enviar a lo largo de la ejecución del programa son los siguientes:

ERROR 1 : El número de vértices o aristas es mayor que el permitido.

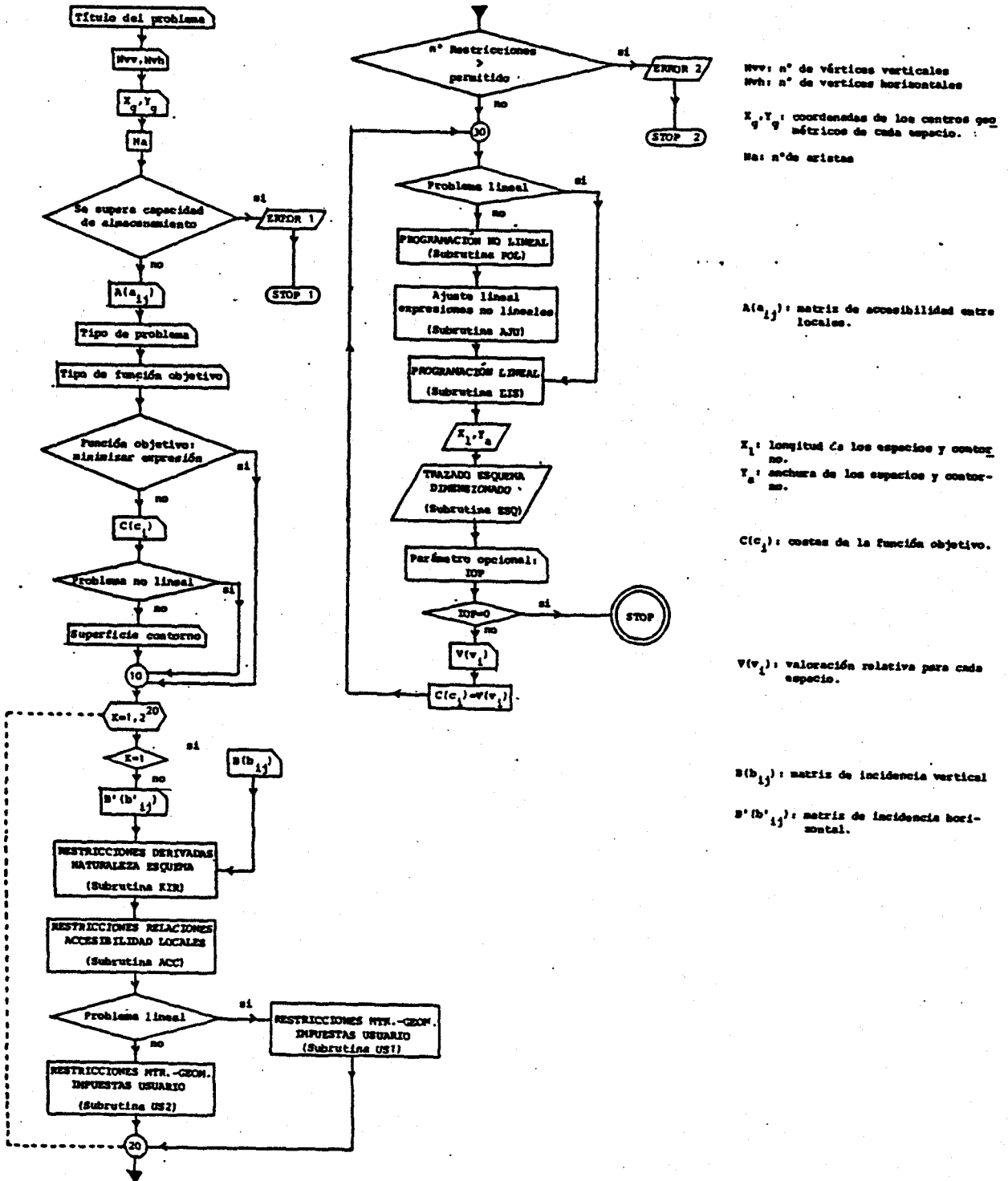
ERROR 2 : El número de restricciones supera la capacidad dispuesta.

ERROR 3 : El problema no está acotado.

ERROR 4 : El problema no tiene solución admisible.

El programa que se presenta está elaborado para un máximo de 60 aristas, 40 vértices y 240 expresiones para restricciones.

A.II.2.2. Organigrama general



A.II.2.3. Listado del Programa

FTN4	DIN-000
SENA(IJK,3)	DIN-001
C	DIN-002
C	DIN-003
C	DIN-004
C *****	DIN-005
C PROGRAM DIN	DIN-006
C *****	DIN-007
C	DIN-008
C-----	DIN-009
C PROGRAM PARA DIMENSIONAMIENTO DE ESQUEMAS DISTRIBUTIVOS EN PLANTA	DIN-010
C-----	DIN-011
C	DIN-012
C	DIN-013
C TABLAS UTILIZADAS	DIN-014
C	DIN-015
C BB(60,40) ... MATRIZ DE INCIDENCIA HORIZONTAL	DIN-016
C BA(60,40) ... MATRIZ DE INCIDENCIA VERTICAL	DIN-017
C IR(250,120) . MATRIZ DE RESTRICCIONES	DIN-018
C IA(60,60) ... MATRIZ DE ACCESIBILIDAD	DIN-019
C	DIN-020
C	DIN-021
C LISTADO DE ERRORES	DIN-022
C	DIN-023
C ERROR 1 NUMERO DE VERTICES O ARISTAS MAYOR QUE EL PERMITIDO	DIN-024
C ERROR 2 EL NUMERO DE RESTRICCIONES SUPERA LA CAPACIDAD DISPUESTA	DIN-024

C PUESTA A CERO	DIN-086
C	DIN-087
20 DO 30 I=1,3*NA	DIN-088
DO 30 J=1,3	DIN-089
B(I,J)=0	DIN-090
30 CONTINUE	DIN-091
DO 35 I=1,3*NA	DIN-092
CODE(I)=0	DIN-093
35 CONTINUE	DIN-094
DO 40 I=1,2*NA	DIN-095
ZETA(I)=0	DIN-096
40 CONTINUE	DIN-097
DO 50 I=1,3*NA	DIN-098
DO 50 J=1,2*NA	DIN-099
IR(I,J)=0	DIN-100
50 CONTINUE	DIN-101
DO 60 I=1,NA	DIN-102
DO 60 J=1,NV	DIN-103
RB(I,J)=0	DIN-104
BA(I,J)=0	DIN-105
60 CONTINUE	DIN-106
DO 70 I=1,NA	DIN-107
DO 70 J=1,NA	DIN-108
IA(I,J)=0	DIN-109
70 CONTINUE	DIN-110
DO 80 I=1,NA	DIN-111
JAU(I)=0	DIN-112
C(I)=0	DIN-113
80 CONTINUE	DIN-114
C	DIN-115
C LECTURA DE LA MATRIZ DE ACCESIBILIDAD	DIN-116
C	DIN-117
WRITE(ISAL,3005)	DIN-118
DO 150 K=1,100	DIN-119
READ(IENT,*) I,(JAU(J),J=1,NVI)	DIN-120
IF(I.EQ.0) GO TO 160	DIN-121
DO 100 J=1,NVI	DIN-122
IF(JAU(J).EQ.0) GO TO 110	DIN-123
II=I-4	DIN-124
JJ=JAU(J)-4	DIN-125
IA(II,JJ)=1	DIN-126
IA(JJ,II)=1	DIN-127
100 CONTINUE	DIN-128
110 WRITE(ISAUX,5006) I,(JAU(JJ),JJ=1,J-1)	DIN-129
DO 120 J=1,NVI	DIN-130
IF(JAU(J).EQ.0) GO TO 150	DIN-131
JAU(J)=0	DIN-132
120 CONTINUE	DIN-133
150 CONTINUE	DIN-134
C	DIN-135
C LECTURA DEL TIPO DE PROBLEMA	DIN-136
C IOP=1, PROBLEMA LINEAL IOP=2, PROBLEMA NO LINEAL.	DIN-137
C	DIN-138
160 WRITE(ISAL,3006)	DIN-139
READ(IENT,*) IOP	DIN-140
IF(IOP.EQ.1) WRITE(ISAUX,5007)	DIN-141
IF(IOP.EQ.2) WRITE(ISAUX,5008)	DIN-142
LL=0	DIN-143
NN=0	DIN-144
JOP=0	DIN-145

C		DIN-146
C	LECTURA DEL PARAMETRO OPCIONAL:	DIN-147
C	PROBLEMA LINEAL:	DIN-148
C	ITIP=0 MINIMIZAR LA LONGITUD Y ANCHURA. ITIP=1 MINIMIZAR UNA FUNCION	DIN-149
C	PROBLEMA NO LINEAL:	DIN-150
C	ITIP=0 MINIMIZACION SUPERFICIE. ITIP=1 MINIMIZACION FUNCION OBJETIVO	DIN-151
C		DIN-152
	WRITE(ISAL,3007)	DIN-153
	READ(IENT,*) ITIP	DIN-154
	IF(ITIP.EQ.0) GO TO 220	DIN-155
	WRITE(ISAX,5009)	DIN-156
C		DIN-157
C	LECTURA DE LOS COSTES DE LA FUNCION OBJETIVO SI SE HA ELEGIDO TIPO 1	DIN-158
C		DIN-159
	WRITE(ISAX,3008)	DIN-160
	READ(IENT,*) (C(I),I=1,NVI)	DIN-161
	WRITE(ISAX,5010) (C(I),I=1,NVI)	DIN-162
	DO 205 I=1,NVI	DIN-163
205	C(I)=-C(I)	DIN-164
	IF(IOP.EQ.1) GO TO 290	DIN-165
C		DIN-166
C	LECTURA EN EL PROBLEMA NO LINEAL DEL AREA TOTAL SI ESTA NO ES DATO	DIN-167
C		DIN-168
	WRITE(ISAL,3009)	DIN-169
	READ(IENT,*) SM	DIN-170
	LL=LL+1	DIN-171
	B(LL,1)=SM	DIN-172
	CODE(LL)=5	DIN-173
	DO 210 I=1,NVI	DIN-174
	IR(LL,I)=1	DIN-175
	IR(LL,I+NA)=1	DIN-176
210	CONTINUE	DIN-177
	WRITE(ISAX,5011) SM	DIN-178
	GO TO 290	DIN-179
220	C(4A)=1.	DIN-180
	IF(IOP.EQ.1) WRITE(ISAX,5012)	DIN-181
	IF(IOP.EQ.2) WRITE(ISAX,5013)	DIN-182
C		DIN-183
C	SERIE DE OPERACIONES A REALIZAR EN DOS CICLOS.	DIN-184
C	PRIMER CICLO: GRAFO DIRIGIDO VERTICAL	DIN-185
C	SEGUNDO CICLO: GRAFO DIRIGIDO HORIZONTAL:	DIN-186
C		DIN-187
C	LOS CODIGOS DE LAS RESTRICCIONES SON:	DIN-188
C	PROBLEMA LINEAL: 0 MENOR O IGUAL	DIN-189
C	1 MAYOR O IGUAL	DIN-190
C	2 IGUAL	DIN-191
C	PROBLEMA NO LINEAL: 1 RELACION LINEAL DE ADYACENCIA (MAYOR O IGUAL)	DIN-192
C	2 RELACION LINEAL DE USUARIO (MAYOR O IGUAL)	DIN-193
C	3 RELACION NO LINEAL DE USUARIO (MAYOR O IGUAL)	DIN-194
C	4 RELACION LINEAL KIRCHHOFF (IGUAL)	DIN-195
C	5 RELACION NO LINEAL USUARIO (IGUAL)	DIN-196
C		DIN-197
290	DO 500 K=1,2	DIN-198
C		DIN-199
C	LECTURA DE LA MATRIZ DE INCIDENCIA VERTICAL U HORIZONTAL	DIN-200
C	SE REFLEJAN ADEMAS LAS RESTRICCIONES IMPUESTAS POR EL USUARIO	DIN-201
C		DIN-202
	IF(K.EQ.1) WRITE(ISAL,3010)	DIN-203
	IF(K.EQ.2) WRITE(ISAL,3011)	DIN-204
	WRITE(ISAL,3012)	DIN-205
	IF(K.EQ.1.AND.IOP.EQ.1) WRITE(ISAL,3013)	DIN-206

IF(K.EQ.2.AND.IDP.EQ.1) WRITE(ISAL,3014)	DIN-207
IF(K.EQ.1.AND.IDP.EQ.2) WRITE(ISAL,3015)	DIN-208
IF(K.EQ.2.AND.IDP.EQ.2) WRITE(ISAL,3016)	DIN-209
DO 300 I=1,70	DIN-210
READ(IENT,*) LA,LO,LF,RO,RF	DIN-211
IF(LA.EQ.0) GO TO 310	DIN-212
IF(IDP.EQ.1) WRITE(ISAUX,5014) LA,LO,LF,RO,RF	DIN-213
IF(IDP.EQ.2.AND.K.EQ.1) WRITE(ISAUX,5014) LA,LO,LF,RO,RF	DIN-214
IF(IDP.EQ.2.AND.K.EQ.2) WRITE(ISAUX,5015) LA,LO,LF,RO	DIN-215
IF(RO.EQ.0) RO=1.01	DIN-216
IF(RF.EQ.0) RF=1.01	DIN-217
LA=LA-4	DIN-218
IF(K.EQ.1) BA(LA,LO)=RO	DIN-219
IF(K.EQ.2) BB(LA,LO)=RO	DIN-220
IF(K.EQ.1) BA(LA,LF)=-RF	DIN-221
IF(K.EQ.2) BB(LA,LF)=-RF	DIN-222
RF=0.	DIN-223
RO=0.	DIN-224
300 CONTINUE	DIN-225
C	DIN-226
C ** LLANADA A LA SUBROUTINA KIR QUE OBTIENE RESTRICCIONES POR LA PRIME-	DIN-227
C RA LEY DE KIRCHHOFF **	DIN-228
C	DIN-229
C 310 IF(K.EQ.1.AND.IDP.EQ.1) CALL KIR(NA,MVV,LL,1,IOP,B,CODE,BA,IR)	DIN-230
IF(K.EQ.1.AND.IDP.EQ.2) CALL KIR(NA,MVV,LL,K,IOP,B,CODE,BA,IR)	DIN-231
IF(K.EQ.2.AND.IDP.EQ.1) CALL KIR(NA,MVH,LL,1,IOP,B,CODE,BB,IR)	DIN-232
IF(K.EQ.2.AND.IDP.EQ.2) CALL KIR(NA,MVH,LL,K,IOP,B,CODE,BB,IR)	DIN-233
C	DIN-234
C ** LLANADA A LA SUBROUTINA ACC QUE OBTIENE LAS RESTRICCIONES DE LAS	DIN-235
C RELACIONES DE ACCESIBILIDAD **	DIN-236
C	DIN-237
IF(K.EQ.1.AND.IDP.EQ.1) CALL ACC(MVV,MVI,NA,LL,1,CODE,B,Y,X,BA,	DIN-238
IR,IA)	DIN-239
IF(K.EQ.1.AND.IDP.EQ.2) CALL ACC(MVV,MVI,NA,LL,K,CODE,B,Y,X,BA,	DIN-240
IR,IA)	DIN-241
IF(K.EQ.2.AND.IDP.EQ.1) CALL ACC(MVH,MVI,NA,LL,1,CODE,B,X,Y,BB,	DIN-242
IR,IA)	DIN-243
IF(K.EQ.2.AND.IDP.EQ.2) CALL ACC(MVH,MVI,NA,LL,K,CODE,B,X,Y,BB,	DIN-244
IR,IA)	DIN-245
C	DIN-246
C ** LLANADA A LA SUBROUTINA US1 O US2 QUE OBTIENEN LAS RESTRICCIONES	DIN-247
C DIMENSIONALES DEL USUARIO SEGUN SEA EL TIPO DE PROBLEMA **	DIN-248
C	DIN-249
IF(K.EQ.1.AND.IDP.EQ.1) CALL US1(NA,MVV,LL,B,CODE,BA,IR)	DIN-250
IF(K.EQ.2.AND.IDP.EQ.1) CALL US1(NA,MVH,LL,B,CODE,BB,IR)	DIN-251
IF(K.EQ.1.AND.IDP.EQ.2) CALL US2(NA,MVV,LL,K,B,CODE,BA,IR)	DIN-252
IF(K.EQ.2.AND.IDP.EQ.2) CALL US2(NA,MVH,LL,K,B,CODE,BB,IR)	DIN-253
IF(K.EQ.1) NN=LL	DIN-254
500 CONTINUE	DIN-255
C	DIN-256
C CONTROL POR SI EL NUMERO DE RESTRICCIONES ES MAYOR QUE EL PERMITIDO	DIN-257
C	DIN-258
IF(LL.LE.240.AND.LL.LE.5+NA) GO TO 501	DIN-259
IERR=NER(ISAL,2)	DIN-260
STOP 0002	DIN-261
C	DIN-262
C ESCRITURA OPCIONAL DE LA MATRIZ DE RESTRICCIONES. TERMINOS INDEPEN -	DIN-263
C DIENTES Y CODIGOS.	DIN-264
C	DIN-265
501 IF(ISSU(1).LT.0) 350,505	DIN-266

350	WRITE(ISAUK,5016) K	DIN-267
	DO 400 I=1,K	DIN-268
	IF(IOP.EQ.2) WRITE(ISAUK,5016) (IR(I,J),J=1,2*NA)	DIN-269
	IF(IOP.EQ.1) WRITE(ISAUK,5016) (IR(I,J),J=1,NA)	DIN-270
400	CONTINUE	DIN-271
	WRITE(ISAUK,5016) K	DIN-272
	DO 450 I=MM+1,LL	DIN-273
	WRITE(ISAUK,5010) B(I,1),CODE(I)	DIN-274
450	CONTINUE	DIN-275
505	IF(IOP.EQ.2) GO TO 610	DIN-276
510	NET=0	DIN-277
	NLET=0	DIN-278
	NGET=0	DIN-279
	DO 550 I=1,MM	DIN-280
	IF(CODE(I).EQ.0) NLET=NLET+1	DIN-281
	IF(CODE(I).EQ.1) NGET=NGET+1	DIN-282
	IF(CODE(I).EQ.2) NET=NET+1	DIN-283
550	CONTINUE	DIN-284
	INDB=0	DIN-285
C		DIN-286
C	** LLANADA A LA SUBROUTINA QUE SOLUCIONA EL PROBLEMA DE PROGRAMACION	DIN-287
C	LINEAL UTILIZANDO EL ALGORITMO SIMPLEX **	DIN-288
C		DIN-289
	CALL LIS(ISAL,NA,MM,1,NET,NGET,NLET,ITIP,CODE,B,NON,C,KB,IR(1,1),	DIN-290
	IR(1,61),ZETA,INDB)	DIN-291
	INDB=1	DIN-292
	NET=0	DIN-293
	NLET=0	DIN-294
	NGET=0	DIN-295
	DO 600 I=MM+1,LL	DIN-296
	IF(CODE(I).EQ.0) NLET=NLET+1	DIN-297
	IF(CODE(I).EQ.1) NGET=NGET+1	DIN-298
	IF(CODE(I).EQ.2) NET=NET+1	DIN-299
600	CONTINUE	DIN-300
	CALL LIS(ISAL,NA,LL,MM+1,NET,NGET,NLET,ITIP,CODE,B,NON,C,KB,IR(1,1,	DIN-301
	IR(1,61),ZETA,INDB)	DIN-302
601	WRITE(ISAL,4001)	DIN-303
	DO 605 J=1,NA	DIN-304
	JJ=J+4	
	WRITE(ISAL,4002) JJ,ZETA(J+NA),JJ,ZETA(J)	DIN-306
605	CONTINUE	DIN-307
	WRITE(ISAL,4003)	DIN-308
C		DIN-309
C	** LLANADA A LA SUBROUTINA QUE DIBUJA EL ESQUEMA DE PLANTA YA DINEN-	DIN-310
C	SIGNADO **	DIN-311
C		DIN-312
	CALL ESQ(ISAL,NA,HVH,HVV,ZETA(NA+1),ZETA(1),JAUX(2*NA+1),JAUX(1),	DIN-313
	LJAUX(NA+1),BA,BB,K,Y)	DIN-314
	IF(JOP.EQ.0) 620,350	DIN-315
C		DIN-316
C	** LLANADA A LA SUBROUTINA QUE RESUELVE EL PROBLEMA DE PROGRAMACION NO	DIN-317
C	LINEAL UTILIZANDO EL ALGORITMO DE POLAK-RIVIERE **	DIN-318
C		DIN-319
610	IF(IOP.EQ.2) CALL POL(ISAL,NA,LL,C,CODE,B,ZETA,IR)	DIN-320
C		DIN-321
C	** LLANADA A LA SUBROUTINA QUE AJUSTA LOS VALORES OBTENIDOS PARA LOS	DIN-322
C	ESPACIOS SI SE HA EMPLEADO LA PROGRAMACION NO LINEAL.	DIN-323
C		DIN-324
C	CALL AJJ(NA,LL,NET,NGET,NLET,ITIP,MM,CODE,B,NON,C,KB,ZETA,INDB,	DIN-325
	IR,JAUX)	DIN-326
	MM=MM-1	DIN-327

GO TO 601	DIN-328
620 WRITE(18AL,3017)	DIN-329
C	DIN-330
C LECTURA DE UN PARAMETRO OPCIONAL POR SI SE QUIERE AJUSTAR LOS VALORES	DIN-331
C HALLADOS	DIN-332
C	DIN-333
READ(18AUX,*) JOP	DIN-334
IF(JOP.EQ.0) GO TO 850	DIN-335
WRITE(18AL,3018)	DIN-336
READ(18AUX,*) (C(I),I=1,NVI)	DIN-337
WRITE(18AUX,3010) (C(I),I=1,NVI)	DIN-338
DO 630 I=1,NVI	DIN-339
C(I)=-C(I)	DIN-340
630 CONTINUE	DIN-341
LL=LL+1	DIN-342
IR(LL,NA)=1	DIN-343
B(LL,1)=ZETA(2*NA)	DIN-344
CODE(LL)=2	DIN-345
DO 750 I=LL,MM+1,-1	DIN-346
B(I+1,1)=B(I,1)	DIN-347
CODE(I+1)=CODE(I)	DIN-348
DO 700 J=1,NA	DIN-349
IR(I+1,J)=IR(I,J)	DIN-350
700 CONTINUE	DIN-351
750 CONTINUE	DIN-352
DO 800 J=1,NA	DIN-353
IR(MM+1,J)=0	DIN-354
800 CONTINUE	DIN-355
MM=MM+1	DIN-356
IR(MM,NA)=1	DIN-357
B(MM,1)=ZETA(MM)	DIN-358
CODE(MM)=2	DIN-359
LL=LL+1	DIN-360
GO TO 510	DIN-361
850 STOP	DIN-362
C	DIN-363
C	DIN-364
C FORMATOS	DIN-365
C	DIN-366
1001 FORMAT(1H1)	DIN-367
2001 FORMAT(40A2)	DIN-368
3001 FORMAT(/1X*MENSAJE :1:ED: TITULA EL PROBLEMA"/)	DIN-369
5001 FORMAT(1X,40A2,71X,79*"/)	DIN-370
3002 FORMAT(1X*MENSAJE :1:ED: NUMERO DE VERTICES VERTICALES Y HORIZONTA	DIN-371
LES"/)	DIN-372
5002 FORMAT(1X*NUMERO DE VERTICES VERTICALES*13)	DIN-373
5003 FORMAT(1X*NUMERO DE VERTICES HORIZONTALES ..*13,/))	DIN-374
3004 FORMAT(1X*MENSAJE :1:ED: COORDENADAS DEL CENTRO GEOMETRICO DE LOS	DIN-375
ESPACIOS. "/16X*SE HARA A PARTIR DEL ESQUERA ADIMENSIONAL, DANDO UN	DIN-376
8AS DIMENSIONES"/16X*ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARAN	DIN-377
8A PARTIR DEL 5. "/16X*AL FINAL DE LA LISTA INTRODUCE: 0"/1X*ESP	DIN-378
8 XC YC"/1X*--- -- --)	DIN-379
5005 FORMAT(14,2F7.2)	DIN-380
3005 FORMAT(/1X*MENSAJE :1:ED: CONDICIONES DE ACCESIBILIDAD ENTRE LOCAL	DIN-381
LES"/16X*SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES	DIN-382
8 A EL. "/16X*AL FINAL DE LA LISTA INTRODUCE: 0"/1X* L L1 L2 L3	DIN-383
8 L4 L5"/1X* - - - - -)	DIN-384
5006 FORMAT(13,3019,/))	DIN-385
3006 FORMAT(/1X*MENSAJE :1:ED: TIPO DE PROBLEMA: 1 .. LINEAL / 2 .. N	DIN-386
80 LINEAL"/)	DIN-387

C		KIR-449
C	ELABORACION DE LA RESTRICCIONES POR KIRCHHOFF	KIR-450
C		KIR-451
	DO 100 I=1,NA	KIR-452
	DO 50 J=1,NV-1	KIR-453
	IF(BB(I,J).EQ.0) GO TO 50	KIR-454
	IR(JJ+J,I+(K-1)*NA)=BB(I,J)/ABS(BB(I,J))	KIR-455
	50 CONTINUE	KIR-456
	100 CONTINUE	KIR-457
	LL=JJ+NV-1	KIR-458
C		KIR-459
C	VECTORES CODIGO Y TERMINOS INDEPENDIENTES	KIR-460
C		KIR-461
	DO 150 I=1,NV-1	KIR-462
	IF(10P.EQ.2) CODE(I+JJ)=4	KIR-463
	IF(10P.EQ.1) CODE(I+JJ)=2	KIR-464
	B(I+JJ,1)=0.	KIR-465
	150 CONTINUE	KIR-466
	RETURN	KIR-467
	END	KIR-468
C		ACC-469
C		ACC-470
	SENA(IJK,3)	ACC-471
	SUBROUTINE ACC(NV,NVI,NA,LL,K,CODE,B,K,Y,BB,IR,IA)	ACC-472
	*****	ACC-473
C		ACC-474
C	SUBROUTINA QUE OBTIENE LAS RESTRICCIONES DERIVADAS DE LAS RELACIONES DE	ACC-475
C	ACCESIBILIDAD IMPUESTAS POR EL USUARIO.	ACC-476
C		ACC-477
	ENA BB(50,40),IR(240,120),IA(60,60)	ACC-478
	INTEGER CODE(240)	ACC-479
	DIMENSION B(240,3),X(NA),Y(NA)	ACC-480
	NN=LL	ACC-481
C		ACC-482
C	ELABORACION DE LAS RESTRICCIONES	ACC-483
C		ACC-484
	DO 250 I=1,NVI-1	ACC-485
	DO 200 J=I+1,NVI	ACC-486
	IF(IA(I,J).EQ.0) GO TO 200	ACC-487
	DO 50 JJ=1,NV	ACC-488
	IF(BB(I,JJ).EQ.0.OR.BB(J,JJ).EQ.0) GO TO 50	ACC-489
	IF(BB(I,JJ).GT.0.AND.BB(J,JJ).GT.0) GO TO 50	ACC-490
	IF(BB(I,JJ).LT.0.AND.BB(J,JJ).LT.0) GO TO 50	ACC-491
	GO TO 60	ACC-492
	50 CONTINUE	ACC-493
	GO TO 200	ACC-494
	60 LL=LL+1	ACC-495
	IR(LL,I+(K-1)*NA)=1	ACC-496
	IR(LL+1,J+(K-1)*NA)=1	ACC-497
	DO 150 II=1,NA-1	ACC-498
	IF(II.EQ.1.OR.II.EQ.J) GO TO 150	ACC-499
	IF(BB(II,JJ).EQ.0) GO TO 150	ACC-500
	IF(K.EQ.1) 70,80	ACC-501
	70 IF(Y(J).GT.Y(I)) 110,120	ACC-502
	80 IF(Y(J).GT.Y(I)) 120,110	ACC-503
	110 IF(BB(II,JJ).LT.0) 130,140	ACC-504
	120 IF(BB(II,JJ).GT.0) 130,140	ACC-505
	130 IF(K(II).LT.K(I)) GO TO 150	ACC-506
	IR(LL,II+(K-1)*NA)=1	ACC-507
	IR(LL+1,II+(K-1)*NA)=-1	ACC-508

	GO TO 150	ACC-509
140	IF(X(I1).LT.X(J)) GO TO 150	ACC-510
	IR(LL,II+(K-1)*NA)=-1	ACC-511
	IR(LL+1,II+(K-1)*NA)=1	ACC-512
150	CONTINUE	ACC-513
	K1=0	ACC-514
	K2=0	ACC-515
C		ACC-516
C	SI TODOS LOS TERMINOS DE LA FILA SON POSITIVOS ESTA SE ELIMINA	ACC-517
C		ACC-518
	DO 155 II=1,NA*K	ACC-519
	IF(IR(LL,II).LT.0) K1=K1+1	ACC-520
	IF(IR(LL+1,II).LT.0) K2=K2+1	ACC-521
155	CONTINUE	ACC-522
	IF(K1.NE.0.OR.K2.EQ.0) GO TO 165	ACC-523
	DO 160 II=1,NA*K	ACC-524
	IR(LL,II)=IR(LL+1,II)	ACC-525
	IR(LL+1,II)=0.	ACC-526
160	CONTINUE	ACC-527
	GO TO 200	ACC-528
165	IF(K1.EQ.0.OR.K2.NE.0) GO TO 175	ACC-529
	DO 170 II=1,NA*K	ACC-530
	IR(LL+1,II)=0.	ACC-531
170	CONTINUE	ACC-532
	GO TO 200	ACC-533
175	IF(K1.NE.0.AND.K2.NE.0) GO TO 185	ACC-534
	DO 180 II=1,NA*K	ACC-535
	IR(LL,II)=0.	ACC-536
	IR(LL+1,II)=0.	ACC-537
180	CONTINUE	ACC-538
	LL=LL-1	ACC-539
185	IF(K1.NE.0.AND.K2.NE.0) LL=LL+1	ACC-540
200	CONTINUE	ACC-541
250	CONTINUE	ACC-542
	IF(LL.EQ.NN) GO TO 300	ACC-543
	DO 300 I=NN+1,LL	ACC-544
	CODE(I)=1	ACC-545
	B(I,1)=1.	ACC-546
300	CONTINUE	ACC-547
360	RETURN	ACC-548
	END	ACC-549
C		US1-550
C		US1-551
	ENNA(IJK,3)	US1-552
	SUBROUTINE US1(NA,NV,LL,B,CODE,BB,IR)	US1-553
C	*****	US1-554
C		US1-555
C	EXPRESA LAS RESTRICCIONES DIMENSIONALES IMPUESTAS POR EL USUARIO EN EL	US1-556
C	PROBLEMA LINEAL.	US1-557
C		US1-558
	ENNA BB(60,40),IR(240,120)	US1-559
	INTEGER CODE(240)	US1-560
	DIMENSION B(240,3)	US1-561
	NN=LL	US1-562
C		US1-563
C	DETERMINACION DE LAS COTAS SUPERIOR E INFERIOR PARA UNA VARIABLE.	US1-564
C	SI LAS DOS SON IGUALES LA RESTRICCION ES DEL TIPO IGUAL.	US1-565
C		US1-566
	DO 100 I=1,NA	US1-567
	J1=0	US1-568
	J2=0	US1-569

DD 50 J=1,NV	US1-570
IF(BB(I,J).EQ.0) GO TO 50	US1-571
IF(BB(I,J).GT.0) J1=J	US1-572
IF(BB(I,J).LT.0) J2=J	US1-573
50 CONTINUE	US1-574
IF(BB(I,J1).NE.-BB(I,J2)) GO TO 60	US1-575
NN=NN+1	US1-576
C	US1-577
C SI UNA VARIABLE NO POSEE COTA INFERIOR, SE HACE ESTA IGUAL A CERO.	US1-578
C	US1-579
IF(BB(I,J1).EQ.1.01) GO TO 70	US1-580
IF(BB(I,J1).NE.1.01) B(MN,1)=BB(I,J1)	US1-581
CODE(MN)=2	US1-582
IR(MN,1)=1	US1-583
GO TO 100	US1-584
60 NN=NN+1	US1-585
70 IF(BB(I,J1).EQ.1.01) GO,90	US1-586
80 B(MN)=0.	US1-587
GO TO 95	US1-588
90 B(MN,1)=BB(I,J1)	US1-589
95 CODE(MN)=1	US1-590
IR(MN,1)=1	US1-591
IF(BB(I,J2).EQ.-1.01) GO TO 100	US1-592
B(MN+1,1)=-BB(I,J2)	US1-593
CODE(MN+1)=0	US1-594
NN=NN+1	US1-595
IR(MN,1)=1	US1-596
100 CONTINUE	US1-597
LL=NN	US1-598
RETURN	US1-599
END	US1-600
C	US2-601
C	US2-602
*ENA(IJK,3)	US2-603
SUBROUTINE US2(MA,NV,LL,K,B,CODE,BB,IR)	US2-604
*****	US2-605
C	US2-606
C SUBROUTINA QUE EXPRESA LAS RESTRICCIONES DIMENSIONALES DEL USUARIO EN	US2-607
C EL PROBLEMA NO LINEAL	US2-608
C	US2-609
ENA BB(60,40),IR(240,120)	US2-610
INTEGER CODE(240)	US2-611
DIMENSION B(240,3)	US2-612
NN=LL	US2-613
C	US2-614
C EN LA MATRIZ DE INCIDENCIA EL TERMINO POSITIVO INDICA LA COTA INFERIOR	US2-615
C DE LA VARIABLE Y EL TERMINO NEGATIVO LA COTA INFERIOR DE LA SUPERFICIE	US2-616
C DE LA ESTANCIA.	US2-617
C	US2-618
DD 100 I=1,NA	US2-619
J1=0	US2-620
J2=0	US2-621
DD 50 J=1,NV	US2-622
IF(BB(I,J).EQ.0) GO TO 50	US2-623
IF(BB(I,J).GT.0) J1=J	US2-624
IF(BB(I,J).LT.0) J2=J	US2-625
50 CONTINUE	US2-626
NN=NN+1	US2-627
C	US2-628
C SI UNA VARIABLE NO POSEE COTA INFERIOR SE HACE ESTA IGUAL A CERO.	US2-629

C	IF(BB(I,J1).EQ.1.01) B(MN,1)=0.	U92-630
	IF(BB(I,J1).NE.1.01) B(MN,1)=BB(I,J1)	U92-631
	CODE(MN)=2	U92-632
	IR(MN,1+(K-1)*NA)=1	U92-633
	IF(K.EQ.2) GO TO 100	U92-634
C		U92-635
C	SI UNA ESTANCIA NO TIENE AREA MINIMA SE HACE ESTA IGUAL A CERO.	U92-636
C		U92-637
C	IF(BB(I,J2).EQ.-1.01) GO TO 100	U92-638
	B(MN+1,1)=-BB(I,J2)	U92-639
	CODE(MN+1)=3	U92-640
	MN=MN+1	U92-641
	IR(MN,1)=1	U92-642
	IR(MN,1+NA)=1	U92-643
100	CONTINUE	U92-644
	LL=MN	U92-645
	RETURN	U92-646
	END	U92-647
C		NER-648
C		NER-649
C	FUNCION NER(ISAL,IERR)	NER-650
C	*****	NER-651
C		NER-652
C	FUNCION QUE DETERMINA EL TIPO DE ERROR	NER-653
C		NER-654
C	WRITE(ISAL,3001) IERR	NER-655
	NER=IERR	NER-656
	RETURN	NER-657
C		NER-658
C		NER-659
C	FORMATS	NER-660
C		NER-661
C	3001 FORMAT(//"ERROR *13)	NER-662
	END	NER-663
C		LIS-664
C		LIS-665
C	SUBROUTINE LIS(ISAL,K,N,IJ,NET,NGET,NLET,NTYPE,CODE,B,TITLE,	LIS-666
	&C,XB,IR,A,ZETA,INDS)	LIS-667
C	*****	LIS-668
C		LIS-669
C	SUBROUTINA QUE SOLUCIONA EL PROBLEMA DE PROGRAMACION LINEAL, UTILIZANDO	LIS-670
C	EL ALGORITMO SIMPLEX.	LIS-671
C		LIS-672
	INTEGER CODE(240),XB(120),C1(120),BASICS,OPTSOL	LIS-673
	REAL TITLE(20)	LIS-674
	DIMENSION B(240,3),C(60),ZETA(120),B1(120),C2(60)	LIS-675
	EMA IR(240,60),A(120,50)	LIS-676
	DO 10 I=1,120	LIS-677
	XB(I)=0	LIS-678
	C1(I)=0	LIS-679
	B1(I)=0	LIS-680
10	CONTINUE	LIS-681
	DO 20 I=1,K	LIS-682
	C2(I)=C(I)	LIS-683
20	CONTINUE	LIS-684
	DO 100 I=N,IJ,-1	LIS-685
	B1(I-IJ+1)=B(I,1)	LIS-686
	C1(I-IJ+1)=CODE(I)	LIS-687
	DO 50 J=K,1,-1	LIS-688
	A(I-IJ+1,J)=IR(I,J)	LIS-689
50	CONTINUE	LIS-690

100	CONTINUE	LIS-691
	NI=N-IJ+1	LIS-692
C		LIS-693
C	** LLAMADA A LA SUBROUTINA QUE PROPORCIONA VARIABLES DE HOLGURA, DE -	LIS-694
C	FECTO Y ARTIFICIALES CUANDO SEAN NECESARIAS **	LIS-695
C		LIS-696
	CALL SSA(KPI,NPI,K,NI,NGET,NLET,NET,NP1,N,NC1,NC,INDEXC,INDEXL,	LIS-697
	SINDEXE,NTYPE,SUM,C2,XB,C1,B1,A)	LIS-698
	BASICS=0	LIS-699
	DPTSOL=0	LIS-700
C		LIS-701
C	** LLAMADA A LA SUBROUTINA QUE EFECTUA EL ALGORITMO SIMPLEX **	LIS-702
C		LIS-703
	CALL SINCSAL,BASICS,NI,K,SUM,DPTSOL,N,NP1,NPI,NC1,C2,XB,A,ZETA,	LIS-704
	SINDS)	LIS-705
	IF(NTYPE.EQ.1) GO TO 220	LIS-706
	SUM=-SUM	LIS-707
220	RETURN	LIS-708
	END	LIS-709
C		SSA-710
C		SSA-711
	SUBROUTINE SSA(KPI,NPI,K,N,NGET,NLET,NET,NP1,N,NC1,NC,INDEXC,	SSA-712
	SINDEXL,INDEXE,NTYPE,SUM,C,XB,CODE,B,A)	SSA-713
	*****	SSA-714
C		SSA-715
C	SUBROUTINA QUE PROPORCIONA VARIABLES DE HOLGURA, DEFECTOS Y ARTIFICIA-	SSA-716
C	LES CUANDO SEAN NECESARIAS.	SSA-717
C		SSA-718
	INTEGER CODE(120),XB(120)	SSA-719
	DIMENSION B(120),C(60),ARTV(120)	SSA-720
	END A(120,60)	SSA-721
C		SSA-722
C	INICIALIZACION DE VARIABLES	SSA-723
C		SSA-724
	DO 1 I=1,120	SSA-725
	ARTV(I)=0	SSA-726
1	CONTINUE	SSA-727
	IA=1	SSA-728
	KPI=K+1	SSA-729
	NPI=N+1	SSA-730
	N=K+2*NGET+NLET+NET	SSA-731
	NPI=N+1	SSA-732
	NC=K+NGET+1	SSA-733
	NC1=NC+NLET	SSA-734
	INDEXC=C+1	SSA-735
	INDEXL=C+NGET+1	SSA-736
	INDEXE=K+NGET+NLET+1	SSA-737
	DO 69 I=1,NPI	SSA-738
	DO 69 J=KPI,NPI	SSA-739
69	A(I,J)=0.	SSA-740
	DO 5 I=1,N	SSA-741
5	A(I,NPI)=B(I)	SSA-742
	DO 4 I=1,N	SSA-743
	IF(CODE(I).EQ.0) GO TO 6	SSA-744
	IF(CODE(I).EQ.1) GO TO 8	SSA-745
	ARTV(IA)=1	SSA-746
	IA=IA+1	SSA-747
	XB(I)=INDEXE	SSA-748
	A(I,INDEXE)=1.	SSA-749
	INDEXE=INDEXE+1	SSA-750

GO TO 4	SSA-751
8 KB(I)=INDEKE	SSA-752
ARTV(IA)=I	SSA-753
IA=IA+1	SSA-754
INDEKE=INDEKE+1	SSA-755
A(I,INDEKE)=-1.	SSA-756
INDEKE=INDEKE+1	SSA-757
GO TO 4	SSA-758
6 KB(I)=INDEKL	SSA-759
A(I,INDEKL)=1.	SSA-760
INDEKL=INDEKL+1	SSA-761
4 CONTINUE	SSA-762
C	SSA-763
C COMPROBAR MAXIMIZACION	SSA-764
C	SSA-765
IF(NTYPE.EQ.0) GO TO 12	SSA-766
DO 60 J=1,K	SSA-767
60 A(NP1,J)=-C(J)	SSA-768
GO TO 50	SSA-769
12 DO 55 J=1,K	SSA-770
55 A(NP1,J)=C(J)	SSA-771
50 DO 61 J=KP1,HP1	SSA-772
A(NP1,J)=0.	SSA-773
61 C(J)=0.	SSA-774
DO 62 J=1,K	SSA-775
62 C(J)=-A(NP1,J)	SSA-776
DO 63 J=NC1,N	SSA-777
63 C(J)=-10.E2	SSA-778
IF(NGET+NET.EQ.0) RETURN	SSA-779
IA=IA-1	SSA-780
KPSTE=K+NGET	SSA-781
DO 64 J=1,KPSTE	SSA-782
SUM=0.	SSA-783
DO 65 I=1,IA	SSA-784
Q000=ARTV(I)	SSA-785
65 SUM=SUM+A(Q000,J)	SSA-786
64 A(NP1,J)=A(NP1,J)-10.E2*SUM	SSA-787
SUM=0.	SSA-788
DO 66 I=1,IA	SSA-789
Q000=ARTV(I)	SSA-790
66 SUM=SUM+A(Q000,HP1)	SSA-791
A(NP1,HP1)=A(NP1,HP1)-10.E2*SUM	SSA-792
RETURN	SSA-793
END	SSA-794
C	SIN-795
C	SIN-796
SUBROUTINE SIN(ISA,BASICS,N,K,SUM,OPTSOL,N,NP1,HP1,NC1,C,KB,A,	SIN-797
ZETA,KRS)	SIN-798
*****	SIN-799
C	SIN-800
C SUBROUTINA QUE REALIZA EL ALGORITMO SIMPLEX.	SIN-801
C	SIN-802
INTEGER KB(120),BASICS,OPTSOL	SIN-803
DIMENSION C(60),ZETA(120)	SIN-804
ENA A(120,60)	SIN-805
C	SIN-806
C PASO 1: OBTENER UNA SOLUCION BASICA ADMISIBLE.	SIN-807
C	SIN-808
100 BASICS=BASICS+1	SIN-809
GO TO 200	SIN-810
105 SUM=0.	SIN-811

DD 112 I=1,N	SIN-812
IF(KB(I).LE.K) ZETA(KB(I)+KRS+K)=A(I, NP1)	SIN-813
PPP=KB(I)	SIN-814
112 SUM=SUM+C(PPP)*A(I, NP1)	SIN-815
IF(OPTSOL.EQ.1)GO TO 920	SIN-816
C	SIN-817
C PASO 2,3: ESCOGER LA VARIABLE NO BASICA DE COSTO RELATIVO MAS NEGATIVO	SIN-818
C PARA INTRODUCIRLA EN LA BASE. SI NINGUNO ES NEGATIVO, SE HA ALCANZADO	SIN-819
C EL OPTIMO.	SIN-820
C IR AL PASO 9	SIN-821
C	SIN-822
200 NEG=0	SIN-823
CNEG=0.	SIN-824
DD 210 J=1,N	SIN-825
IF(A(NP1,J).GE.CNEG) GO TO 210	SIN-826
CNEG=A(NP1,J)	SIN-827
NEG=J	SIN-828
210 CONTINUE	SIN-829
IF(NEG.EQ.0) GO TO 900	SIN-830
C	SIN-831
C PASO 4: COMPROBAR SI LA FUNCION OBJETIVO ES NO ACOTADA.	SIN-832
C	SIN-833
400 SPR=10.E10	SIN-834
DD 410 I=1,N	SIN-835
IF(A(I,NEG).LE.0.00001) GO TO 410	SIN-836
IF(A(I, NP1)/A(I,NEG).GE.SPR) GO TO 410	SIN-837
SPR=A(I, NP1)/A(I,NEG)	SIN-838
NSPR=I	SIN-839
410 CONTINUE	SIN-840
IF(SPR.LE.10.E8) GO TO 510	SIN-841
IERR=NER(ISAL,3)	SIN-842
STDP 0003	SIN-843
C	SIN-844
C PASO 5: PIVOTEAR	SIN-845
C	SIN-846
510 PELE=A(NSPR,NEG)	SIN-847
DD 500 J=1, NP1	SIN-848
500 A(NSPR,J)=A(NSPR,J)/PELE	SIN-849
KB(NSPR)=NEG	SIN-850
C	SIN-851
C PASOS 6,7,8: REALIZAR TRANSFORMACIONES ELENENTALES E IR AL PASO 1 PARA	SIN-852
C IMPRINIR LA NUEVA SOLUCION BASICA ADMISIBLE.	SIN-853
C	SIN-854
600 DD 610 I=1, NP1	SIN-855
IF(I.EQ.NSPR) GO TO 610	SIN-856
HOLD=A(I,NEG)	SIN-857
DD 620 J=1, NP1	SIN-858
620 A(I,J)=A(I,J)-HOLD*A(NSPR,J)	SIN-859
610 CONTINUE	SIN-860
GO TO 100	SIN-861
C	SIN-862
C PASOS 9,10: SI UNA VARIABLE ARTIFICIAL ES POSITIVA NO EXISTE SOLUCION	SIN-863
C ADMISIBLE. EN CASO CONTRARIO SE HA OBTENIDO EL OPTIMO.	SIN-864
C	SIN-865
900 OPTSOL=1.	SIN-866
GO TO 105	SIN-867
920 DD 930 I=1,N	SIN-868
IF(KB(I).LT.NC1) GO TO 930	SIN-869
IF(A(I, NP1).LE.0) GO TO 930	SIN-870
IERR=NER(ISAL,4)	SIN-871

STDP 0004	SIN-072
930 CONTINUE	SIN-073
KRS=1	SIN-074
RETURN	SIN-075
C	SIN-076
C FDRMATS	SIN-077
C	SIN-078
END	SIN-079
C	POL-080
C	POL-081
SENA(IJK,3)	POL-082
SUBROUTINE PDL(19AL,N,K,C,CODE,B,ZETA,IR)	POL-083
*****	POL-084
C	POL-085
C SUBROUTINA QUE RESUELVE EL PROBLEMA DE PROGRAMACION NO LINEAL UTILIZAND	POL-086
C EL ALGORITMO DE POLAK-RIVIERE.	POL-087
C ESTE ALGORITMO PENALIZA LA FUNCION OBJETIVO CON UN PENALTY EXTERIOR Y	POL-088
C OTRO INTERIOR Y UTILIZA EL METODO DEL GRADIENTE CONJUGADO.	POL-089
C	POL-090
INTEGER CODE(240)	POL-091
DIMENSION C(60),B(240,3),ZETA(120),GF(120),GP(120),GPP(120)	POL-092
DIMENSION GF1(120),G(120),H(120),Z(120)	POL-093
ENA IR(240,120)	POL-094
JJS=0	POL-095
10 NN=0	POL-096
C	POL-097
C VALORES DE LOS DISTINTOS FACTORES:	POL-098
C CAMPO DE ACCION RECOMENDABLE:	POL-099
C	POL-900
C ONE (0.2,5)	POL-901
C BETA(0.5,0.8)	POL-902
C EX1 0.05	POL-903
C EX2 0.0025	POL-904
C EXI (0.5,1)	POL-905
C	POL-906
ONE=1.	POL-907
BETA=0.58	POL-908
EX1=0.05	POL-909
EX2=0.025	POL-910
EXI=0.7	POL-911
SN=0.	POL-912
C	POL-913
C PASO 01 SE TOMA COMO VALORES INICIALES LA COTA INFERIOR DE CADA UNA DE	POL-914
C LAS VARIABLES. PARA LA LONGITUD Y ANCHURA DEL CONTORNO, SI NO SON DA-	POL-915
C TOS, SE TOMA LA RAIZ CUADRADA DE LA SUMA DE LOS CUADRADOS DE LOS VALO-	POL-916
C RES INICIALES DE LAS RESTANTES VARIABLES.	POL-917
C	POL-918
DO 150 I=1,K	POL-919
IF(CODE(I).NE.2) GO TO 140	POL-920
DO 100 J=1,2*N	POL-921
IF(IR(I,J).NE.0) GO TO 110	POL-922
100 CONTINUE	POL-923
110 ZETA(J)=0(I,1)	POL-924
140 IF(CODE(I).NE.5) GO TO 150	POL-925
SN=B(I,1)	POL-926
150 CONTINUE	POL-927
IF(SN.NE.0) GO TO 210	POL-928
DO 200 I=1,N-1	POL-929
SN=SN+ZETA(I)+ZETA(I+N)	POL-930
200 CONTINUE	POL-931
210 ZETA(N)=SQRT(SN)	POL-932
ZETA(2*N)=SQRT(SN)	

<pre> KONT=0 C C PASO 1: ITERACIONES PARA AJUSTAR LA SOLUCION OPTIMA. C DO 850 L=0,10 JONT=0 C C PASO 2: DIVISION DE LAS RESTRICCIONES UNA VEZ SUSTITUIDAS LAS VARIA- C BLES EN ELLAS POR LOS VALORES INICIALES (EN RESTRICCIONES MAYORES O C IGUALES QUE CERO Y RESTRICCIONES MENORES QUE CERO). LAS PRIMERAS SE C PENALIZAN EXTERIORMENTE Y LAS SEGUNDAS INTERIORMENTE. C DO 300 I=1,K B(I,2)=B(I,1) IF(CODE(I).EQ.5) GO TO 270 DO 250 J=1,2*N IF(IR(I,J).EQ.0) GO TO 250 IF(CODE(I).NE.1.AND.CODE(I).NE.4) GO TO 260 B(I,2)=B(I,2)-IR(I,J)*ZETA(J) 250 CONTINUE GO TO 300 260 IF(CODE(I).EQ.2) B(I,2)=B(I,2)-ZETA(J) IF(CODE(I).EQ.3) B(I,2)=B(I,2)-ZETA(J)*ZETA(J+N) IF(CODE(I).NE.5) GO TO 300 270 DO 280 JJ=1,N-1 B(I,2)=B(I,2)-ZETA(JJ)*ZETA(JJ+N) 280 CONTINUE 300 CONTINUE C C PASO 3: C ** LLAMADA A LA SUBROUTINA QUE OBTIENE LOS VALORES DE LA FUNCION OBJE- C TIVO Y FUNCIONES PENALTIS ** C 310 CALL FUN(N,2*N,K,F,P,PP,CODE,C,ZETA,B,IR) C C PASO 4: C ** LLAMADA A LA SUBROUTINA QUE OBTIENE EL GRADIENTE DE LA FUNCION OB- C JETIVO ** C CALL GRI(N,2*N,GF,C,ZETA) C C PASO 5: C ** LLAMADA A LA SUBROUTINA QUE OBTIENE LOS GRADIENTES DE LAS FUNCIONES C PENALTIS ** C CALL GR2(N,2*N,K,CODE,SP,GPP,ZETA,B,IR) C C PASO 6: DETERMINACION DEL VALOR DE LA FUNCION OBJETIVO PENALIZADA. C F1=F+P/EX1+EX2*PP C C PASO 7: DETERMINACION DEL GRADIENTE DE LA FUNCION PENALIZADA. C DO 400 J=1,2*N GF(J)=GF(J)+CP(J)/EX1+EX2*GPP(J) 400 CONTINUE IF(JONT.EQ.1) GO TO 660 401 DO 405 J=1,2*N G(J)=-GF(J) H(J)=-CF1(J) </pre>	<pre> POL-933 POL-934 POL-935 POL-936 POL-937 POL-938 POL-939 POL-940 POL-941 POL-942 POL-943 POL-944 POL-945 POL-946 POL-947 POL-948 POL-949 POL-950 POL-951 POL-952 POL-953 POL-954 POL-955 POL-956 POL-957 POL-958 POL-959 POL-960 POL-961 POL-962 POL-963 POL-964 POL-965 POL-966 POL-967 POL-968 POL-969 POL-970 POL-971 POL-972 POL-973 POL-974 POL-975 POL-976 POL-977 POL-978 POL-979 POL-980 POL-981 POL-982 POL-983 POL-984 POL-985 POL-986 POL-987 POL-988 POL-989 POL-990 POL-991 POL-992 </pre>
---	--

405	CONTINUE	POL-993
	NM=0	POL-994
C		POL-995
C	PASO 8: DETERMINACION DE LA MINIMA LONGITUD DEL PASO.	POL-996
C		POL-997
410	NM=0	POL-998
	K=0.	POL-999
420	DO 450 J=1,2*N	POL-000
	Z(J)=ZETA(J)+K*H(J)	POL-001
450	CONTINUE	POL-002
C		POL-003
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE EL VALOR DE LA FUNCION OBJETIVO	POL-004
C	Y LAS FUNCIONES PENALTIIS **	POL-005
C		POL-006
	CALL FUN(N,2*N,K,F,P,PP,CODE,C,Z,B,IR)	POL-007
	TET=F+P/EX1+EX2*PP-F1	POL-008
C		POL-009
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE EL GRADIENTE DE LA FUNCION OBJE	POL-010
C		POL-011
	CALL GR1(N,2*N,GF,C,Z)	POL-012
C		POL-013
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE LOS GRADIENTES DE LAS FUNCIONES	POL-014
C	PENALTIIS **	POL-015
C		POL-016
	CALL GR2(N,2*N,K,CODE,SP,GPP,Z,B,IR)	POL-017
	TETT=0	POL-018
	DO 500 J=1,2*N	POL-019
	GF(J)=GF(J)+SP(J)/EX1+EX2*GPP(J)	POL-020
	TETT=TETT+GF(J)*H(J)	POL-021
500	CONTINUE	POL-022
	IF(TETT.EQ.0.) GO TO 580	POL-023
	TAND=1.	POL-024
510	DO 550 J=1,2*N	POL-025
	Z(J)=ZETA(J)+(X-TAND*TETT)*H(J)	POL-026
550	CONTINUE	POL-027
C		POL-028
C	** LLAMADA A LA SUBROUTINA QUE OBTIENE EL VALOR DE LA FUNCION OBJETIVO	POL-029
C	Y DE LAS FUNCIONES PENALTIIS **	POL-030
C		POL-031
	CALL FUN(N,2*N,K,F,P,PP,CODE,C,Z,B,IR)	POL-032
	F2=F+P/EX1+EX2*PP	POL-033
	TET1=F2-F1	POL-034
	DELTA=TET1-TET+TAND*TETT**2/2.	POL-035
	IF(DELTA.LE.0.001) GO TO 560	POL-036
	TAND=BETA*TAND	POL-037
	GO TO 510	POL-038
560	NM=NM+1	POL-039
	IF(NM.GE.2) GO TO 570	POL-040
	K=K-TAND*TETT	POL-041
	GO TO 420	POL-042
570	UN=K-TAND*TETT	POL-043
	GO TO 590	POL-044
580	UN=K	POL-045
C		POL-046
C	PASO 9: DETERMINACION DE LOS NUEVOS VALORES DE LAS VARIABLES.	POL-047
C		POL-048
590	DO 600 J=1,2*N	POL-049
	Z(J)=ZETA(J)	POL-050
	ZETA(J)=ZETA(J)+UN*H(J)	POL-051
600	CONTINUE	POL-052
	KONT=KONT+1	POL-053

JONT=1	POL-054
GO TO 310	POL-055
C	POL-056
C PASO 10: DETERMINACION DE LA NORMA DEL NUEVO GRADIENTE DE LA FUNCION	POL-057
C OBJETIVO PENALIZADA	POL-058
C	POL-059
C	POL-060
660 RCF=0	POL-061
DO 700 J=1,2*M	POL-062
RCF=RCF+CF1(J)**2	POL-063
700 CONTINUE	POL-064
RCF=SQRT(RCF)	POL-065
IF(RCF.GT.EXI) 710,860	POL-066
710 NN=NN+1	POL-067
IF(ISSU(0).LT.0) WRITE(ISAL,3001) JJS,NN	POL-068
3001 FORMAT(2I3)	POL-069
C	POL-070
C PASO 11: COMPROBACION SI SE HA REALIZADO TANTOS CICLOS COMO VARIABLES.	POL-071
C	POL-072
IF(NN.LT.2*M) 719,715	POL-073
C	POL-074
C PASO 12: GRADIENTE CONJUGADO.	POL-075
C	POL-076
715 JJS=JJS+1	POL-077
IF(JJS.EQ.2) 860,401	POL-078
719 RC=0.	POL-079
RCF=0.	POL-080
DO 720 J=1,2*M	POL-081
RC=RC+C(J)**2	POL-082
RCF=RCF+CF1(J)**2	POL-083
720 CONTINUE	POL-084
IF(RCF.LE.RC) GO TO 740	POL-085
DO 730 J=1,2*M	POL-086
ZETA(J)=ONE*ZETA(J)+(1-ONE)*Z(J)	POL-087
730 CONTINUE	POL-088
740 RC=0	POL-089
ESC=0	POL-090
DO 750 J=1,2*M	POL-091
ESC=ESC+(CF1(J)+C(J))*GF1(J)	POL-092
RC=RC+C(J)**2	POL-093
G(J)=-CF1(J)	POL-094
750 CONTINUE	POL-095
ESC=ESC/RC	POL-096
DO 800 J=1,2*M	POL-097
H(J)=G(J)+ESC*H(J)	POL-098
800 CONTINUE	POL-099
GO TO 410	POL-100
850 CONTINUE	POL-101
860 RETURN	POL-102
END	POL-103
C	FUN-104
C	FUN-105
#ENR(IJK,3)	FUN-106
SUBROUTINE FUN(N,NN,K,F,P,PP,CODE,C,Z,B,IR)	FUN-107
*****	FUN-108
C	FUN-109
C SUBROUTINA QUE OBTIENE LOS VALORES DE LAS FUNCIONES OBJETIVO, PENALTY	FUN-110
C EXTERIOR Y PENALTY INTERIOR.	FUN-111
C	FUN-112
INTEGER CODE(K)	FUN-113

DIMENSION C(N),Z(MN),B(240,3)	FUN-114
EMA IR(240,120)	FUN-115
F=0	FUN-116
DO 50 J=1,M	FUN-117
F=F+C(J)*Z(J)+Z(J+N)	FUN-118
50 CONTINUE	FUN-119
P=0	FUN-120
PP=0	FUN-121
DO 200 I=1,K	FUN-122
B(I,3)=B(I,1)	FUN-123
IF(CODE(I).EQ.5) GO TO 120	FUN-124
DO 100 J=1,MN	FUN-125
IF(IR(I,J).EQ.0) GO TO 100	FUN-126
IF(CODE(I).NE.1.AND.CODE(I).NE.4) GO TO 110	FUN-127
B(I,3)=B(I,3)-IR(I,J)*Z(J)	FUN-128
100 CONTINUE	FUN-129
GO TO 200	FUN-130
110 IF(CODE(I).EQ.2) B(I,3)=B(I,3)-Z(J)	FUN-131
IF(CODE(I).EQ.3) B(I,3)=B(I,3)-Z(J)+Z(J+N)	FUN-132
IF(CODE(I).NE.5) GO TO 200	FUN-133
120 DO 150 JJ=1,M-1	FUN-134
B(I,3)=B(I,3)-Z(JJ)+Z(JJ+N)	FUN-135
150 CONTINUE	FUN-136
200 CONTINUE	FUN-137
DO 250 I=1,K	FUN-138
IF(CODE(I).LE.3.AND.B(I,3).LT.0) GO TO 210	FUN-139
IF(CODE(I).GT.3.OR.B(I,2).GE.0) P=P+B(I,3)+B(I,3)	FUN-140
210 IF(CODE(I).LE.3.AND.B(I,2).LT.0) PP=PP-1./B(I,3)	FUN-141
250 CONTINUE	FUN-142
RETURN	FUN-143
END	FUN-144
C	
C	
SUBROUTINE GR1(N,MN,GF,C,Z)	GR1-145
*****	GR1-146
C	GR1-147
C	GR1-148
C SUBROUTINA QUE OBTIENE EL GRADIENTE DE LA FUNCION OBJETIVO	GR1-149
C	GR1-150
DIMENSION C(N),GF(MN),Z(MN)	GR1-151
DO 50 J=1,MN	GR1-152
GF(J)=0.	GR1-153
IF(J.LE.N) GF(J)=C(J)+Z(J+N)	GR1-154
IF(J.GT.N) GF(J)=C(J-N)+Z(J-N)	GR1-155
50 CONTINUE	GR1-156
RETURN	GR1-157
END	GR1-158
C	GR1-159
C	GR2-160
SENA(IJK,3)	GR2-161
SUBROUTINE GR2(N,MN,K,CODE,GP,GPP,Z,B,IR)	GR2-162
*****	GR2-163
C	GR2-164
C	GR2-165
C SUBROUTINA QUE OBTIENE LOS GRADIENTES DE LA FUNCION PENALTY EXTERIOR E	GR2-166
C INTERIOR	GR2-167
C	GR2-168
INTEGER CODE(K)	GR2-169
DIMENSION Z(MN),B(240,3),GP(MN),GPP(MN)	GR2-170
EMA IR(240,120)	GR2-171
DO 100 J=1,MN	GR2-172
GP(J)=0	GR2-173
GPP(J)=0	GR2-174

	DO 50 I=1,K	GR2-175
	IF(IR(I,J).EQ.0) GO TO 50	GR2-176
	IF(CODE(I).EQ.3.OR.CODE(I).EQ.5) GO TO 20	GR2-177
	IF(CODE(I).LE.3.AND.B(I,3).LT.0) GO TO 10	GR2-178
	IF(CODE(I).GT.3.OR.B(I,2).GE.0) GP(J)=GP(J)-B(I,3)*IR(I,J)	GR2-179
	10 IF(CODE(I).LE.3.AND.B(I,2).LT.0) GPP(J)=GPP(J)-IR(I,J)/B(I,3)**2	GR2-180
	GO TO 50	GR2-181
	20 IF(CODE(I).EQ.3.AND.B(I,3).LT.0) GO TO 30	GR2-182
	IF(CODE(I).GT.3.OR.B(I,2).GE.0.AND.J.LE.N) GP(J)=GP(J)-B(I,3)*Z(J+	GR2-183
	&N)	GR2-184
	IF(CODE(I).GT.3.OR.B(I,2).GE.0.AND.J.GT.N) GP(J)=GP(J)-B(I,3)*Z(J-	GR2-185
	&N)	GR2-186
	30 IF(CODE(I).LE.3.AND.B(I,2).LT.0.AND.J.LE.N) GPP(J)=GPP(J)-Z(J+H)/B	GR2-187
	&(I,3)**2	GR2-188
	IF(CODE(I).LE.3.AND.B(I,2).LT.0.AND.J.GT.N) GPP(J)=GPP(J)-Z(J-H)/B	GR2-189
	&(I,3)**2	GR2-190
	50 CONTINUE	GR2-191
	GP(J)=2.*GP(J)	GR2-192
	100 CONTINUE	GR2-193
	RETURN	GR2-194
	END	GR2-195
C		AJU-196
C		AJU-197
	SUBROUTINE AJU(N,K,MET,NGET,NLET,ITIP,LL,CODE,B,NON,C,XB,Z,INDS,	AJU-198
	IR,JAUX)	AJU-199
C	*****	AJU-200
C		AJU-201
C	SUBROUTINA PARA AJUSTAR LOS VALORES OBTENIDOS POR PROGRAMACION LINEAL	AJU-202
C	APLICANDO EL ALGORITMO SIMPLEX A PARTIR DE LOS RESULTADOS OBTENIDOS	AJU-203
C		AJU-204
	ENA IR(240,120)	AJU-205
	INTEGER CODE(240),NON(40),XB(120),JAUX(240)	AJU-206
	DIMENSION Z(120),C(60),B(240,3)	AJU-207
C		AJU-208
C	VAMOS A ELIMINAR DE LA MATRIZ DE RESTRICCIONES LAS NO LINEALES	AJU-209
C		AJU-210
	DO 200 I=1,K	AJU-211
	IF(CODE(I).NE.3.AND.CODE(I).NE.5) GO TO 200	AJU-212
	DO 100 J=1,K	AJU-213
	DO 50 JJ=1,2*N	AJU-214
	IR(J,JJ)=IR(J+1,JJ)	AJU-215
	50 CONTINUE	AJU-216
	B(J,1)=B(J+1,1)	AJU-217
	CODE(J)=CODE(J+1)	AJU-218
	100 CONTINUE	AJU-219
	DO 150 J=1,2*N	AJU-220
	IR(K,J)=0	AJU-221
	150 CONTINUE	AJU-222
	B(K,1)=0	AJU-223
	CODE(K)=0	AJU-224
	K=K-1	AJU-225
	200 CONTINUE	AJU-226
C		AJU-227
C	SE AGRUPA LA MATRIZ DE RESTRICCIONES	AJU-228
C		AJU-229
	LL=0	AJU-230
	DO 350 I=1,K	AJU-231
	KT=0	AJU-232
	DO 250 J=1,N	AJU-233
	IF(IR(I,J).EQ.0) GO TO 250	AJU-234

	KT=KT+1	AJU-235
250	CONTINUE	AJU-236
	IF(KT.NE.0) GO TO 350	AJU-237
	IF(LL.EQ.0) LL=1	AJU-238
	DO 300 J=1,N	AJU-239
	IR(I,J)=IR(I,J+N)	AJU-240
	IR(I,J+N)=0	AJU-241
300	CONTINUE	AJU-242
350	CONTINUE	AJU-243
C		AJU-244
C	SE ACOTAN LAS VARIABLES DE NUEVO DE ACUERDO A LOS RESULTADOS OBTENIDOS	AJU-245
C		AJU-246
	DO 500 I=1,K	AJU-247
	IF(CODE(I).NE.2) GO TO 500	AJU-248
	DO 450 J=1,N	AJU-249
	IF(IR(I,J).NE.0) GO TO 460	AJU-250
450	CONTINUE	AJU-251
460	IF(LL.LE.1) GO TO 470	AJU-252
	IF(Z(J).GT.B(I,1)) B(I,1)=Z(J)	AJU-253
	GO TO 500	AJU-254
470	IF(Z(J+N).GT.B(I,1)) B(I,1)=Z(J+N)	AJU-255
500	CONTINUE	AJU-256
	ITIP=0	AJU-257
	NET=0	AJU-258
	NLET=0	AJU-259
	NGET=0	AJU-260
	INDS=0	AJU-261
C		AJU-262
C	TRANSFORMACION DE LOS CODIGOS DE LAS RESTRICCIONES PARA A AJUSTARLOS	AJU-263
C	A UNO DE LOS TIPOS DE LA PROGRAMACION LINEAL	AJU-264
C		AJU-265
	DO 550 I=1,K	AJU-266
	IF(CODE(I).EQ.2) CODE(I)=1	AJU-267
	IF(CODE(I).EQ.4) CODE(I)=2	AJU-268
550	CONTINUE	AJU-269
	DO 600 I=1,LL-1	AJU-270
	IF(CODE(I).EQ.1) NGET=NGET+1	AJU-271
	IF(CODE(I).EQ.2) NET=NET+1	AJU-272
600	CONTINUE	AJU-273
C		AJU-274
C	** LLAMADA A LA SUBROUTINA QUE APLICA EL ALGORITMO DE PROGRAMACION LI-	AJU-275
C	NEAL **	AJU-276
C		AJU-277
	CALL LIS(ISAL,N,LL-1,1,NET,NGET,NLET,ITIP,CODE,B,NON,C,XB,	AJU-278
	IR(1,1),IR(1,61),Z,INDS)	AJU-279
	INDS=1	AJU-280
	ITIP=0	AJU-281
	NLET=0	AJU-282
	NET=0	AJU-283
	NGET=0	AJU-284
	DO 650 I=LL,K	AJU-285
	IF(CODE(I).EQ.1) NGET=NGET+1	AJU-286
	IF(CODE(I).EQ.2) NET=NET+1	AJU-287
650	CONTINUE	AJU-288
C		AJU-289
C	** LLAMADA A LA SUBROUTINA QUE APLICA EL ALGORITMO DE PROGRAMACION LI-	AJU-290
C	NEAL **	AJU-291
C		AJU-292
	CALL LIS(ISAL,N,K,LL,NET,NGET,NLET,ITIP,CODE,B,NON,C,XB,IR(1,1),	AJU-293
	IR(1,61),Z,INDS)	AJU-294
	RETURN	AJU-295

	END	AJU-296
C		ESQ-297
C		ESQ-298
	SUBROUTINE ESQ(ISA,NA,NVH,NVV,XL,YL,NUNE,IPIL,ISTA,BA,BB,X,Y)	ESQ-299
	*****	ESQ-300
C		ESQ-301
C	SUBROUTINA QUE TRAZA EL ESQUEMA YA DIMENSIONADO	ESQ-302
C		ESQ-303
	ENA BB(60,40),BA(60,40)	ESQ-304
	INTEGER VI,VF,VE,VA,WI,WF,WE,WA	ESQ-305
	DIMENSION X(60),Y(60),XL(60),YL(60),NUNE(60),ISTA(60),IPIL(60)	ESQ-306
	DIMENSION XG(60),YG(60),IAUX(60),JAU(60)	ESQ-307
C		ESQ-308
C	PUESTA A CERO	ESQ-309
C		ESQ-310
	DO 10 I=1,NA	ESQ-311
	NUNE(I)=0	ESQ-312
	ISTA(I)=0	ESQ-313
	IAUX(I)=0	ESQ-314
	JAU(I)=0	ESQ-315
	XG(I)=0	ESQ-316
	YG(I)=0	ESQ-317
	10 CONTINUE	ESQ-318
	DO 20 I=1,NVH	ESQ-319
	IPIL(I)=0	ESQ-320
	20 CONTINUE	ESQ-321
C		ESQ-322
C	HACENOS TODOS LOS TERMINOS DE LA MATRIZ DE INCIDENCIA HORIZONTAL: 0,1,	ESQ-323
C	HACENOS TODOS LOS TERMINOS DE LA MATRIZ DE INCIDENCIA VERTICAL: 0,1,-1	ESQ-324
C		ESQ-325
	45 DO 100 I=1,NA	ESQ-326
	DO 50 J=1,NVH	ESQ-327
	IF(BB(I,J).EQ.0) GO TO 50	ESQ-328
	BB(I,J)=BB(I,J)/ABS(BB(I,J))	ESQ-329
	50 CONTINUE	ESQ-330
	DO 80 J=1,NVV	ESQ-331
	IF(BA(I,J).EQ.0) GO TO 80	ESQ-332
	BA(I,J)=BA(I,J)/ABS(BA(I,J))	ESQ-333
	80 CONTINUE	ESQ-334
	100 CONTINUE	ESQ-335
C		ESQ-336
C	NUMERACION DE LAS ARISTAS Y OBTENCION DE LAS NUEVAS COORDENADAS DEL C.	ESQ-337
C		ESQ-338
	DO 120 J=1,NVV	ESQ-339
	IF(BA(NA,J).EQ.-1) WI=J	ESQ-340
	IF(BA(NA,J).EQ.1) WF=J	ESQ-341
	IF(WI.NE.0.AND.WF.NE.0) GO TO 130	ESQ-342
	120 CONTINUE	ESQ-343
	130 DO 150 J=1,NVH	ESQ-344
	IF(BB(NA,J).EQ.1) VI=J	ESQ-345
	IF(BB(NA,J).EQ.-1) VF=J	ESQ-346
	IF(VI.NE.0.AND.VF.NE.0) GO TO 160	ESQ-347
	150 CONTINUE	ESQ-348
	160 IORT=0	ESQ-349
	KORT=1	ESQ-350
	VE=VI	ESQ-351
	IPIL(KORT)=VE	ESQ-352
	JORT=0	ESQ-353
	IA1=0	ESQ-354
	IA2=0	ESQ-355

260	II=0	ES0-356
	KNIN=100.	ES0-357
	DO 300 I=1,NA-1	ES0-358
	IF(BB(I,VE).NE.-1.OR.HUNE(I).NE.0) GO TO 300	ES0-359
	IF(K(I).GT.KNIN) GO TO 300	ES0-360
	II=I	ES0-361
	KNIN=K(II)	ES0-362
300	CONTINUE	ES0-363
	IF(II.EQ.0) GO TO 440	ES0-364
	DO 320 J=1,NVH	ES0-365
	IF(BB(II,J).EQ.1) GO TO 325	ES0-366
320	CONTINUE	ES0-367
325	VA=J	ES0-368
	DO 330 J=1,NVV	ES0-369
	IF(BA(II,J).EQ.-1) ME=J	ES0-370
	IF(BA(II,J).EQ.1) MA=J	ES0-371
330	CONTINUE	ES0-372
	IF(VE.EQ.VI) IA1=II	ES0-373
	IF(VE.NE.VI) IA1=0	ES0-374
	IF(MA.EQ.MI) IA2=II	ES0-375
	IF(MA.NE.MI) IA2=0	ES0-376
	JONT=JONT+1	ES0-377
	HUNE(II)=JONT	ES0-378
	IAUX(II)=IA1	ES0-379
	J AUX(II)=IA2	ES0-380
	IF(JONT.EQ.NA-1) GO TO 450	ES0-381
	IOAT=IOAT+1	ES0-382
	ISTA(IOAT)=II	ES0-383
	VE=VA	ES0-384
	IF(VE.EQ.VF) GO TO 445	ES0-385
	KONT=KONT+1	ES0-386
	IPIL(KONT)=VE	ES0-387
	GO TO 260	ES0-388
440	IPIL(KONT)=0	ES0-389
	KONT=KONT-1	ES0-390
445	VE=IPIL(KONT)	ES0-391
	GO TO 260	ES0-392
450	CALL COP(MA,NVV,X,Y,HUNE,IAUX,BA)	ES0-393
	CALL COP(MA,NVH,Y,X,HUNE,J AUX,BB)	ES0-394
	DO 520 K=1,10	ES0-395
	DO 500 I=1,NA-1	ES0-396
	DO 490 J=1,NA-1	ES0-397
	IF(HUNE(J).NE.I) GO TO 490	ES0-398
	II=IAUX(J)	ES0-399
	IF(II.NE.J) GO TO 460	ES0-400
	IF(YG(J).EQ.0) YG(J)=YL(J)/2.	ES0-401
	GO TO 470	ES0-402
460	IF(YG(II).EQ.0.OR.YG(J).NE.0) GO TO 470	ES0-403
	IF(Y(II).GT.Y(J)) YG(J)=YG(II)-YL(II)/2.-YL(J)/2.	ES0-404
	IF(Y(II).LT.Y(J)) YG(J)=YG(II)+YL(II)/2.+YL(J)/2.	ES0-405
470	JJ=JAUX(J)	ES0-406
	IF(JJ.NE.J) GO TO 480	ES0-407
	IF(XG(J).EQ.0) XG(J)=XL(J)/2.	ES0-408
	GO TO 500	ES0-409
480	IF(XG(JJ).EQ.0.OR.XG(J).NE.0) GO TO 500	ES0-410
	IF(X(JJ).GT.X(J)) XG(J)=XG(JJ)-XL(JJ)/2.-XL(J)/2.	ES0-411
	IF(X(JJ).LT.X(J)) XG(J)=XG(JJ)+XL(JJ)/2.+XL(J)/2.	ES0-412
490	CONTINUE	ES0-413
500	CONTINUE	ES0-414
	DO 510 I=1,NA-1	ES0-415
	IF(XG(I).EQ.0.OR.YG(I).EQ.0) GO TO 520	ES0-416

510 CONTINUE	ESQ-417
GO TO 530	ESQ-418
520 CONTINUE	ESQ-419
530 WRITE(18AL,3001)	ESQ-420
CALL PLTLU(10)	ESQ-421
CALL PLOT(0.,0.,-3)	ESQ-422
DO 600 I=1,NA-1	ESQ-423
RI=I+4	ESQ-424
CALL HUMB(KG(I),YG(I),0.15,RI,0.,-1)	ESQ-425
Z1=KG(I)+XL(I)/2.	ESQ-426
Z2=YG(I)+YL(I)/2.	ESQ-427
Z3=KG(I)-XL(I)/2.	ESQ-428
Z4=YG(I)-YL(I)/2.	ESQ-429
CALL PLOT(Z1,Z2,3)	ESQ-430
CALL PLOT(Z3,Z2,2)	ESQ-431
CALL PLOT(Z3,Z4,2)	ESQ-432
CALL PLOT(Z1,Z4,2)	ESQ-433
CALL PLOT(Z1,Z2,2)	ESQ-434
600 CONTINUE	ESQ-435
CALL PLOT(0.,0.,3)	ESQ-436
STDP	ESQ-437
C	ESQ-438
C FORMAT3S	ESQ-439
C	ESQ-440
3001 FORMAT(//1X'MENSAJE 1:SR:1 SE DIBUJA EN EL PLOTTER')	ESQ-441
END	ESQ-442
C	COP-443
C	COP-444
SUBROUTINE COP(NA,NV,X,Y,NUNE,IAUX,BA)	COP-445
*****	COP-446
C	COP-447
C	COP-448
C	COP-449
C	COP-450
SUSRUTINA PARA COMPLETAR PARAMETROS PARA EL TRAZADO POSTERIOR	COP-451
DIMENSION BA(60,40)	COP-452
DIMENSION X(60),Y(60),IAUX(60),NUNE(60)	COP-453
DO 250 K=1,10	COP-454
DO 230 L=1,NA-1	COP-455
DO 200 I=1,NA-1	COP-456
IF(NUNE(I).NE.L) GO TO 200	COP-457
IF(IAUX(I).NE.0) GO TO 230	COP-458
DO 50 J=1,NV	COP-459
IF(BA(I,J).EQ.1) VA=J	COP-460
IF(BA(I,J).EQ.-1) VE=J	COP-461
50 CONTINUE	COP-462
KK=0	COP-463
DNIN=1.000.	COP-464
DO 100 J=1,NA-1	COP-465
IF(J.EQ.1) GO TO 100	COP-466
IF(Y(J).GT.Y(I)) GO TO 100	COP-467
IF(BA(J,VA).NE.1.AND.BA(J,VE).NE.-1) GO TO 100.	COP-468
IF((X(J)-X(I))**2+(Y(J)-Y(I))**2.GE.DNIN) GO TO 100	COP-469
KK=J	COP-470
DNIN=(X(J)-X(I))**2+(Y(J)-Y(I))**2	COP-471
100 CONTINUE	COP-472
IF(KK.NE.0) GO TO 180	COP-473
DO 150 J=1,NA-1	COP-474
IF(J.EQ.1) GO TO 150	COP-475
IF(BA(J,VA).NE.-1.AND.BA(J,VE).NE.1) GO TO 150	COP-476
IF((X(J)-X(I))**2+(Y(J)-Y(I))**2.GE.DNIN) GO TO 150	
IF(IAUX(J).EQ.0.OR. IAUX(J).EQ.1) GO TO 150	

```
IF(Y(IAUX(J)).GT.Y(I)) GO TO 150
KK=J
DNIN=(X(J)-X(I))*2+(Y(J)-Y(I))*2
150 CONTINUE
IAUX(I)=IAUX(KK)
GO TO 230
180 IAUX(I)=KK
GO TO 230
200 CONTINUE
230 CONTINUE
DO 240 I=1,NA-1
IF(IAUX(I).EQ.0) GO TO 250
240 CONTINUE
GO TO 260
250 CONTINUE
260 RETURN
END
```

```
COP-477
COP-478
COP-479
COP-480
COP-481
COP-482
COP-483
COP-484
COP-485
COP-486
COP-487
COP-488
COP-489
COP-490
COP-491
COP-492
COP-493
COP-494
```

A.II.2.4. Ejemplos

MENSAJE 11ED1: TITULO EL PROBLEMA
 EJEMPLO 1 (DIMENSIONADO)

 MENSAJE 11ED1: NUMERO DE VERTICES VERTICALES Y HORIZONTALES
 NUMERO DE VERTICES VERTICALES 6
 NUMERO DE VERTICES HORIZONTALES ... 8
 MENSAJE 11ED1: COORDENADAS DEL CENTRO GEOMETRICO DE LOS ESPACIOS.
 SE HAYA A PARTIR DEL ESQUEMA ADIMENSIONAL, DANDO UNAS DIMENSIONES
 ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARAN PARTIR DEL 5.
 AL FINAL DE LA LISTA INTRODUCE: 0

ESP	XC	YC
5	8.00	11.00
6	7.50	8.00
7	3.00	9.00
8	9.50	7.00
9	9.50	9.00
10	5.00	4.50
11	2.00	5.00
12	2.00	2.00
13	8.00	5.00
14	8.00	2.00
15	4.50	1.50
16	5.50	1.50

MENSAJE 11ED1: CONDICIONES DE ACCESIBILIDAD ENTRE LOCALES
 SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES A EL.
 AL FINAL DE LA LISTA INTRODUCE: 0

L	L1	L2	L3	L4	L5
-	-	-	-	-	-
6	9	8			
10	7	11	12	14	13
5	6	7			

MENSAJE 11ED1: TIPO DE PROBLEMA: 1 ... LINEAL / 2 ... NO LINEAL

PROBLEMA NO LINEAL

MENSAJE 11ED1: TIPO DE FUNCION OBJETIVO:
 PROBLEMA LINEAL 01 MINIMIZACION DE LA LONGITUD Y ANCHURA
 PROBLEMA NO LINEAL 01 MINIMIZACION DE UNA EXPRESION
 01 MINIMIZACION DE LA SUPERFICIE
 11 MINIMIZACION DE UNA EXPRESION

FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE

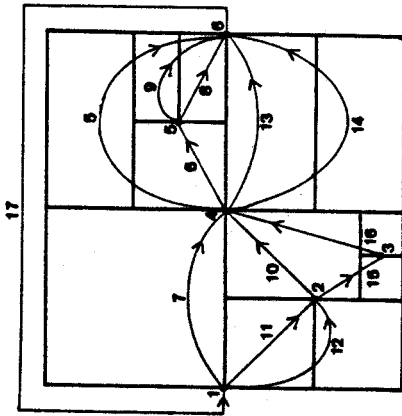
MENSAJE (IED): MATRIZ DE INCIDENCIA VERTICAL

LA NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 5.
LA ARISTA QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:

- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA ANCHURA
- COTA INFERIOR DE LA SUPERFICIE

A	VO	VE	CI
5	4	6	1.1
6	4	5	2.0
7	1	4	3.0
8	5	6	1.0
9	5	6	1.0
10	2	4	1.0
11	1	2	1.5
12	1	2	2.0
13	4	6	2.0
14	4	6	3.0
15	2	3	.7
16	3	4	.7
17	6	1	0.0



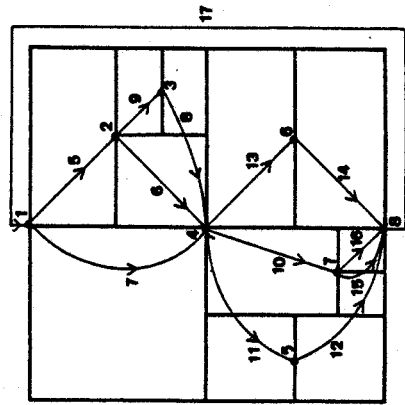
MENSAJE (IED): MATRIZ DE INCIDENCIA HORIZONTAL

IGUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.
LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:

- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA LONGITUD

A	VO	VE	CI
5	1	2	1.1
6	2	4	2.0
7	1	4	3.0
8	3	4	1.0
9	2	2	1.0
10	4	7	1.0
11	4	5	1.5
12	5	8	3.0
13	4	6	2.0
14	6	8	3.0
15	7	8	.7



16 7 8 7
17 6 1 0.0

----- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONITORNO SON: -----

KC (5) = 3.28 VC (5) = 1.10
 KC (6) = 2.28 VC (6) = 2.50
 KC (7) = 4.35 VC (7) = 3.60
 KC (8) = 1.00 VC (8) = 1.26
 KC (9) = 1.00 VC (9) = 1.24
 KC (10) = 1.40 VC (10) = 3.29
 KC (11) = 3.00 VC (11) = 1.63
 KC (12) = 3.00 VC (12) = 3.20
 KC (13) = 3.33 VC (13) = 2.00
 KC (14) = 3.33 VC (14) = 2.00
 KC (15) = .70 VC (15) = 1.72
 KC (16) = .70 VC (16) = 1.72
 KC (17) = 7.73 VC (17) = 0.60

 MENSAJE (18): SE DIBUJA EN EL PLOTTER

MENSAJE (19): SI SE DESEA AJUSTAR LOS RESULTADOS INTRODUCIR 1
 SI QUIERES TERMINAR INTRODUCIR 0

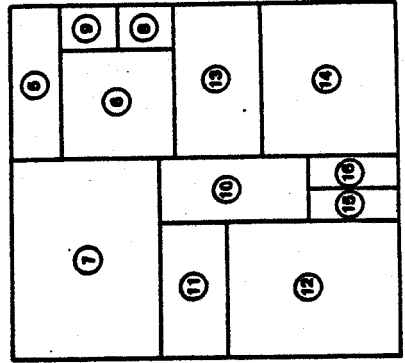
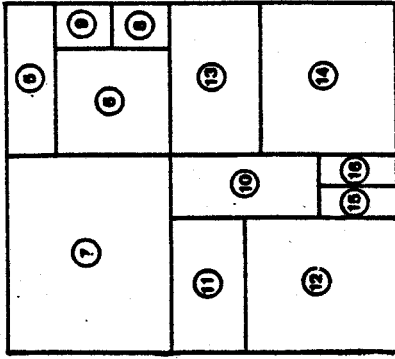
----- MENSAJE (10): VALORACION RELATIVA PARA CADA ESPACIO.

LOS VALORES ASIGNADOS SERAN PROPORCIONALES A LA IMPORTANCIA QUE
 SE ATRIBUYE A LA SUPERFICIE DE CADA ESPACIO.

V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
 1.0 5.0 0.0 1.0 1.0 1.0 5.0 9.0 9.0 9.0 5.0 5.0

----- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONITORNO SON: -----

KC (5) = 3.28 VC (5) = 1.10
 KC (6) = 2.28 VC (6) = 2.50
 KC (7) = 4.35 VC (7) = 3.20
 KC (8) = 1.00 VC (8) = 1.26
 KC (9) = 1.00 VC (9) = 1.24
 KC (10) = 1.40 VC (10) = 3.32
 KC (11) = 3.00 VC (11) = 1.50
 KC (12) = 3.00 VC (12) = 2.02
 KC (13) = 3.33 VC (13) = 2.00
 KC (14) = 3.33 VC (14) = 2.00
 KC (15) = .70 VC (15) = 2.00
 KC (16) = .70 VC (16) = 2.00
 KC (17) = 7.73 VC (17) = 0.60



MENSAJE (IED) TITULO EL PROBLEMA

EJEMPLO 2 (DIMENSIONADO)

NUMERO DE VERTICES VERTICALES Y HORIZONTALES

NUMERO DE VERTICES VERTICALES 7
NUMERO DE VERTICES HORIZONTALES 9

NUMERO DE VERTICES VERTICALES Y HORIZONTALES

SE HAYA A PARTIR DEL ESQUEMA ADIMENSIONAL, DADO UNAS DIMENSIONES
ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARAN PARTIR DEL 5.
AL FINAL DE LA LISTA INTRODUCIR: *

ESP.	XG	YG
5	2.00	12.00
6	6.00	11.00
7	9.00	11.00
8	1.00	9.00
9	5.00	7.00
10	9.00	6.00
11	3.00	6.00
12	1.00	6.00
13	1.00	3.00
14	6.00	3.00
15	9.00	2.00
16	2.00	1.00
17	5.00	.50

NUMERO DE VERTICES VERTICALES Y HORIZONTALES

SE FIJA UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES A EL.
AL FINAL DE LA LISTA INTRODUCIR: *

L	L1	L2	L3	L4	L5
17	14				
16	11	15			
10	7	15			
7	6				
11	5	6	12	16	9

NUMERO DE VERTICES VERTICALES Y HORIZONTALES

PROBLEMA NO LINEAL

NUMERO DE VERTICES VERTICALES Y HORIZONTALES

PROBLEMA LINEAL 0: MINIMIZACION DE LA LONGITUD Y ANCHURA
1: MINIMIZACION DE UNA EXPRESION

PROBLEMA NO LINEAL ... 0: MINIMIZACION DE LA SUPERFICIE
 1: MINIMIZACION DE UNA EXPRESION

FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE

EMBAJAJE (1ED): MATRIZ DE INCIDENCIA VERTICAL

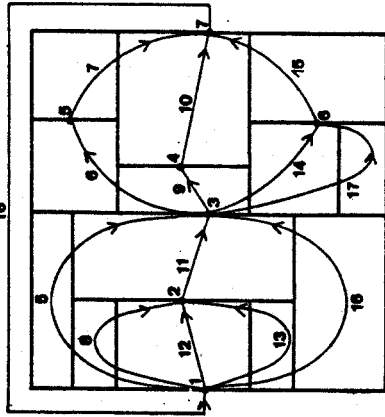
LA NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 3.
 LA ARISTA QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.

LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:

- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA ANCHURA
- COTA SUPERIOR DE LA ANCHURA

A	VO	VE	CI	CU
5	1	2	2.0	10.0
6	3	5	2.0	6.0
7	5	7	2.0	12.0
8	1	2	1.0	9.0
9	3	4	1.6	3.0
10	4	7	1.6	6.0
11	2	3	1.0	0.0
12	1	2	2.0	6.0
13	1	2	1.0	0.0
14	3	6	1.1	1.5
15	6	7	2.0	10.0
16	1	2	2.0	10.0
17	3	6	1.1	1.5
18	7	1	5.0	0.0



EMBAJAJE (1ED): MATRIZ DE INCIDENCIA HORIZONTAL

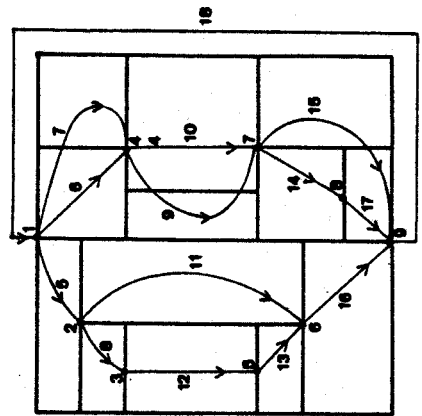
IGUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.

LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:

- ARISTA
- VERTICE ORIGEN
- VERTICE EXTREMO
- COTA INFERIOR DE LA LONGITUD

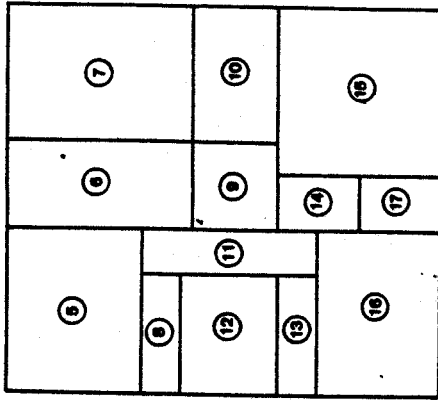
A	VO	VE	CI	CL
5	1	2	2.0	
6	1	4	2.0	
7	1	4	2.0	
8	2	3	1.6	
9	4	7	1.6	
10	4	7	1.6	
11	2	6	1.0	



12 3 5 2.0
 13 5 6 1.6
 14 7 8 1.1
 15 7 9 2.0
 16 6 9 2.0
 17 8 9 1.1
 18 9 1 7.0

----- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONTORNO SON: -----

K(5)= 3.70 V(5)= 2.97
 K(6)= 2.90 V(6)= 4.10
 K(7)= 2.90 V(7)= 4.16
 K(8)= 2.70 V(8)= 1.95
 K(9)= 1.88 V(9)= 1.98
 K(10)= 3.12 V(10)= 1.92
 K(11)= 1.90 V(11)= 3.95
 K(12)= 2.70 V(12)= 2.20
 K(13)= 2.70 V(13)= 1.95
 K(14)= 1.24 V(14)= 1.97
 K(15)= 3.76 V(15)= 2.79
 K(16)= 3.70 V(16)= 2.79
 K(17)= 1.24 V(17)= 1.95
 K(18)= 8.70 V(18)= 9.46



----- MENSAJE (18): SE DIBUJA EN EL PLOTTER

MENSAJE (12): SI SE DESEA AJUSTAR LOS RESULTADOS INTRODUCIR 1
 SI QUIERES TERMINAR INTRODUCIR 0

Mensaje 11ED: TITULO DEL PROBLEMA

EJEMPLO 3 (DIMENSIONADO)

Mensaje 11ED: NUMERO DE VERTICES VERTICALES Y HORIZONTALES

NUMERO DE VERTICES VERTICALES 8
 NUMERO DE VERTICES HORIZONTALES . . . 8

Mensaje 11ED: COORDENADAS DEL CENTRO GEOMETRICO DE LOS ESPACIOS.

SE HARA A PARTIR DEL ESQUEMA ADIMENSIONAL, DANDO UNAS DIMENSIONES
 ARBITRARIAS A CADA ESPACIO. ESTOS SE NUMERARAN PARTIR DEL 5.
 AL FINAL DE LA LISTA INTRODUCIR 0

ESP	KC	YC
5	5.00	1.00
6	8.00	2.00
7	7.00	5.00
8	3.00	4.00
9	7.00	7.00
10	5.00	9.00
11	6.00	9.00
12	8.00	12.00
13	2.00	11.00
14	3.00	1.00
15	1.00	1.00
16	5.00	12.00
17	5.00	11.00

Mensaje 11ED: CONDICIONES DE ACCESIBILIDAD ENTRE LOCALES

SE PONE UN LOCAL Y SE INDICA CUALES DEBEN SER ACCESIBLES A EL.
 AL FINAL DE LA LISTA INTRODUCIR 0

L	L1	L2	L3	L4	L5
5	0	0	14		
6	7	11			
14	9	10	12	17	
14	13				

Mensaje 11ED: TIPO DE PROBLEMA: 1 ... LINEAL / 2 ... NO LINEAL

PROBLEMA NO LINEAL

Mensaje 11ED: TIPO DE FUNCION OBJETIVO:

PROBLEMA LINEAL 0: MINIMIZACION DE LA LONGITUD Y ANCHURA
 1: MINIMIZACION DE UNA EXPRESION

PROBLEMA NO LINEAL ... 0: MINIMIZACION DE LA SUPERFICIE
 1: MINIMIZACION DE UNA EXPRESION

FUNCION OBJETIVO: MINIMIZACION DE LA SUPERFICIE

Mensaje 11ED1: MATRIZ DE INCIDENCIA VERTICAL

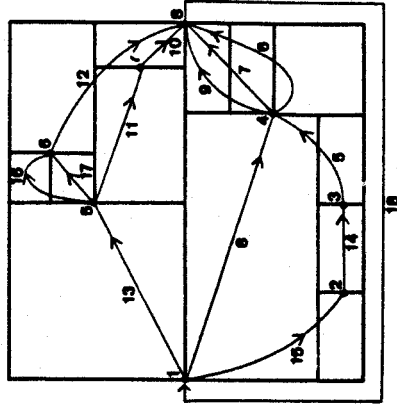
LA NUMERACION DE LOS ESPACIOS SE HARA A PARTIR DEL 9.
 LA ARISTA QUE REPRESENTA AL CONTORNO SE NUMERARA EN ULTIMO LUGAR.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.

LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:

- ARISTA
- VERTICE BRINER
- VERTICE EXTREMO
- COTA INFERIOR DE LA ANCHURA
- COTA INFERIOR DE LA SUPERFICIE

A	VO	VE	CI	CC
3	1	1.1	1.5	
4	8	2.0	6.0	
7	4	3.0	10.0	
8	1	4	12.0	
9	4	8	2.0	6.0
10	7	8	1.5	0.0
11	5	7	1.1	0.0
12	6	9	3.0	10.0
13	1	5	3.0	10.0
14	2	3	1.5	0.0
15	1	2	1.5	0.0
16	5	6	1.0	0.0
17	5	6	1.0	0.0
18	8	1	0.0	0.0



Mensaje 11ED1: MATRIZ DE INCIDENCIA HORIZONTAL

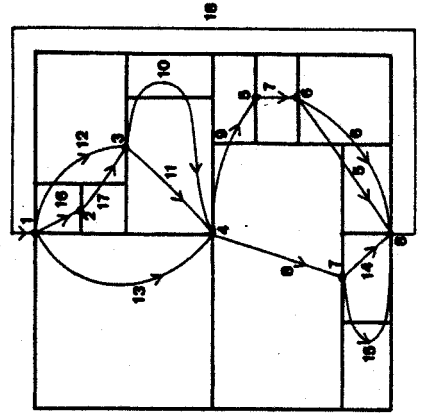
IGUAL CRITERIO DE NUMERACION QUE EN LA VERTICAL.

SE RECOGEN AQUI LOS REQUISITOS METRICO-GEOMETRICOS.

LOS DATOS SE DAN EN EL SIGUIENTE ORDEN:

- ARISTA
- VERTICE BRINER
- VERTICE EXTREMO
- COTA INFERIOR DE LA LONGITUD

A	VO	VE	CI
3	7	8	1.1
6	6	8	2.0
7	5	6	3.0
8	4	7	3.0
9	4	5	2.0
10	3	4	1.5
11	3	4	2.0
12	1	3	3.0



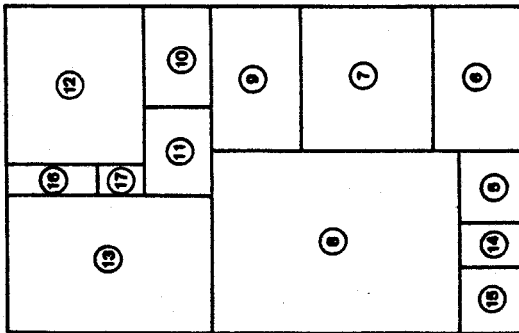
13 1 4 2.0
 14 7 8 1.0
 15 7 8 1.5
 16 1 2 .6
 17 2 3 .6
 18 8 1 0.0

----- LOS RESULTADOS OBTENIDOS PARA CADA ESPACIO Y EL CONTORNO SON: -----

K(5)= 1.61 Y(5)= 1.50
 K(6)= 3.24 Y(6)= 2.06
 K(7)= 3.24 Y(7)= 3.00
 K(8)= 4.11 Y(8)= 3.56
 K(9)= 3.24 Y(9)= 2.00
 K(10)= 2.24 Y(10)= 1.50
 K(11)= 2.00 Y(11)= 1.50
 K(12)= 3.55 Y(12)= 3.10
 K(13)= 3.11 Y(13)= 4.60
 K(14)= 1.00 Y(14)= 1.50
 K(15)= 1.50 Y(15)= 1.50
 K(16)= .69 Y(16)= 2.05
 K(17)= .59 Y(17)= 1.05
 K(18)= 7.35 Y(18)= 11.66

MENSAJE 'ERR' SE DIBUJA EN EL PLOTTER

MENSAJE 'ED' SI SE DESEA AJUSTAR LOS RESULTADOS INTRODUCIR 1
 SI QUIERES TERMINAR INTRODUCIR 0



1. Orden alfabético de autores

Alexander, C. ENSAYO SOBRE LA SÍNTESIS DE LA FORMA. Ed. Infinito, Buenos Aires, 1976, (4ª ed.). Trad. de NOTE 1966
ON THE SYNTHESIS OF FORM. Harvard University - Press, Cambridge (Mass.), 1966.

Avondo-Bodino, G. "Graph Theory and operations research". En 1979
APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 223-253.

Baglivo, J.A. y Graver J.E. INCIDENCE AND SYMMETRY IN DESIGN 1983
AND ARCHITECTURE. Cambridge University Press, col. Cambridge Urban and Architectural Studies, vol. 7, Cambridge, 1983.

Berge, C. GRAPHERS, Bordas, 1983 (3ª ed.), 1ª ed. Dunod, 1970. 1970

Bloch, C.J. y Krishnamurti. "The counting of rectangular 1978
dissections". En ENVIRONMENT AND PLANNING B, Vol. 5, 1978, pp. 207-214.

-
- Broadbent, G. DISEÑO ARQUITECTÓNICO, ARQUITECTURA Y CIENCIAS
1974 HUMANAS. Ed. Gustavo Gili, col. Arquitectura/
Perspectivas, Barcelona, 1976. Trad. de DESIGN IN
ARCHITECTURE. ARCHITECTURE AND THE HUMAN SCIENCES.
John Wiley & Sons Ltd., Londres 1974.
- Brooks, R.L. ed. al. "The dissection of rectangles into squa
1940 res". En DUKE MATHEMATICAL JOURNAL, Vol. 7, 1940,
pp. 312-341.
- Bruno, J. ed. al. "A new planarity test based on 3-Connecti-
1970 vity". En IEEE. TRANSACTIONS ON CIRCUIT THEORY,
Vol. 17, n°2, mayo, 1970, pp. 197-206.
- Bryant, P.R. "Graph Theory and electrical networks". En
1979 APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y L.W.
Beineke ed.). Academic Press, Londres, 1979, pp.
81-119.
- Cattermole, K.W. "Graph Theory and Communications Networks".
1979 En APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y
L.W. Beineke ed.). Academic Press, Londres, 1979,
pp. 17-57.
- Cea, J. OPTIMISATION THEORIE ET ALGORITHMES. Ed. Dunod, col.
1971 Méthodes Mathématiques de l'informatique, Vol. 2,
París, 1971.
- Charalambous, C. "Non linear least pth. optimization and non
1977 linear programming". En MATHEMATICAL PROGRAMMING,
Vol. 12, 1977, pp. 195-225.
- Cousin, J. "Topological organization of Architectural space".
1970 Architectural Design, Vol. 10, Octubre, 1970, pp.
491-493.
-

-
- Earl, C.F. "A note on the generation of rectangular dissections". En ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 241-246.
- Earl, C.F. y March, L.J. "Architectural applications of graph theory". En APPLICATIONS OF GRAPH THEORY, (R.J. - Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 327-355.
- Eastman, C.M. "Representations for space planning". En COMMUNICATIONS OF THE ACM, Vol. 13, n°4, Abril, 1970, pp. 242-250.
- Eastman, C.M. "Preliminary Report on a system for general space planning". En COMMUNICATIONS OF THE ACM, Vol. 15, n°2, Febrero, 1972, pp. 76-87.
- Eastman, C.M. "Automated Space Planning". En ARTIFICIAL INTELLIGENCE, Vol. 4, 1973, pp. 41-64.
- Fisher, G.J. "Computer recognition and extraction of planar graphs from the incidence matrix". En IEEE TRANSACTIONS ON CIRCUIT THEORY, Vol. 13, n°2, Junio, 1966, pp. 154-163.
- Gero, J.S. "Note on 'Synthesis and optimization of small rectangular floor plans' of Mitchell, Steadman and Liggett". En ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 81-88.
- Gill, P.E. ed al. PRACTICAL OPTIMIZATION. Academic Press, Londres, 1981.
- Grooms, H.R. "Algorithm for matrix bandwidth reduction". En JOURNAL OF THE STRUCTURAL DIVISION, Enero, 1972, pp. 203-214.
-

-
- Harary, F. ed al. "On enumerating certain design problems in
1978 terms of bicoloured graphs with no isolates". En
ENVIRONMENT AND PLANNING B, Vol. 5, 1978, pp. 31-
43.
- Hock, W. y Schittokowski, K. TEST EXAMPLES FOR NON LINEAR
1981 PROGRAMMING CODES. Ed. Board, col. Lecture Notes
in Economics and Mathematical Systems, Vol. 187,
Nueva York, 1981.
- Hopcroft, J. y Tarjan, R. "Efficient algoritms for graph ma-
1973,a nipulation". En COMMUNICATIONS OF THE ACM, Vol.
16, n°6, Junio 1973, pp. 372-378.
- Hopcroft, J. y Tarjan, R. "Dividing a graph into triconnec-
1973,b ted components". En SIAM J. COMPUT., Vol. 2, 1973,
pp. 135-158.
- Hopcroft, J. y Tarjan, R. "Efficient Planarity Testing". En
1974 JOURNAL OF THE ASSOCIATION FOR COMPUTING MACHINE-
RY, Vol. 21, n°4, Octubre, 1974, pp. 549-568.
- Hutton, G. y Roston, M. COMPUTER PROGRAMS FOR THE BUILDING
1979 INDUSTRY. Architectural Press. Londres, 1979, -
(2a ed.).
- Kaufmann, A. y Cruon, R. LA PROGRAMACIÓN DINÁMICA. Cía. Ed.
1967 Continental, México, 1967, Trad, de LA PROGRAMMA-
TION DYNAMIQUE. Ed. Dunod, París, 1964.
- Kaufmann, A. PUNTOS Y FLECHAS. TEORÍA DE LOS GRAFOS. Marcom-
1976 bo S.A., Boixerau Ed., Barcelona, 1976. Trad. de -
DES POINTS ET DES FLECHES... LA THEORIE DES GRA-
PHES, Ed. Dunod, París.
-

-
- Korf, R.E. "A shape independent theory of space allocation".
1977 En ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp.
 37-50.
- Krejcirik, M. "Computer-aided plant layout". En COMPUTER
1969 AIDED DESIGN, Otoño, 1969, pp. 7-19.
- Krishnamurti, R. y O'N Roe, P.H. "Algorithmic aspects of -
1978 plan generation and enumeration". En ENVIRONMENT
 AND PLANNING B, Vol. 5, 1978, pp. 157-177.
- Krishnamurti, R. "3-rectangulations: an algorithm to genera-
1979 te box packings". En ENVIRONMENT AND PLANNING B,
 Vol. 6, 1979, pp. 331-352.
- Kuratowski, K. "Sur le problème des courbes gauches en topo-
1930 logie". En FUNDAMENTA MATH., Vol. 15, 1930, pp.
 271-283.
- Kuratowski, K. INTRODUCCIÓN A LA TEORÍA DE CONJUNTOS Y A LA
1961 TOPOLOGÍA. Ed. Vicens-Vives, Barcelona, 1966.
 Trad, de WSTEP DO TEORII MNOGOSCI I TOPOLOGII,
 Panstowowe Wyda Wnictow Naukowe. Varsovia, 1961.
- Larrañeta, J. PROGRAMACIÓN LINEAL Y GRAFOS. Secretariado de
1977 Publicaciones de la Universidad de Sevilla, Sevi-
 lla, 1977.
- Levin, P.H. "Use of graphs to decide the optimun layout of -
1964 buildings". En THE ARCHITECT'S JOURNAL INFORMA-
 TION, Vol. 7, Octubre 1964, pp. 809-815.
- March, L. y Steadman, P. THE GEOMETRY OF ENVIRONMENT, RIBA -
1974 Publications Ltd, Londres, 1971. Trad. al italia-
 no como LA GEOMETRIA DELL'AMBIENTE: UNA INTRODUC-
-

ZIONE ALLA ORGANIZZAZIONE SPAZIALE NELLA PROGETTA
ZIONE. Gabrielle Mazzota ed. Milán, 1974.

March, L. y Earl, C.F. "On counting architectural plans". En
1977 ENVIRONMENT AND PLANNING B, Vol. 4, 1977, pp. 57-
80.

Mitchell, W.J. ed. al. "Synthesis and optimization of small
1976 rectangular floor plans". En ENVIRONMENT AND
PLANNING B, Vol. 3, 1976, pp. 37-70.

Mitchell, W.J. COMPUTER-AIDED ARCHITECTURAL DESIGNS, Mason/
1977 Charter Publishers, Nueva York, 1977.

Moucka, J. "The use of propositional calculus in architectu-
1979 ral design". En ENVIRONMENT AND PLANNING B, Vol.
6, 1979, pp. 263-268.

Negroponte, N. LA MACHINA PER L'ARCHITETTURA. Ed. Il Saggia
1970 tore, col. Struttura e forma urbana, Milán, 1974.
Trad. de THE ARCHITECTURE MACHINE, The M.I.T.
Press, Cambridge (Mass.), 1970.

Nielsen, F. "Flowgraphs". En APPLICATIONS OF GRAPH THEORY,
1979 (R.J. Wilson y L.W. Beineke ed.), Academic Press,
Londres, 1979, pp. 59-80.

Polak, E. COMPUTATIONAL METHODS IN OPTIMIZATION: A UNIFIED
1971 APPROACH, Academic Press, col. Mathematics in
science and engineering, Vol. 77, Londres, 1971.

Portlock, P.C. y Whitehead, B. "Three Dimensional Layout
1974 Planning". En BUILD SCIENCE, Vol. 9, 1974, pp.
45-53.

-
- Rao, S.S. OPTIMIZATION THEORY AND APPLICATIONS, Wiley Eastern Limited, Nueva Delhi, 1978.
- 1978
- Read, R. "Algorithms in graph theory". En APPLICATIONS OF GRAPH THEORY, (R.J. Wilson y L.W. Beineke ed.), Academic Press, Londres, 1979, pp. 381-417.
- 1979
- Ritzman, L.P. "The efficiency of computer algorithms for plant layout". En MANAGEMENT SCIENCE, Vol. 18, n°5, Enero, Part. I, 1972, pp. 240-247.
- 1972
- Rodrigues, J.S. "Node numbering optimization in structural analysis". En JOURNAL OF THE STRUCTURAL DIVISION, Febrero, 1975, pp. 361-376.
- 1975
- Scarano, R. PROGETTAZIONE PER OTTIMIZZAZIONE, Liguori, Editore, col. La società e la scienza, Vol. 6, Nápoles, 1979.
- 1979
- Seguí, J. "Investigación sobre procesos de diseño. 1- Planteamiento general". En PUBLICACIÓN DEL SEMINARIO DE ARQUITECTURA Y AUTOMÁTICA EN LA E.T.S.A. DE MADRID, Madrid, Junio, 1972.
- 1972,a
- Seguí, J. y Gutiérrez Guitián, M.V. "Ideas para el análisis y simulación de procesos artísticos". En PUBLICACIÓN DEL SEMINARIO DE ARQUITECTURA Y AUTOMÁTICA EN LA E.T.S.A. DE MADRID, Madrid, Junio, 1972.
- 1972,b
- Seguí, J. y Gutiérrez Guitián, M.V. "Investigación en procesos de diseño. Modelo operativo de formalización". EN BOLETÍN DEL CENTRO DE CÁLCULO DE LA UNIVERSIDAD DE MADRID, n°24, Madrid, Enero, 1974, pp. 1-38.
- 1974

-
- Sevilla, C. "Investigación sobre procesos de diseño. 2-Modelo de Organismo (I y II partes)". En PUBLICACIÓN DEL SEMINARIO DE ARQUITECTURA Y AUTOMÁTICA EN LA E.T.S.A. DE MADRID, Madrid, Junio, 1972.
- Sevilla, C. "Notas sobre la informatización y el diseño de Arquitectura". En INFORMES DE LA CONSTRUCCIÓN, nº337, 1981, pp. 5-20.
- Shaviv, E. y Gali, D. "A Model for space allocation in complex buildings: A computer graphic approach". En BUILD. INTERNATIONAL, Vol. 7, 1974, pp. 493-518.
- Turner, J.C. MATEMÁTICA MODERNA APLICADA, PROBABILIDADES, ESTADÍSTICA E INVESTIGACIÓN OPERATIVA, Alianza, - Edit., Madrid, 1974, Trad, de MODERN APPLIED MATHEMATICS: PROBABILITY-STATISTICS-OPERATIONAL RESEARCH, The English Universities Press Ltd., Londres, 1970.
- Tutte, W.T. "A census of planar maps". En CANADIAN JOURNAL OF MATHEMATICS, Vol. 15, 1962, pp.249-271.
- Tutte, W.T. "How to draw a graph". En PROC. LONDON MATH. SOC., Vol. 3, nº13, 1963, pp. 743-768.
- Varios autores. ACTAS DEL CONGRESO DE LA U.I.A., Madrid, 1975.
- Wilson, R.J. y Beineke L.W. APPLICATIONS OF GRAPH THEORY, Academic, Press, Londres, 1979.
- Wright, R. ed. al. "Constraint processing in design". En JOURNAL OF THE STRUCTURAL DIVISION, Enero, 1971, pp. 481-494.
-