

GENERACION AUTOMATICA DE DISTRIBUCIONES EN PLANTA: PARTE I

JACINTO CANIVELL

*Profesor del Departamento de
Expresión Gráfica Arquitectónica
de la Escuela Técnica Superior de
Arquitectura de la Universidad de Sevilla.*

RESUMEN

El presente trabajo pretende ser una contribución al campo de la distribución automática en planta asistida por ordenador. La elaboración del método de generación que se expone a continuación, ha exigido el desarrollo de varios algoritmos originales para el manejo de grafos, como por ejemplo, el que determina las caras de un grafo planar, o el que comprueba la planaridad de un grafo ante la introducción de nuevas aristas. Mediante este nuevo procedimiento es posible la representación gráfica de una distribución en planta que satisface las restricciones impuestas por el diseñador, aunque el grafo correspondiente a tales restricciones no sea planar máximo, únicamente ha de ser biconexo y planar.

SUMMARY

This paper means to be a contribution to the field of automatic space-planing. The method of generation we expose has required the development of several new algorithms, for instance the one that computes the faces of a planar graph and the one that tests the planarity of a graph when new sides are added. Our method makes possible the graphics representations of a floor plan layout satisfying the restrictions imposed by the designer when the graph associated to these restrictions is planar and biconected but not necessarily a planar maximal graph.

INTRODUCCION

En la década de los años setenta fueron numerosos los investigadores que mostraron su entusiasmo e interés por la aplicación del ordenador en el campo del diseño arquitectónico y auguraban su radical transformación. Sin embargo, este cambio no se ha producido hasta el momento; mientras, en otras disciplinas —ajenas al quehacer arquitectónico—, el ordenador se ha convertido en una poderosa herramienta de ayuda al diseño.

Las primeras aplicaciones prácticas aparecidas en el mercado para la utilización del ordenador en las tareas propias de la elaboración del proyecto arquitectónico, fueron principalmente dirigidas al campo del cálculo de estructuras, confección de mediciones

Recibido: Enero 1987

y presupuestos, dibujo, etc., no satisfaciendo —ninguno de estos— las condiciones que un programa de CAD para el diseño arquitectónico debe poseer.

Según Mitchell¹⁰ un programa de este tipo debe ser capaz de realizar las siguientes operaciones:

1. Almacenamiento, recuperación y manipulación de los datos que nos describen el objeto de diseño.
2. Generación automática de soluciones al problema de diseño.
3. Evaluación de las distintas soluciones generales según las condiciones, restricciones y objetivos impuestos.

En general, los programas de CAD que se ofrecen en el campo de la arquitectura son únicamente capaces de realizar las operaciones descritas en el primer punto aludido por Mitchell, dado que la mayor parte de estos han sido desarrollados, mediante la extrapolación, a partir de programas específicos de ingeniería. Por consiguiente, dichos programas no suelen ajustarse a las necesidades del diseño arquitectónico.

Así pues, nuestra labor de investigación se centra en el desarrollo de procedimientos englobados en los dos últimos puntos señalados por Mitchell, es decir, en los aspectos que más específicamente distinguen el proceso de diseño arquitectónico de otros procesos de diseño. Entre los problemas que pueden ser abordados en esta etapa se encuentran los de generación y optimización de esquemas de distribuciones espaciales en planta que satisfagan una serie de restricciones y requisitos impuestos por el diseñador.

GENERACION AUTOMATICA DE ESQUEMAS DE DISTRIBUCION

Estado actual del problema

En los últimos quince años han sido numerosas las propuestas de métodos para la generación automática —mediante ordenador— de distribuciones espaciales en planta que cumplan con una serie de restricciones impuestas “a priori”.

Estas restricciones, como proponen algunos autores —Levin⁸, Krejcirik⁷, Eastman², Shavid y Gali¹²— son de una naturaleza tal que permiten plantear el problema de la generación de soluciones como un problema de optimización, mientras que otros tratan este problema de forma radicalmente distinta: consideran que dichas restricciones sólo serán las debidas a la adyacencia entre locales, orientación, accesibilidad entre los mismos y limitaciones dimensionales. Así tratan el problema, entre otros, Grason¹³, Mitchell, Steadman y Ligget⁹, Flemming³, Gillear⁴ y Sendra¹¹.

Todos estos autores citados en el segundo grupo coinciden en diferenciar dos etapas muy claras en el proceso generador global: una primera etapa “adimensional” en la que se generan una o todas las soluciones de distribución en planta que cumplan con los requisitos impuestos, y una segunda etapa “dimensional” en la que se dimensionan los esquemas adimensionales generados.

De igual forma, casi todos los autores se basan en la teoría de grafos para expresar y procesar las condiciones de adyacencia y orientación entre los distintos locales que componen la planta. Además tratan con plantas de contorno rectangular constituidas

por locales o espacios también rectangulares. No obstante, esta limitación se puede superar mediante la introducción de rectángulos ficticios, consiguiéndose así plantas en forma de L,U,T u otras formas irregulares.

Descripción general del método desarrollado

El trabajo que se presenta supone un nuevo método de distribución espacial en planta de edificios asistido por el ordenador, mediante el cual es posible mejorar el análisis previo en el proceso de diseño; e indudablemente repercute en una mejora del resultado final.

Pasemos pues, a describir este método. Podríamos considerarlo dividido en las siguientes etapas:

1. Introducción de las condiciones iniciales. Se definen las relaciones más importantes de adyacencia y orientación entre los espacios componentes. Todos estos datos se estructuran según un grafo abstracto inicial.
2. Test de conectividad. Se comprueba si el grafo inicial es conexo. Si no lo fuera, se introducirían nuevas relaciones de adyacencia para lograr que lo sea.
3. Comprobación de la biconectividad. De igual forma que en la etapa anterior, si el grafo no es biconexo, se introducirán nuevas relaciones para que tenga esta propiedad.
4. Comprobación de la planaridad. Si el grafo no es planar, se sustituirán o eliminarán las adyacencias que lo impiden.
5. Generación de las caras. Una vez comprobada la planaridad del grafo de adyacencia se genera un posible conjunto de caras.
6. Obtención del grafo dual o del pseudo dual. Se comprueba si el grafo de adyacencia posee una única representación en el plano y al mismo tiempo se extrae su dual. Si el grafo no está triangulado, se obtiene además su pseudo dual. En esta etapa se ofrece la posibilidad de generar todas las representaciones planas del grafo de adyacencia, en el caso de que no fuese única.
7. Trazado del pseudo-dual o del dual. Si el grafo de adyacencia no es un grafo triangulado trazaremos su pseudo-dual, en caso contrario, representaremos su dual. Estos trazados serán los primeros esquemas de distribución en planta. Habría que destacar en esta etapa, que los trazados de las plantas que se obtienen, no han de tener un contorno rectangular, sino que automáticamente se genera una planta de contorno irregular y con espacios interiores vacíos (patios) que cumple con los requisitos impuestos; si el grafo de adyacencia no es un grafo triangulado.
8. Introducción de nuevos requisitos. A la vista de los esquemas de distribución en planta se introducirán nuevos requisitos de adyacencia, comprobándose al mismo tiempo si dichos requisitos son compatibles con la planaridad del grafo de adyacencia. Si lo fuera, trazaríamos el nuevo esquema de planta mediante la repetición de las etapas 6 y 7.

Todo este procedimiento general, que hemos descrito, se ha desarrollado mediante la elaboración de un programa de ordenador que realiza todas las operaciones necesarias

para la generación de los esquemas de distribución en planta, de forma totalmente automática, interactiva y flexible, necesitando únicamente que el operador defina las relaciones de adyacencia entre los distintos locales que integran la planta, así como sus características geométricas límites.

DESCRIPCION DE LOS ALGORITMOS DESARROLLADOS

Condiciones iniciales

Las condiciones iniciales del problema adimensional las forman el conjunto de espacios o locales que se van a ordenar, así como todas las relaciones de adyacencia y orientación que se establezcan entre ellos. Estos primeros datos se estructuran como un *grafo abstracto, simple y no dirigido* G_1 : los locales son sus *vértices* y las relaciones —de adyacencia u orientación— entre ellos sus *aristas*; comúnmente se le denomina “grafo de adyacencia”

Dicho grafo abstracto queda definido mediante el conjunto de aristas que enlazan cada par de vértices del mismo: en el conjunto —o lista— $ADY(v)$ figurarán todos aquellos vértices w adyacentes al v mediante la arista (v, w) .

Los cuatro primeros vértices representarán a las orientaciones Norte, Oeste, Sur y Este, siempre en este orden; serán, por consiguiente, los vértices exteriores del mencionado grafo.

Dado que el procedimiento que se presenta contempla la adición de nuevas relaciones de adyacencia a lo largo del proceso, resulta conveniente introducir como datos iniciales aquellas relaciones que se consideran más importantes, para posteriormente, y a la vista de sucesivos trazados planos, añadir aquellas otras que completen la serie de requisitos impuestos a la organización en planta —resultado final del proceso de generación.

Conecividad

La técnica de representación de una distribución espacial en planta mediante el *dual* del grafo de adyacencia exige que éste sea *biconexo y planar*.

El algoritmo elaborado para el análisis de la conectividad —así como el desarrollado para el estudio de la biconectividad, en la siguiente etapa— está basado en los trabajos realizados por Hopcroft y Tarjan⁵ en este sentido. Dichos autores utilizan una potente técnica de exploración de grafos denominada *depth first search* (D.F.S.)

Esta técnica transforma el grafo original en otro grafo denominado *árbol palmera*, de estructura más sencilla y ordenada que la del primitivo; de esta forma, se facilita su manipulación en posteriores operaciones.

Una vez finalizado el proceso, se informa sobre el estado de conectividad del grafo y se describen las componentes conexas —caso de que existan— para que el operador las conecte entre sí mediante la introducción de las aristas oportunas. En esta etapa, por consiguiente, se puede producir la primera adición de nuevos requisitos por parte del operador, requisitos que tendrán por objetivo relacionar las componentes conexas para que ésta sea única.

La técnica D.F.S. seguida para la exploración de todos los vértices se ilustra en el ejemplo de la Figura 1.

Como puede apreciarse en la figura, este tipo de examen impone una dirección a cada arista atravesada y una nueva numeración ($NUM(v)$) de los vértices según se alcanzan. De igual forma, partitiona el conjunto de aristas en dos clases: las *ramas* que unen vértices en orden ascendente —según la nueva numeración— y nos definen un árbol T de G_1 , y las *frondas* que unen vértices en orden descendente. A este tipo de árbol con ramas y frondas, se le denomina *árbol palmera*.

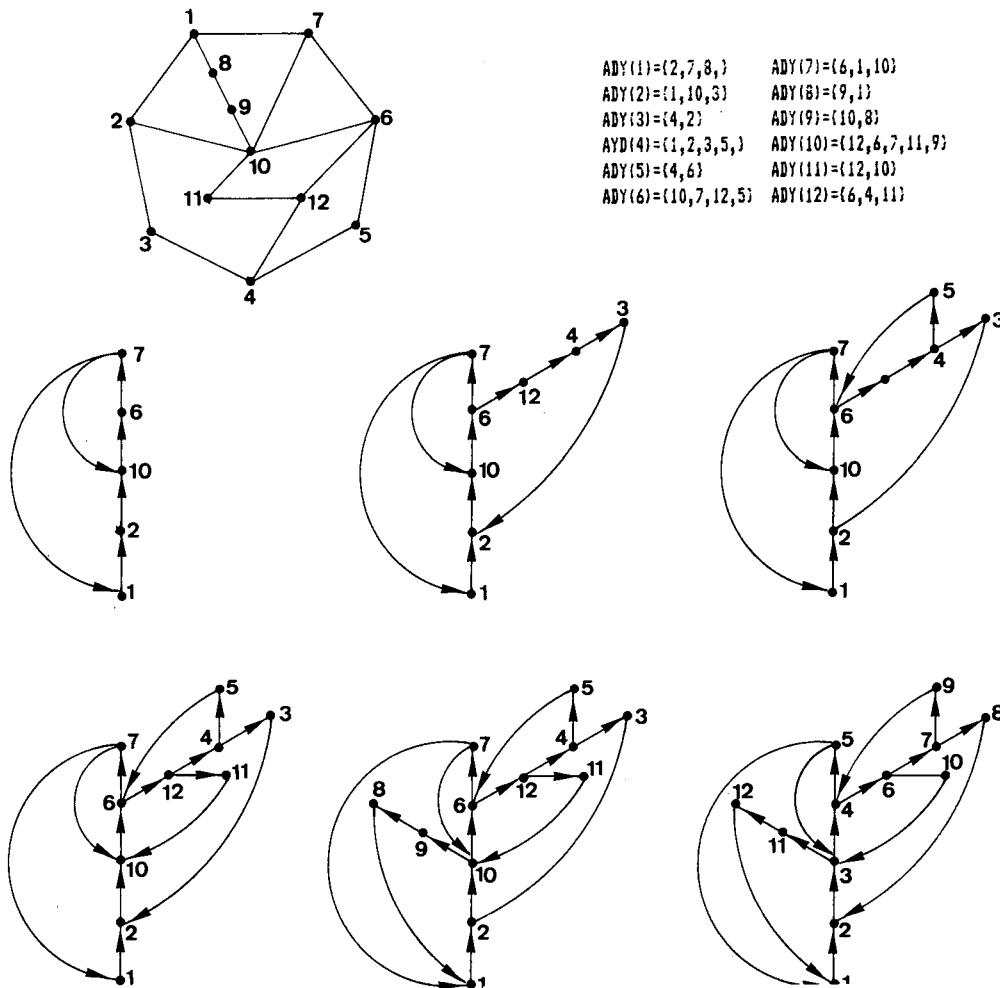


Figura 1.

Biconectividad

El algoritmo elaborado para verificar la planaridad de un grafo —que se describe en la siguiente etapa— sólo se puede aplicar sobre grafos *biconexos*. Así pues, se necesita asegurar la biconectividad del grafo de adyacencia antes de comprobar su planaridad.

El análisis de la biconectividad se realiza durante la misma D.F.S. descrita en la etapa anterior, e igualmente se basa en el algoritmo propuesto por Hopcroft y Tarjan⁵. Por lo tanto, al término de la D.F.S. obtendremos todos los posibles vértices que articulan componentes biconexas (Figura 2).

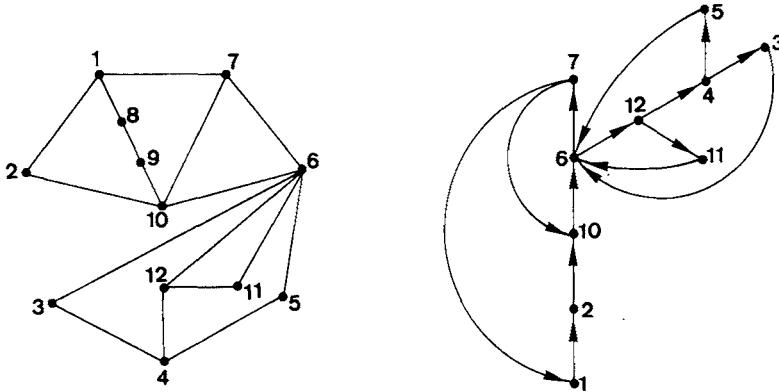


Figura 2.

Del mismo modo que en el apartado anterior, el programa informará del estado de biconnectividad del grafo, describiendo las componentes biconexas, para que el operador las conecte entre sí mediante la introducción de las aristas oportunas.

Al término de estas dos etapas, el grafo inicial G_1 ha pasado a ser un grafo G_2 dirigido, remunerado según $NUM(v)$ y unas nuevas listas de adyacencia para cada uno de ellos: si en la lista $ADY(v)$ de G_1 figuraban todos los vértices que eran adyacentes al v , en la nueva lista $ADY(v)$ de G_2 solamente figurarán aquellos vértices w tales que: $v \rightarrow w$ o $v \dashrightarrow w$.

Primera comprobación de planaridad

Como cuestión previa al procedimiento general propuesto para verificar la planaridad del grafo G_3 , realizaremos una sencilla comprobación mediante la aplicación de la fórmula de Euler: para que un grafo sea planar, es condición necesaria que cumpla la siguiente expresión:

$$|A| \leq 3.|V| - 6$$

Así pues, el número de relaciones impuestas —aristas en el grafo— se encuentra limitado superiormente a la cantidad que figura en la relación de euler.

Una vez superado el test previo de planaridad, podemos ya aplicar el algoritmo desarrollado para el estudio de la planaridad. Dicho algoritmo está basado en los trabajos realizados por Hopcroft y Tarjan⁶. Es del tipo constructivo, es decir, lo que se intenta conseguir es una representación planar del grafo mediante la adición consecutiva y ordenada de partes del mismo.

El procedimiento general de comprobación de planaridad de un grafo biconexo (G_2) en síntesis, incluye las siguientes operaciones:

- a) Obtención de una *ruta* $R : v \dashrightarrow v$ en el árbol palmera que comience y termine en el mismo vértice, a la que llamaremos *círculo original*.
 Este circuito original estará formado por una ruta sobre el árbol T extendido sobre G_2 ($R : v \rightarrow w$) más una fronda que lo cierra ($w \dashrightarrow v$).
- b) Eliminación de las aristas de este circuito. tras dicha eliminación el grafo quedará escindido en varios *segmentos*. Cada segmento escindido constará, o bien de una sola fronda, o de un subárbol y todas las frondas que salen de él y conducen a vértices del circuito.
- c) Repetición de las dos operaciones anteriores a) y b) con todos y cada uno de los segmentos generados: en cada segmento escindido se obtiene un circuito original, y la supresión de sus aristas produce, a su vez, nuevos segmentos escindidos. Esta operación de tipo recursivo finalizaría cuando todos los segmentos escindidos sean única y exclusivamente del tipo frondas.
- d) Ubicación en el plano de todos y cada uno de los segmentos en el mismo orden con el que han sido generados.

Por el teorema de la Curva de Jordán, un segmento puede ser representado en el plano bien por el exterior, o por el interior de su circuito original, sin que se produzcan cruces de aristas. La ubicación en el plano de un segmento por el interior (exterior) puede provocar el cambio al exterior (interior) de otros segmentos ya situados, y este cambio, a su vez, puede ser la causa de nuevos giros de otros segmentos del exterior al interior, o viceversa.

El grafo será planar si se pueden compatibilizar todos estos cambios mencionados en la última operación para que, en su representación plana, no se produzcan cruces de aristas. En caso contrario se trataría de un grafo no planar. El procedimiento se ilustra en la Figura 3a y 3b.

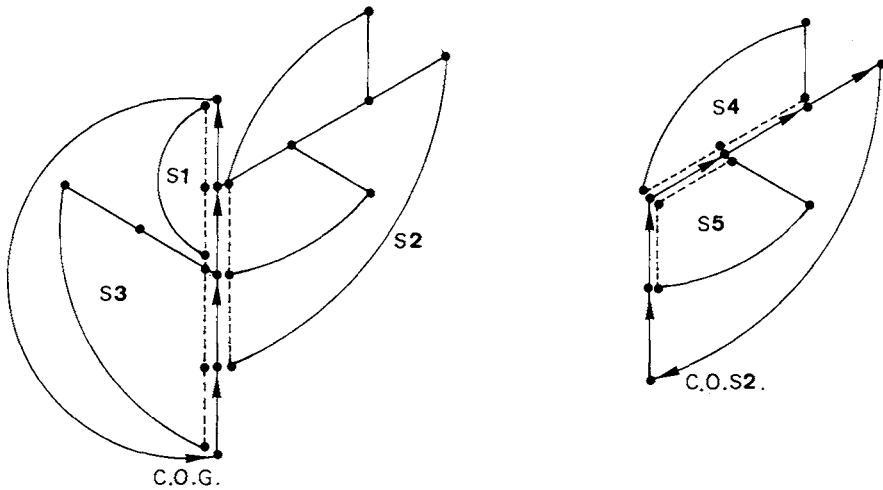


Figura 3a.

Todo este proceso descrito se realiza utilizando el algoritmo una sola vez, mediante

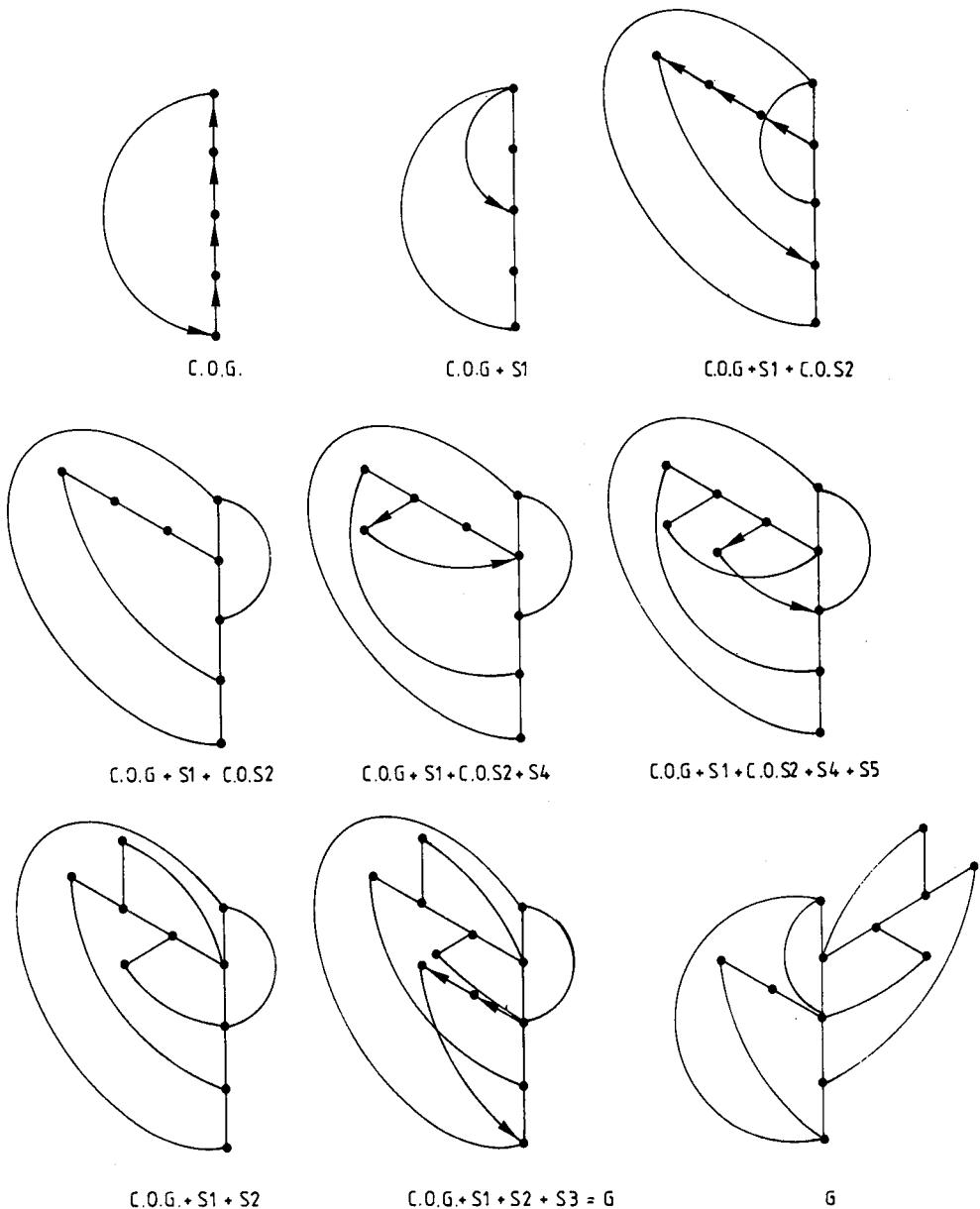


Figura 3b.

la aplicación de una nueva D.F.S. que: genera todas las *rutas disjuntas* $R : u \dashrightarrow v$ (que no contienen aristas en común) del grafo G_2 , y las sitúa en el plano conforme se van obteniendo.

Al término de esta etapa, comienza a configurarse ya un grafo con unas características más concretas que el inicial. Conocemos ya los suficientes datos como para poderlo representar geométricamente en el plano.

Tendremos, por tanto, un grafo G_3 que será: dirigido, ordenado, y sobre todo, planar. Conocemos, además, donde se han de ubicar todas sus aristas para que, en una

representación plana del mismo, no se produzca ningún cruce entre ellas.

Formación de las caras

En esta sección exponemos un procedimiento original que obtiene el conjunto de caras del grafo planar G_3 , entiéndase por *cara* de un grafo planar la región del plano limitada por las aristas de un *ciclo*, tal que dos puntos arbitrarios de la citada región pueden unirse por un trayecto continuo que no atravesese ni aristas ni vértices.

Este conjunto de caras se genera a partir de los resultados obtenidos en la anterior etapa que estudia la planaridad. En ella, quedan definidas las posiciones que han de ocupar todas las *frondas* para que el grafo pueda representarse en el plano sin que sus aristas se corten.

Hay que hacer notar, que el conjunto de caras de un grafo planar no tiene porqué ser único, como puede apreciarse en la Figura 4.

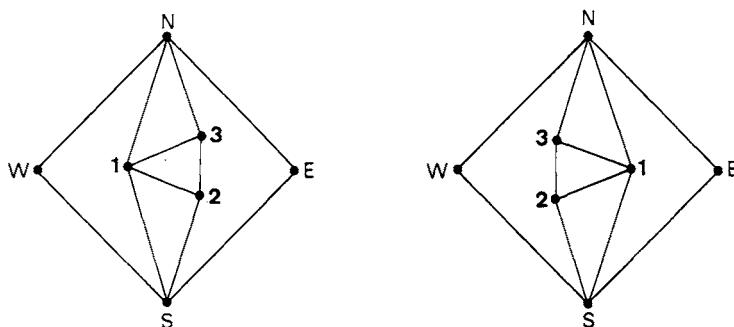


Figura 4.

El conjunto de caras de un grafo planar, en el que se ha definido la *cara exterior*, será único si cumple con una de las dos condiciones siguientes:

- a) El grafo es *triconexo*. (Whitney 1933)
- b) Si es *biconexo* y todo par de vértices de corte $[a, b]$ se encuentra en uno de los casos siguientes:

- 1- Son vértices de la cara exterior y definen exactamente dos *componentes*. El conjunto de caras sería único, pues el giro de cualquiera de las componentes alrededor del par de corte sería imposible, ya que nos llevaría a la pérdida de la cara exterior (Figura 5).
- 2- Son vértices de la cara exterior y definen exactamente tres *componentes*, una de las cuales consiste en una *ruta simple* (de extremos a y b) no contenida en la cara exterior.

También sería único el conjunto de caras, puesto que la componente *ruta simple* es imposible situarla en la cara exterior, ya que implica la pérdida de vértices de dicha cara que consideramos fija. Para las otras dos componentes nos remitimos al caso b1 (Figura 5)

- 3- Uno de los vértices (o ambos) no pertenece a la cara exterior y definen únicamente dos *componentes*, una de las cuales consiste en una *ruta simple* de extremos a y b .

La componente *ruta simple* no puede ubicarse en otra cara sin que se produzcan cruces de aristas, ya que implicaría la existencia de más de dos componentes para dicho par.

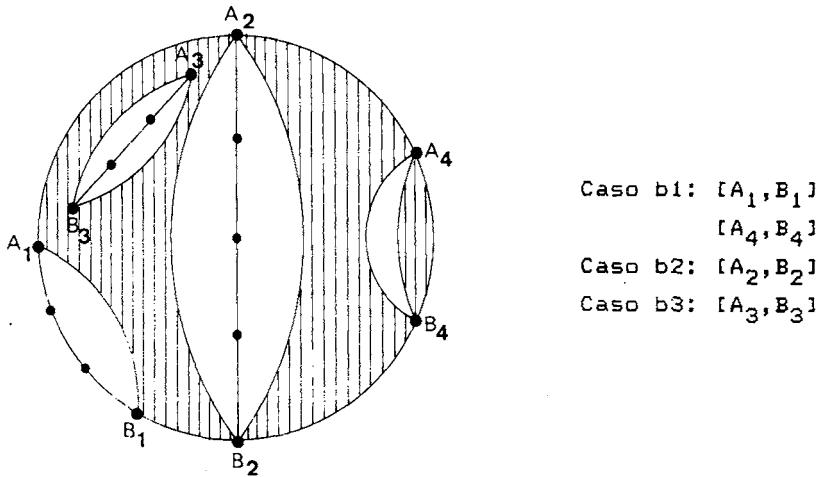


Figura 5.

No obstante, si el grafo G_3 no posee un único conjunto de caras, por no cumplir con las condiciones anteriormente expresadas, el algoritmo generará uno de los conjuntos posibles.

El procedimiento propuesto para la formación de caras consiste, en esencia, en recorrer todas las aristas del grafo G_3 según un orden determinado, de forma que se vayan generando ciclos que sean siempre caras del grafo.

Para ello, supongamos que ante un trazado plano (sin cruce de aristas) del grafo G_3 , deseamos “recorrer” todas las caras de dicho grafo. El procedimiento sería muy simple en síntesis: escoger siempre la arista siguiente —según un sentido prefijado, el antihorario por ejemplo— a la arista por la que hemos alcanzado un vértice v . De esta forma, los ciclos que se generan son siempre caras, de acuerdo con la definición que hemos dado de éstas (Figura 6).

Por lo tanto, el procedimiento debe realizar dos operaciones: una ordenación de las adyacencias de cada vértice v —aristas que inciden sobre v — en el sentido prefijado, y una exploración de estas nuevas listas de adyacencias para generar el conjunto de caras.

Describamos a continuación cada una de estas operaciones:

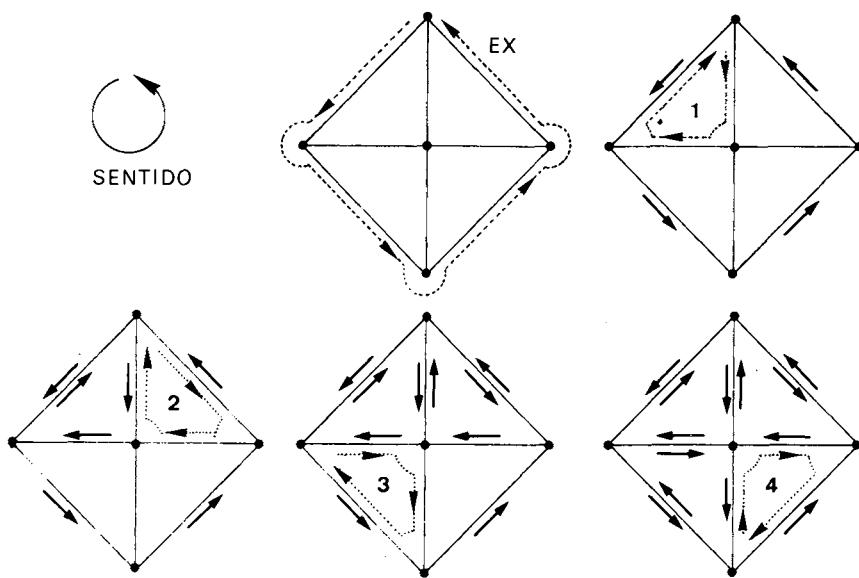


Figura 6.

1. Ordenación de las adyacencias

En esta etapa se elabora un nuevo conjunto de listas de adyacencia $AYD - ORD(v)$ a partir del conjunto de listas $ADY(v)$ que definen el grafo G_3 .

Si la lista $ADY(v)$ contenía sólo aquellos vértices w adyacentes al v tales que $v \rightarrow w$ o $v \dashrightarrow w$, en $ADY - ORD(v)$ figurarán todos los vértices que son adyacentes al v , y además, ordenados según el sentido antes mencionado (antihorario) alrededor del vértice v .

Para ello, exploramos el grafo G_3 en el mismo orden que lo hicimos en el estudio de la planaridad, es decir, realizamos una D.F.S. de forma que se generen de nuevo y en el mismo orden, todas y cada una de las frondas. Durante esta D.F.S. se forman las nuevas listas $ADY - ORD(v)$ del siguiente modo:

- Si la arista atravesada es del tipo *rama* $v \rightarrow w$: se añade a la lista $ADY - ORD(v)$ como *vértice activo superior* y v se añade a $ADY - ORD(w)$ como *vértice activo inferior*.
- Si la arista atravesada es del tipo *fronda* $v \dashrightarrow w$: se introduce a la izquierda o derecha del vértice activo *superior* de $ADY - ORD(v)$ y v se introduce a la izquierda o derecha del vértice activo *inferior* de $ADY - ORD(w)$. La adición a izquierda o derecha depende de la posición asignada a dicha fronda en el estudio de la planaridad.

Al término de esta operación el grafo G_3 habrá pasado a ser un nuevo grafo G_4 no dirigido, pues en las listas de adyacencia $ADY - ORD(v)$ que lo definen, aparecen de nuevo todos los vértices que son adyacentes al v , y además, todas estas listas se encuentran ordenadas según el sentido antes mencionado.

2. Generación de las caras

Mediante una nueva exploración sobre el grafo G_4 estructurado en el paso anterior, obtendremos el conjunto de todas las caras. El orden y regla de búsqueda será como sigue:

- a) Elección de un vértice arbitrario u_0 que sea extremo de aristas no exploradas (u_0, u) . Esta arista sería la primera arista de la cara c_i .
- b) Supongamos que hemos recorrido una arista (u, v) de la cara c_i , la siguiente arista de dicha cara sería entonces la (v, w) , siendo w el vértice que sigue al u por la derecha en la lista ordenada de adyacencia del v :

$$ADY - ORD(v) = \{\dots, u, w, \dots\}$$

- c) Dado que cada arista pertenece a dos caras, será recorrida sólo dos veces y en sentidos opuestos. Para evitar que una arista se recorra más de dos veces, o en un sentido no apropiado, se irán marcando sucesivamente las aristas exploradas y el sentido de recorrido.
- d) Repetimos los pasos b y c hasta que se alcance el vértice u_0 donde comenzó la generación de la cara c_i . En ese momento se habrá completado dicha cara.
- e) Una vez completada la cara c_i , comenzamos una nueva cara por cualquier arista no explorada desde dicho vértice u_0 . Cuando desde el vértice u_0 no queden más aristas sin explorar, se habrán completado todas las caras a las que pertenece el vértice u_0 , entonces comenzaríamos de nuevo por el paso a para elegir un nuevo vértice u_0 que contenga aristas sin explorar.

El proceso finaliza cuando todas las aristas han sido exploradas en los dos sentidos: obtendríamos las $2 + |A| - |V|$ caras según la fórmula de Euler.

El procedimiento propuesto para la generación de las caras se ilustra en la Figura 7.

Hay que hacer notar —como se observa en la figura— que el sentido de las caras obtenidas (excepto la exterior) es el contrario al considerado en la ordenación de las adyacencias.

El grafo G_4 , no dirigido y con las listas de adyacencias ordenadas, no ha sufrido alteración alguna en este segundo paso. Así pues, esa será su estructura al finalizar esta etapa.

Estudio de las componentes escindidas

Una vez generado un conjunto de caras en la etapa anterior, y antes de introducir nuevos requisitos de adyacencia u orientación, comprobaremos si el grafo G_4 contiene alguna componente cuyo giro alrededor de un par de vértices alterase el conjunto de caras obtenido anteriormente, es decir, lo que comprobamos en definitiva es si el grafo G_4 cumple con los requisitos señalados anteriormente en 1 para que el conjunto de caras fuese único.

Antes de pasar a describir el algoritmo original propuesto para obtener cada

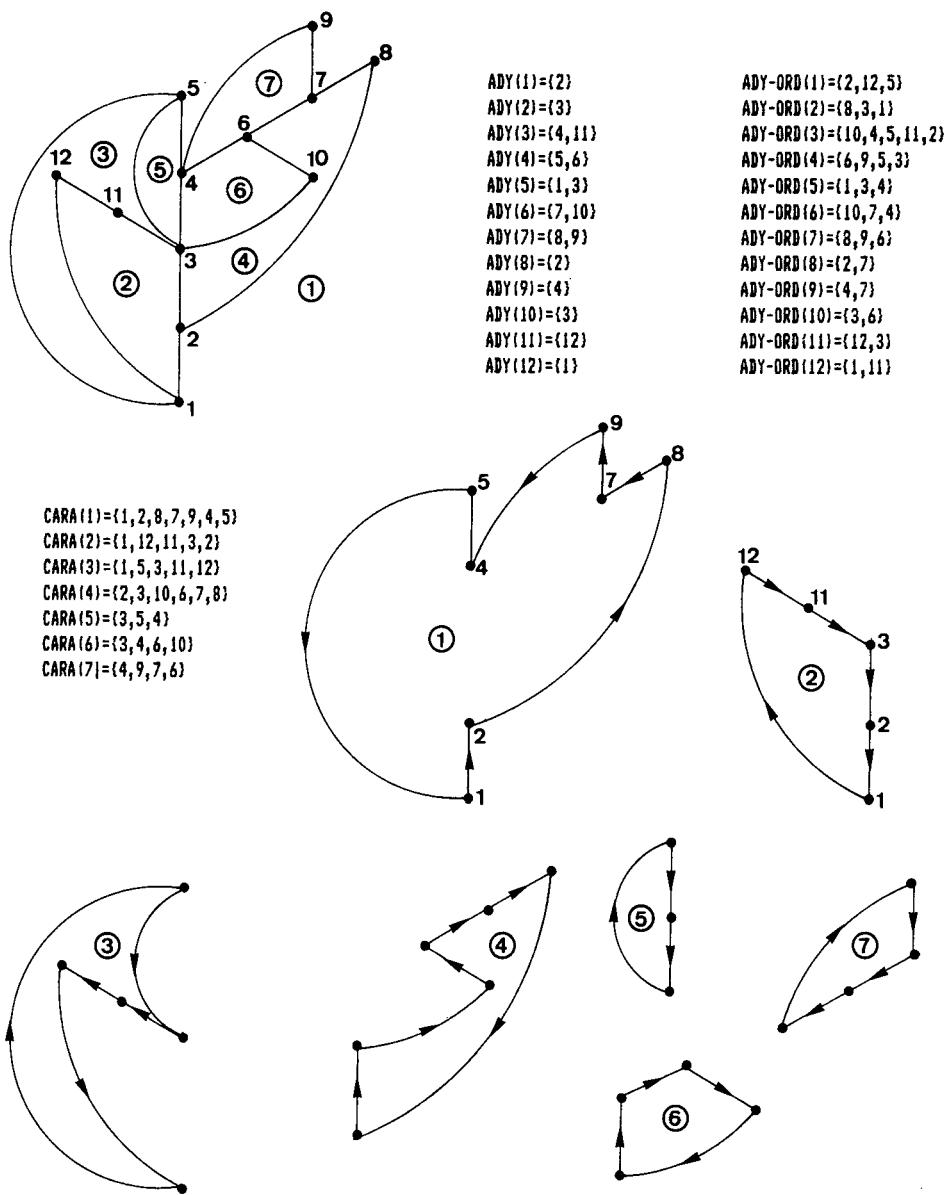


Figura 7.

una de estas componentes, conviene expresar algunos conceptos que utilizaremos posteriormente en dicha descripción:

- Dos caras C_k, C_l son *adyacentes* si poseen en común al menos dos vértices enlazados mediante una arista.
- Si dos caras no adyacentes C_k, C_l poseen más de un vértice en común, éstos — agrupados de dos en dos — serán vértices de corte, y ambas caras formarán lo que denominaremos *par de caras separador* $PC_i = [C_k, C_l]$ (Figura 8A)
- Si dos caras adyacentes poseen en común vértices no consecutivos, de igual forma,

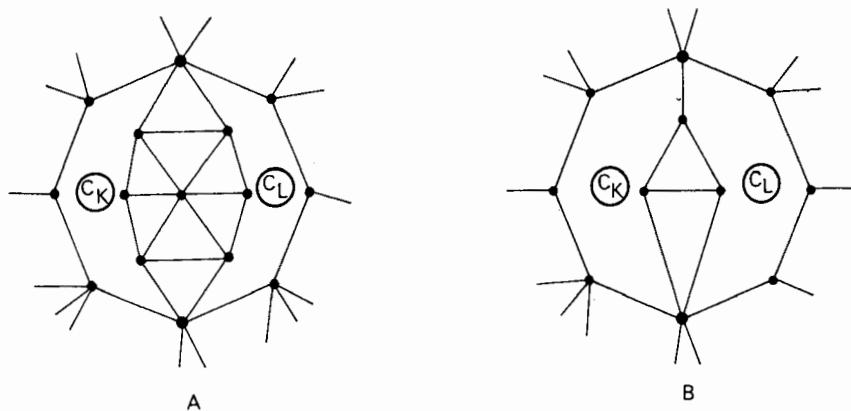


Figura 8

formarán un *par separador* PC_i (Figura 8B).

- Si en el grafo dual de G_4 eliminamos los vértices correspondientes al par de caras separador PC_i y las aristas que inciden sobre ellos, dicho grafo contendrá el vértice que representa a la cara exterior del grafo G_4 . El conjunto de las restantes lo denominaremos *componente escindida* CM_i por el par de caras separador PC_i (Figura 9).

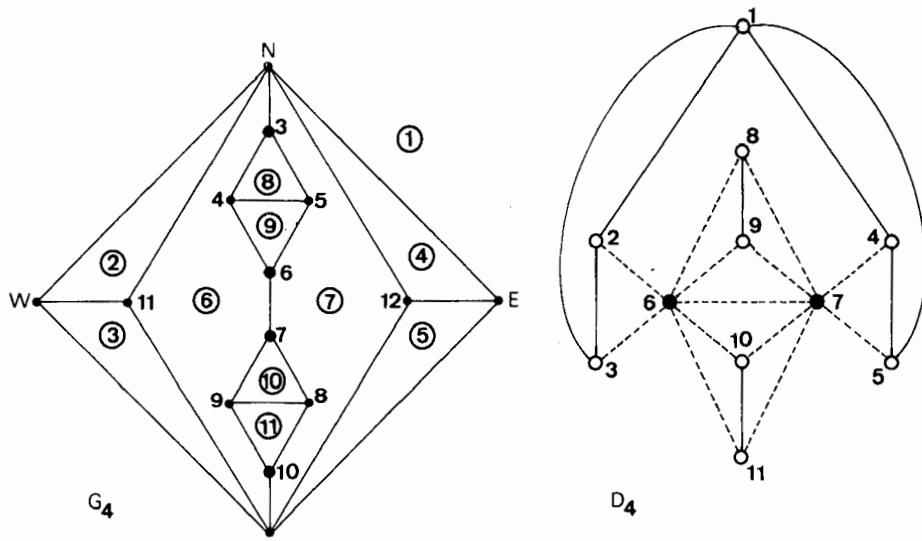


Figura 9

Una vez expresados estos conceptos, podemos pasar a describir el algoritmo original propuesto para la obtención del conjunto de componentes escindidas y de los pares de corte alrededor de los cuales pueden girar dichas componentes. Dicho algoritmo realiza las operaciones siguientes:

- a) Comparación de pares de caras. Mediante la comparación de todos los pares de caras del grafo G_4 obtenemos:

- a.1) Vértices comunes de cada par comparado.
 - a.2) Listas de caras adyacentes: si dos caras C_k, C_l son adyacentes se introducirán en las listas $ADY - C(C_l)$ y $ADY - C(C_k)$ respectivamente.
Estas listas, precisamente, serán las que definirán al final de todo este proceso al grafo dual D_4 del grafo de adyacencia G_4 .
 - a.3) Conjunto de los pares de caras separadores: si este conjunto fuese el vacío, sería único el conjunto de caras del grafo G_4 —tendría una única representación en el plano— y pasaríamos a la siguiente etapa: introducción de nuevos requisitos.
- b) Obtención de los vértices *pares de corte*. Esta operación sólo se realizará si el conjunto de pares de caras separadores no fuese el vacío, como se ha indicado anteriormente.
Por cada par de caras separador $PC_i = [C_k, C_l]$, se calcula el conjunto de componentes conexas escindidas CM_i del grafo dual D_4 , mediante la eliminación de sus vértices correspondientes a las caras C_k y C_l y, las aristas que incidan sobre ellos, siendo:

$$\begin{aligned} CM_i &= \{CM_{i1}, CM_{i2}, \dots\} && \text{conjunto de componentes escindidas.} \\ CM_{ij} &= \{C_p, C_{p+1}, \dots\} && \text{componente escindida i, j.} \end{aligned}$$

Si designamos por $\{v_j\}$ el conjunto de vértices comunes del par de caras separador $PC_i = [C_k, C_l]$, y $\{w_{ij}\}$ el conjunto de vértices de la componente escindida CM_{ij} por dicho par de caras separador, el par de vértices de corte PV_{ij} correspondiente a la componente CM_{ij} es aquel que:

$$PV_{ij} = [u, v] = \{v_i\} \cap \{w_{ij}\}$$

Estos pares de corte PV_{ij} constituyen los vértices alrededor de los cuales puede girar la componente escindida CM_{ij} . Para ilustrar esta operación, los pares de corte del ejemplo de la Figura 9 serían:

$$\{v_i\} = \{N, S, 3, 6, 7, 10\}, \quad \{w_{11}\} = \{3, 4, 5, 6\}$$

$$\{v_i\} \cap \{w_{11}\} = \{3, 6\} \quad \text{luego :}$$

$$PV_{11} = [3, 6]$$

$$\{w_{12}\} = \{7, 8, 9, 10\}, \quad \{v_1\} \cap \{w_{1,2}\} = \{7, 10\} \quad \text{luego :}$$

$$PV_{12} = [7, 10]$$

De esta forma, al término del algoritmo descrito, tendremos una información completa sobre la estructura del grafo G_4 —y de su dual D_4 — que nos permitirá, en etapas posteriores, obtener todas las representaciones planas posibles del grafo de adyacencia G_4 y su dual D_4 , además de permitirnos el poder añadir nuevos requisitos de adyacencia u orientación sin alterar la planaridad del grafo de adyacencia.

En esta etapa el grafo G_4 no sufre en su estructura cambio alguno.

Obtención del pseudo-dual

Con el fin de facilitar la introducción de nuevas relaciones de adyacencia u orientación entre los espacios componentes del problema, representaremos gráficamente un esquema de distribución en planta que responda a las adyacencias introducidas hasta el momento.

No obstante, para que dicha representación pueda efectuarse mediante el trazado del grafo dual, se requiere que el grafo de adyacencia sea un grafo *triangulado*, si todos los locales han de ser rectangulares.

Por consiguiente, utilizaremos la técnica de representación del pseudo-dual — Baybars y Eastman¹— mientras que el grafo G_4 no cumpla con la citada condición de ser triangular.

El procedimiento para obtener dicho pseudo-dual comprende las siguientes operaciones:

- 1- Introducción de un vértice ficticio en cada cara del grafo G_4 cuya valencia sea distinta de tres, excepto en la cara exterior (Figura 10).

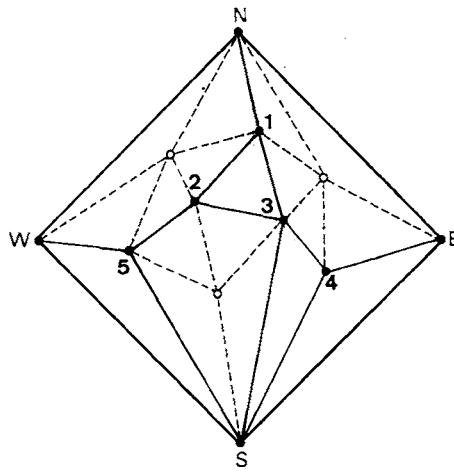


Figura 10

- 2- Adición de aristas ficticias enlazando cada vértice ficticio en el interior de una cara con todos y cada uno de sus vértices (Figura 10).
Después de esta operación se obtendrá un grafo GF_4 ficticio con todas sus caras de valencia tres (triángulos), excepto la cara exterior que continuará teniendo valencia cuatro.

- 3- Obtención del *dual* de GF_4 . Será el grafo resultante PD_4 de considerar un vértice por cada una de las caras de GF_4 , y una arista por cada dos vértices que representen a dos caras de GF_4 que sean adyacentes. En la Figura 11 se ilustra la obtención del dual del grafo GF_4 de la Figura 10.

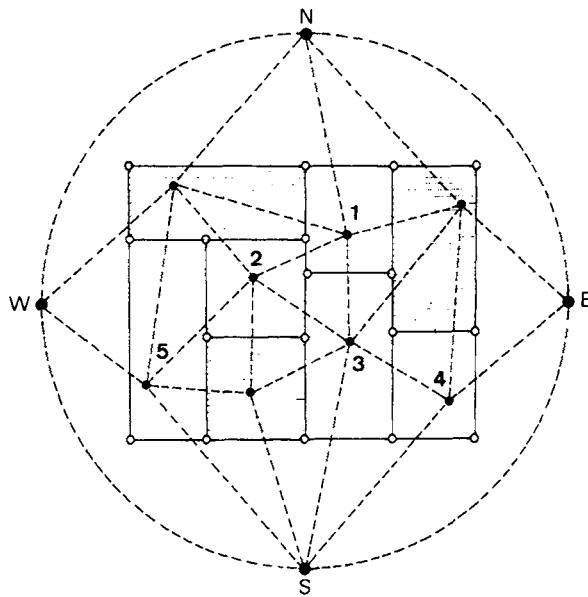


Figura 11

Este grafo PD_4 es el *pseudo-dual* de G_4 , y por tanto, un grafo simple cuyas aristas representan las particiones o tabiques que separan los distintos espacios entre sí o con el exterior, y sus vértices los puntos de encuentro entre dichas particiones; será pues, una representación de un esquema de distribución en planta.

No obstante, para la representación gráfica de PD_4 , suprimiremos el vértice que representa a la cara exterior de GF_4 y todas las aristas que incidan sobre el mismo.

De esta forma, a la vista del trazado de un esquema de planta, podremos introducir con más precisión las condiciones de adyacencia que deseemos, incluso decidir transformar uno o más locales ficticios en reales, consiguiéndose, en este último supuesto, que las condiciones del contorno del esquema de planta sea totalmente irregular, como se puede apreciar en el ejemplo de la Figura 11.

Introducción de nuevos requisitos

La extracción del pseudo-dual PD_4 y su trazado en el plano, así como la obtención del conjunto de componentes escindidas y de los pares de corte alrededor de los cuales pueden girar dichas componentes, facilitarán la introducción de nuevas relaciones de adyacencia u orientación —nuevas aristas—, restringiendo, pues, el número de soluciones finales en el problema de generación de organizaciones espaciales en planta.

La adición de aristas, sin embargo, no ha de realizarse aleatoriamente, sino de acuerdo a un cierto método —propuesto por nosotros originalmente— que,

fundamentalmente, pretende asegurar la no alteración de la planaridad del grafo de adyacencia con la introducción de nuevos requisitos.

Sea pues (u, v) la nueva arista que deseamos introducir. La primera operación a realizar es la de buscar en todas las caras la pareja de vértices u y v , pudiéndose representar entonces los tres siguientes casos:

- 1) Si existe una cara C_i que contiene al par de vértices u y v , podemos asegurar que el grafo resultante es planar y la cara C_i se escindirá en dos por la nueva arista (u, v) , aumentando en una unidad el número total de caras y aristas del grafo G_4 .
- 2) Por el contrario, si los vértices extremos de la arista (u, v) no pertenecen a una misma cara C_j del grafo G_4 y es único el conjunto de caras de dicho grafo —por cumplir las condiciones señaladas en 1—, podemos afirmar que la introducción de dicha arista (u, v) en el grafo de adyacencia hará que este no sea planar.
- 3) El caso de análisis más complejo, y al que vamos a dedicar una especial atención, se presentará cuando los dos vértices extremos — u y v — de la arista no pertenezcan a una misma cara del grafo G_4 , pero, por no ser único el conjunto de caras de dicho grafo, puede darse la posibilidad de que los vértices señalados aparezcan en una misma cara, mediante el giro de alguna componente escindida CM_{ij} alrededor de su par de corte PV_{ij} . Si esto es factible, el grafo de adyacencia resultante será planar.

Cada uno de los tres casos mencionados quedan ilustrados en la Figura 12.

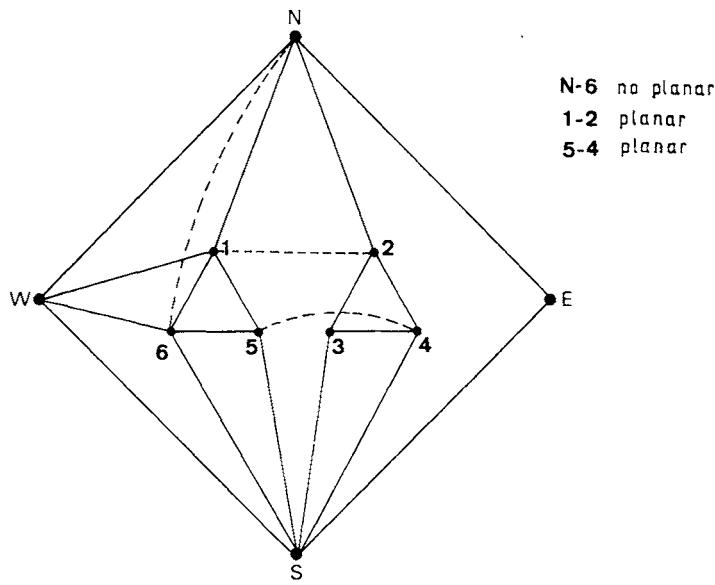


Figura 12

El problema que se plantea en este último caso se reduce, pues, a comprobar si dos vértices cualesquiera que se quieren unir pueden aparecer en una misma cara del grafo, mediante los giros de determinadas componentes. El algoritmo propuesto para

verificar este hecho pretende, en primer lugar, calcular el conjunto de vértices *extras* que pueden aparecer en cada cara, y determinar cuáles son los giros de las componentes que hay que realizar para que eso ocurra.

El algoritmo lo podemos describir detalladamente mediante los tres pasos siguientes:

1. Formación de listas de inclusión de componentes

Consideremos las componentes escindidas:

$$CM_i = \{C_o, C_{o+1}, \dots\} \quad CM_j = \{C_p, C_{p+1}, \dots\}$$

Diremos que CM_i incluye a CM_j si cualquier cara de CM_j pertenece a CM_i . Esta relación de inclusión define sobre el conjunto de componentes escindidas un *árbol dirigido*, cuyo origen será un vértice que representa al grafo G_4 , puesto que dicho grafo incluye a todas las componentes escindidas. Este árbol dirigido queda expresado mediante un conjunto de listas de inclusión: en la lista $INCLU(i)$ figurarán todas las componentes escindidas que incluye la CM_i .

En la figura 13 se ilustra con un ejemplo cuál sería la estructura de los conjuntos que hemos definido hasta el momento.

2. Exploración del árbol de inclusión

Una vez formado el árbol que manifiesta las relaciones de inclusión entre componentes escindidas, podemos obtener ya los conjuntos de vértices extras que pueden aparecer en cada una de las caras del grafo G_4 ; esta operación se realizará mediante los *giros virtuales* de las componentes alcanzadas en una exploración del árbol —con una D.F.S.— partiendo de su origen.

Definamos, pues, las operaciones necesarias para poder efectuar el giro de una componente escindida.

Sean:

$$PC_i = [C_k, C_l]$$

$$CM_i = \{CM_{i1}, CM_{i2}, \dots\}$$

$$CM_{ij} = \{C_o, C_{o+1}, \dots\}$$

$$PV_{ij} = [a, b]$$

$$C_o = \{v_i, v_{i+1}, \dots\}$$

un par de caras de separación.

el conjunto de componentes escindidas por el par PC_i .

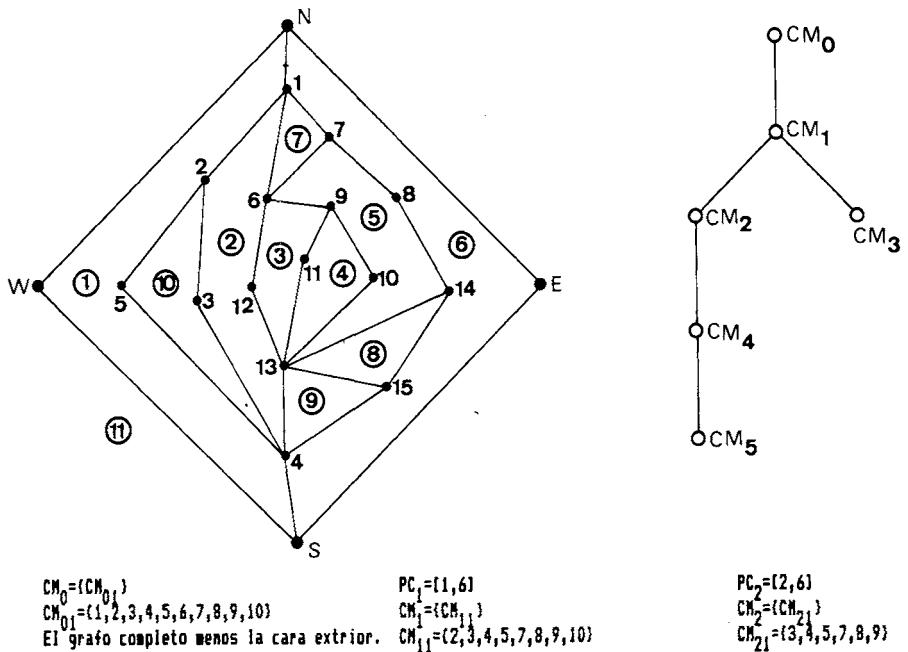
componente escindida j perteneciente al conjunto CM_i .

el par de vértices de corte de la componente CM_{ij} .

una cara de la componente CM_{ij} .

El giro de la componente CM_{ij} alrededor de su par de corte PV_{ij} , implica un intercambio de vértices entre las caras del par de separación PC_i —entre las caras C_k y C_l —, de forma que:

- La cara C_k transmite los vértices comprendidos entre el par de corte (a, b) hacia la cara C_l entre los vértices (b, a) y,



$$\begin{aligned}PC_3 &= \{1, 2\} \\CM_3 &= \{CM_3\} \\CM_{2,1} &= \{10\}\end{aligned}$$

$$\begin{aligned}PC_4 &= \{2, 5\} \\CM_4 &= \{CM_{41}\} \\CM_{41} &= \{3, 4\}\end{aligned}$$

$$\begin{aligned}PC_5 &= [3, 5] \\CM_5 &= \{CM_{51}\} \\CM_{51} &= \{4\}\end{aligned}$$

El conjunto de caras y los vértices extras que pueden aparecer en cada una de ellas, más los giros necesarios para que esto ocurra son:

```

C(1)={1,2,5,4,S,W,N}
C(2)={6,11,13,4,3,2,1}
C(3)={6,9,12,13,11}
C(4)={9,10,13,12}
C(5)={7,8,14,13,10,9,6}
C(6)={4,15,14,8,7,1,W,E,S}
C(7)={1,7,6}
C(8)={13,14,15}
C(9)={4,13,15}
C(10)={2,3,4,5}
C(11)={W,S,E}

```

```

VE(1)={7,8,14,15,6,11,13,9,10,12,3}
VE(2)={7,8,14,15,9,10,12,5}
VE(3)={10}
VE(4)={}
VE(5)={1,12}
VE(6)={2,5,6,11,13,9,10,12,3}
VE(7)={}
VE(8)={}
VE(9)={}
VE(10)={}
VE(11)={}

```

GV(1,7)={1}	GV(2,10)={4}
GV(1,8)={1}	GV(2,12)={5,4}
GV(1,14)={1}	GV(2,5)={3}
GV(1,15)={1}	GV(3,10)={5}
GV(1,6)={2,1}	GV(5,11)={4}
GV(1,11)={2,1}	GV(5,12)={5}
GV(1,13)={2,1}	GV(4,2)={1}
GV(1,9)={4,2,1}	GV(6,5)={1}
GV(1,10)={4,2,1}	GV(6,6)={2}
GV(1,12)={5,4,2,1}	GV(6,11)={2}
GV(1,3)={3}	GV(6,13)={23}
GV(2,7)={2}	GV(6,9)={4,2}
GV(2,8)={2}	GV(6,10)={4,2}
GV(2,14)={2}	GV(6,12)={5,4,2}
GV(2,15)={2}	GV(6,3)={3,1}
GV(2,9)={4}	

Figura 13

- La cara C_l transmite vértices hacia la C_k del mismo modo.

Así pues, las caras:

$$C_k = \{\dots, \mathbf{a}, v_i, v_{i+1}, \dots, v_n, \mathbf{b}, \dots\}$$

$$C_l = \{\dots, \mathbf{b}, v_j, v_{j+1}, \dots, v_n, \mathbf{a}, \dots\}$$

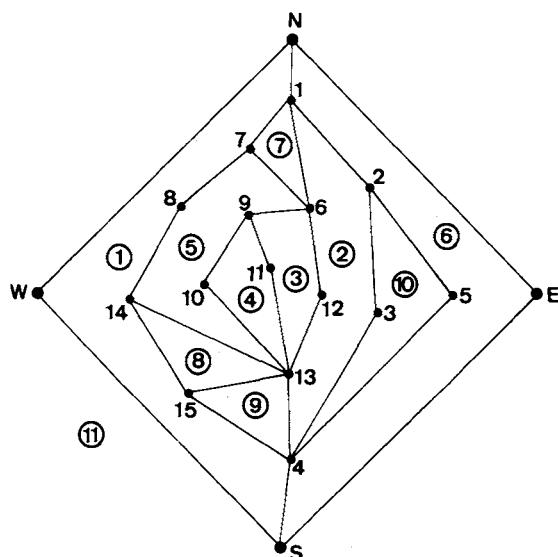
después del giro pasarían a ser:

$$C_k = \{\dots, \mathbf{a}, v_n, \dots, v_{i+1}, v_i, \mathbf{b}, \dots\}$$

$$C_l = \{\dots, \mathbf{b}, v_n, \dots, v_{i+1}, v_i, \mathbf{a}, \dots\}$$

Además cambiarían de sentido todas las caras de las componentes que incluya la CM_{ij} .

La operación de giro de una componente escindida alrededor de su par de corte se ilustra en la Figura 14.



Si giramos la componente CM_2 alrededor de su par de vértices (1,4), habrá un intercambio de vértices entre las caras 2 y 6 y cambiarán de sentido las caras 3,4,5,7,8,9.

```

C(2)={2,1,6,11,13,4,3)
C(6)={S,4,15,14,8,7,1,N)
C(3)={11,6,9,12,13}
C(4)={9,10,13,12}
C(5)={7,8,14,13,10,9,6}
C(7)={1,7,6}
C(8)={13,14,15}
C(9)={4,13,15}

```

después del giro se convierte en ->

```

C(2)={2,1,7,8,14,15,4,3}
C(6)={5,4,13,11,6,1,N}
C(3)={13,12,9,6,11}
C(4)={12,13,10,9}
C(5)={6,9,10,13,14,8,7}
C(7)={6,7,1}
C(8)={15,14,13}
C(9)={15,13,4}

```

Figura 14

Si hemos calificado como virtuales estos giros de componentes es porque, en realidad, no se llegan a efectuar; sólo sirven para determinar los vértices que pueden aparecer en una cara del grafo G_4 (conjunto de vértices extras en esa cara) y los giros necesarios para que esto ocurra.

Una vez que la cara C_k ha transmitido virtualmente los vértices oportunos hacia la C_l , habría que buscar hacia atrás en el árbol para comprobar si existe alguna componente CM_{ij} cuyo par de separación PC_{ij} contiene a la cara C_l , en cuyo caso la cara C_l transmitiría vértices hacia la otra cara del par PC_{ij} encontrado y así sucesivamente.

Al término de este paso quedan definidos los siguientes conjuntos:

$$VE_p = \{v_i, v_{i+1}, \dots\}$$

conjunto de vértices extras que
pueden aparecer en la cara C_p .

$$GV_{pi} = \{j, k, \dots\}$$

conjunto de giros necesarios de
componentes para que el vértice
 i aparezca en la cara C_p .

3. Búsqueda del par de vértices a unir

Si el nuevo requisito que se desea introducir supone enlazar mediante una arista los vértices u y v —vértices que no aparecían en una misma cara del grafo G_4 —, habrá que buscar, en primer lugar, todas aquellas caras C_p que cumplan con las siguientes condiciones:

$$u \in \{C_p \cup VE_p\} \quad \text{y} \quad v \in \{C_p \cup VE_p\}$$

De todas las caras C_p que las verifiquen, si existe una tal que: la pareja u, v no esté contenida en ninguna componente CM_{ij} cuyo par separador PC_i contenga a la cara C_p , entonces es posible la unión de dichos vértices mediante los giros contenidos en GV_{pi} . En caso contrario, la unión de u y v mediante una nueva arista se declararía no planar.

4. Giro de las componentes

En este paso se realizan los *giros efectivos* necesarios GV_{pi} para que los vértices u y v aparezcan en la cara C_p . El procedimiento de giro de componentes ha sido descrito anteriormente con los *giros virtuales*, y sólo se diferencia de aquellos, en que aquí quedan modificadas las caras por las transmisiones efectivas de vértices entre ellas, y por los cambios de sentido de las caras que pertenecen a una componente girada.

Así pues, una vez comprobada la planaridad del grafo de adyacencia ante la nueva relación introducida, se presenta geométricamente su dual —o el pseudo-dual en caso de no estar triangulado el grafo de adyacencia—. De esta forma el diseñador posee siempre información gráfica de la distribución en planta que cumple con todos los requisitos que ha introducido hasta el momento.

CONCLUSIONES

Con este trabajo se aporta un método completo y coherente de generación mediante ordenador de distribuciones de edificios en planta, del cual podríamos destacar —además de los algoritmos originales sobre manipulación de grafos— los siguientes puntos:

- 1) Casi desde el comienzo del proceso —cuando el grafo de adyacencia que estructura la naturaleza del problema sea biconexo y planar— el método que se propone es capaz de obtener y representar gráficamente una distribución en planta que satisfaga todos los requisitos introducidos por el operador hasta ese momento. De esta forma, se facilita al operador la introducción de nuevos requisitos.
- 2) Esas distribuciones en planta generadas automáticamente pueden tener un contorno irregular y espacios vacíos interiores; esto puede sugerir nuevas formas de plantas al diseñador distintas a la rectangular.

Con todo esto, creemos aportar un nuevo método de generación y optimización de distribuciones en planta de edificios, apoyado sobre una base de nuevas formulaciones teóricas, a partir de las cuales, pueden desarrollarse futuras investigaciones en el campo del diseño arquitectónico asistido por ordenador, que hagan de éste una poderosa herramienta de diseño al servicio del arquitecto.

EJEMPLO

Consideremos que deseamos diseñar la planta de una vivienda unifamiliar aislada con el siguiente programa:

- | | | |
|----------------|----------------|------------|
| – Vestíbulo | – Dormitorio 1 | – Cocina |
| – Distribuidor | – Dormitorio 2 | – Lavadero |
| – Aseo | – Dormitorio 3 | – Garage |
| – Baño | – Comedor | – Estar |

Con una superficie y dimensiones mínimas de:

Local	Long. min.	Sup. min.
Planta total	6.00	—
Vestíbulo	1.50	—
Distribuidor	1.00	—
Aseo	1.40	—
Baño	1.70	—
Dormitorio 1	3.00	12.00
Dormitorio 2	3.00	12.00
Dormitorio 3	3.00	—
Comedor	3.00	—
Cocina	2.50	10.00
Lavadero	2.00	—
Garage	2.50	15.00
Estar	3.00	18.00

Numeramos las estancias a partir del cinco —pues del uno al cuatro quedan reservados a las orientaciones básicas—:

- | | | | |
|----------|-----------------|------------------|--------------|
| 1. Norte | 5. Vestíbulo | 9. Dormitorio 1 | 13. Cocina |
| 2. Oeste | 6. Distribuidor | 10. Dormitorio 2 | 14. Lavadero |
| 3. Sur | 7. Aseo | 11. Dormitorio 3 | 15. Garage |
| 4. Este | 8. Baño | 12. Comedor | 16. Estar |

Una vez numeradas las estancias introducimos las relaciones de adyacencia siguientes:

Locales adyacentes

5-6	5-13	7-8	9-10	11-6	12-16
5-7	7-6	7-9	10-11	6-12	16-1 11-16

A continuación solicitamos que nos informe del estado del grafo de adyacencia introducido hasta el momento, y se obtiene el siguiente resultado:

Grafo NO CONEXO.

Vértices de la componente conexa 1:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16.

Vértices de la componente conexa 2:

14.

Vértices de la componente conexa 3:

15.

La componente conexa 1 no es BICONEXA.

Vértice de articulación en “1” que articula las componentes:

2, 3, 4 con 5, 6, 7, 8, 9, 10, 11, 12, 13, 16.

Vértice de articulación en “16” que articula las componentes:

1, 2, 3, 4 con 5, 6, 7, 8, 9, 10, 11, 12, 13.

Vértice de articulación en “5” que articula las componentes:

1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 16 con 13.

Así pues, introducimos a continuación las siguientes relaciones de adyacencia:
 13-12 13-15 15-4 14-3 10-2 2-9 13-14.

Solicitamos de nuevo información sobre el estado del grafo de adyacencia introducido hasta el momento y obtenemos el resultado de que es PLANAR. Por consiguiente, se pasa automáticamente a representar el esquema adimensional de distribución en planta que figura en la página siguiente. Sobre este esquema adimensional se irán introduciendo todos los requisitos adicionales que se deseen, y se irán representando sucesivos esquemas adimensionales de distribuciones que satisfacen los requisitos introducidos (Figuras 1-9).

En este paso decidimos no introducir más requisitos de adyacencia, y deseamos convertir el esquema representado en el definitivo, es decir, los espacios “ficticios” vacíos que se han generado automáticamente pasan a ser “reales”.

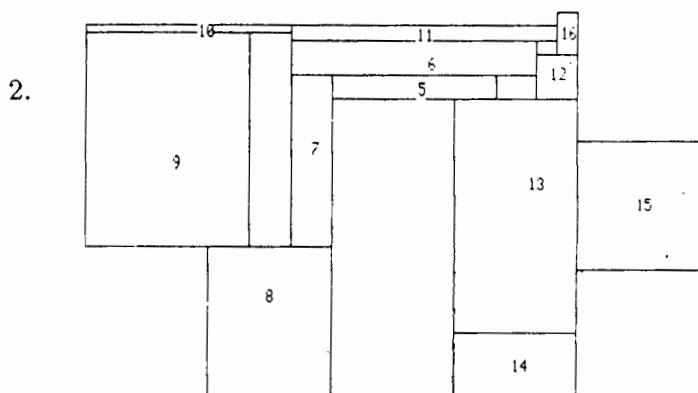
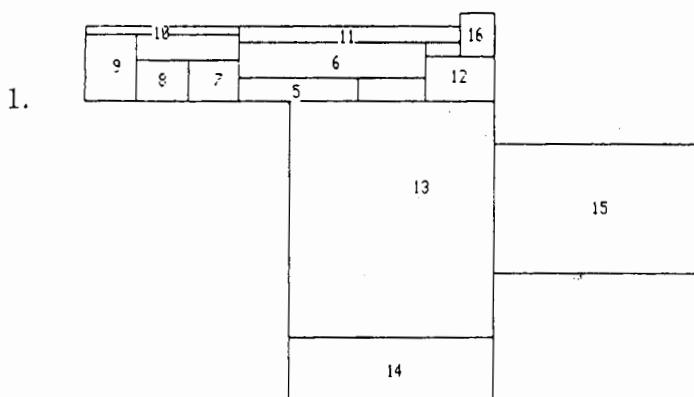
A continuación se representa un esquema adimensional de distribución en planta, en la que quedan señaladas aquellas particiones que poseen una orientación fija (Figura 10).

En este caso podemos, además, fijar la orientación de tantas particiones como se desee. En el ejemplo concreto que se trata, fijamos las siguientes orientaciones:

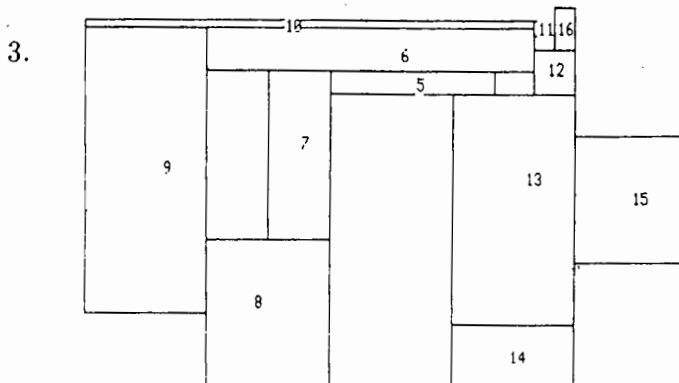
Partición entre locales	Orientación
6-13	Horizontal
6-12	Vertical
6-11	Horizontal
6-10	Vertical
6-9	Horizontal
13-12	Horizontal
12-15	Vertical
10-11	Vertical

Se finaliza el ejemplo con la generación de todos los esquemas adimensionales que satisfacen los requisitos impuestos, y sus correspondientes esquemas optimizados dimensionalmente. La función objetivo elegida en este ejemplo ha sido:

$$\min.: \sum_{i=1,n} x_i + \sum_{j=1,n} y_j$$



Introducir nueva arista (<0 salir, =0 real->pseudo): -6 - 3

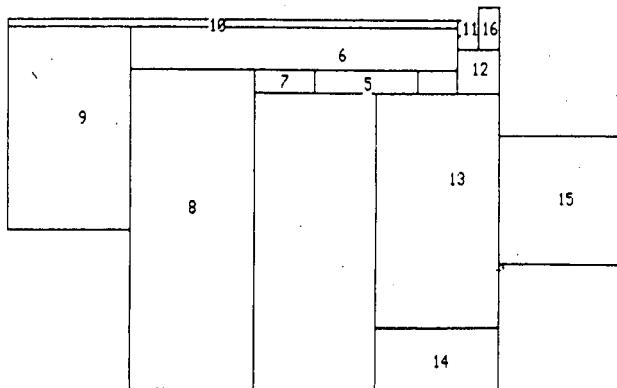


Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 3

Introducir nueva arista (<0 salir, =0 real->pseudo): 9 - 6

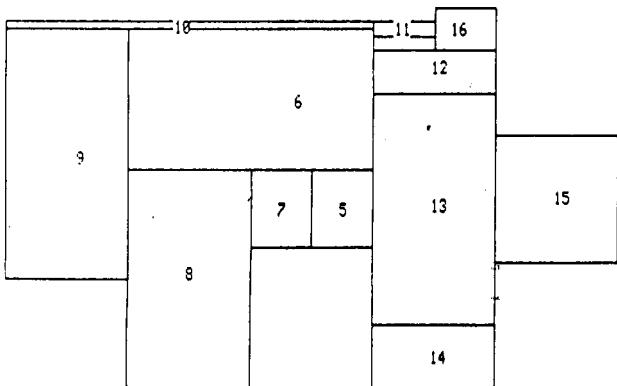
Introducir nueva arista (<0 salir, =0 real->pseudo): -10 - 6

4.



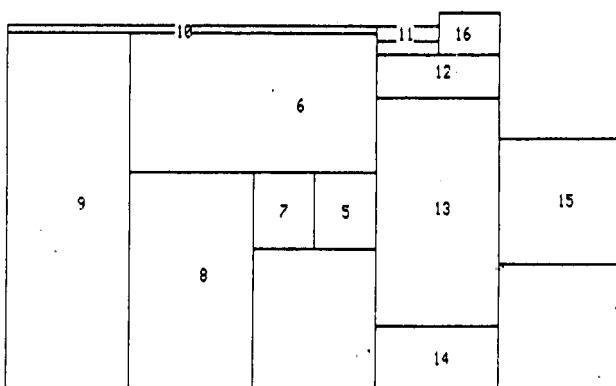
Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 3
 Introducir nueva arista (<0 salir, =0 real->pseudo): 9 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -10 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 6

5.



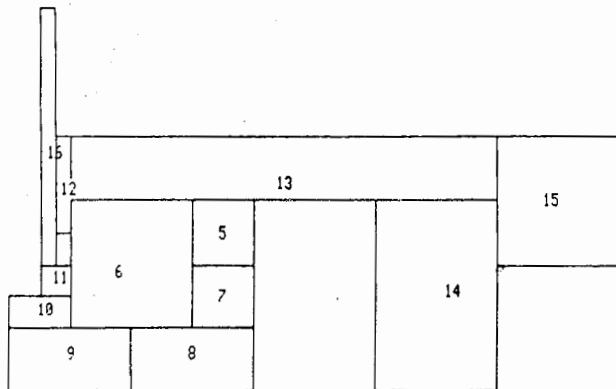
Introducir nueva arista (<0 salir, =0 real->pseudo): 9 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -10 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -6 - 13

6.



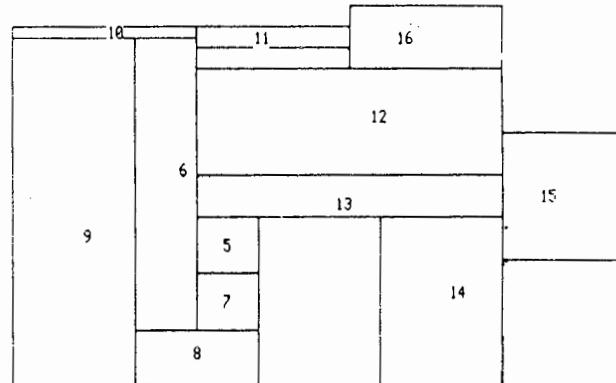
Introducir nueva arista (<0 salir, =0 real->pseudo): -10 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 6
 Introducir nueva arista (<0 salir, =0 real->pseudo): -6 - 13
 Introducir nueva arista (<0 salir, =0 real->pseudo): -9 - 3

7.



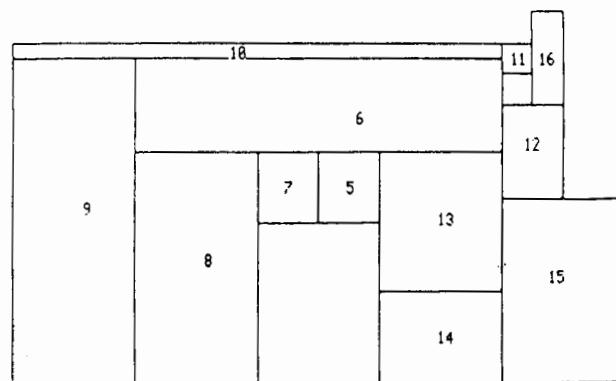
Introducir nueva arista (<>0 salir, =0 real->pseudo): -8 - 6
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -6 - 13
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -9 - 3
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -15 - 14

8.



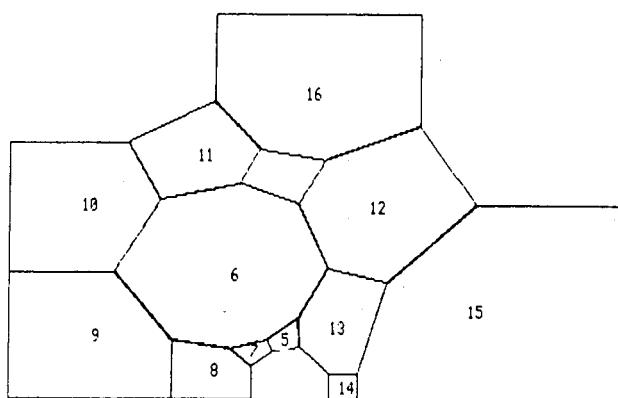
Introducir nueva arista (<>0 salir, =0 real->pseudo): -6 - 13
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -9 - 3
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -15 - 14
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -15 - 12

9.

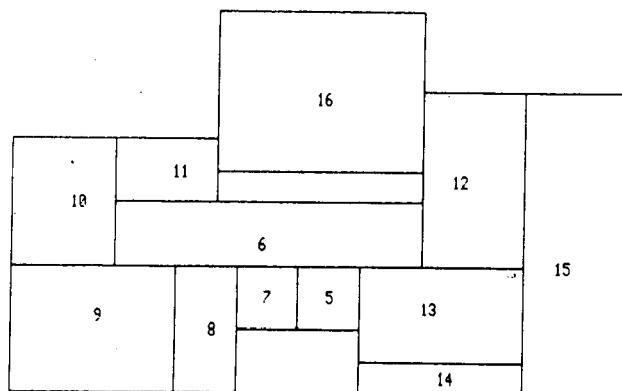


Introducir nueva arista (<>0 salir, =0 real->pseudo): -9 - 3
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -15 - 14
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -15 - 12
 Introducir nueva arista (<>0 salir, =0 real->pseudo): -15 - 3

10.

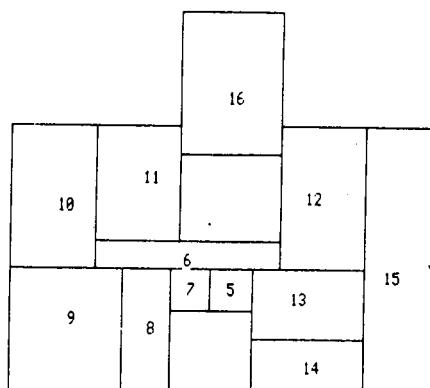


11.



SOLUCION 1

12.

SOLUCION 1
(dimensionada)

REFERENCIAS

1. I. Baybars y C.M. Eastman, "Enumerating architectural arrangements by generating their underlying graphs", *Environment & Planning B*, Vol. 7, pp. 289-310, (1980).
2. C.M. Eastman, "Representations for space planning", *Communications of the ACM*, Vol. 13, no.4, pp. 242-250, (1970).
3. U. Flemming, "Wall representations or rectangular dissections and their use in automated space allocation", *Environment & Planning B*, Vol. 5, pp. 215-232, (1978).
4. J. Gilleard, "LAYOUT —a heretical computer model for the production of architectural floor plans", *Environment & Planning B*, Vol. 5, pp. 233-241, (1978).
5. J. Hopcroft y R. Tarjan, "Efficient algorithms for graph manipulation", *Communications of the ACM*, Vol. 16, pp. 372-378, (1973).
6. J. Hopcroft y R. Tarjan, "Efficient Planarity Testing", *Journal of the association for computing machinery*. Vol. 21, no.4, pp. 549-568, (1974).
7. M. Krejcirik, "Computer-aided plant layout", *Computer Aided Design*, Otoño, pp. 7-19, (1969).
8. P.H. Levin, "Use the graphs to decide the optimun layout of buildings", *The Architect's Journal Information*, Vol. 7, pp. 809-815, (1964).
9. W.J. Mitchell, J.P. Steadman y R.S. Ligget, "Synthesis and optimisation of small regular floor plans", *Environment & Planning B*, Vol. 3, pp. 37-70, (1976).
10. W.J. Mitchell, *Computer-Aided Architectural Design*, Mason/Charter Publisher, Nueva York, (1977).
11. J.J. Sendra, "Generación y optimización automática de esquemas de distribución de edificios en planta", *Tesis doctoral*, E.T.S. de Arquitectura, Universidad de Sevilla, (1984).
12. E. Shavid y D. Gali, "A model for space allocation in complex buildings: A computer graphic approach", *Build International*, Vol. 7, pp. 439-518, (1984).
13. J.P. Steadman, *Architectural Morphology*, Pion Limited, Londres, (1983).
14. W.T. Tutte, "How to draw a graph", *Proc. London Math. Soc.*, Vol.3, no.13, pp. 743-768, (1963).