

~~TESIS  
CAN~~

Tesis  
CAN

PROCEDIMIENTO INTERACTIVO PARA LA GENERACION Y  
OPTIMIZACION DE ORGANIZACIONES  
ESPACIALES EN PLANTA

Tesis doctoral presentada por:

Jacinto Canivell Rodriguez

Dirigida por:

Juan José Sendra Salas

Dr. Arquitecto

Profesor titular de Construcciones Arquitectónicas.

E.T.S.A. de Sevilla.

---

# INDICE

---

I. INTRODUCCION .....	5
II. GENERACION AUTOMATICA DE ESQUEMAS DE DISTRIBUCION ..	9
II.1. ESTADO ACTUAL DEL PROBLEMA .....	9
II.1.1. Etapa adimensional .....	10
II.1.2. Dimensionamiento .....	11
II.2. PROPUESTA DE TRABAJO .....	13
II.2.1. Generación de esquemas adimensionales.	14
II.2.2. Dimensionamiento .....	17
III. DESCRIPCION DE LOS ALGORITMOS DESARROLLADOS .....	20
III.1. GENERACION DE ESQUEMAS ADIMENSIONALES .....	21
III.1.1. Condiciones iniciales .....	21
III.1.2. Conectividad .....	22
III.1.3. Biconectividad .....	26
III.1.4. Primera comprobación de la planaridad	28
III.1.5. Obtención de las caras .....	33

III.1.5.1. Ordenación de las adyacencias	36
III.1.5.2. Generación de las caras .....	37
III.1.6. Estudio de las componentes escindidas	40
III.1.7. Obtención del pseudo-dual .....	44
III.1.8. Introducción de nuevos requisitos ...	47
III.1.8.1. Formación de las listas de inclusión de componentes .....	49
III.1.8.2. Exploración del árbol de inclusión .....	49
III.1.8.3. Búsqueda del par de vértices a unir .....	53
III.1.8.4. Giro de las componentes .....	53
III.1.9. Generación de esquemas adimensionales	54
III.1.9.1. Obtención de las aristas del dual con orientación fija .....	55
III.1.9.2. Generación de soluciones admisibles .....	57
III.1.10. Trazado de esquemas adimensionales .	62
III.1.10.1. Obtención de las clases .....	63
III.1.10.2. Obtención de las subclases ..	64
III.1.10.3. Trazado de la subclases horizontales y verticales .....	64
III.2. DIMENSIONAMIENTO .....	67
III.2.1. Restricciones métrico-geométricas ...	68
III.2.2. Restricciones de rectangularidad de planta y locales .....	69
III.2.3. Restricciones de accesibilidad .....	72
III.2.4. Función objetivo .....	75

III.2.5. Resolución del problema de optimización 76

IV. CONCLUSIONES .....	77
ANEXO I. PROGRAMA DE ORDENADOR .....	80
ANEXO II. EJEMPLOS .....	142
ANEXO III. ALGUNOS CONCEPTOS DE LA TEORIA DE GRAFOS .....	223
BIBLIOGRAFIA .....	228

---

## I. INTRODUCCION

---

Fueron numerosos los investigadores que en la década de los años sesenta mostraron su entusiasmo acerca de la aplicación del ordenador al diseño arquitectónico, y auguraban una radical transformación en la práctica de la arquitectura. Sin embargo, esta transformación anunciada no se ha producido hasta el momento presente. Mientras tanto, en otras disciplinas ajenas al diseño arquitectónico el ordenador se ha convertido en una herramienta poderosa de ayuda al diseño, reemplazando al diseñador en unos casos o asistiéndolo en otros.

El desarrollo de los primeros programas de diseño asistido por ordenador -Computer Aided Design (CAD)- comienza en los años sesenta, siendo su principal pionero Ivan E. Sutherland, quien en el año 1963 presentaba en el Spring Joint Computer Conference un programa elaborado en el Instituto Tecnológico de Massachusetts -denominado SKETCHPAD SYSTEM- que generó un gran interés dentro del ámbito de la ingeniería. Este era un programa de CAD orientado al diseño en la ingeniería, y permitía la manipulación de cualquier objeto mediante un terminal gráfico interactivo.

Paralelamente se desarrollaron otros trabajos, entre ellos cabe destacar el elaborado por la IBM para la General Motors denominado DAC-1, para el campo específico del diseño de automóviles, y que fue el precursor de otros muchos trabajos que se desarrollaron posteriormente sobre este tema. Desde entonces, los programas interactivos de CAD han sido ampliamente usados en los diversos sectores de la mecánica, óptica,

química y otras industrias.

Sin embargo, la aplicación de programas de CAD en la arquitectura se ha retrasado considerablemente en relación con otras disciplinas. Varias son las razones que parecen haber contribuido a este retraso, entre ellas: la hostilidad o desconfianza mostrada hacia la nueva máquina por el colectivo de arquitectos, la ignorancia del potencial de la tecnología computacional en el campo del diseño arquitectónico, y la dificultad de implementación sobre un ordenador -por la propia naturaleza del diseño arquitectónico- de un programa capaz de emular las decisiones del diseñador.

Pero además, una razón que indudablemente ha influido en este sentido ha sido la económica: las grandes firmas automovilísticas y aeroespaciales han sido capaces de realizar, por su gran capacidad económica, grandes inversiones en investigación y equipos costosos para el desarrollo de sistemas de diseño, control y manufacturación asistidos por ordenador, mientras que, la capacidad económica media de los estudios de arquitectura no ha sido la suficiente como para permitir realizar este esfuerzo.

No obstante, el enorme interés despertado entre algunos grupos de arquitectos por el potencial de los programas de CAD aplicados a la arquitectura no ha decrecido durante todos estos años. Cuando en el año 1964 aparece el libro de C. Alexander titulado "Notes on the Syntesis of Form" produce una auténtica revolución en la comunidad académica y en todo el ámbito arquitectónico, generando un considerable entusiasmo e interés en la utilización de métodos sistemáticos computacionales de diseño en la arquitectura.

Durante esos años se desarrollaron los primeros programas experimentales de diseño arquitectónico asistido por ordenador, como pueden ser COPLANNER de Souder & Clark 1964 y URBAN 5 de Negroponte & Groisser 1970 -ambos citados por Mitchell

(1977)-. Sin embargo, las primeras aplicaciones prácticas aparecidas en el mercado fueron principalmente dirigidas al campo del cálculo de estructuras, mediciones y presupuestos, etc., pues los programas de aplicación a la arquitectura se desarrollaron paralelamente e incluso formando parte de los desarrollados para la ingeniería. Por esta razón, además, un gran número de programas de CAD que se ofrecen a los arquitectos no satisfacen las condiciones que un programa de diseño arquitectónico asistido por ordenador habría de cumplir.

No obstante, hay que destacar una cuestión que nadie parece poner en duda, como ya comentara Alexander (1964), y es la capacidad limitada del ordenador para reemplazar al arquitecto en ciertas etapas decisorias del proceso de diseño, y en ciertas labores evaluativas del mismo; el ordenador nunca será capaz de reconocer que se encuentra ante una solución de singular valor creativo, facultad exclusiva del buen diseñador.

Según Mitchell (1974) un programa de diseño arquitectónico asistido por ordenador debe ser capaz de realizar las siguientes operaciones:

- 1) Almacenamiento, recuperación y manipulación de los datos que nos describen el objeto de diseño.

En esta operación la potencia del ordenador se emplea en la eficiente recuperación de la información requerida y su transformación, por ejemplo, en una lista alfanumérica ordenada, o en un gráfico en pantalla. De esta forma, el ordenador reemplazaría o ayudaría a la labor de delineación, elaboración de presupuestos, etc., en otras palabras, apoyaría las fases de mano de obra de producción.

- 2) Generación automática de soluciones al problema de diseño definido.

En este caso, el ordenador reemplaza o ayuda, en las primeras etapas del proceso de diseño, a alguna de las tradicionales funciones del arquitecto.

- 3) Evaluación de las distintas soluciones generadas según las condiciones, restricciones y objetivos impuestos.

Aquí el ordenador realiza tareas evaluativas aprendidas y desarrolladas tradicionalmente por el arquitecto u otros especialistas.

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, dado que la mayor parte de estos han sido desarrollados, mediante la extrapolación, a partir de programas específicos de ingeniería y por ingenieros. 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. Se centra, pues, en la etapa inferior del proceso de diseño; etapa en la que casi todo el mundo parece admitir que se producen los aspectos decisivos y creativos más importantes de todo el proceso en cuestión.

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 arquitecto diseñador. En definitiva, con esta investigación pretendemos aumentar las posibilidades del ordenador como instrumento válido en el proceso de diseño arquitectónico.

---

**II. GENERACION AUTOMATICA DE  
ESQUEMAS DE DISTRIBUCION**

---

---

## II. 1. 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 -Levin (1964), Krejcirik (1969), Eastman (1970), Shaviv y Gali (1974)- son de una naturaleza tal que permiten plantear el problema de la generación de una distribución en planta como un problema de optimización, mediante el planteamiento de una función objetivo a minimizar, en la que intervienen factores como la longitud del recorrido entre dos estancias, frecuencia de dicho recorrido, etc.. Además de este tipo de restricciones, pueden aparecer otras limitaciones: adyacencia entre locales, orientación, acceso, etc.

Sin embargo, otros autores tratan el problema de la generación de una forma totalmente diferente a las citados anteriormente. Consideran que dichas restricciones sólo serán las debidas a: la adyacencia o no adyacencia entre locales, condiciones de accesibilidad entre los mismos, condiciones de orientación, y limitaciones dimensionales de la organización en planta y de los espacios que la componen. Así tratan el problema, entre otros, Grason (1970), Mitchell, Steadman y Liggett (1976), Flemming (1978), Gilleard (1978) y Sendra (1984).

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 genera una o todas las soluciones de distribución en planta que cumplan con los requisitos impuestos, y una segunda etapa en la que se dimensionan los esquemas adimensionales generados.

Veamos a continuación como suele ser abordada cada una de estas dos etapas.

### II.1.1 ETAPA ADIMENSIONAL

En esta fase se generan los esquemas adimensionales de distribución en planta que cumplan con las limitaciones impuestas. Atendiendo al número de soluciones generadas, estos métodos podrían clasificarse en:

a) *Métodos exhaustivos*: si generan todas aquellas soluciones que satisfagan las condiciones impuestas inicialmente. Así lo tratan, entre otros, Grason (1970), Gilleard (1978), Mitchell, Steadman y Liggett (1976).

El método propuesto por Mitchell, Steadman y Liggett consiste, en esencia, en generar todas las posibles distribuciones en planta para "n" locales, y una vez generadas seleccionar aquéllas que cumplan con los requisitos de

adyacencia y orientación entre locales.

En estos métodos el número de soluciones posibles crece de forma casi exponencial con el número de locales que se considere, por lo que el número de estos, prácticamente, queda limitado a no más de ocho o nueve.

b) Métodos *constructivos o heurísticos*: si se genera una solución -Sendra (1984)- o varias -Flemming (1978)- de todas las posibles distribuciones espaciales en planta que cumplan con las condiciones impuestas inicialmente.

La solución o soluciones que se generan se van modelando a lo largo del proceso mediante la introducción gradual de nuevos requisitos de adyacencia u orientación, comenzando por aquéllos que se estimen más importantes.

Casi todos los autores citados se han basado en la teoría de grafos. En efecto, desde los primeros trabajos -Levin (1964)- se parte de la idea de que las condiciones de adyacencia de los espacios a ordenar pueden expresarse mediante un grafo, cuyo dual será un esquema de distribución que cumpla con las condiciones fijadas. No obstante, Fleming (1978) utiliza una técnica de representación distinta, basada en una propuesta propia ("wall representations").

Otra característica, común a todos estos trabajos, es que manejan 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.

### II.1.2. DIMENSIONAMIENTO

Todos los trabajos realizados coinciden en tratar esta etapa como un problema de optimización. La formulación general de

un problema de este tipo viene dada por el establecimiento de una función objetivo y unas restricciones a cumplir por las variables que en ella intervienen. Estas variables, en el caso que nos ocupa, son la longitud y anchura de cada uno de los espacios que componen la planta, o también, las superficies tanto de cada local como de la planta completa.

La función objetivo se establece en función del aspecto que se desee maximizar o minimizar del esquema de distribución generado. Las más usuales son: minimizar el perímetro, su longitud o anchura, su proporción o su superficie. Así pues, unas funciones objetivos serán lineales y otras no lineales.

Las restricciones que han de cumplir las variables antes citadas las podemos clasificar en:

a) Condiciones de rectangularidad del contorno de la planta y de cada uno de los locales que la componen.

Para expresar estas restricciones se reduce el esquema adimensional a dos grafos orientados, dirigidos y ortogonales entre sí -uno vertical y otro horizontal-, a los que se aplican las leyes de Kirchhoff para las redes eléctricas.

b) Restricciones que traducen las condiciones de accesibilidad entre parejas de locales adyacentes.

c) Limitaciones impuestas a las dimensiones de los locales y de la planta general, ya se refieran a su longitud, anchura o superficie.

La naturaleza lineal o no lineal de la función objetivo y de las restricciones establecidas nos determinan el método de resolución a emplear en el problema de optimización.

---

## II.2. PROPUESTA DE TRABAJO

---

Este trabajo pretende continuar la línea de investigación iniciada por J. J. Sendra con su tesis doctoral "Generación y optimización automática de esquemas de distribución de edificios en planta".

Sendra proponía un método *constructivo* de generación subdividido en las dos fases señaladas en II.1: en la primera se obtienen esquemas adimensionales que cumplen con las condiciones impuestas inicialmente, y en la segunda se obtienen los esquemas dimensionados mediante la definición de las restricciones métrico-geométricas.

El procedimiento propuesto por Sendra introduce gradualmente los requisitos de adyacencia y/u orientación entre los distintos espacios que componen la planta, hasta conseguir, mediante la triangulación total del grafo de adyacencia, que la representación de dicho grafo sea única y, por tanto, su dual -esquema adimensional de la planta- sea también único.

Este método obliga a introducir gradualmente restricciones -a lo largo del proceso- para obtener una solución única, lo cual lleva consigo eliminar otras soluciones que pudieran ser, asimismo, contempladas por el operador. Aunque cada una de estas últimas pudieran ser obtenidas modificando la naturaleza de las restricciones, sólo se podría generar una en cada ciclo iterativo, lo que a la larga supone restar operatividad al mismo.

Además, todos estos requisitos se introducen sobre el trazado

geométrico del grafo de adyacencia -modelo de difícil lectura en algunos casos- y no mediante la observación directa de la organización espacial en planta que se está elaborando.

Así pues, lo que se pretende con este trabajo es proporcionar un método de generación de distribuciones espaciales en planta que obtenga no una, sino todas las posibles soluciones que cumplan con las condiciones introducidas hasta un determinado momento. De esta forma, cualquier decisión de adición de nuevas restricciones se verá enormemente asistida al disponer de una o varias representaciones de la organización u organizaciones espaciales posibles hasta ese momento.

El procedimiento que se presenta lo dividiremos, como el anterior, en las dos fases ya mencionadas:

#### II.2.1. GENERACION DE ESQUEMAS ADIMENSIONALES

En esta fase se generarán todos los posibles esquemas adimensionales de distribuciones espaciales en planta que cumplan con los requisitos impuestos.

Este procedimiento de generación se divide, a su vez, en las siguientes etapas:

##### 1) Introducción de las condiciones iniciales.

Se definen el número y naturaleza de los locales que van a formar parte de la futura organización en planta, así como las relaciones de adyacencia y orientación entre los mismos que se consideren más importantes. 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 cumpla 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.

9) Generación de todos los esquemas adimensionales.

En el momento que el operador haya introducido todos los requisitos que crea necesarios, o cuando el grafo de adyacencia sea un grafo triangulado (planar máximo), se pasaría a generar todas las posibles soluciones de planta que cumplan con los requisitos impuestos hasta ese momento.

Así pues, las diferencias principales con el procedimiento propuesto por Sendra son las siguientes:

- a) Genera más de una solución: todas aquellas que son compatibles con los requisitos impuestos.
- b) No hace falta esperar a que el grafo de adyacencia sea un grafo triangulado para poder obtener y trazar su dual, sino que desde el momento en que dicho grafo de adyacencia sea biconexo y planar, podemos trazar ya un primer esquema de planta que responda a los requisitos introducidos hasta entonces.
- c) La introducción de nuevas condiciones se realiza, pues,

sobre el trazado gráfico de un esquema de distribución en planta, y no sobre el trazado del grafo de adyacencia.

- d) Se propone un algoritmo más sencillo para el trazado geométrico de los esquemas adimensionales de distribución en planta.
- e) Los esquemas de distribución en planta que se generan no se limitan a aquéllos de contorno rectangular, sino que automáticamente pueden generarse distribuciones en planta de contorno irregular y con espacios interiores vacíos que satisfagan las condiciones impuestas por el operador. Estos esquemas corresponderían a los trazados de los grafo pseudo-duales.

Todo el proceso descrito queda detallado ampliamente en el capítulo siguiente.

### II.2.2. DIMENSIONAMIENTO

En esta fase se desarrollan los métodos de dimensionamiento de los esquemas adimensionales de planta que se generaron en la fase anterior. Hemos entendido su resolución, como la gran mayoría de los autores, como un problema de "optimización", y emplearemos únicamente para su resolución la programación lineal, como propone Fleming (1978), mediante la linealización de las restricciones y función objetivo que no sean lineales.

El procedimiento que se propone para la optimización dimensional de los esquemas de distribución puede dividirse, pues, en las siguientes etapas:

- 1) Formulación de la función objetivo.

Las funciones objetivos contempladas se reducen sólo a aquellas que son lineales, como pueden ser: la suma del

perímetro exterior, longitud, anchura, etc.

- 2) Introducción de las restricciones métrico-geométricas de la planta y los locales.

Serán aquéllas que limitan superior o inferiormente las dimensiones de la planta y los locales. También se contempla la posibilidad de limitar las superficies, pero en este caso, estas expresiones -no lineales- se aproximan automáticamente a otras que son lineales.

- 3) Obtención de las restricciones que reflejan la condición de rectangularidad de las estancias.

Se procede en esta etapa a obtener automáticamente, a partir del grafo de adyacencia, dos grafos dirigidos y ortogonales que nos definen las direcciones de cada partición o tabique. Sobre estos grafos se aplican las leyes de Kirchhoff para redes eléctricas.

- 4) Obtención de las expresiones que traducen las condiciones de accesibilidad entre locales.

A partir de los dos grafos dirigidos citados anteriormente, se obtienen las expresiones que aseguran la accesibilidad entre dos locales que sean adyacentes, a través de un hueco de paso.

- 4) Resolución del problema de optimización.

Emplearemos métodos de programación lineal para la resolver el problema de optimización, dado que tanto la función objetivo como las restricciones que se plantean son siempre lineales. De esta forma, aprovechamos la enorme potencia de resolución que posee dicha técnica.

Todo este proceso termina con los trazados de los esquemas ya

dimensionados.

Las diferencias fundamentales con el método de dimensionamiento propuesto por Sendra son las siguientes:

- a) La función-objetivo que se contempla siempre será lineal: minimización del perímetro, del largo o ancho de la planta, etc. Las restricciones métrico-geométricas impuestas que no sean lineales son, a su vez, linealizadas.
- b) Efectúa de forma automática el paso de la etapa adimensional a la dimensional. El operador no necesita elaborar los dos grafos dirigidos mencionados anteriormente: el programa los extrae automáticamente.
- c) Se ofrecen algoritmos distintos y más sencillos para la elaboración automática de las expresiones que traducen las condiciones métrico-geométricas de la planta y las de accesibilidad entre locales adyacentes, a partir de los dos grafos dirigidos ortogonales.
- d) También se presenta un sencillo algoritmo para el trazado de los esquemas ya dimensionados a partir del trazado de los esquema adimensionales.

Todos 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.

Por último, indicar que el programa completo ha sido elaborado e implementado en un micro-ordenador Sinclair QL con sistema operativo Qdos y en lenguaje Superbasic interpretado.

---

**III. DESCRIPCION DE LOS  
ALGORITMOS DESARROLLADOS**

---

---

## III.1. ESQUEMAS ADIMENSIONALES

---

### III.1.1. 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.

Así pues, si un local  $v_i$  se desea orientar al Norte -por ejemplo- en su lista  $ADY(v_i)$  aparecerá el vértice 1 (N). Del mismo modo, las listas de adyacencias de los locales que se

deseen ubicar en las esquinas de la futura organización en planta contendrán, necesariamente, los dos vértices correspondientes a las dos orientaciones de los límites de dichos espacios (orientación compuesta).

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

### III.1.2. CONECTIVIDAD

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 Tarjan (1973) 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.

Pasemos a continuación a describir el procedimiento completo para la comprobación de la conectividad. Comenzaremos explorando el grafo por un vértice inicial arbitrario  $u_1$  y continuaremos realizando las siguientes operaciones consecutivas:

a) Se marca cada vértice  $u_n$  con el número de orden  $n$  que ha sido alcanzado durante la exploración ( $\text{NUM}(u_n)=n$ ) y se guarda sobre un conjunto ordenado, al que llamaremos PILA, definido como:

$$\text{PILA}=\{u_1, \dots, u_n\}$$

siendo:  $n$  el número de vértices explorados hasta el momento,  $u_1$  el vértice inicial, y  $u_n$  el último vértice explorado.

Naturalmente, al inicio del algoritmo el conjunto PILA consistirá en un único vértice, que será precisamente  $u_1$ , por consiguiente, será también el último vértice alcanzado  $u_n$ .

b) Para todo  $v_i \in \text{ADY}(u_n)$ :

b1) Si  $v_i \notin \text{PILA}$ : la arista  $(u_n, v_i)$  no ha sido recorrida en la D.F.S.; dicha arista será una *rama* y la denotaremos como  $u \rightarrow v$ . Entonces, aplicamos de nuevo la operación  $a$  para el vértice  $v_i$ , por lo que pasará a ser el último vértice alcanzado  $u_n$  del conjunto PILA y se marcará con el número de orden en el que ha sido alcanzado.

b2) Si  $v_i \in \text{PILA}$  y la arista  $(u_n, v_i)$  no ha sido recorrida, dicha arista será una *fronda* y la denotaremos como  $u \rightarrow v$ .

El vértice  $v_i$  no pasa a formar parte del conjunto PILA ni se marca con el orden en el que ha sido alcanzado, ya que, por ser extremo de una fronda, ha sido explorado con anterioridad. Así pues, se selecciona el siguiente vértice de la lista  $\text{ADY}(u_n)$  y repetimos la operación  $b$ . Si se ha llegado al final de la lista

ADY( $u_n$ ), se suprime  $u_n$  del conjunto PILA y se repite la operación  $b$  para ADY( $u_{n-1}$ ), siendo ahora  $u_{n-1}$  el último vértice alcanzado.

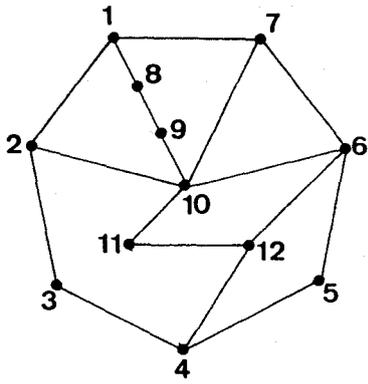
Finaliza este proceso recursivo cuando el conjunto PILA no contiene a ningún vértice como elemento. Entonces, si todos los vértices del grafo han sido marcados, el grafo de adyacencia será conexo. En caso contrario sería inconexo y el conjunto de vértices marcados formaría una componente conexa.

Para la obtención de las restantes componentes conexas se comenzaría de nuevo el proceso por cualquier vértice no marcado. Se obtendrán, así, tantas componentes conexas como veces haya que repetir dicho procedimiento, hasta que todos los vértices del grafo queden marcados.

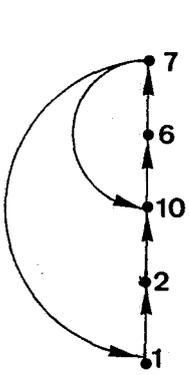
Una vez finalizado el proceso, se informa sobre el estado de conectividad del grafo y se describen las componentes conexas -caso de que existan- para que el operador las conecte entre sí mediante la introducción de las aristas oportunas. En esta etapa, por consiguiente, se puede producir la primera adición 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.

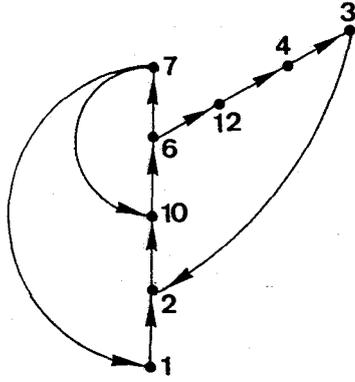
Como puede apreciarse en la figura, este tipo de examen impone una dirección a cada arista atravesada y una nueva numeración ( $NUM(v)$ ) de los vértices según se alcanzan. De igual forma, particiona el conjunto de aristas en dos clases: las *ramas* que unen vértices en orden ascendente -según la nueva 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*.



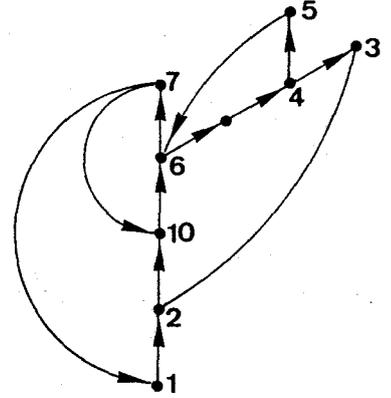
ADY(1)=(2,7,8,3)    ADY(7)=(6,1,10)  
 ADY(2)=(1,10,3)    ADY(8)=(9,1)  
 ADY(3)=(4,2)        ADY(9)=(10,8)  
 ADY(4)=(1,2,3,5,)    ADY(10)=(12,6,7,11,9)  
 ADY(5)=(4,6)        ADY(11)=(12,10)  
 ADY(6)=(10,7,12,5)    ADY(12)=(6,4,11)



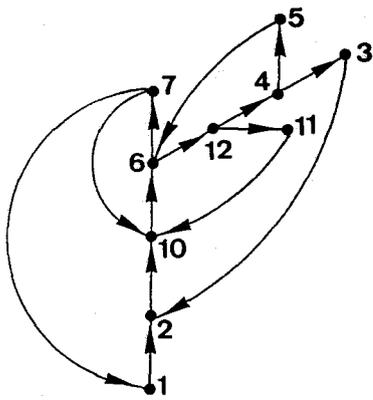
PILA=(1,2,10,6,7)  
 NUM(1)=1  
 NUM(2)=2  
 NUM(10)=3  
 NUM(6)=4  
 NUM(7)=5



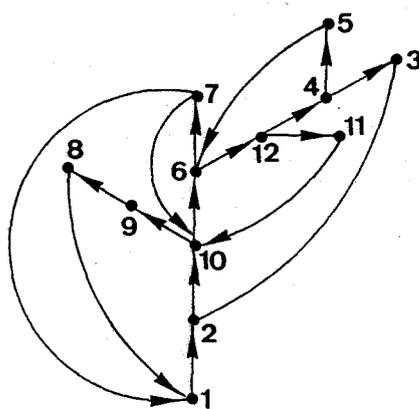
PILA=(1,2,10,6,12,4,3)  
 NUM(12)=6  
 NUM(4)=7  
 NUM(3)=8



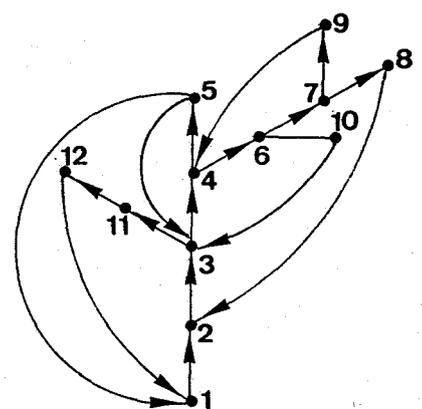
PILA=(1,2,10,6,12,4,5)  
 NUM(5)=9



PILA=(1,2,10,6,12,11)  
 NUM(11)=10  
 NUM(8)=12



PILA=(1,2,10,9,8)  
 NUM(9)=11



ADY(1)=(2)            ADY(7)=(8,9)  
 ADY(2)=(3)            ADY(8)=(2)  
 ADY(3)=(4,11)        ADY(9)=(4)  
 ADY(4)=(5,6)        ADY(10)=(3)  
 ADY(5)=(1,3)        ADY(11)=(12)  
 ADY(6)=(7,10)        ADY(12)=(1)

Figura 1

### III.1.3. 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. No obstante, además de los conjuntos PILA y NUM antes señalados, esta nueva verificación exige definir otros dos nuevos conjuntos:

- Conjunto  $S(v)$  de vértices alcanzados por frondas desde vértices *descendientes* de  $v$ :

$$S(v) = \{w \mid \exists u (R: v \rightarrow u \text{ y } u \rightarrow w)\}$$

siendo  $R: v \rightarrow u$  una *ruta* sobre el árbol  $T$  de  $G_1$  que comienza en  $v$  y termina en  $u$ .

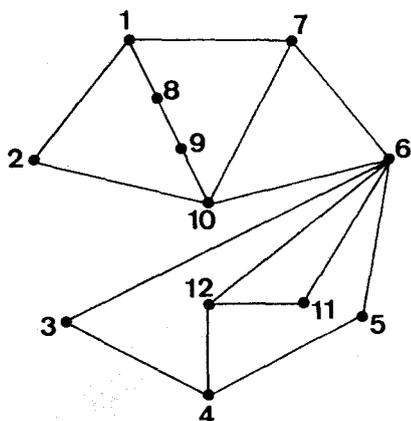
- Conjunto  $B1(v)$  formado por el vértice de menor numeración alcanzado por una fronda desde vértices descendientes de  $v$ :

$$B1(v) = \text{MINIMO}(\{v\} \cup S(v))$$

Definidos estos dos conjuntos, consideremos un cierto estado en la D.F.S en el que la configuración del conjunto PILA sea  $\{u_1, u_2, \dots, u_{n-1}, u_n\}$ . Si se verifica que:

$$\text{NUM}(B1(u_n)) \geq \text{NUM}(u_{n-1})$$

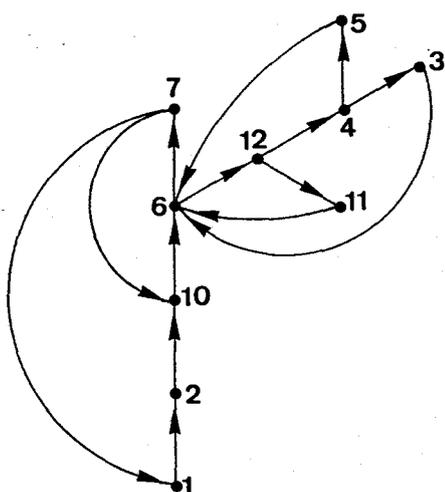
el grafo no es biconexo, y el vértice  $u_{n-1}$   $\in$  PILA es un *vértice de articulación* que nos delimita al menos dos componentes biconexas (Hopcroft-Tarjan 1973). Por lo tanto, al término de la D.F.S. obtendremos todos los posibles vértices que articulan componentes biconexas. (Figura 2).



ADY(1)={2,7,8}  
 ADY(2)={1,10}  
 ADY(3)={4,6}  
 ADY(4)={12,3,5}  
 ADY(5)={4,6}  
 ADY(6)={10,7,12,5,3,11}  
 ADY(7)={6,1,10}  
 ADY(8)={9,1}  
 ADY(9)={10,8}  
 ADY(10)={2,6,7,9}  
 ADY(11)={12,6}  
 ADY(12)={6,4,11}

NUM(1)=1  
 NUM(2)=2  
 NUM(3)=8  
 NUM(4)=7  
 NUM(5)=9  
 NUM(6)=4  
 NUM(7)=5  
 NUM(8)=12  
 NUM(9)=11  
 NUM(10)=3  
 NUM(11)=10  
 NUM(12)=6

BI(1)=1  
 BI(2)=1  
 BI(3)=6  
 BI(4)=6  
 BI(5)=6  
 BI(6)=1  
 BI(7)=1  
 BI(8)=1  
 BI(9)=1  
 BI(10)=1  
 BI(11)=6  
 BI(12)=6



En un cierto estado de la D.F.S. la PILA tiene la siguiente configuración:

PILA={1,2,10,6,12}

El vértice 6 es un vértice de articulación pues:

NUM(BI(12))=NUM(6)

Figura 2

Del mismo modo que en el apartado anterior, el programa informará del estado de biconectividad 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  $G_1$  a pasado a ser un grafo  $G_2$  dirigido, reenumerado 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:  $v \rightarrow w$  ó  $v \dashrightarrow w$ .

#### III.1.4. PRIMERA COMPROBACION DE LA 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 se cumpla la siguiente expresión:

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

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 Tarjan (1974), que suponen una mejora al inicialmente propuesto por Auslander y Parter y posteriormente revisado por Goldstein. 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.

Antes de describir el algoritmo, definamos un nuevo conjunto en el grafo, a añadir a los hasta ahora enunciados:

$$B2(v) = \text{MINIMO}(\{v\} \cup (S(v) - \{B1(v)\}))$$

siendo  $B2(v)$  el vértice alcanzado por una fronda desde vértices descendientes de  $v$ , pero no el de menor numeración -que figuraría en  $B1(v)$ - sino el segundo de menor numeración en la exploración D.F.S.

Una vez definido este conjunto, podemos pasar a describir el procedimiento general de comprobación de planaridad de un

grafo biconexo ( $G_2$ ) que, en síntesis, incluiría las siguientes operaciones:

- a) Obtención de una *ruta*  $R: v \rightarrow v$  en el árbol palmera que comience y termine en el mismo vértice, a la que llamaremos *círculo original*.

Este *círculo original* estará formado por una ruta sobre el árbol  $T$  extendido sobre  $G_2$  ( $R: v \rightarrow w$ ) más una fronda que lo cierra ( $w \rightarrow v$ ).

- b) Eliminación de las aristas de este *círculo*.

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 *círculo*.

- 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 *círculo 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 Jordan, un *segmento* puede ser representado en el plano bien por el exterior, o por el interior de su *círculo original*, sin que se produzcan cruces de aristas. La ubicación en el plano de un *segmento* por el interior (exterior) de su *círculo* 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 completo se ilustra en la figura 3a y 3b.

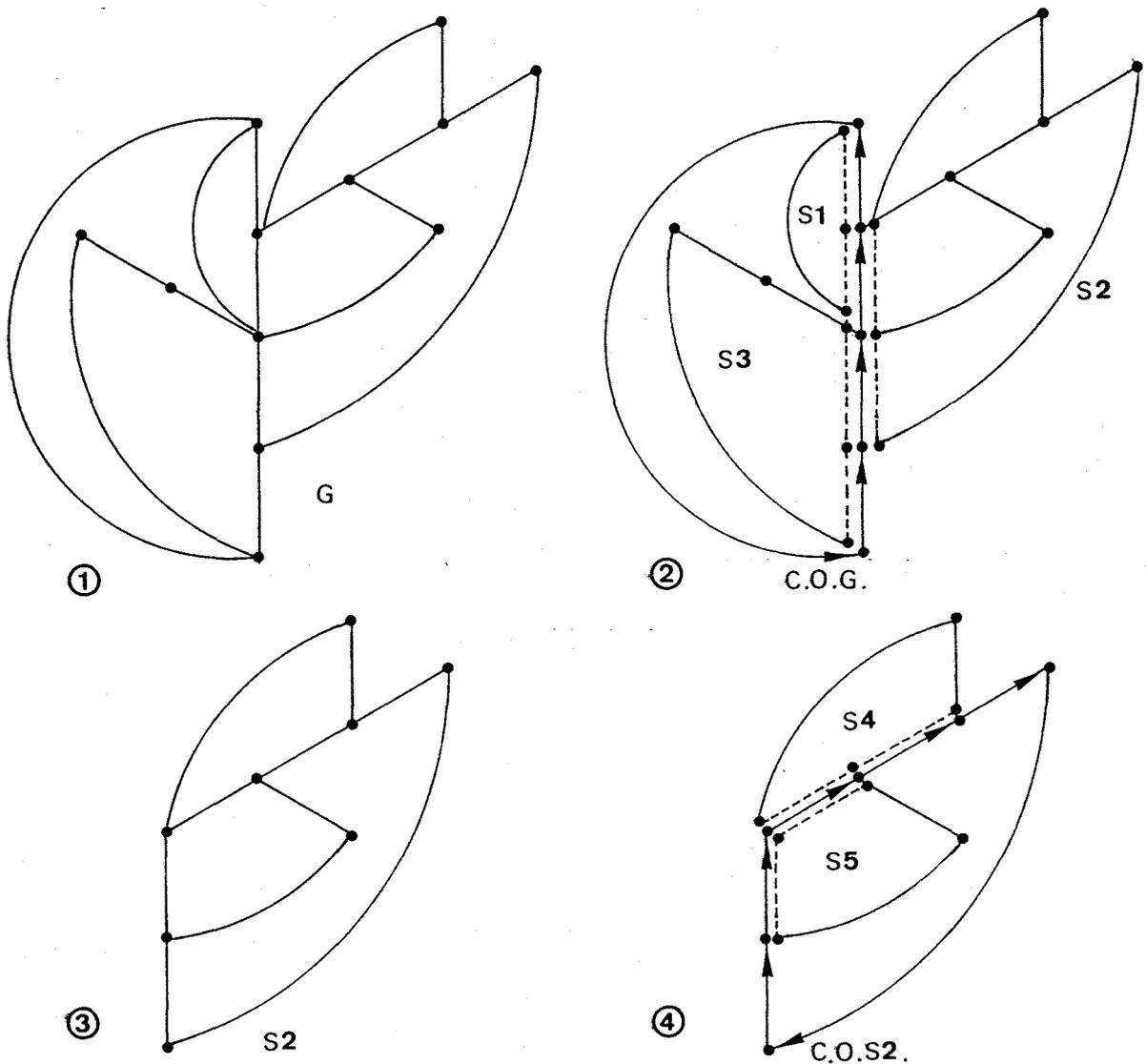


Figura 3a

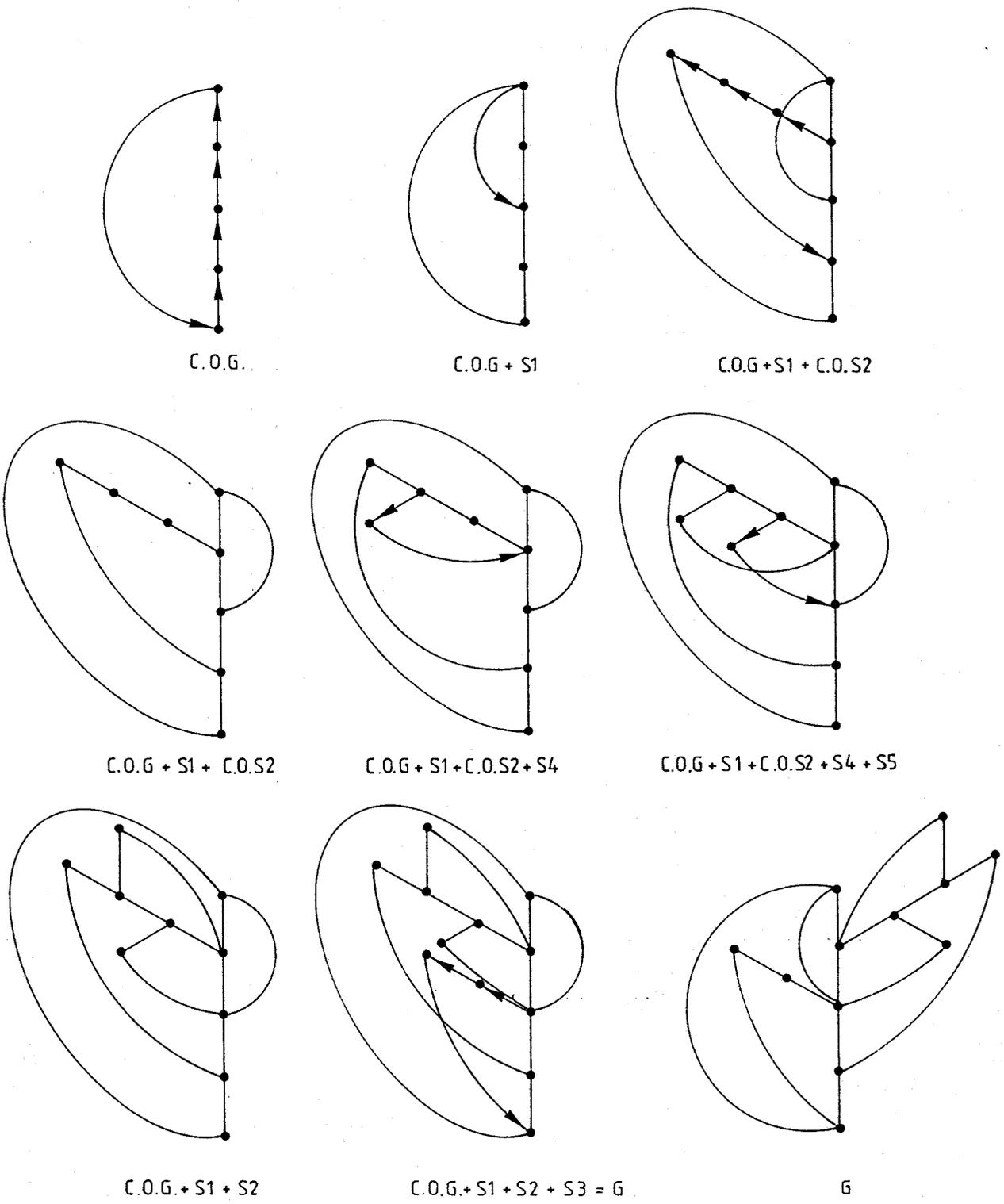


Figura 3b

Todo este proceso descrito se realiza utilizando el algoritmo una sola vez, mediante la aplicación de una nueva D.F.S que: genera todas las *rutas disjuntas*  $R:u \rightarrow v$  (que no contienen aristas en común) del grafo  $G_2$ , y las situa en el plano conforme se van obteniendo.

Para el correcto funcionamiento del algoritmo, las rutas han de ser generadas en un orden determinado. Así pues, se realizará una ordenación de las listas de adyacencias  $ADY(v)$  según una función FI que se define como sigue -Hopcroft, Tarjan (1974):

$$FI((v,w)) = \begin{cases} 2 \times w & \text{si } v \rightarrow w \\ 2 \times B1(w) & \text{si } v \rightarrow w \text{ y } B2(w) \geq v \\ 2 \times B1(w) + 1 & \text{si } v \rightarrow w \text{ y } B2(w) < v \end{cases}$$

Las listas  $ADY(v)$  se ordenarán en orden creciente de la citada función: si  $ADY(v) = (v_1, v_2, \dots)$ , se ha de cumplir que:

$$FI(v, v_1) \leq FI(v, v_2) \leq \dots$$

Recordemos que los vértices en el grafo  $G_2$  se encontraban reenumerados según  $NUM(v)$ , por lo que el valor de la función FI se calcula según la nueva numeración.

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 geoméricamente en el plano.

Tendremos, por tanto, un grafo  $G_3$  que será: dirigido, ordenado según la función FI, y sobre todo, planar. Conocemos, además, dónde se han de ubicar todas sus aristas para que, en una representación plana del mismo, no se produzca ningún cruce entre ellas.

### III.1.5. FORMACION DE CARAS

En esta sección proponemos 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 atraviese 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.

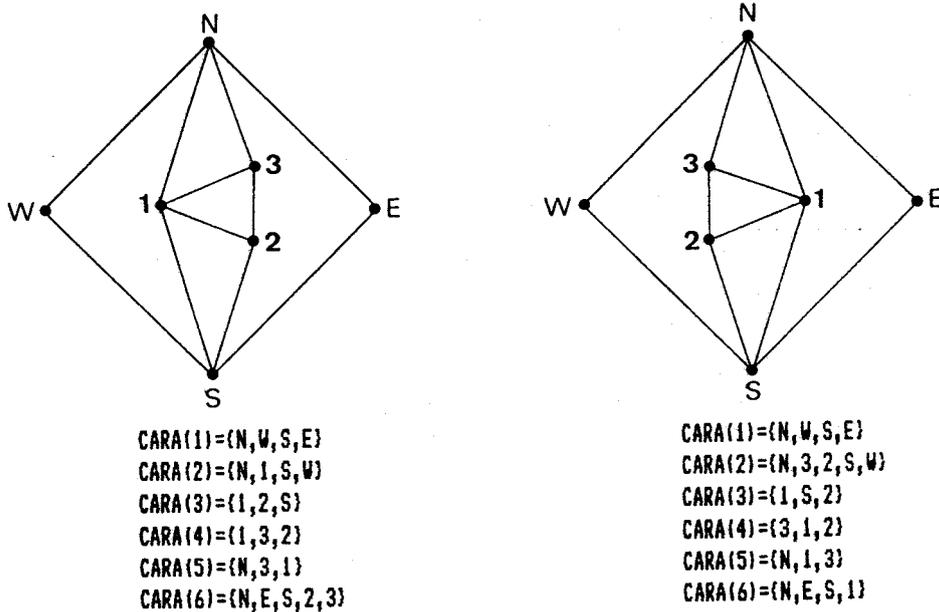


Figura 4

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*.

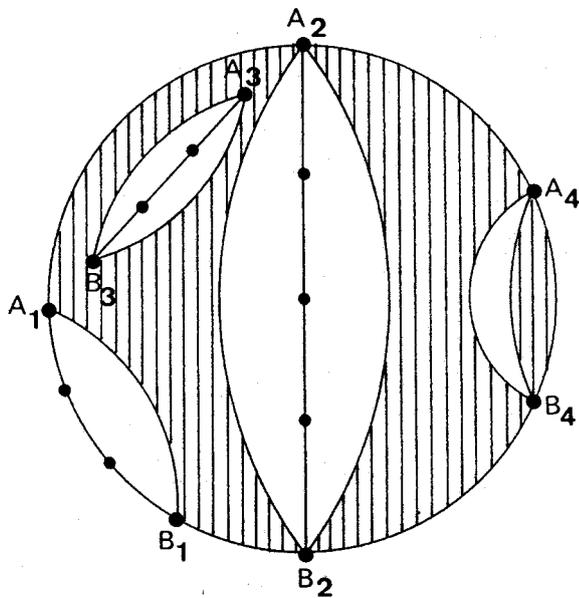
El conjunto de caras sería único, pues el giro de cualquiera de las componentes alrededor del par de corte sería imposible, ya que nos llevaría a la pérdida de la cara exterior (Figura 5).

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).



Caso b1:  $[A_1, B_1]$   
 $[A_4, B_4]$   
 Caso b2:  $[A_2, B_2]$   
 Caso b3:  $[A_3, B_3]$

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)

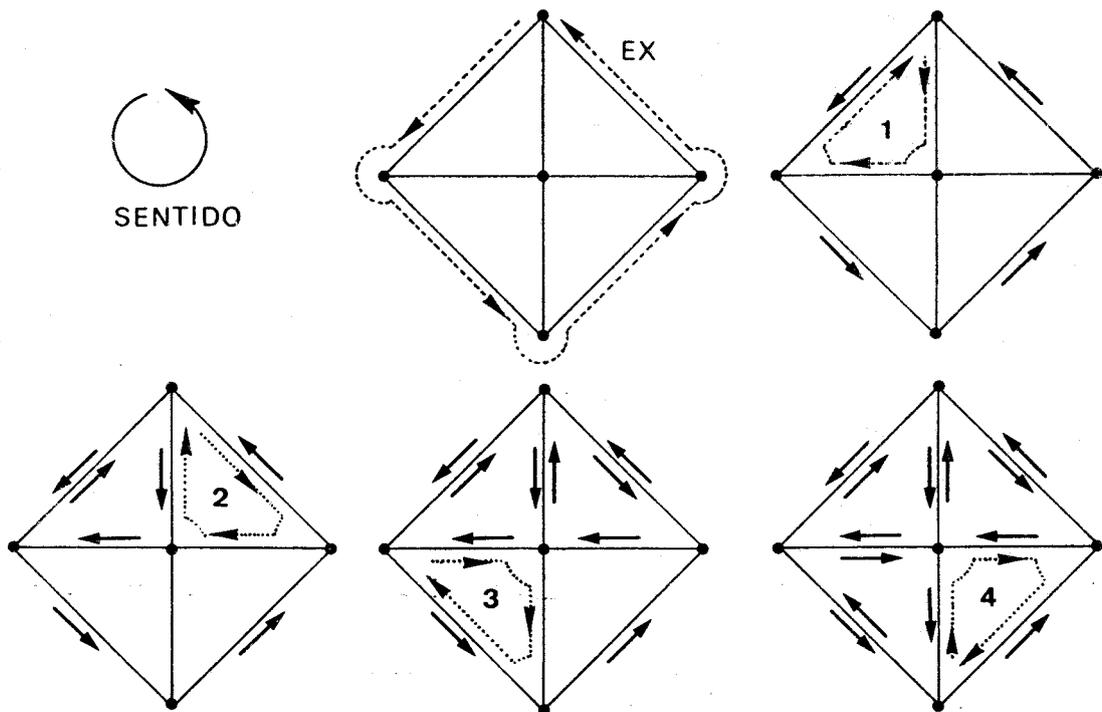


Figura 6

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 adyacencia para generar el conjunto de caras.

Describamos a continuación cada una de estas operaciones:

### III.1.5.1 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 al grafo  $G_3$ .

Si la lista  $ADY(v)$  contenía sólo aquellos vértices  $w$  adyacentes al  $v$  tales que  $v \rightarrow w$  o  $v \leftarrow w$ , en  $AYD-ORD(v)$  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*  $v \rightarrow w$ : 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*  $v \rightarrow w$ : 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.

### III.1.5.2. 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 es el vértice que sigue al u por la

derecha en la lista ordenada de adyacencia del  $v$ :

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

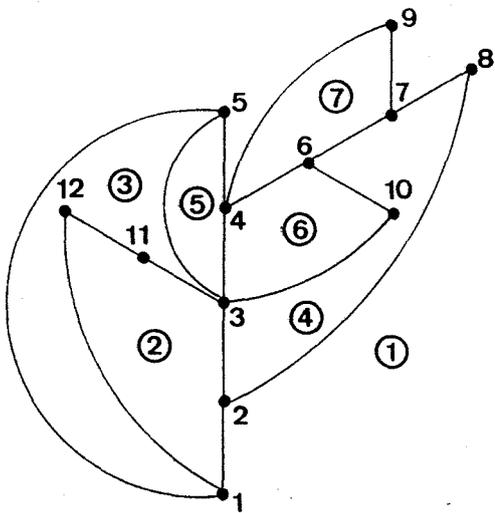
- c) Dado que cada arista pertenece a dos caras, será recorrida sólo dos veces y en sentidos opuestos. Para evitar que una arista se recorra más de dos veces, o en un sentido no apropiado, se irán marcando sucesivamente las aristas exploradas y el sentido de recorrido.
- d) Repetimos los pasos b y c 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 en 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 adyacencia ordenadas, no ha sufrido alteración alguna en este segundo paso. Así pues, esa será su estructura al finalizar esta etapa.



ADY(1)=(2)  
 ADY(2)=(3)  
 ADY(3)=(4,11)  
 ADY(4)=(5,6)  
 ADY(5)=(1,3)  
 ADY(6)=(7,10)  
 ADY(7)=(8,9)  
 ADY(8)=(2)  
 ADY(9)=(4)  
 ADY(10)=(3)  
 ADY(11)=(12)  
 ADY(12)=(1)

ADY-ORD(1)=(2,12,5)  
 ADY-ORD(2)=(8,3,1)  
 ADY-ORD(3)=(10,4,5,11,2)  
 ADY-ORD(4)=(6,9,5,3)  
 ADY-ORD(5)=(1,3,4)  
 ADY-ORD(6)=(10,7,4)  
 ADY-ORD(7)=(8,9,6)  
 ADY-ORD(8)=(2,7)  
 ADY-ORD(9)=(4,7)  
 ADY-ORD(10)=(3,6)  
 ADY-ORD(11)=(12,3)  
 ADY-ORD(12)=(1,11)

CARA(1)=(1,2,8,7,9,4,5)  
 CARA(2)=(1,12,11,3,2)  
 CARA(3)=(1,5,3,11,12)  
 CARA(4)=(2,3,10,6,7,8)  
 CARA(5)=(3,5,4)  
 CARA(6)=(3,4,6,10)  
 CARA(7)=(4,9,7,6)

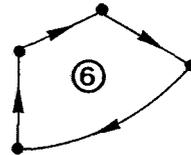
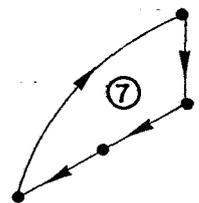
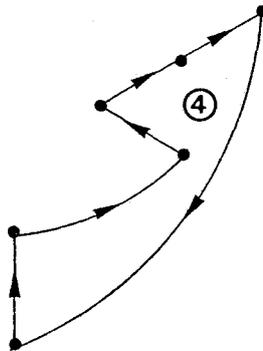
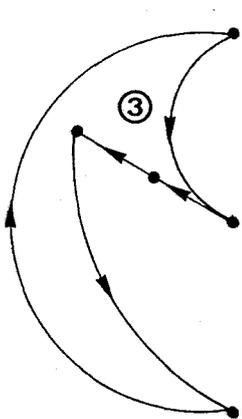
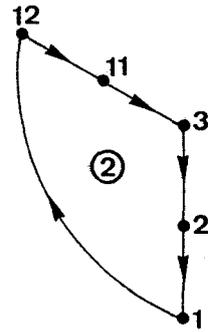
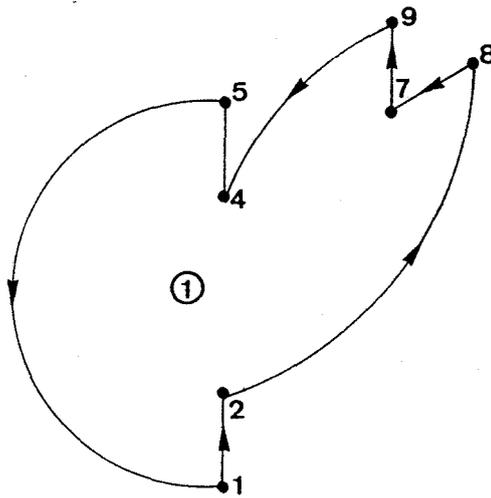


Figura 7

### III.1.6. 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 1.5 para que el conjunto de caras fuese único.

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

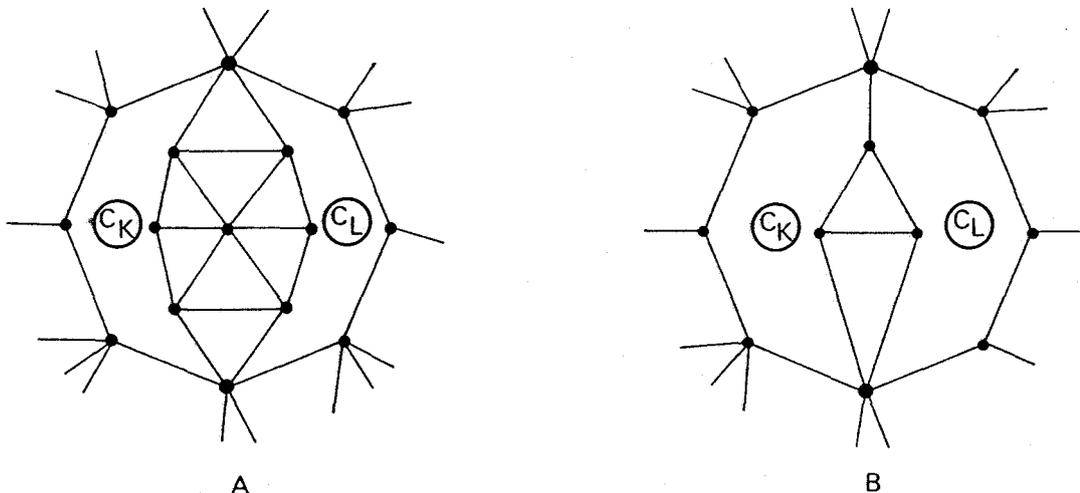
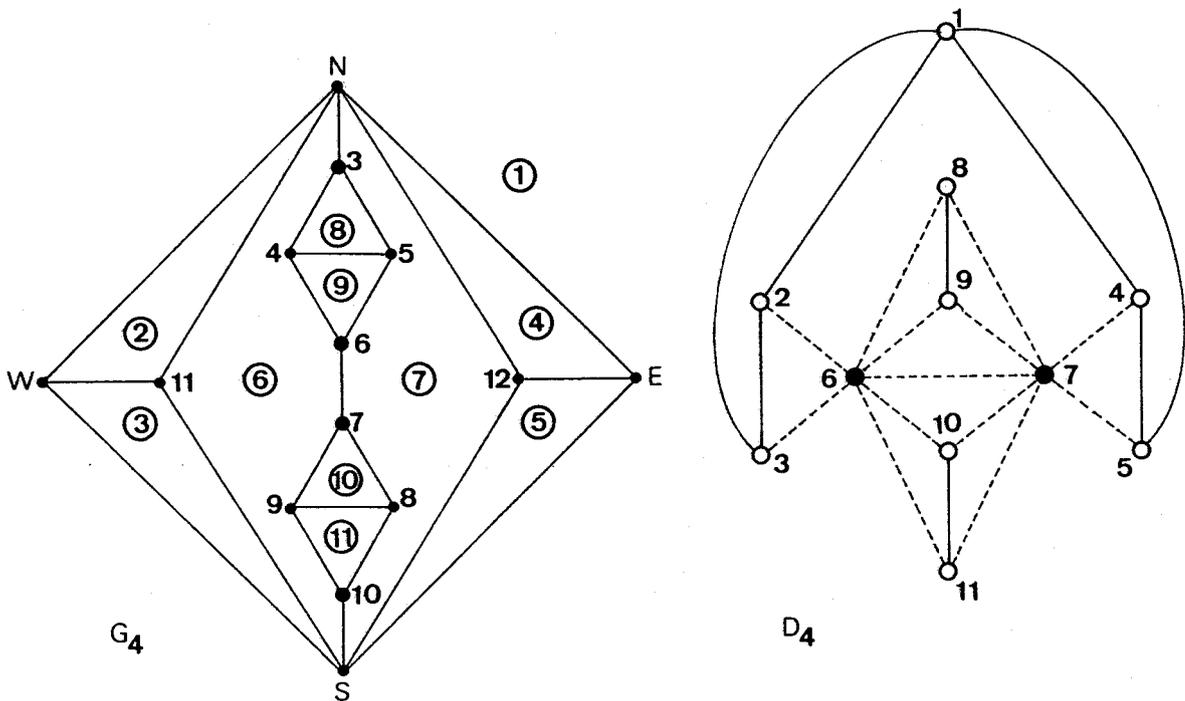


Figura 8

- Si dos caras adyacentes poseen en común vértices no consecutivos, de igual forma, 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 quedará escindido al menos en dos componentes conexas: una de ellas 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)



Las caras 6 y 7 forman el par de corte  $PC_1 = \{6, 7\}$ , el conjunto de componentes escindidas por este par es el  $CM_1 = \{CM_{11}, CM_{12}\}$ , siendo  $CM_{11} = \{8, 9\}$ ,  $CM_{12} = \{10, 11\}$ , pues, la componente conexa que contiene a la cara exterior es  $\{1, 2, 3, 4, 5\}$ .

Figura 9

Una vez expresados estos conceptos, podemos pasar a describir el algoritmo 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 directamente 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:

$CM_i = \{CM_{i1}, CM_{i2}, \dots\}$  conjunto de componentes escindidas.

$CM_{ij} = \{C_p, C_{p+1}, \dots\}$  componente escindida  $i, j$ .

Si designamos por  $\{V_i\}$  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:

$$\{v_1\} = \{N, S, 3, 6, 7, 10\}, \quad \{w_{11}\} = \{3, 4, 5, 6\}$$

$$\{v_1\} \cap \{w_{11}\} = \{3, 6\} \text{ luego:}$$

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

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

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

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.

### III.1.7. OBTENCION 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 Eastman (1984)- 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).

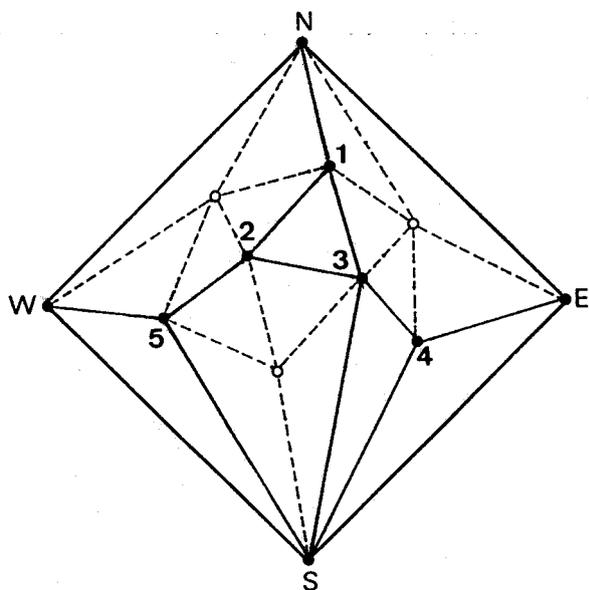


Figura 10

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.

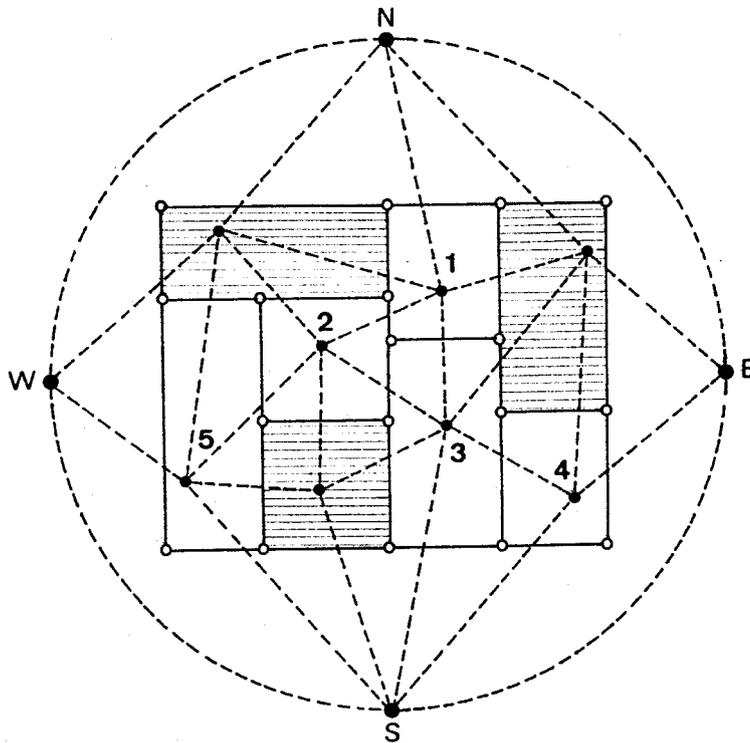


Figura 11

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.

El procedimiento del trazado gráfico del pseudo-dual, se comentará más adelante cuando se describa el trazado del dual.

### III.1.8. INTRODUCCION 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 presentar 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  $C_i$  del grafo  $G_4$  y es único el conjunto de caras de dicho grafo -por cumplir las condiciones señaladas en 1.5-, podemos afirmar que la introducción de dicha arista  $(u,v)$  en el grafo de adyacencia hará que éste sea no planar.
- 3) El caso más difícil de analizar, 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.

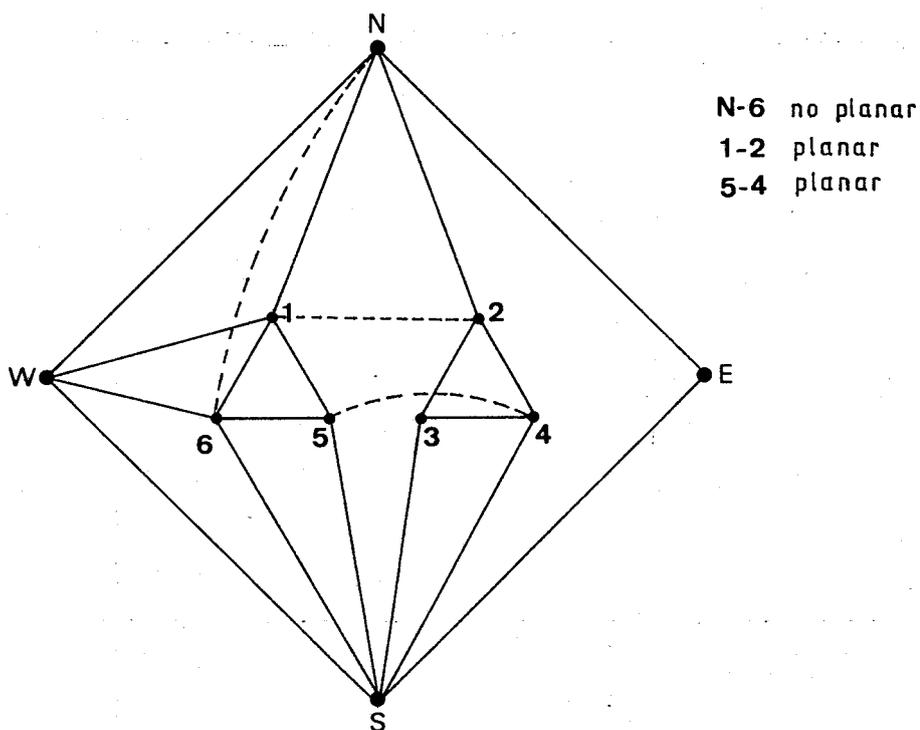


Figura 12

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:

### III.1.8.1. Formación de listas de inclusión de componentes.

Consideremos las componentes escindidas:

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

$$CM_j = \{C_p, C_{p+1}, \dots\}$$

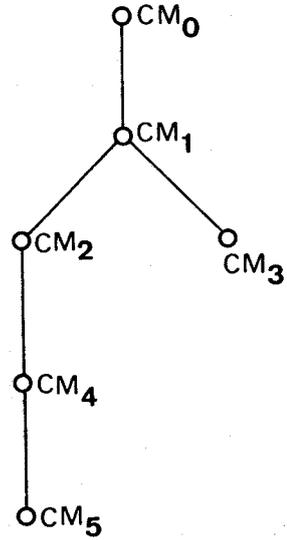
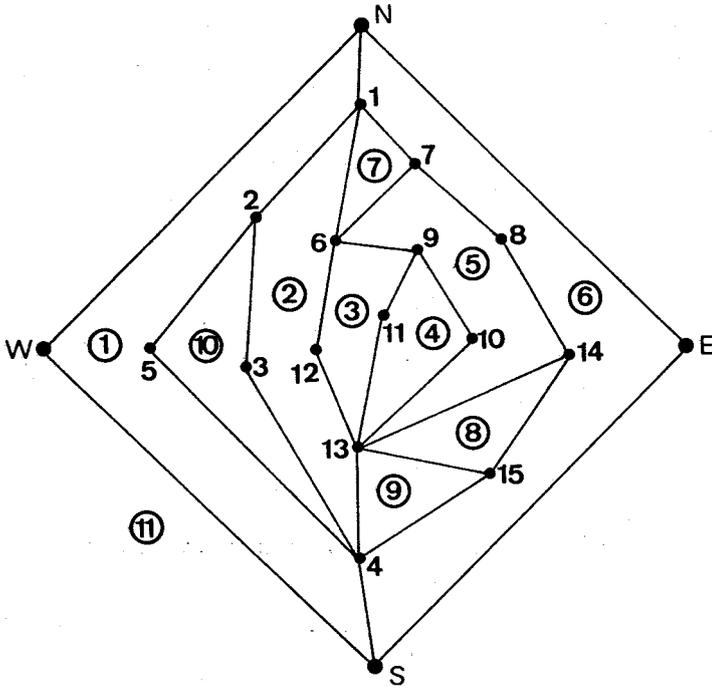
Diremos que  $CM_i$  incluye a  $CM_j$  si cualquier cara de  $CM_j$  pertenece a  $CM_i$ . Esta relación de inclusión define sobre el conjunto de componentes escindidas un *árbol dirigido*, cuyo origen será un vértice que representa al grafo  $G_4$ , puesto que dicho grafo incluye a todas las componentes 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 cual sería la estructura de los conjuntos que hemos definido hasta el momento.

### III.1.8.2. 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.



$CM_0 = \{CM_{01}\}$   
 $CM_{01} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

El grafo completo menos la cara exterior.

$PC_1 = \{1, 6\}$

$CM_1 = \{CM_{11}\}$

$CM_{11} = \{2, 3, 4, 5, 7, 8, 9, 10\}$

$PC_2 = \{2, 6\}$

$CM_2 = \{CM_{21}\}$

$CM_{21} = \{3, 4, 5, 7, 8, 9\}$

$PC_3 = \{1, 2\}$

$CM_3 = \{CM_{31}\}$

$CM_{31} = \{10\}$

$PC_4 = \{2, 5\}$

$CM_4 = \{CM_{41}\}$

$CM_{41} = \{3, 4\}$

$PC_5 = \{3, 5\}$

$CM_5 = \{CM_{51}\}$

$CM_{51} = \{4\}$

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:

$C(1) = \{1, 2, 5, 4, S, W, N\}$

$C(2) = \{6, 11, 13, 4, 3, 2, 1\}$

$C(3) = \{6, 9, 12, 13, 11\}$

$C(4) = \{9, 10, 13, 12\}$

$C(5) = \{7, 8, 14, 13, 10, 9, 6\}$

$C(6) = \{4, 15, 14, 8, 7, 1, N, E, S\}$

$C(7) = \{1, 7, 6\}$

$C(8) = \{13, 14, 15\}$

$C(9) = \{4, 13, 15\}$

$C(10) = \{2, 3, 4, 5\}$

$C(11) = \{N, W, S, E\}$

$VE(1) = \{7, 8, 14, 15, 6, 11, 13, 9, 10, 12, 3\}$

$VE(2) = \{7, 8, 14, 15, 9, 10, 12, 5\}$

$VE(3) = \{10\}$

$VE(4) = \{\}$

$VE(5) = \{1, 12\}$

$VE(6) = \{2, 5, 6, 11, 13, 9, 10, 12, 3\}$

$VE(7) = \{\}$

$VE(8) = \{\}$

$VE(9) = \{\}$

$VE(10) = \{\}$

$VE(11) = \{\}$

$GV(1, 7) = \{1\}$

$GV(1, 8) = \{1\}$

$GV(1, 14) = \{1\}$

$GV(1, 15) = \{1\}$

$GV(1, 6) = \{2, 1\}$

$GV(1, 11) = \{2, 1\}$

$GV(1, 13) = \{2, 1\}$

$GV(1, 9) = \{4, 2, 1\}$

$GV(1, 10) = \{4, 2, 1\}$

$GV(1, 12) = \{5, 4, 2, 1\}$

$GV(1, 3) = \{3\}$

$GV(2, 7) = \{2\}$

$GV(2, 8) = \{2\}$

$GV(2, 14) = \{2\}$

$GV(2, 15) = \{2\}$

$GV(2, 9) = \{4\}$

$GV(2, 10) = \{4\}$

$GV(2, 12) = \{5, 4\}$

$GV(2, 5) = \{3\}$

$GV(3, 10) = \{5\}$

$GV(5, 11) = \{4\}$

$GV(5, 12) = \{5\}$

$GV(6, 2) = \{1\}$

$GV(6, 5) = \{1\}$

$GV(6, 6) = \{2\}$

$GV(6, 11) = \{2\}$

$GV(6, 13) = \{2\}$

$GV(6, 9) = \{4, 2\}$

$GV(6, 10) = \{4, 2\}$

$GV(6, 12) = \{5, 4, 2\}$

$GV(6, 3) = \{3, 1\}$

Figura 13

Sean:

$PC_i = [C_k, C_l]$  un par de caras de separación.

$CM_i = \{CM_{i1}, CM_{i2}, \dots\}$  el conjunto de componentes escindidas por el par  $PC_i$ .

$CM_{ij} = \{C_o, C_{o+1}, \dots\}$  componente escindida  $j$  perteneciente al conjunto  $CM_i$ .

$PV_{ij} = [a, b]$  el par de vértices de corte de la componente  $CM_{ij}$ .

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

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

Así pues, las caras:

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

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

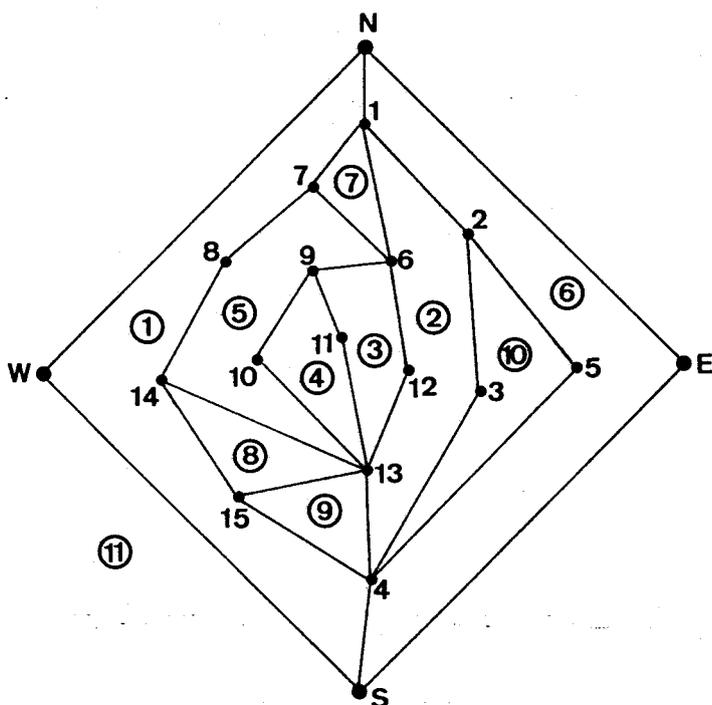
después del giro pasarían a ser:

$$C_k = \{\dots, a, v_m, \dots, v_{j+1}, v_j, b, \dots\}$$

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

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  $CM_7$  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.

$C(2) = \{2, 1, 6, 11, 13, 4, 3\}$	después del giro se convierte en ->	$C(2) = \{2, 1, 7, 8, 14, 15, 4, 3\}$
$C(6) = \{5, 4, 15, 14, 8, 7, 1, N\}$	·	$C(6) = \{5, 4, 13, 11, 6, 1, N\}$
$C(3) = \{11, 6, 9, 12, 13\}$	·	$C(3) = \{13, 12, 9, 6, 11\}$
$C(4) = \{9, 10, 13, 12\}$	·	$C(4) = \{12, 13, 10, 9\}$
$C(5) = \{7, 8, 14, 13, 10, 9, 6\}$	·	$C(5) = \{6, 9, 10, 13, 14, 8, 7\}$
$C(7) = \{1, 7, 6\}$	·	$C(7) = \{6, 7, 1\}$
$C(8) = \{13, 14, 15\}$	·	$C(8) = \{15, 14, 13\}$
$C(9) = \{4, 13, 15\}$	·	$C(9) = \{15, 13, 4\}$

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  $C_1$ , 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  $C_1$ , en cuyo caso, la cara  $C_1$  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  $C_p$ .

### III.1.8.3. 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.

### III.1.8.4. 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 solo 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.

### III.1.9. GENERACION DE ESQUEMAS ADIMENSIONALES

Una vez ampliado el grafo de adyacencia inicial con todos los requisitos introducidos a lo largo del proceso, y habiendo garantizado la planaridad del mismo, podemos generar ya aquellos esquemas de distribución de los espacios que cumplan todas las restricciones impuestas.

Para ello, hemos de obtener el dual del grafo de adyacencia, grafo que ha de estar triangulado para que la representación de su dual responda a un esquema de distribución en planta donde todos los locales sean rectangulares. En el caso de que el grafo no fuese triangulado, recurriríamos a la obtención de su pseudo-dual, tal y como se describe en 1.7.

No obstante, el dual de un grafo *triangulado* no posee una única representación en el plano que responda a una distribución de locales con forma rectangular, dado que pueden existir aristas del dual que admitan más de una orientación (horizontal o vertical), como puede apreciarse en el ejemplo de la figura 15.

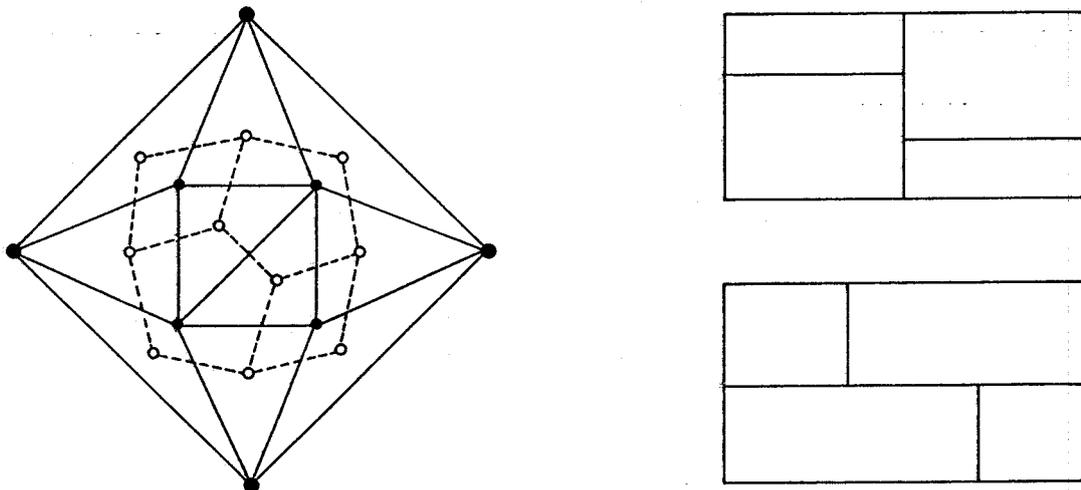


Figura 15

Así pues, vamos a describir a continuación un procedimiento original que genera todas las representaciones en el plano de un grafo dual dado. Dicho procedimiento incluye los siguientes pasos:

### III.1.9.1. Obtención de aristas del dual con orientación fija

Aquellas aristas de un grafo dual  $D$  que dada, su posición respecto de las demás, no admitan más que una orientación (horizontal o vertical) las llamaremos *aristas orientadas*. En caso contrario, las denotaremos como *no orientadas*.

Veamos a continuación en qué casos -como propone Guilleard (1978)- la orientación de una arista queda fijada:

- a) Son aristas orientadas las pertenecientes al perímetro exterior, es decir, todas aquellas que unan centros de caras del grafo  $G_4$  en las que aparezcan uno de los cuatro vértices N,S,E,W. (Figura 16)

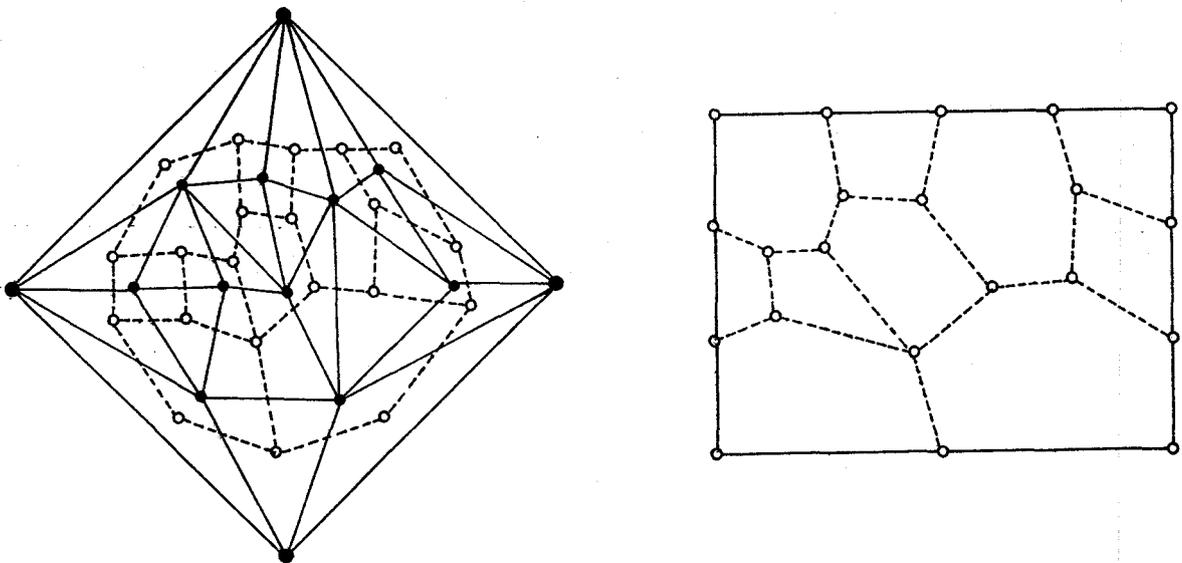


Figura 16

- b) Siempre que sobre un vértice del dual de valencia tres, incidan dos aristas con la misma orientación, la tercera arista queda con una orientación fija perpendicular a las anteriores. (Figura 17)

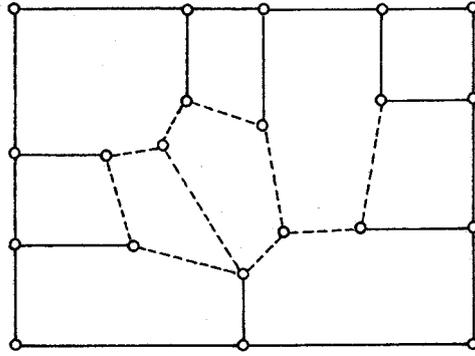


Figura 17

c) Si sobre un vértice del dual con valencia cuatro incide una arista orientada, la orientación de las tres restantes quedan fijadas. (Figura 18)

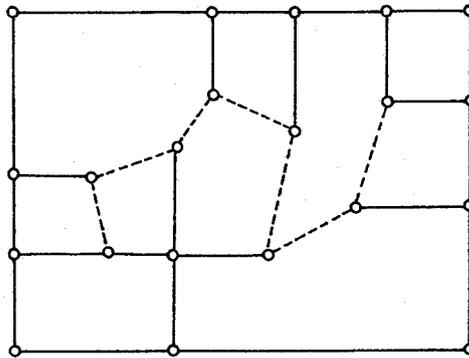


Figura 18

d) Si una cara del dual de valencia cuatro posee orientadas más de una de sus aristas, la orientación de las restantes queda fijada. (Figura 19)

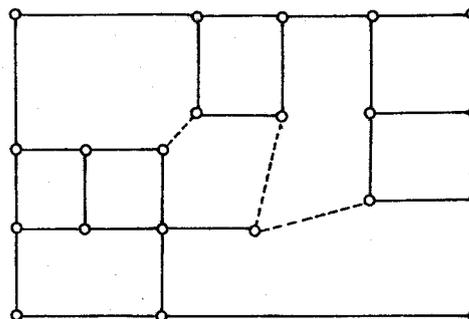


Figura 19

e) Si una cara del dual posee dos o más aristas orientadas es posible, en determinados casos, orientar las restantes. (Figura 20)

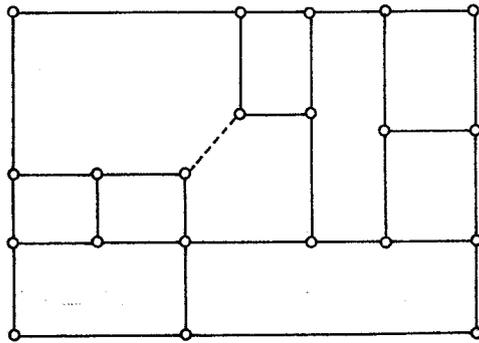


Figura 20

En la figura 21 proponemos un ejemplo de grafo cuyo dual posee una única representación en el plano, pues, mediante la aplicación de los casos anteriormente mencionados, sus aristas quedan todas con una orientación fija.

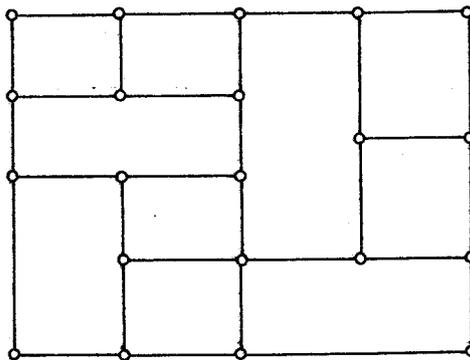


Figura 21

### III.1.9.2. Generación de soluciones admisibles.

La generación de todos los esquemas adimensionales, que cumplan con todos los requisitos impuestos, sigue un proceso

ramificado de exploración de todos aquellos vértices del dual sobre los que incidan aristas no orientadas.

Dado que el número de soluciones distintas aumenta, de forma considerable, con el número de aristas no orientadas, se ofrece la posibilidad de fijar las orientaciones de todas aquellas particiones (aristas) que se deseen.

El algoritmo podríamos describirlo del siguiente modo:

1) Por cada vértice del dual alcanzado en la búsqueda, se consideran todos los estados de orientaciones compatibles de sus aristas no orientadas. Cada estado de orientación del vértice constituye una bifurcación en el proceso ramificado de búsqueda.

Estos estados de compatibilidad son los siguientes:

a) Vértices de valencia tres: admiten sólo seis estados compatibles de orientación de sus aristas. Sea el vértice  $u$  y sus adyacentes  $v_1, v_2, v_3$ , es decir,  $ADY-C(u) = \{v_1, v_2, v_3\}$ ; los estados de compatibilidad posibles del vértice  $u$  serían:

Estados	1	2	3	4	5	6
$(u, v_1)$	V	V	H	H	H	V
$(u, v_2)$	V	H	V	H	V	H
$(u, v_3)$	H	V	V	V	H	H

b) Vértices de valencia cuatro: admiten únicamente dos estados. Si  $ADY-C(u) = \{v_1, v_2, v_3, v_4\}$ , entonces:

Estados	1	2
$(u, v_1)$	V	H
$(u, v_2)$	H	V
$(u, v_3)$	V	H
$(u, v_4)$	H	V

- c) Caras: si recorremos ordenadamente las aristas de las caras del dual  $D_4$ , el número de cambios de orientación a lo largo del recorrido ha de ser igual a cuatro para estancias rectangulares, tal y como se observa en la figura 22.

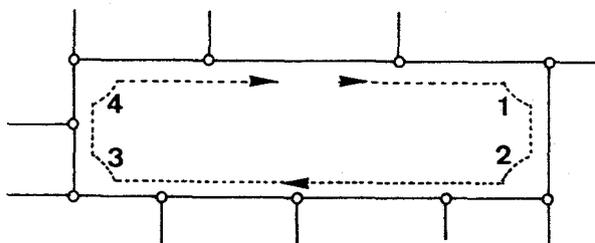


Figura 22

- 2) Por cada arista  $(u,v)$  que se oriente desde el vértice  $u$  se comprueba:
- a) Si su orientación implica la de otras aristas en el vértice  $v$ , y así sucesivamente.
  - b) Si estas orientaciones inducidas son compatibles con el estado de orientación de los vértices sobre los que incide y de las caras a las que pertenece. En caso de incompatibilidad, se abandonaría la exploración de esa rama para continuar por otra.
- 3) Cuando se logre alcanzar un estado del dual  $D_4$  -mediante la repetición de los pasos 1 y 2-, tal que todos sus vértices tengan sus aristas orientadas, pasaremos entonces a su representación gráfica.

Cada uno de estos estados daría origen a una solución posible y, por tanto, a un esquema de distribución distinto.

El dimensionamiento óptimo y automático de cada uno de los esquemas de distribución en planta puede realizarse, si se desea, conforme éstas son generadas. Por consiguiente, anteriormente se habrán definido las dimensiones (máxima o mínima) de todas las estancias, así como el criterio de optimización que se requiera para su dimensionamiento. Esta fase de optimización dimensional será descrita más adelante.

En la figura 24 se ilustra la generación de los posibles trazados planos del dual propuesto en la figura 23.

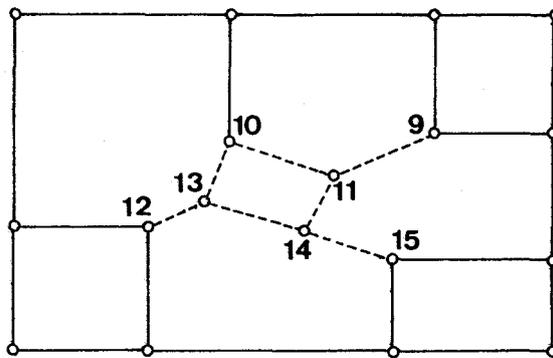
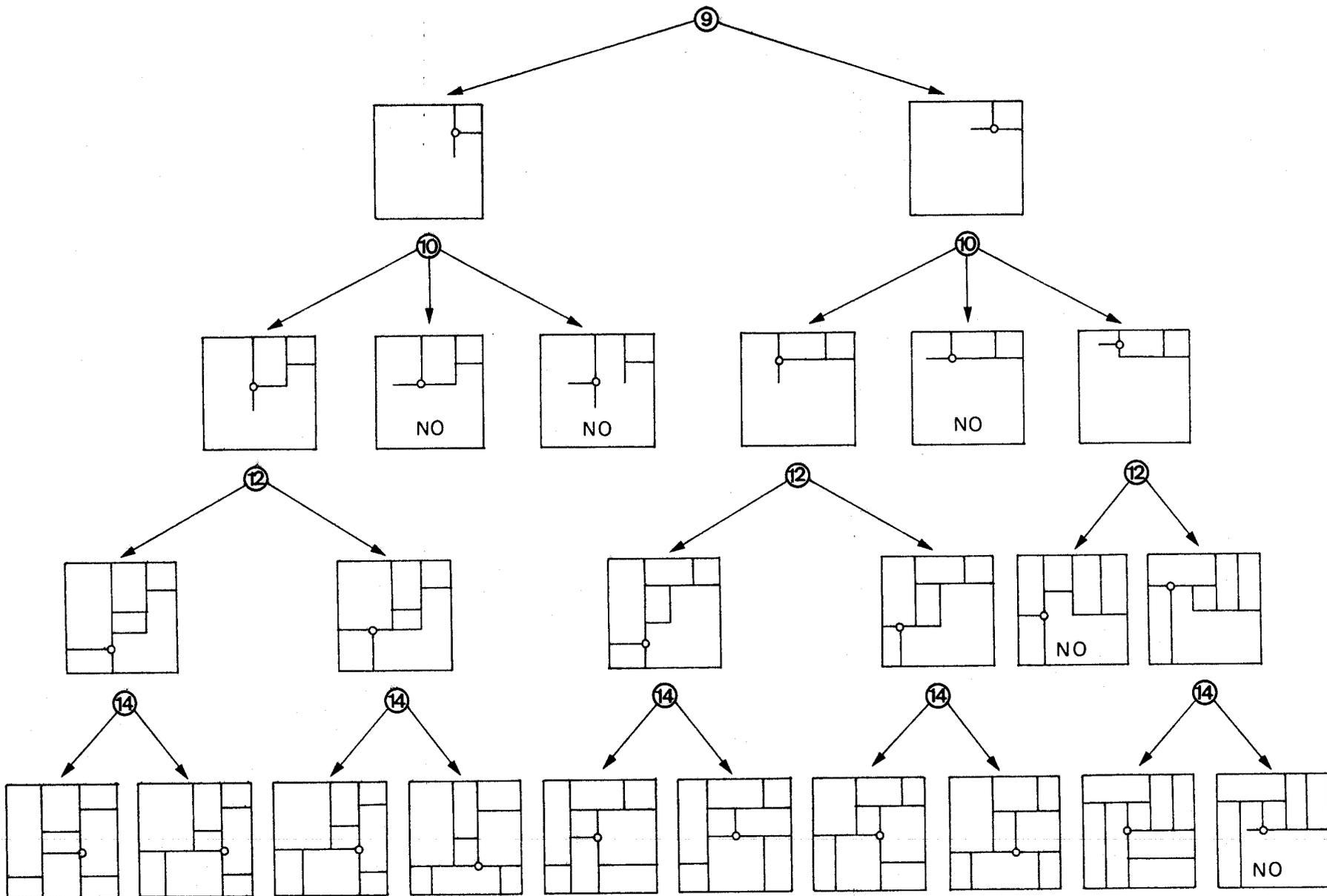


Figura 23

Figura 24



### III.1.10. TRAZADO DE LOS ESQUEMAS ADIMENSIONALES

La última etapa del procedimiento establecido para la generación de esquemas de distribución de edificios en planta consiste en el trazado de los mismos, es decir, en el trazado de todas las soluciones posibles obtenidas en la etapa anterior.

El procedimiento original propuesto para dicho trazado está basado en los estudios realizados por Tutte (1963) sobre el dibujo en dos dimensiones de un grafo planar.

En síntesis, el algoritmo propuesto por Tutte se desarrolla según los pasos siguientes:

- a) Generación de un circuito formado por la cara exterior.
- b) Asignación de coordenadas fijas a los vértices de dicho circuito, de modo que sus aristas configuren un polígono convexo en el plano.
- c) Obtención de las coordenadas de los restantes vértices mediante el planteamiento y resolución de los siguientes sistemas de ecuaciones, para un vértice  $v_i(x_i, y_i)$ :

$$\begin{aligned} b_i x_i &= \sum_{j=1, |V|} a_{ij} x_j \\ b_i y_i &= \sum_{j=1, |V|} a_{ij} y_j \end{aligned}$$

siendo:

$(a_{ij})$       matriz de adyacencia del grafo.  
 $b_i$           valencia del vértice  $v_i$   
 $|V|$           número de vértices.

Si este procedimiento lo aplicamos al grafo dual (o al pseudo dual), obtendremos -fijando las coordenadas de los vértices pertenecientes al perímetro exterior sobre un rectángulo- el trazado que se observa en la figura 25.

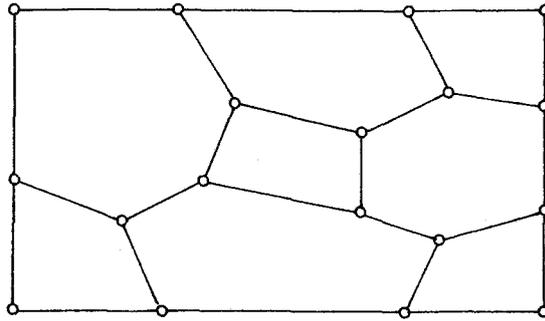


Figura 25

No obstante, aplicando el citado algoritmo obtenemos unas orientaciones arbitrarias para las aristas del dual y, por tanto, unas direcciones arbitrarias para los muros, particiones o tabiques. Es por esta razón por la que proponemos un procedimiento original que evita este extremo, y que permite conseguir el trazado de todas las representaciones planas posibles. Dicho procedimiento, aplicado a cada una de las soluciones generadas en la etapa anterior, se desarrolla en los tres pasos siguientes:

III.1.10.1. Obtención de las clases.

Particionamos el conjunto de las aristas del grafo dual en dos clases: las horizontales y las verticales. De forma que: una arista pertenece a la clase horizontal (ó vertical), si su orientación ha sido fijada como horizontal (ó vertical). (Figura 26)

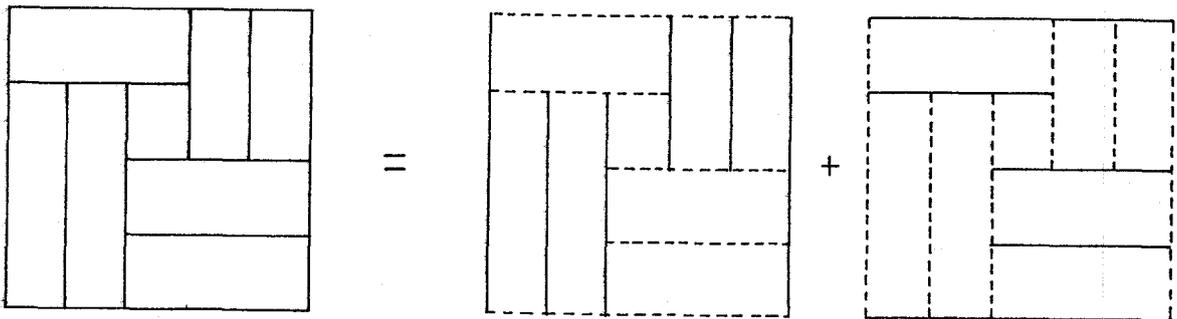


Figura 26

### III.1.10.2. Obtención de las subclases.

Sean  $\{H\}$ ,  $\{V\}$  las clases *horizontales* y *verticales* obtenidas anteriormente y establezcamos en cada una de ellas la siguiente relación: dos aristas  $ah_i, ah_j \in \{H\}$  -ó  $av_i, av_j \in \{V\}$ - son equivalentes, si existe una ruta en  $\{H\}$  -ó  $\{V\}$ - que las una. De esta forma,  $\{H\}$  y  $\{V\}$  quedan divididas en subclases de equivalencia  $\{HP\}_k$  y  $\{VP\}_k$ , correspondiendo cada una de ellas a las distintas particiones horizontales y verticales del esquema de distribución en planta a trazar. (Figura 27)

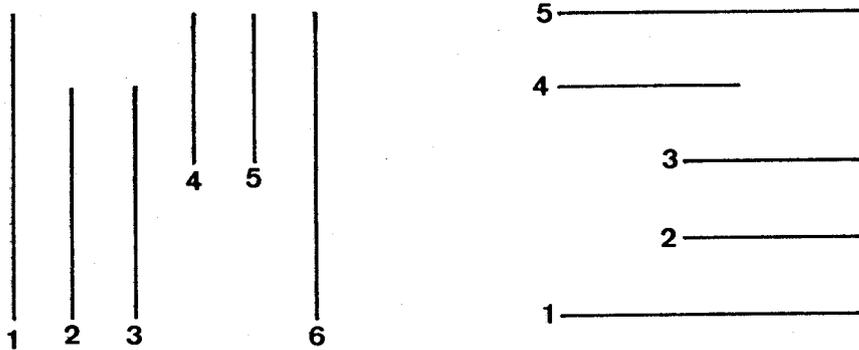


Figura 27

### III.1.10.3. Trazado de las subclases verticales y horizontales.

El trazado de cada uno de los esquemas de distribución se efectuará a partir del cálculo de las coordenadas de los vértices del dual que representan los puntos de encuentro de las particiones o tabiques, es decir, de los vértices pertenecientes a cada subclase vertical u horizontal.

Para el cálculo de las coordenadas de los vértices de cada subclase necesitamos definir las relaciones de adyacencia de cada una de ellas. Definamos para ello los siguientes conceptos:

a) Entenderemos por *extremo* de una subclase, el vértice de comienzo de la ruta que recorra todas sus aristas en un solo sentido. Este sentido se fija el mismo para todas las subclases. (Figura 28)

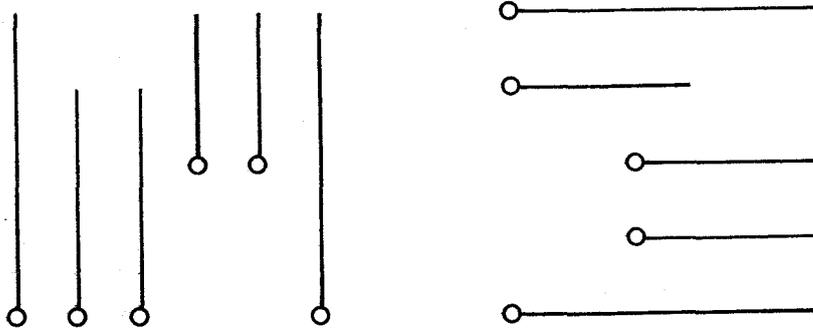


Figura 28

b) Una subclase horizontal (ó vertical) es adyacentes a otra si el extremo de la primera es adyacente a un vértice cualquiera de la segunda. (Figura 29)

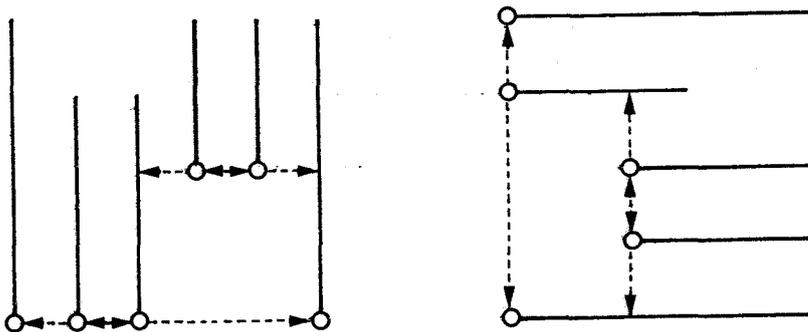


Figura 29

En ese caso, las coordenadas X de todos los vértices pertenecientes a la subclase vertical  $\{VP\}_i$  serán la solución del sistema de ecuaciones:

$$i=1,n : \quad b_i x_i = \sum_{j=1,n} v_{ij} x_j$$

y las coordenadas  $Y$  de los vértices pertenecientes a la subclase horizontal  $\{HP\}_i$ :

$$i=1,m : \quad b_i y_i = \sum_{j=1,m} h_{ij} y_j$$

siendo:

- $n$  número de clases verticales.
- $m$  número de clases horizontales.
- $b_i$  valencia de la subclase  $i$ .
- $v_{ij}$  matriz de adyacencia de las subclases verticales.
- $h_{ij}$  matriz de adyacencia de las subclases horizontales.

Con esta etapa se culmina el proceso de generación y trazado de esquemas adimensionales de todas organizaciones espaciales en planta que respondan a las restricciones de adyacencia y orientación iniciales y a las impuestas a lo largo del mismo.

Como se ha podido apreciar a lo largo de todo el proceso, se trata de un método de generación flexible e interactivo, capaz de generar automáticamente no sólo las soluciones de contorno rectangular, sino incluso aquellas de contorno irregular y espacios interiores vacíos (patios). Así pues, se ofrece al operador una mayor variedad en la tipología de las distribuciones espaciales en planta que cumplan con las condiciones impuestas.

Veamos a continuación el procedimiento que habría que seguir si se pretende obtener como solución final una planta optimizada dimensionalmente.

---

## III.2. DIMENSIONAMIENTO

---

La segunda parte del procedimiento que proponemos consiste en el dimensionamiento óptimo y automático de los esquemas de organizaciones espaciales en planta generados en 1.9. La resolución de este problema obliga, pues, a emplear métodos de optimización, es decir, métodos que permitan calcular los valores que han de tener las distintas variables que intervienen en el problema, por la naturaleza de la función objetivo y de las restricciones impuestas a la misma.

Como ya hemos comentado con anterioridad, utilizaremos únicamente restricciones y funciones objetivos lineales ó aproximadas linealmente; de esta forma emplearemos la potente técnica de la programación lineal en la resolución rápida y eficiente del problema de optimización.

Analicemos, por consiguiente, las restricciones y las funciones objetivo que van a ser consideradas en el problema de dimensionamiento, así como las expresiones matemáticas que lo representan.

### III.2.1. RESTRICCIONES METRICO-GEOMETRICAS

Estas restricciones son las que traducen las limitaciones métrico-geométricas impuestas por el operador a cada uno de los locales que componen las distribución espacial en planta, y pueden ser las siguientes:

a) Dimensiones máximas y mínimas de cada local i:

$$d_i \leq x_i \leq D_i$$

$$d_i \leq y_i \leq D_i$$

b) Area máxima y mínima de cada local i:

$$s_i \leq x_i y_i \leq S_i$$

Este último tipo de restricciones automáticamente se aproximan linealmente mediante la sustitución de cada rama hiperbólica -que representa cada limitación superficial- por cuatro rectas secantes, como se ilustra en la figura 30.

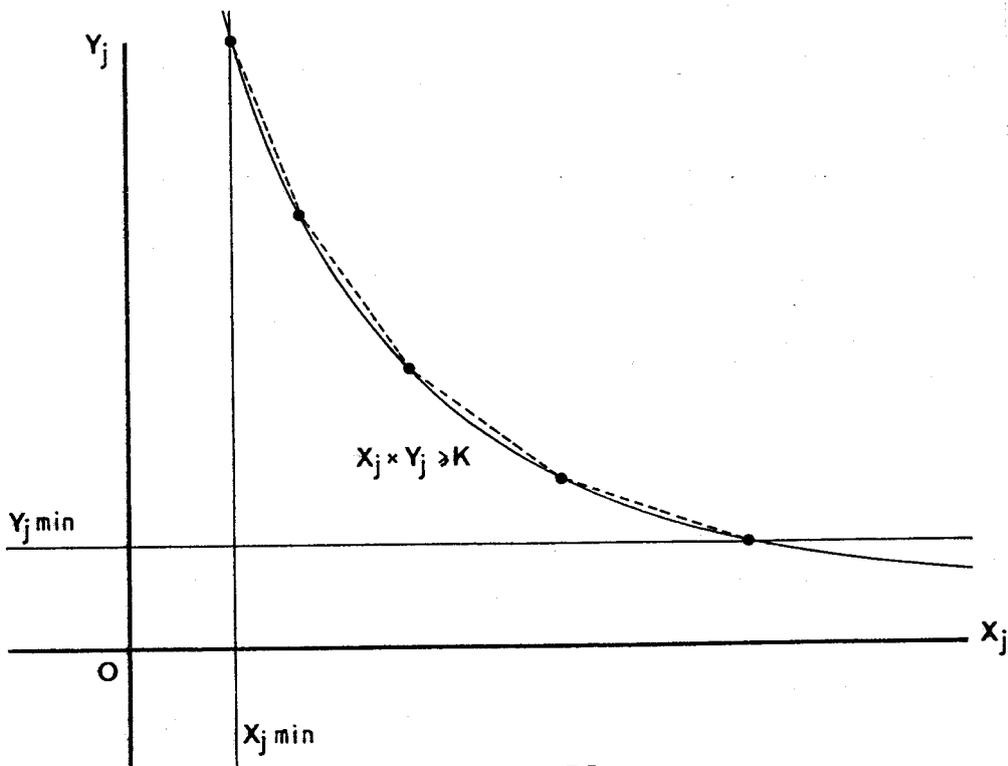


Figura 30

### III.2.2. RESTRICCIONES DE RECTANGULARIDAD DE PLANTA Y LOCALES

Como su propio nombre indica, con este tipo de restricción lo que pretendemos es que la organización espacial en planta final, ya dimensionada, cumpla que todos los espacios componentes así como el contorno tengan forma rectangular. En el caso descrito en la etapa adimensional de contorno de planta no rectangular, consideraremos que su contorno es un rectángulo circunscrito a dicha planta irregular (Figura 31).

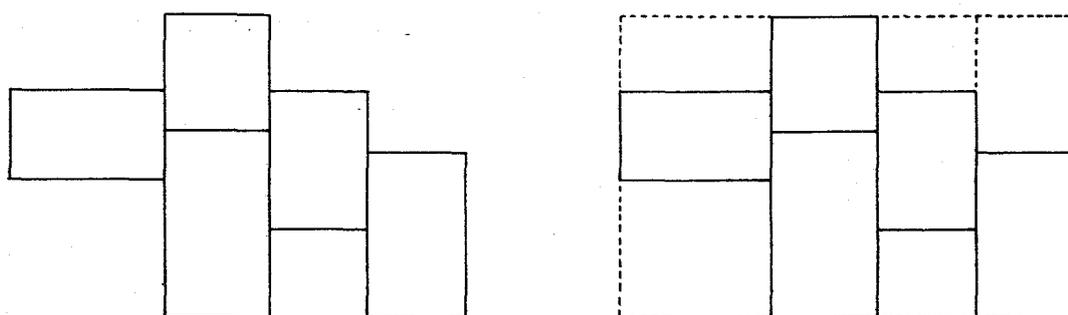


Figura 31

Para obtener las expresiones de las restricciones de rectangularidad hemos de descomponer, previamente, el grafo de adyacencia  $G_4$  en dos grafos dirigidos, uno vertical  $GV_4$  y otro horizontal  $GH_4$ . Esta división se realiza del modo siguiente:

Sea la arista  $(v_m, v_n) \in G_4$  y su correspondiente arista  $(C_j, C_l)$  en el dual  $D_4$ , entonces:

$$(v_m, v_n) \in GH_4 \text{ si } (C_j, C_l) \in \{H\}$$

ó

$$(v_m, v_n) \in GV_4 \text{ si } (C_j, C_l) \in \{V\}$$

En la figura 32 se ilustra cuáles serían los grafos dirigidos vertical y horizontal correspondientes a un determinado grafo de adyacencia.

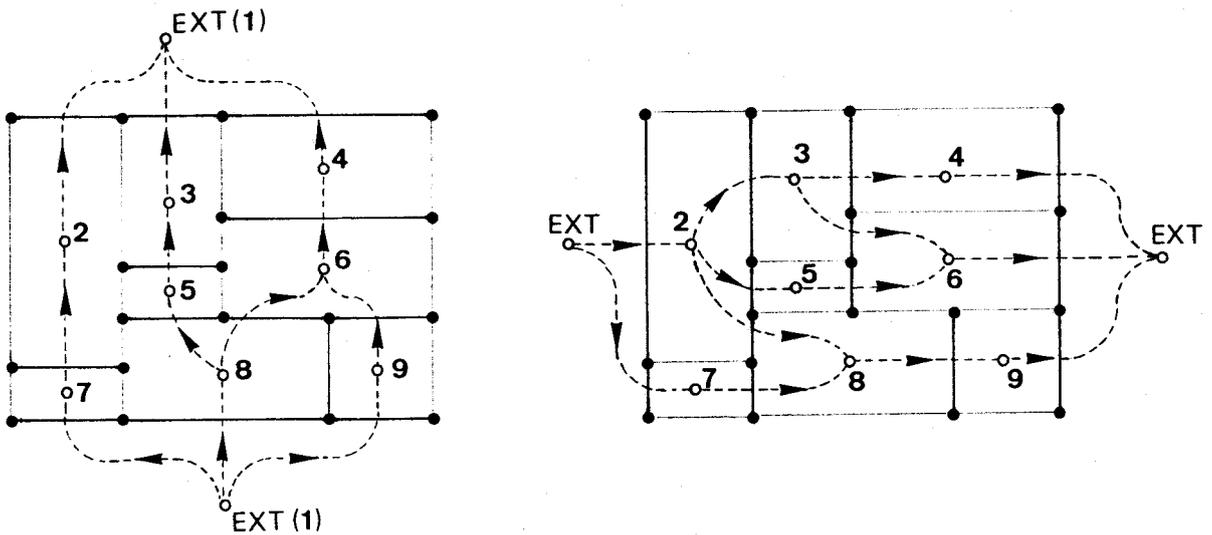


Figura 32

Hay que hacer notar que los sentidos de las aristas del grafo  $G_4$  y las de su dual  $D_4$  se encuentran ligados por la forma en que se obtuvo el dual, por lo tanto, si orientamos las aristas del dual en un sentido, quedan orientadas las de  $G_4$ , y viceversa. (Figura 33).

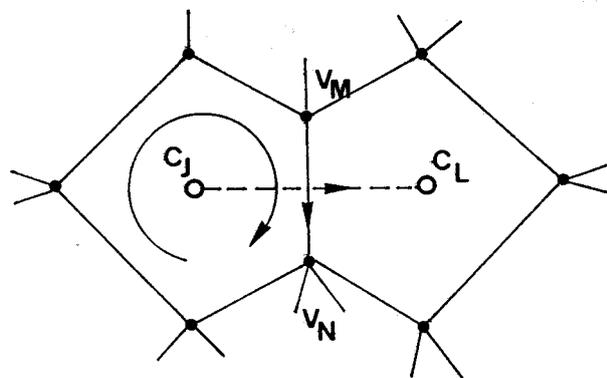


Figura 33

Así pues, si  $X(C_j), Y(C_j)$  son las coordenadas  $x$  e  $y$  obtenidas del vértice  $C_j$  de  $D_4$  en el trazado del dual, y orientamos las aristas del dual  $D_4$  de forma que:

$$\begin{aligned} ah_i = (C_j, C_1) &\rightarrow X(C_j) < X(C_1) \\ av_i = (C_j, C_1) &\rightarrow Y(C_j) < Y(C_1) \end{aligned}$$

y teniendo en cuenta que tanto la planta como todos los

locales han de ser rectangulares, se ha de cumplir entonces para cada subclase horizontal  $\{HP\}_k$ :

si  $\{HP\}_k = \{ah_1, ah_2, \dots, ah_p\}$  :

$$\frac{\sum_{m=1, p} x_m - \sum_{n=1, p} x_n = 0}{}$$

siendo  $ah_i = (C_j, C_1) \rightarrow (v_{im}, v_{in}) \in GH_4 \rightarrow x_m, x_n$

Del mismo modo, para cada subclase vertical  $\{VP\}_k$ :

si  $\{VP\}_k = \{av_1, av_2, \dots, av_q\}$  :

$$\frac{\sum_{m=1, q} y_m - \sum_{n=1, q} y_n = 0}{}$$

siendo  $av_i = (C_j, C_1) \rightarrow (v_{im}, v_{in}) \in GV_4 \rightarrow y_m, y_n$

Las restricciones de rectangularidad obtenidas a partir de los grafos dirigidos horizontal y vertical, respectivamente, del grafo de la figura 32 serían:

Para las clases horizontales:

$$\begin{aligned} x_1 - (x_2 + x_3 + x_4) &= 0 \\ x_4 - x_6 &= 0 \\ x_3 - x_5 &= 0 \\ x_5 + x_6 - (x_8 + x_9) &= 0 \\ x_2 - x_7 &= 0 \\ x_7 + x_8 + x_9 - x_1 &= 0 \text{ (sobra)} \end{aligned}$$

Para las clases verticales:

$$\begin{aligned} y_1 - (y_7 + y_2) &= 0 \\ y_7 + y_2 - (y_8 + y_5 + y_3) &= 0 \\ y_8 - y_9 &= 0 \\ y_5 + y_3 - (y_6 + y_4) &= 0 \\ y_9 + y_6 + y_4 - y_1 &= 0 \text{ (sobra)} \end{aligned}$$

De esta forma obtendremos las  $(h-1)+(v-1)$  ecuaciones que nos definen las restricciones debidas a la rectangularidad de la planta y locales, siendo  $h$  y  $v$  el número de subclases o particiones horizontales y verticales respectivamente.

### III.2.3. ACCESIBILIDAD ENTRE LOCALES

Con este tipo de restricciones se pretende establecer los requisitos de comunicación de los locales a través de huecos de paso con una anchura mayor que la que se entiende como mínima. La accesibilidad, pues, entre dos locales adyacentes depende de la dimensión de la partición o tabique común a ambos: si esta dimensión es mayor o igual que 1 metro, diremos que dichos locales son accesibles entre sí. (Figura 34).

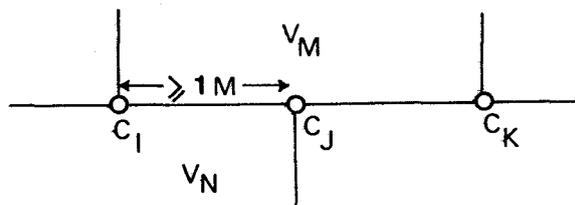
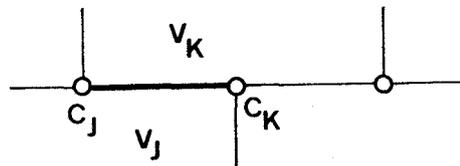


Figura 34

Por lo tanto, si los locales adyacentes han de ser además accesibles, se elaborarán unas inequaciones, a partir del análisis de las subclases horizontales y verticales, que nos definan esta condición. Para ello, procederemos del siguiente modo:

Si  $(v_j, v_k)$  es la arista de  $GH_4$  correspondiente a la arista del dual  $(C_j, C_k)$  -orientada y con el sentido impuesto anteriormente en 2.2-, designaremos al vértice  $v_j$  como local inferior de  $(C_j, C_k)$ , y al vértice  $v_k$  como local superior de  $(C_j, C_k)$ , y lo denotaremos:

$$\begin{aligned} \text{LOC}_{\text{inf}}(C_j, C_k) &= v_j \\ \text{LOC}_{\text{sup}}(C_j, C_k) &= v_k \end{aligned}$$



Sea  $\{HP\}_k$  una subclase horizontal cualquiera, cuyas aristas, ordenadas según la coordenada  $x$  en el plano de sus vértices extremos y leídas de izquierda a derecha, son:

$$\dots, (C_i, C_j), (C_j, C_k), (C_k, C_l), \dots$$

tal y como se ilustra en la figura 35, se cumple que:

- si  $\text{LOC}_{\text{inf}}(C_i, C_j) < \text{LOC}_{\text{inf}}(C_j, C_k) \rightarrow \Sigma X_{\text{SUP}k} - \Sigma X_{\text{INF}j} > 1$ , (A)
- si  $\text{LOC}_{\text{inf}}(C_j, C_k) < \text{LOC}_{\text{inf}}(C_k, C_l) \rightarrow \Sigma X_{\text{INF}k} - \Sigma X_{\text{SUP}j} > 1$ , (B)
- si  $\text{LOC}_{\text{sup}}(C_i, C_j) < \text{LOC}_{\text{sup}}(C_j, C_k) \rightarrow \Sigma X_{\text{INF}k} - \Sigma X_{\text{SUP}j} > 1$ , (C)
- si  $\text{LOC}_{\text{sup}}(C_j, C_k) < \text{LOC}_{\text{sup}}(C_k, C_l) \rightarrow \Sigma X_{\text{SUP}k} - \Sigma X_{\text{INF}j} > 1$ , (D)

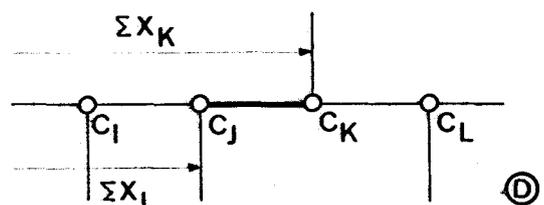
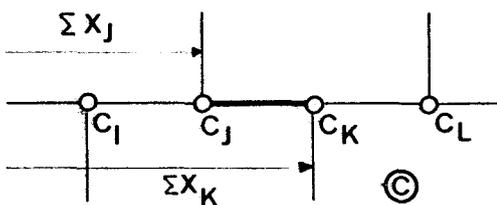
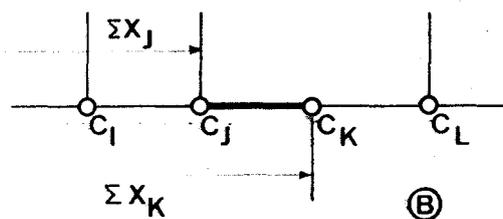
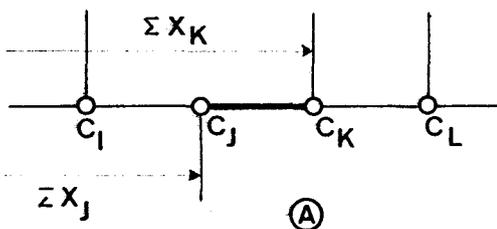


Figura 35

siendo:

$\Sigma X_{SUPk}$  la suma de las longitudes de todos los locales superiores que están situados a la izquierda de la partición vertical límite que contiene al vértice  $C_k$  del dual.

$\Sigma X_{INFj}$  la misma suma que en el caso anterior, pero referida a los locales inferiores.

El término independiente 1 nos indica la longitud mínima del hueco de paso entre dos locales.

En definitiva, para las subclases horizontales obtenemos unas inecuaciones del tipo:

$$\frac{+\Sigma X_{SUPk} - \Sigma X_{INFj} \geq 1}{}$$

que nos definen la longitud mínima de adyacencia entre dos locales que han de ser accesibles entre sí.

Siguiendo el mismo razonamiento con las subclases verticales, llegaríamos a las inecuaciones:

$$\frac{+\Sigma Y_{SUPk} - \Sigma Y_{INFj} \geq 1}{}$$

Para el ejemplo ilustrado en la figura 32, las restricciones de accesibilidad serían:

$$x_8 - x_5 \geq 1$$

$$y_8 - y_7 \geq 1$$

$$y_6 - y_5 \geq 1$$

### III.2.4. FUNCION OBJETIVO

La función objetivo considerada es una expresión lineal muy general que engloba todos los casos posibles que se plantean. Su expresión sería la siguiente:

$$\text{Min. } \sum c_i x_i + \sum d_i y_i$$

Variando convenientemente los coeficientes  $c_i$ ,  $d_i$  -costes ó pesos- podremos conseguir:

- a) Minimizar una de las dos dimensiones del contorno, haciendo igual a cero todos los pesos excepto el  $c_1$  ó el  $d_1$ :

$$\text{Min. } x_1 \quad \text{ó} \quad \text{Min. } y_1$$

- b) Minimizar el perímetro exterior. En este caso la función objetivo se convertiría en:

$$\text{Min. } c_1 x_1 + d_1 y_1$$

Si  $c_1 = d_1$  la forma del contorno tendería a ser de proporción cuadrada, en caso contrario se primaría una de las dos dimensiones.

- c) Minimizar cualquier dimensión o el perímetro de un local  $i$ , o de varios locales.

Como puede apreciarse, no se incluye ninguna expresión que contemple la minimización del área, pues como ya hemos indicado, pretendemos aprovechar la potencia y efectividad de la programación lineal para la resolución del problema de optimización. Consideramos que las funciones objetivos que se proponen producen unos resultados de esquemas de plantas ya dimensionados, con una aproximación suficiente para el fin que se pretende: generar todas las soluciones de esquemas de

distribución en planta que satisfagan unas condiciones de adyacencia y dimensionales previamente impuestas.

### III.2.5. RESOLUCION DEL PROBLEMA DE OPTIMIZACION

La estructura general del problema sería:

Min.  $C X$

s.a.  $A X = 0$  (Rectangularidad de planta y locales)  
 $B X \geq 1$  (Accesibilidad)  
 $X \geq D$  (Métrico-geométricas)  
 $X \leq E$

Por la naturaleza de las restricciones y de la función objetivo planteadas, el problema de optimización se resuelve mediante el clásico método de programación lineal del "Simplex".

---

#### **IV. CONCLUSIONES**

---

Este trabajo pretende continuar la línea de investigación iniciada en la E.T.S. de Arquitectura de Sevilla por J.J. Sendra sobre el diseño arquitectónico asistido por ordenador, y más concretamente sobre la generación y optimización de esquemas de distribución en planta. Para lo cual, y tras el análisis de las propuestas que diversos autores han elaborado sobre el tema, se ha llegado al desarrollo de un método de generación de distribuciones en planta del que destacamos principalmente los siguientes aspectos:

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) Dicho método puede, además, generar y trazar todas las

soluciones posibles de las distribuciones en planta que cumplen con las restricciones introducidas por el operador, sin más limitación que el tiempo requerido para obtener y representar todas y cada una de las mismas.

- 3) 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.

En cuanto a nuevos algoritmos aportados originalmente por este trabajo podemos destacar los siguientes:

- 1) Comprobación de la planaridad y obtención del grafo dual.

Mediante el algoritmo descrito para la generación de las caras de un grafo, se facilita, en etapas posteriores, la obtención del grafo dual y la comprobación de la planaridad del grafo de adyacencia ante la introducción de nuevos requisitos.

- 2) Generación de esquemas adimensionales.

Se presenta un método que genera todas las posibles representaciones planares del grafo dual mediante la consideración de los estados compatibles de orientación de sus aristas.

- 3) Trazado del grafo dual.

Se desarrolla un método, a partir del propuesto por Tutte para el trazado en el plano de un grafo planar, que obtiene la representación gráfica del esquema adimensional de distribución en planta generado.

- 4) Paso automático de la etapa adimensional a la dimensional.

Dicha transición se efectúa mediante un algoritmo que permite la obtención automática de las restricciones debidas tanto a la rectangularidad de los locales como a las de accesibilidad entre los mismos, a partir de un esquema adimensional de distribución en planta generado.

En el dimensionamiento de los esquemas distributivos, se ha considerado la programación lineal como el método idóneo por su potencia en la resolución de problemas de optimización, por consiguiente, tanto la función objetivo como las restricciones que se consideran son lineales o son linealizadas automáticamente.

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.

---

**ANEXO I. PROGRAMA DE ORDENADOR**

---



```

1036 REPEAT entrada
1037   INPUT "Locales adyacentes:"!a;
1038   IF a>0
1039     INPUT "-";b
1040     ady%(a,0)=ady%(a,0)+1:ady%(a,ady%(a,0))=b
1041     ady%(b,0)=ady%(b,0)+1:ady%(b,ady%(b,0))=a
1042     numero_de_aristas=numero_de_aristas+1
1043   ELSE
1044     SELECT ON a
1045       =-1
1046         DIM numero%(v),romenu%(v),bajo1%(v),bajo2%(v)
1047         n=0:arista$=""
1048         conexo=0:biconexo=1:plano=1
1049 :
1050         REMark Se llama al procedimiento que comprueba
           la Conectividad y Biconectividad.
1051 :
1052         DFS 1,0
1053 :
1054         REMark Si no es conexo, imprime las componentes
1055 :
1056         IF n<v
1057           temp2=0
1058           PRINT "NO CONEXO"
1059           REPEAT conex
1060             temp1=0
1061             FOR i=1 TO v
1062               IF numero%(i)>temp2
1063                 PRINT !i!
1064               ELSE
1065                 IF NOT temp1 AND numero%(i)=0:temp1=i
1066               END IF
1067             END FOR i
1068             temp2=n:PRINT
1069             IF n>=v:EXIT conex
1070             DFS temp1,0
1071           END REPEAT conex

```

```

1072         NEXT entrada
1073     ELSE
1074         conexo=1
1075     END IF
1076     IF NOT biconexo:NEXT entrada
1077 :
1078     REMark Ordenar las adyacencias segun la funcion
        FI, como preparacion para PLANAR.

1079 :
1080     ORD_FI
1081 :
1082     REMark Comprobar la planaridad.
1083 :
1084     PLANAR
1085     IF NOT plano:NEXT entrada
1086 :
1087     REMark Se ordenan las adyacencias para la ob-
        tencion de las caras.

1088 :
1089     ORDENAR
1090     CARAS
1091     DIM ady%(1), adya%(1), pila%(1), prox%(1), f%(1),
        ruta%(1), bloque%(1), sup%(1), inf%(1)
1092     num_par_corte=1: posible_par_corte%=0
1093     salir%=1
1094     REPEAT nueva
1095         IF num_par_corte OR posible_par_corte%
            :TRICONEXO
1096         IF num_par_corte
1097             COMPONENTES_PAR_CARAS
1098             GIROS
1099         ELSE
1100             DIM par_cara%(1), vert_comune$(1), numero$(1)
                , comp_par_cara$(1)
1101             DIM par_vert$(1), inclu_comp$(1),
                extra_vert$(1), guarda_giro$(1),
                indice%(1)

```



```

1133 n=n+1:numero%(v)=n:romenu%(n)=v
1134 bajo1%(v)=numero%(v):bajo2%(v)=numero%(v)
1135 FOR i=1 TO ady%(v,0)
1136   w=ady%(v,i)
1137   IF numero%(w)=0
1138     arista$=arista$&CHR$(0)&CHR$(v)&CHR$(w)
1139     DFS (w),(v)
1140     IF bajo1%(w)<bajo1%(v)
1141       bajo2%(v)=MIN(bajo1%(v),bajo2%(w))
1142       bajo1%(v)=bajo1%(w)
1143     ELSE
1144       IF bajo1%(w)=bajo1%(v)
1145         bajo2%(v)=MIN(bajo2%(v),bajo2%(w))
1146       ELSE
1147         bajo2%(v)=MIN(bajo2%(v),bajo1%(w))
1148       END IF
1149     END IF
1150   ELSE
1151     IF numero%(w)<numero%(v) AND w<>u
1152       arista$=arista$&CHR$(0)&CHR$(v)&CHR$(w)
1153       IF numero%(w)<bajo1%(v)
1154         bajo2%(v)=bajo1%(v)
1155         bajo1%(v)=numero%(w)
1156       ELSE
1157         IF numero%(w)>bajo1%(v)
1158           bajo2%(v)=MIN(bajo2%(v),numero%(w))
1159         END IF
1160       END IF
1161     END IF
1162   END IF
1163 END FOR i
1164 :
1165 REMark Se comprueba la biconectividad
1166 :
1167 IF bajo1%(v)>=numero%(u) AND u<>0 AND v<>2
1168   PRINT"Vertice de corte en"!u,v
1169   FOR j=numero%(v) TO n

```







```

1275 :
1276 DEFine PROCEDURE PATHFINDER (v)
1277 LOCAL i,w,lazo
1278 FOR i=1 TO adya%(v,0)
1279     w=adya%(v,i)
1280     IF v<w
1281         IF s=0:s=v:p=p+1
1282         ruta%(w)=p
1283         PATHFINDER (w):IF NOT plano:RETURN
1284         IF j<>0:IF pila%(bloque%(j-1),1)>=v
                    :bloque%(j-1)=0
1285         IF j<>0:IF pila%(bloque%(j),1)>=v:bloque%(j)=0
1286         REPEAT lazo
1287             IF j=0:EXIT lazo
1288             IF bloque%(j-1)<>0 OR bloque%(j)<>0:EXIT lazo
1289             j=j-2
1290         END REPEAT lazo
1291         IF j<>0:IF pila%(bloque%(j-1),1)>=v
                    :bloque%(j-1)=0
1292         IF j<>0:IF pila%(bloque%(j),1)>=v:bloque%(j)=0
1293         REPEAT lazo
1294             IF prox%(0)=0 OR pila%(prox%(0),1)<v:EXIT lazo
1295             IF bloque%(j)<>0
1296             EMPOTRAR (pila%(prox%(0),0)),
                    (pila%(bloque%(j),0)),1
1297         ELSE
1298             IF pila%(prox%(prox%(0)+1),1)<>0
1299             EMPOTRAR (pila%(prox%(0),0)),
                    (pila%(prox%(prox%(0)+1),0)),1
1300         ELSE
1301             IF bloque%(j-1)<>0
                    :EMPOTRAR (pila%(prox%(0),0)),
                    (pila%(bloque%(j-1),0)),-1
                    :ELSE
                    IF f%(pila%(prox%(0),0),0)=0
                            :bloc=bloc+1
                            :f%(pila%(prox%(0),0),0)=-bloc

```

```

1302         END IF
1303     END IF
1304         prox%(0)=prox%(prox%(0)+1)
1305     END REPEAT lazo
1306     REPEAT lazo
1307         IF prox%(1)=0 OR pila%(prox%(1),1)<v:EXIT lazo
1308         IF bloque%(j-j<>0)<>0
1309             EMPOTRAR (pila%(prox%(1),0)),
                    (pila%(bloque%(j-1),0)),1
1310     ELSE
1311         IF bloque%(j)<>0
1312             EMPOTRAR (pila%(prox%(1),0)),
                    (pila%(bloque%(j),0)),-1
1313     ELSE
1314         IF pila%(prox%(prox%(1)+1),1)<>0
                    :EMPOTRAR (pila%(prox%(1),0)),
                    (pila%(prox%(prox%(1)+1),0)),1
                    :ELSE
                    IF f%(pila%(prox%(1),0),0)=0
                    :bloc=bloc+1
                    :f%(pila%(prox%(1),0),0)=bloc
1315     END IF
1316     END IF
1317     prox%(1)=prox%(prox%(1)+1)
1318     END REPEAT lazo
1319     IF ruta%(w)<>ruta%(v)
1320         lp=0
1321     REPEAT lazo
1322         IF j=0 THEN EXIT lazo
1323         IF pila%(bloque%(j-1),1)<=f%(ruta%(w),1)
                    AND (pila%(bloque%(j),1)<=f%(ruta%(w),1)
                    OR pila%(prox%(0),1)=0):EXIT lazo
1324         IF pila%(bloque%(j-1),1)>f%(ruta%(w),1)
1325         IF pila%(bloque%(j),1)>f%(ruta%(w),1)
                    :PRINT'NO PLANAR 1':plano=0:RETURN
1326         lp=bloque%(j-1)
1327     ELSE

```

```

1328         guarda=prox%(lp+1)
1329         prox%(lp+1)=prox%(0)
1330         prox%(0)=prox%(bloque%(j)+1)
1331         prox%(bloque%(j)+1)=guarda
1332         lp=bloque%(j)
1333     END IF
1334     j=j-2
1335 END REPEAT lazo
1336 IF j<>0
1337     IF bloque%(j-1)=0
1338         IF lp<>0 OR bloque%(j)<>0
1339             bloque%(j-1)=lp
1340         ELSE
1341             j=j-2
1342         END IF
1343     END IF
1344 END IF
1345 prox%(0)=prox%(prox%(0)+1)
1346 END IF
1347 ELSE
1348     IF s=0:p=p+1:s=v
1349     f%(p,1)=w:f%(p,2)=v:f%(p,3)=s
1350     lp=0
1351     rp=-1
1352     REPEAT lazo
1353         IF (prox%(lp+1)=0 OR pila%(prox%(lp+1),1)<=w)
1354             AND (prox%(rp+1)=0 OR pila%(prox%(rp+1),1)<=w)
1355             OR j=0 THEN EXIT lazo
1356         IF bloque%(j-1)<>0 AND bloque%(j)<>0
1357             IF pila%(prox%(lp+1),1)>w
1358                 IF pila%(prox%(rp+1),1)>w
1359                     PRINT "NO PLANAR"
1360                     PRINT "Aristas: "!romenu%(f%(p,1));"-";
1361                         romenu%(f%(p,2))!"y"!
1362                     PRINT !romenu%(f%(pila%(bloque%(j-1),0)
1363                         ,1));"-";
1364                     PRINTromenu%(f%(pila%(bloque%(j-1),0),2))

```

```

1361         plano=0
1362         RETurn
1363     END IF
1364     EMPOTRAR (p), (pila%(bloque%(j-1),0)), -1
1365     guarda=prox%(rp+1)
1366     prox%(rp+1)=prox%(lp+1)
1367     prox%(lp+1)=guarda
1368     guarda=prox%(bloque%(j-1)+1)
1369     prox%(bloque%(j-1)+1)=prox%(bloque%(j)+1)
1370     prox%(bloque%(j)+1)=guarda
1371     lp=bloque%(j)
1372     rp=bloque%(j-1)
1373     ELSE
1374         lp=bloque%(j-1)
1375         rp=bloque%(j)
1376     END IF
1377     ELSE
1378         IF bloque%(j-1)<>0
1379             EMPOTRAR (p), (pila%(bloque%(j-1),0)), -1
1380             guarda=prox%(bloque%(j-1)+1)
1381             prox%(bloque%(j-1)+1)=prox%(rp+1)
1382             prox%(rp+1)=prox%(lp+1)
1383             prox%(lp+1)=guarda
1384             rp=bloque%(j-1)
1385         ELSE
1386             IF bloque%(j)<>0:rp=bloque%(j)
1387         END IF
1388     END IF
1389     j=j-2
1390 END REPEAT lazo
1391 IF f%(ruta%(s),1)<w
1392     IF lp=0:lp=libre
1393     pila%(libre,1)=f%(p,1):pila%(libre,0)=p
1394     prox%(libre+1)=prox%(1)
1395     prox%(1)=libre
1396     libre=libre+1
1397 END IF

```









```

1536         IF siguiente=0
1537             vertice=ady%(fila,primero)
1538             ady%(fila,primero)=-ady%(fila,primero)
1539         ELSE
1540             vertice=ady%(fila,siguiente)
1541             ady%(fila,siguiente)=-ady%(fila,siguiente)
1542         END IF
1543         ady%(fila,lugar)=
                ady%(fila,lugar)*(ady%(fila,lugar)>0)
1544     END IF
1545     cara$(num_cara)=cara$(num_cara)&CHR$(vertice)
1546     IF vertice=s:EXIT lazo
1547     temp1=fila
1548     fila=vertice
1549     vertice=temp1
1550 END REPEAT lazo
1551 END IF
1552 END FOR j
1553 END FOR i
1554 FOR i=1 TO p
1555     IF f%(i,0)=0
1556         FOR j=1 TO DIMN(cara$,1)
1557             temp1=CHR$(f%(i,1)) INSTR cara$(j)
1558             temp2=CHR$(f%(i,2)) INSTR cara$(j)
1559             IF temp1 AND temp2
1560                 IF temp1>temp2
1561                     :temp0=temp1:temp1=temp2:temp2=temp0
1562                 REMark Completamos la matriz ADY$.
1563                 k=CODE(cara$(j,temp1))
1564                 temp3=cara$(j,temp1+1) INSTR ady$(k)
1565                 IF temp3=1
1566                     ady$(k)=cara$(j,temp2)&ady$(k)
1567                 ELSE
1568                     ady$(k)=ady$(k,1 TO temp3-1)&
                            cara$(j,temp2)&ady$(k,temp3 TO ady$(k,0))
1569                 END IF
1570                 k=CODE(cara$(j,temp2))

```



```

1603 DIM ady_cara$(num_cara,v/2),par_cara%(num_cara,2),
      vert_comune$(num_cara,v/2)
1604 num_par_corte=0:dual_arist$='':ady_arist$=' '
      :posible_par_corte%=0
1605 FOR i=1 TO num_cara-1
1606   FOR j=i+1 TO num_cara
1607     adyacentes=0:par_corte=0:par1$="":par2$=""
      :comune$=""
1608     FOR k=1 TO cara$(i,0)-1
1609       lado$=cara$(i,k+1)&cara$(i,k)
1610       temp0=lado$ INSTR cara$(j)
1611       IF temp0
1612         IF NOT adyacentes
1613           ady_cara$(i)=ady_cara$(i)&CHR$(j)
1614           ady_cara$(j)=ady_cara$(j)&CHR$(i)
1615           adyacentes=1
1616         ELSE
1617           posible_par_corte%=1
1618         END IF
1619         ady_arist$=ady_arist$&lado$&lado$(1)&CHR$(0)
1620         dual_arist$=dual_arist$&CHR$(i)&CHR$(j)&
                                CHR$(i)&CHR$(0)
1621         comune$=comune$&cara$(i,k+1)
1622       ELSE
1623         temp1=cara$(i,k+1) INSTR cara$(j)
1624         IF temp1
1625           comune$=comune$&cara$(i,k+1)
1626           IF par1$<>" "
1627             par2$=cara$(i,k+1)
1628             temp2=(par1$&par2$ INSTR cara$(i))OR
                    (par2$&par1$ INSTR cara$(i))
1629             temp1=(par1$&par2$ INSTR cara$(j))OR
                    (par2$&par1$ INSTR cara$(j))
1630             temp0=temp2 AND temp1
1631             IF (NOT par_corte) AND (NOT temp0) AND
                i<>cara_exterior AND j<>cara_exterior
1632               num_par_corte=num_par_corte+1

```



```

1665     DFS_CARAS comienzo
1666     IF CHR$(cara_exterior) INSTR
                                comp_par_cara$(puntero)
1667         comp_par_cara$(puntero)=" "
1668         puntero=puntero-1
1669         par_cara%(i,0)=puntero
1670     ELSE
1671         comune$=""
1672         FOR k=1 TO comp_par_cara$(puntero,0)
1673             comune$=comune$&
                                cara$(CODE(comp_par_cara$(puntero,k)))
1674         END FOR k
1675         FOR j=1 TO vert_comune$(i,0)
1676             temp1=vert_comune$(i,j) INSTR comune$
1677             IF temp1
                                :par_vert$(puntero)=par_vert$(puntero)&
                                vert_comune$(i,j)
1678         END FOR j
1679         REMark Modifico las cara$(par_cara%(i)) para
                                que entre el temp1-temp2 se encuentre siempre
                                el bloque a transmitir.
1680         a=par_cara%(i,1):b=par_cara%(i,2):temp0=0
1681         temp1=par_vert$(puntero,1) INSTR cara$(a)
1682         temp2=par_vert$(puntero,2) INSTR cara$(a)
1683         IF temp1>temp2
1684             temp0=temp1:temp1=temp2:temp2=temp0:temp0=0
1685             a$=par_vert$(puntero,1)
                                :par_vert$(puntero,1)=par_vert$(puntero,2)
                                :par_vert$(puntero,2)=a$
1686         END IF
1687         IF temp1<>temp2-1
1688             IF NOT(cara$(a,temp1+1) INSTR comune$)
                                :DAR_VUELTA a,temp1:temp0=1
1689         ELSE
1690             IF temp1=1
1691                 IF cara$(a,3) INSTR comune$
                                :DAR_VUELTA a,temp1:temp0=1

```

```

1692         ELSE
1693             IF cara$(a,1) INSTR comune$
                :DAR_VUELTA a,temp1:temp0=1
1694         END IF
1695     END IF
1696     IF temp0
1697         a$=par_vert$(puntero,1)
                :par_vert$(puntero,1)=par_vert$(puntero,2)
                :par_vert$(puntero,2)=a$
1698     END IF
1699     temp1=par_vert$(puntero,2) INSTR cara$(b)
1700     temp2=par_vert$(puntero,1) INSTR cara$(b)
1701     IF temp1>temp2:DAR_VUELTA b,temp2
1702     REMark ----- DAR_VUELTA-----
1703     REMark Invierte de orden una serie de elementos
1704     DEFine PROCEDURE DAR_VUELTA (a,b)
1705     FOR j=1 TO b
1706         a$=cara$(a,1)
                :cara$(a)=cara$(a,2 TO cara$(a,0))&a$
1707     END FOR j
1708     END DEFine DAR_VUELTA
1709 :
1710     END IF
1711     ELSE
1712         EXIT lazo
1713     END IF
1714     END REPEAT lazo
1715 END FOR i
1716 REMark Modifico par_cara% y vert_comune$ para una
        correspondencia biunivoca con comp_par_cara$.
1717 FOR i=num_par_corte TO 1 STEP -1
1718     IF i=par_cara%(i,0):EXIT i
1719     j=par_cara%(i,0):temp0=par_cara%(i-1,0)
1720     REPEAT lazo
1721         IF j=temp0:EXIT lazo
1722         vert_comune$(j)=vert_comune$(i)
1723         par_cara%(j,0)=par_cara%(i,0)

```











```

1858 vp%=v:num_carap=2
1859 cara$(1)=cara$(1)&cara$(1,1)
1860 carap$(1)=cara$(1)
1861 FOR i=2 TO num_cara
1862   cara$(i)=cara$(i)&cara$(i,1)
      :carap$(num_carap)=cara$(i)
1863   IF LEN(cara$(i))=4
1864     num_carap=num_carap+1
1865     NEXT i:EXIT i
1866   END IF
1867   vp%=vp%+1:k4%=0
1868   FOR j=1 TO cara$(i,0)-1
1869     temp=CODE(cara$(i,j))
1870     IF ady$(temp,0)<>2:NEXT j:EXIT j
1871     IF ady$(temp,1) INSTR ady$(CODE(ady$(temp,2)))
      :k4%=1
      :PRINT "K4 en cara"!i!"vertice"!CODE(cara$(i,j))!
      "con"!CODE(ady$(temp1))!CODE(ady$(temp1,2))
      :EXIT i
1872   END FOR j
1873   b#=cara$(i):carap$(num_carap)=" "
1874   FOR j=1 TO LEN(b#)-1
1875     temp=CODE(b$(j))
1876     adyp$(vp%)=b$(j)&adyp$(vp%)
1877     IF j=1
1878       ant#=b$(LEN(b#)-1)
1879     ELSE
1880       ant#=b$(j-1)
1881     END IF
1882     temp1=ant# INSTR adyp$(temp)
1883     IF temp1=adyp$(temp,0)
1884       adyp$(temp)=adyp$(temp)&CHR$(vp%)
1885     ELSE
1886       adyp$(temp)=adyp$(temp,1 TO temp1)&
      CHR$(vp%)&adyp$(temp,temp1+1 TO adyp$(temp,0))
1887     END IF
1888     carap$(num_carap)=CHR$(vp%)&b$(j TO j+1)&CHR$(vp%)

```

















```

2137 FOR i=1,2:READ posib4$(i)
2138 FOR i=1 TO num_cara
      :solucion$(i)=FILL$("_",ady_cara$(i,0))
      :paso_gene$(i)=FILL$(CHR$(0),ady_cara$(i,0))
2139 FOR i=1 TO v:solucion_vert$(i)=FILL$("_",ady$(i,0))
2140 top=num_cara:bot=2:orient1$=""
      :orient2$="":copia%=0:num_solu%=0:num_graf%=0
2141 REMark Introducimos en SOLUCION$ y ADY% las aristas con
      orientación fijada.
2142 FOR i=2 TO num_cara
2143   borde=0
2144   IF CHR$(1) INSTR cara$(i)
      :orient1$=orient1$&CHR$(i):borde=1:coord(i,2)=100
2145   IF CHR$(2) INSTR cara$(i)
      :orient2$=orient2$&CHR$(i):borde=1:coord(i,1)=5
2146   IF CHR$(3) INSTR cara$(i)
      :orient1$=orient1$&CHR$(i):borde=1:coord(i,2)=5
2147   IF CHR$(4) INSTR cara$(i)
      :orient2$=orient2$&CHR$(i):borde=1:coord(i,1)=161
2148   SELEct ON borde
2149     =0:ind%(top)=i:top=top-1
2150     =1:ind%(bot)=i:bot=bot+1
2151   END SELEct
2152 END FOR i
2153 REMark La matriz CARA$ ha sido ordenada según pertenez-
      ca al BORDE ó no. A continuación vamos a introducir las
      aristas con orientación fija, al mismo tiempo se forma-
      rá la MATRIZ para GAUSS.
2154 paso%=0
2155 FOR i=2 TO bot-1
2156   k=ind%(i)
2157   FOR j=1 TO ady_cara$(k,0)
2158     IF solucion$(k,j)="_"
2159       IF CHR$(k) INSTR orient1$ AND CHR$(k) INSTR
                                                    orient2$
2160         IF ady_cara$(k,j)=CHR$(1)
           :solucion$(k,j)="E":NEXT j

```

```

2161         IF ady_cara$(k,j) INSTR orient1$
2162             IF NOT(ady_cara$(k,j) INSTR orient2$)
2163                 SOLU (k),(j),"H"
2164             ELSE
2165                 temp=CHR$(k)&ady_cara$(k,j) INSTR
2166                                     dual_arist$
2167                 IF ady_arist$(temp) INSTR CHR$(1)&CHR$(3)
2168                 OR ady_arist$(temp+1) INSTR CHR$(1)&CHR$(3)
2169                     SOLU (k),(j),"H"
2170                 ELSE
2171                     SOLU (k),(j),"V"
2172                 END IF
2173             END IF
2174         ELSE
2175             IF ady_cara$(k,j) INSTR orient1$&orient2$
2176                 IF CHR$(k) INSTR orient1$
2177                     SOLU (k),(j),"H"
2178                 ELSE
2179                     IF CHR$(k) INSTR orient2$:SOLU (k),(j),"V"
2180                 END IF
2181             END IF
2182         END IF
2183     END IF
2184 END FOR j
2185 END FOR i
2186 IF dibujar%
2187     DIM clase$(1,num_cara/2+7,num_cara/2+7),num_clase%(1)
2188     CLASES "V",clase$(1),orient2$,1,num_clase%(1)
2189     CLASES "H",clase$(0),orient1$,2,num_clase%(0)
2190     DIBUJAR
2191     INPUT#0,"¿Desea fijar la orientación de algún tabique
2192             ? (s/n)!"a$
2193     IF a$=="S"
2194         PRINT#0,"Introduzca 0 para salir."

```



```

2228 solucion$(1,CHR$(k) INSTR ady_cara$(1))=hv$
2229 paso_gene$(k,j)=CHR$(paso%)
2230 paso_gene$(1,CHR$(k) INSTR ady_cara$(1))=CHR$(paso%)
2231 IF POSIBLE(k,j,l,m,n)
2232     borra%=0
2233     FOR i=k,1
2234         IF "_" INSTR solucion$(i)
2235             IF ady_cara$(i,0)=3
2236                 temp1=solucion$(i) INSTR "_VV_V,_HH_H"
2237                 IF temp1
2238                     SELEct ON temp1
2239                         =1:h=1:a#=h$
2240                         =2:h=3:a#=h$
2241                         =3:h=2:a#=h$
2242                         =7:h=1:a#=v$
2243                         =8:h=3:a#=v$
2244                         =9:h=2:a#=v$
2245                     END SELEct
2246                     temp2=CHR$(i)&ady_cara$(i,h)INSTR dual_arist$
2247                     SOLU (i),(h),(a$)
2248                     IF borra%
2249                         BORRAR (i),(h)
2250                     RETurn
2251                 END IF
2252             END IF
2253         ELSE
2254             VALE4 solucion$
2255             REMark ----- VALE4 -----
2256             DEFIne PROCedure VALE4 (solucion$)
2257                 temp1="V" INSTR solucion$(i)
2258                 temp2="H" INSTR solucion$(i)
2259                 IF temp1 OR temp2
2260                     SELEct ON temp1
2261                         =1:b$="HVH":c$="234"
2262                         :IF CODE(hv$)>100:b$="hvh"
2261                         =3:b$="VHH":c$="124"
2262                         :IF CODE(hv$)>100:b$="vhh"

```

```

2263         =2:b$="HHV":c$="134"
           :IF CODE(hv$)>100:b$="hhv"
2264         =4:b$="HVH":c$="123"
           :IF CODE(hv$)>100:b$="hvh"
2265     END SElect
2266     SElect ON temp2
2267         =1:b$="VHV":c$="234"
           :IF CODE(hv$)>100:b$="vhv"
2268         =3:b$="HVV":c$="124"
           :IF CODE(hv$)>100:b$="hvv"
2269         =2:b$="VVH":c$="134"
           :IF CODE(hv$)>100:b$="vvh"
2270         =4:b$="VHV":c$="123"
           :IF CODE(hv$)>100:b$="vhv"
2271     END SElect
2272     END DEfine VALE4
2273     FOR h=1 TO 3
2274         IF solucion$(i,c$(h))="_"
2275             temp2=CHR$(i)&ady_cara$(i,c$(h)) INSTR
                                                    dual_arist$
2276             SOLU (i),(c$(h)),b$(h)
2277             IF borra%
2278                 BORRAR (i),(c$(h))
2279             RETurn
2280             END IF
2281         END IF
2282     END FOR h
2283     END IF
2284     END IF
2285     END IF
2286     END FOR i
2287     FOR i=m,n
2288         SElect ON i=1 TO 4:EXIT i
2289         IF "_" INSTR solucion_vert$(i)
2290             IF ady$(i,0)=4
2291                 VALE4 solucion_vert$
2292             FOR h=1 TO 3

```





```

2354     solucion_vert$(temp)=
           solucion_vert$(temp,2 TO solucion_vert$(temp,0))&
           solucion_vert$(temp,1)

2355     SELEct ON temp2
2356         =0
2357         IF solucion_vert$(temp,temp1)==hv$
2358             NEXT i
2359         ELSE
2360             temp2=1:NEXT i
2361         END IF
2362     =1,2
2363     IF solucion_vert$(temp,temp1)==hv$
2364         ari$=ari$&CHR$(temp)&ady$(temp,temp1)&CHR$(0)
           :temp2=2
2365     IF NOT(ady$(temp,temp1) INSTR pila$) AND
           numero$(CODE(ady$(temp,temp1)))=CHR$(0)
           AND ady$(temp,temp1)<>fin$
2366         pila$=pila$&ady$(temp,temp1)
2367     END IF
2368     numero$(CODE(ady$(temp,temp1)))=CHR$(temp)
2369     ELSE
2370         IF temp2=2 AND ari$<>"":EXIT i
2371     END IF
2372     END SELEct
2373     END FOR i
2374     IF LEN(pila$)=1:EXIT lazo
2375     pila$=pila$(2 TO)
2376     temp2=0
2377     END REPEAT lazo
2378     num_clase%=0
2379     FOR i=2 TO num_cara
2380     IF numero%(i):NEXT i:EXIT i
2381     num_clase%=num_clase%+1:numero%(i)=1
2382     clase$(num_clase%)=CHR$(i):pert_cl%(i)=num_clase%
2383     FOR j=1 TO ady_cara$(i,0)
2384     IF solucion$(i,j)==hv$
2385     l=CODE(ady_cara$(i,j)):numero%(l)=1

```

```

2386     A\ADIR i,ady_cara$(i,j):pert_c1%(1)=num_clase%
2387     REPEAT lazo
2388         FOR k=1 TO ady_cara$(1,0)
2389             IF solucion$(1,k)==hv$ AND NOT
                numero%(CODE(ady_cara$(1,k)))
2390                 A\ADIR 1,ady_cara$(1,k)
2391                 l=CODE(ady_cara$(1,k)):numero%(1)=1
2392                 pert_c1%(1)=num_clase%
2393             NEXT lazo
2394         END IF
2395     END FOR k
2396     EXIT lazo
2397     END REPEAT lazo
2398 END IF
2399 END FOR j
2400 END FOR i
2401 :
2402 REMark ----- A\ADIR -----
2403 DEFine PROCedure A\ADIR(i,a$)
2404 temp0=CHR$(i)&a$ INSTR dual_arist$
2405 IF ady_arist$(temp0)&ady_arist$(temp0+1) INSTR ari$
2406     clase$(num_clase%)=clase$(num_clase%)&a$
2407 ELSE
2408     clase$(num_clase%)=a$&clase$(num_clase%)
2409 END IF
2410 END DEFine A\ADIR
2411 :
2412 REMark Pasamos a formar MATRIZ para GAUSS
2413 FOR i=1 TO num_cara:numero%(i)=0
2414     l=0
2415     FOR i=1 TO num_clase%
2416         IF NOT(clase$(i,1) INSTR borde$):l=l+1:numero%(i)=1
2417     END FOR i
2418 DIM matriz(num_clase%-2,num_clase%-1)
2419 FOR i=1 TO num_clase%
2420     j=numero%(i):IF NOT j:NEXT i:EXIT i
2421     m=CODE(clase$(i,1))

```





```

2488 temp=solucion$(u) INSTR "_V__VH__HV_H_V____"
2489 SElect ON temp
2490   =1,2,3:a$="hhvh":b$="3":c$="2"
2491   =6,7,11:a$="vvhv":b$="3":c$="2"
2492   =4,5,8,9,10,12:a$="vh":b$="2":c$="1"
2493   =15
2494     IF ady_cara$(u,0)=4
2495       a$="vh":b$="2":c$="1"
2496     ELSE
2497       a$="vvhvhvv":b$="6":c$="3"
2498     END IF
2499   =REMAINDER :STOP
2500 END SElect
2501 FOR i=1 TO b$
2502   temp1=0
2503   FOR j=1 TO c$
2504     IF "_" INSTR solucion$(u)
2505       temp1="_" INSTR solucion$(u,temp1+1 TO)+temp1
2506       SOLU (u),(temp1),(a$(i+j-1))
2507       IF solucion$(u) INSTR ant$:borra%=1:BEEP 1000,10
2508     END IF
2509     IF borra%
2510       BORRAR u,(temp1)
2511       GO TO 2534:REMark Salgo para borrar y retornar.
2512     END IF
2513   END FOR j
2514   ant$=ant$&solucion$(u)&CHR$(0)
2515   w=u
2516   REPeat lazo
2517     IF w=num_cara:EXIT lazo
2518     w=w+1
2519     IF "_" INSTR solucion$(w):EXIT lazo
2520   END REPeat lazo
2521   IF "_" INSTR solucion$(w)
     :GENERAR (w)
     :paso%=paso%-1
     :IF terminar%:RETurn

```







```

2621 IF canal=5
      :PRINT #canal,TO 23;i;TO 27;"-";TO 31;
          tras(numero%(i+3)-3);TO 35;"m"
2622 END FOR i
2623 PRINT #canal,\'Area minima del local en m2.
          (0 si no desea limitarla)\'\
2624 FOR i=1 TO v-3
2625 INPUT TO 23;(i+3);TO 27;"-";TO 31;
          s(numero%(i+3)-3);TO 35;"m2"
2626 IF canal=5
      :PRINT #canal,TO 23;i+3;TO 27;"-";TO 31;
          s(numero%(i+3)-3);TO 35;"m2"
2627 END FOR i
2628 FOR i=1 TO vi%:c(i)=1
2629 PRINT #canal,"La F.O. por defecto es minimizar la suma
          de todos los perimetros, (s/n)!"
2630 INPUT a$
2631 IF a$=="N"
2632 PRINT #canal,"Coeficientes de la Función objetivo:"
2633 FOR i=1 TO v-3
2634 INPUT TO 23;(i+3);TO 27;"-";TO 31;
          c(numero%(i+3)-3);", ";c(numero%(i+v)-3)
2635 IF canal=5
      :PRINT #canal,TO 23;(i+3);TO 27;"-";TO 31;
          c(numero%(i+3)-3);", ";c(numero%(i+v)-3)
2636 END FOR i
2637 END IF
2638 REMark Recuperamos cada solucion para dimensionarla.
2639 FOR j=1 TO num_solu%
2640 RECUPERA_SOLUCION (j)
2641 resti%=2*vi%+1.5*(num_clase%(0)+num_clase%(1))
2642 DIM tip%(resti%),b(resti%),r(resti%,vi%)
2643 resti%=0:rest0%=0:rest1%=0:rest2%=0
2644 FOR i=1 TO v-3
2645 IF s(i)
2646 resti%=resti%+4:rest1%=rest1%+4
2647 b(resti%)=2*tras(i)*(s(i)-tras(i)^2)

```

```

2648      b(resti%-1)=b(resti%)
2649      r(resti%,i)=tras(i)*SQRT(s(i))+s(i)
2650      r(resti%,i+v-3)=2*tras(i)^2
2651      r(resti%-1,i)=r(resti%,i+v-3)
2652      r(resti%-1,i+v-3)=r(resti%,i)
2653      b(resti%-2)=2*tras(i)*SQRT(s(i))+s(i)-3*tras(i)^2
2654      b(resti%-3)=b(resti%-2)
2655      r(resti%-2,i)=tras(i)+SQRT(s(i))
2656      r(resti%-2,i+v-3)=2*tras(i)
2657      r(resti%-3,i)=r(resti%-2,i+v-3)
2658      r(resti%-3,i+v-3)=r(resti%-2,i)
2659      FOR k=0 TO 3:tip%(resti%-k)=1
2660  END IF
2661  END FOR i
2662  CLASIFICAR
2663  RESTRICCIONES resti%
2664  x=(vi%+resti%+rest1%)*2+resti%*2+(vi%+rest1%)*resti%
2665  CLS#0
2666  SIMPLEX 0,x
2667  REMark Trazar las plantas e imprimir resultados.
2668  FOR i=1 TO v-3
2669      r(0,i)=r(0,i)+tras(i)
2670      r(0,i+v-3)=r(0,i+v-3)+tras(i)
2671  END FOR i
2672  b(0)=0
2673  FOR i=1 TO vi%:b(0)=b(0)+c(i)*r(0,i)
2674  escala=r(0,v-2)
2675  SCALE escala,0,0
2676  DIM pert_clase%(1,num_cara)
2677  FOR i=0,1
2678      FOR k=1 TO num_clase%(i)
2679          FOR l=1 TO clase$(i,k,0)
2680              pert_clase%(i,CODE(clase$(i,k,l)))=k
2681          END FOR l
2682      END FOR k
2683  END FOR i
2684  FOR i=2 TO num_cara

```

```

2685     IF coord(i,1)=5
           :coord(i,1)=0:coord(i,0)=coord(i,0)+2
2686     IF coord(i,1)=161
           :coord(i,1)=r(0,1):coord(i,0)=coord(i,0)+2
2687     IF coord(i,2)=5
           :coord(i,2)=0:coord(i,0)=coord(i,0)+1
2688     IF coord(i,2)=100
           :coord(i,2)=escala:coord(i,0)=coord(i,0)+1
2689     END FOR i
2690     FOR i=0,1
2691         pila$=""
2692         FOR k=1 TO num_clase%(i):pila%=pila%&CHR$(k)
2693         REPEAT lazo
2694             DIM lon_sup(1),lon_inf(1)
2695             k=CODE(pila$(1))
2696             temp=CODE(clase$(i,k,1))
2697             IF coord(temp,0)=0 OR coord(temp,0)=i+1
2698                 pila%=pila$(2 TO)&pila$(1)
2699             NEXT lazo
2700         END IF
2701         lon_sup(1)=coord(temp,i+1):lon_inf(1)=lon_sup(1)
2702         FOR l=1 TO clase$(i,k,0)-1
2703             temp=clase$(i,k,l)&clase$(i,k,l+1) INSTR
                                                    dual_arist#
2704             temp1=CODE(ady_arist$(temp))
2705             IF temp1<4:temp1=4
2706             temp2=CODE(ady_arist$(temp+1))
2707             IF temp2<4:temp2=4
2708             IF lon_inf(0)<>temp1
2709                 lon_inf(0)=temp1
2710                 lon_inf(1)=lon_inf(1)+r(0,temp1-3+(v-3)*i)
2711             END IF
2712             IF lon_sup(0)<>temp2
2713                 lon_sup(0)=temp2
2714                 lon_sup(1)=lon_sup(1)+r(0,temp2-3+(v-3)*i)
2715             END IF
2716             temp=CODE(clase$(i,k,l+1))

```



```

2750 REPEAT lazo
2751     cambio%=0
2752     FOR k=1 TO clase$(i,j,0)-1
2753         IF coord(CODE(clase$(i,j,k)),i+1) >
                coord(CODE(clase$(i,j,k+1)),i+1)
2754             a$=clase$(i,j,k)
2755             clase$(i,j,k)=clase$(i,j,k+1)
2756             clase$(i,j,k+1)=a$
2757             cambio%=cambio%+1
2758         END IF
2759     END FOR k
2760     IF NOT cambio%:EXIT lazo
2761     cambio%=0
2762     FOR k=clase$(i,j,0) TO 2 STEP -1
2763         IF coord(CODE(clase$(i,j,k)),i+1) <
                coord(CODE(clase$(i,j,k-1)),i+1)
2764             a$=clase$(i,j,k-1)
2765             clase$(i,j,k-1)=clase$(i,j,k)
2766             clase$(i,j,k)=a$
2767             cambio%=cambio%+1
2768         END IF
2769     END FOR k
2770     IF NOT cambio%:EXIT lazo
2771 END REPEAT lazo
2772 END FOR j
2773 FOR j=1 TO num_clase%(i)
2774     IF coord(CODE(clase$(i,j,1)),(i<>1)+1) =
                (i=0)*100+(i=1)*5
2775         a$=clase$(i,j)
2776         clase$(i,j)=clase$(i,num_clase%(i))
2777         clase$(i,num_clase%(i))=a$
2778     EXIT j
2779 END IF
2780 END FOR j
2781 END FOR i
2782 END DEFINE CLASIFICAR
2783 :

```





```

2848 DEFine PROCedure SIMPLEX (tipo%,x)
2849 LOCal i,j,lazo,v%,l1%,l2%,l3%,l4%,l5%,c1%,crneg,cr,
           ineg%,bamp,ba,jsal%,apiv,apiva,col,vueltas%,a(x)
2850 v%=vi%+restl%
2851 l1%=v%+resti%
2852 l2%=l1%+v%
2853 l3%=l2%+resti%
2854 l4%=l3%+resti%
2855 FOR i=1 TO vi%:a(i)=c(i)
2856 FOR i=l1%+1 TO l1%+resti%:a(i)=tip%(i-l1%)
2857 FOR i=l3%+1 TO l4%:a(i)=b(i-l3%)
2858 FOR i=1 TO resti%
2859     l5%=l4%+v%*(i-1)
2860     FOR j=1 TO vi%:a(l5%+j)=r(i,j)
2861 END FOR i
2862 IF tipo%=1:FOR i=1 TO vi%:a(i)=-a(i)
2863 c1%=0
2864 FOR j=1 TO resti%
2865     temp=a(l1%+j)
2866     SElect ON temp
2867         =1
2868         c1%=c1%+1:a(vi%+c1%)=0:a(v%+j)=1E30
           :a(l4%+v%*(j-1)+vi%+c1%)=-1
2869         =2
2870         a(v%+j)=1E30
2871     END SElect
2872 END FOR j
2873 FOR i=1 TO v%+resti%:a(l1%+i)=i
2874 REPeat lazo
2875     crneg=0
2876     FOR i=1 TO v%
2877         cr=a(a(l1%+i))
2878         FOR j=1 TO resti%
           :l5%=l4%+v%*(j-1):cr=cr-a(a(l2%+j))*a(l5%+i)
2879         IF cr<crneg:crneg=cr:ineg%=i
2880     END FOR i
2881 IF crneg=0:EXIT lazo

```

```

2882  bamp=1E30
2883  FOR j=1 TO resti%
2884    15%=14%+v%*(j-1)
2885    IF a(15%+ineg%)>0
2886      ba=a(13%+j)/a(15%+ineg%)
2887      IF ba>=0
2888        IF ba<bamp
2889          bamp=ba:jsal%=j
2890        END IF
2891      END IF
2892    END IF
2893  END FOR j
2894  IF bamp>=1E29
2895    :PRINT #canal,"No tiene solución.":RETURN
2896  apiv=a(14%+(jsal%-1)*v%+ineg%)
2897  apiva=1/apiv
2898  a(13%+jsal%)=a(13%+jsal%)*apiva
2899  FOR i=1 TO v%
2900    IF i<>ineg%
2901      a(14%+(jsal%-1)*v%+i)=a(14%+(jsal%-1)*v%+i)*apiva
2902    ELSE
2903      a(14%+(jsal%-1)*v%+ineg%)=apiva
2904    END IF
2905  END FOR i
2906  FOR j=1 TO resti%
2907    IF j=jsal%:NEXT j:EXIT j
2908    15%=14%+v%*(j-1)
2909    apiva=-a(15%+ineg%)
2910    a(13%+j)=a(13%+j)+a(13%+jsal%)*apiva
2911    FOR i=1 TO v%
2912      IF i<>ineg%
2913        a(15%+i)=a(15%+i)+a(14%+(jsal%-1)*v%+i)*apiva
2914      ELSE
2915        a(15%+i)=apiva*a(14%+(jsal%-1)*v%+i)
2916      END IF
2917    END FOR i
2918  END FOR j

```



---

**ANEXO II. EJEMPLO**

---

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

- Vestíbulo
- Distribuidor
- Aseo
- Baño
- Dormitorio 1
- Dormitorio 2
- Dormitorio 3
- Comedor
- Cocina
- Lavadero
- Garage
- 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 1 al cuatro quedan reservados a las orientaciones básicas-:

1 - Norte	5 - Vestibulo	9 - Dorm. 1	13 - Cocina
2 - Oeste	6 - Distrib.	10 - Dorm. 2	14 - Lavadero
3 - Sur	7 - Aseo	11 - Dorm. 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	8 - 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

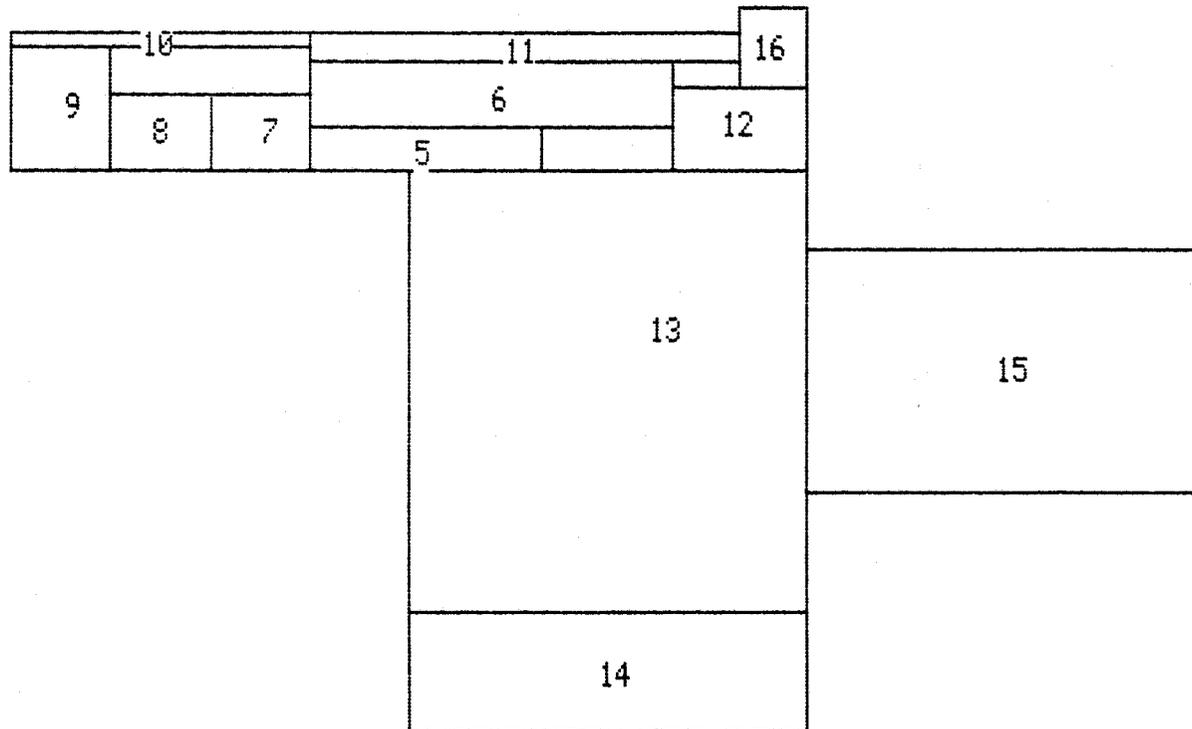
Vértice de articulación en "5" que articula las componentes:

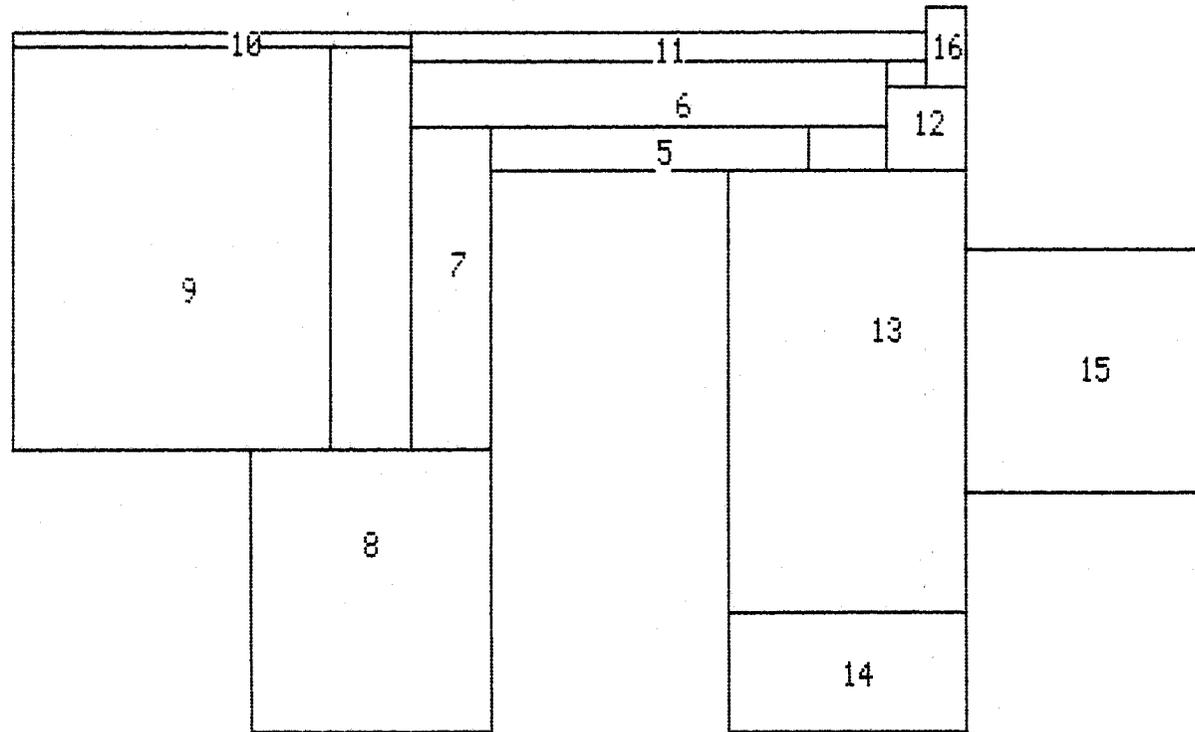
1,2,3,4,6,7,8,9,10,11,12,16 con 13

Así pues, introducimos a continuación las siguientes relaciones de adyacencia:

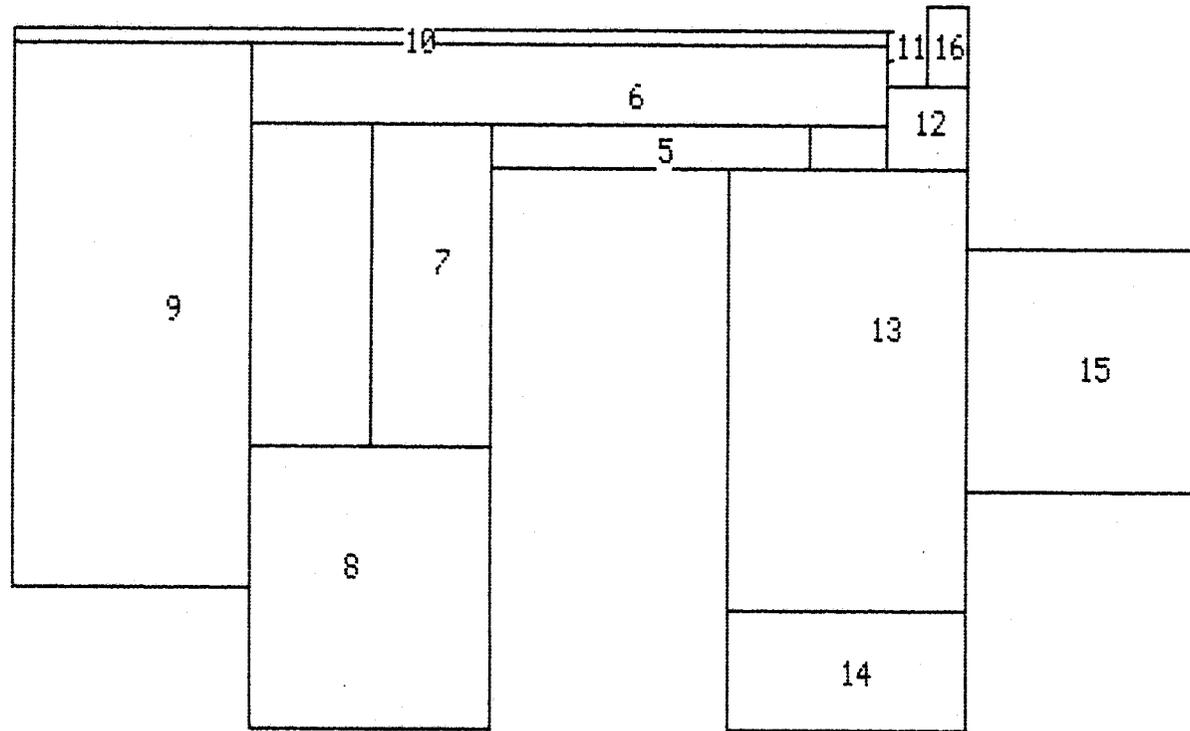
13 - 12    13 - 15    15 - 4    14 - 3    10 - 2    2 - 9  
13 - 14

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.

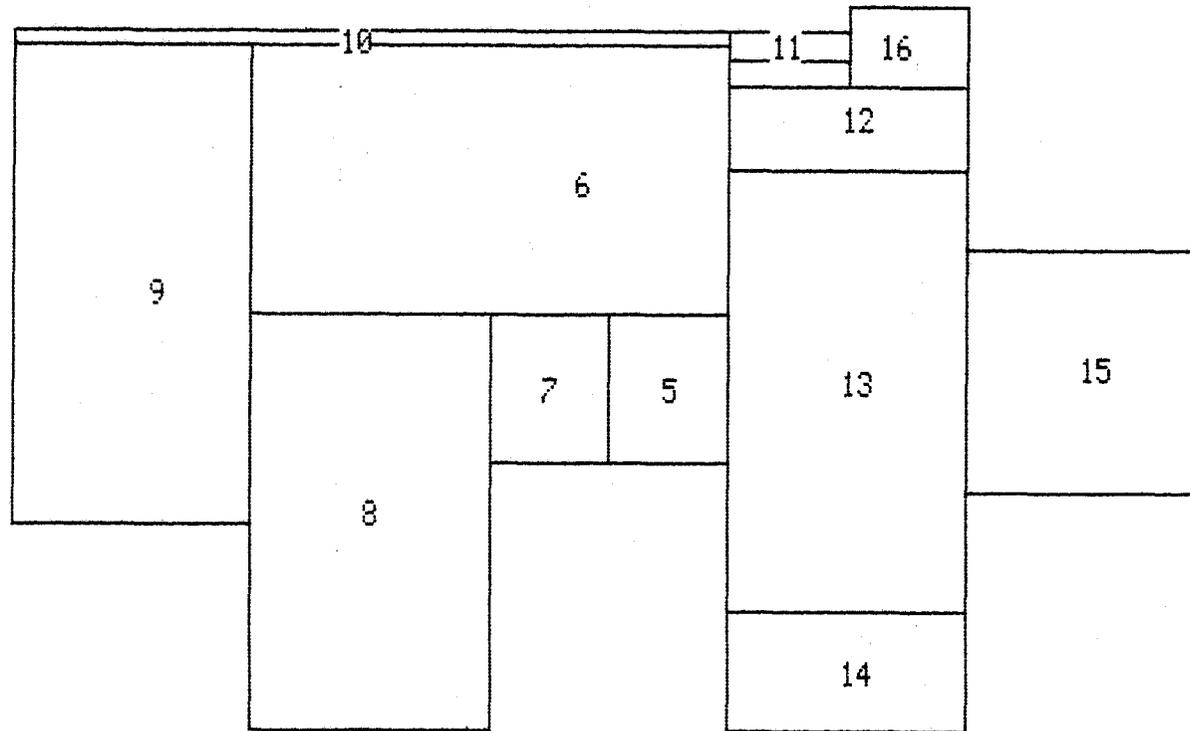




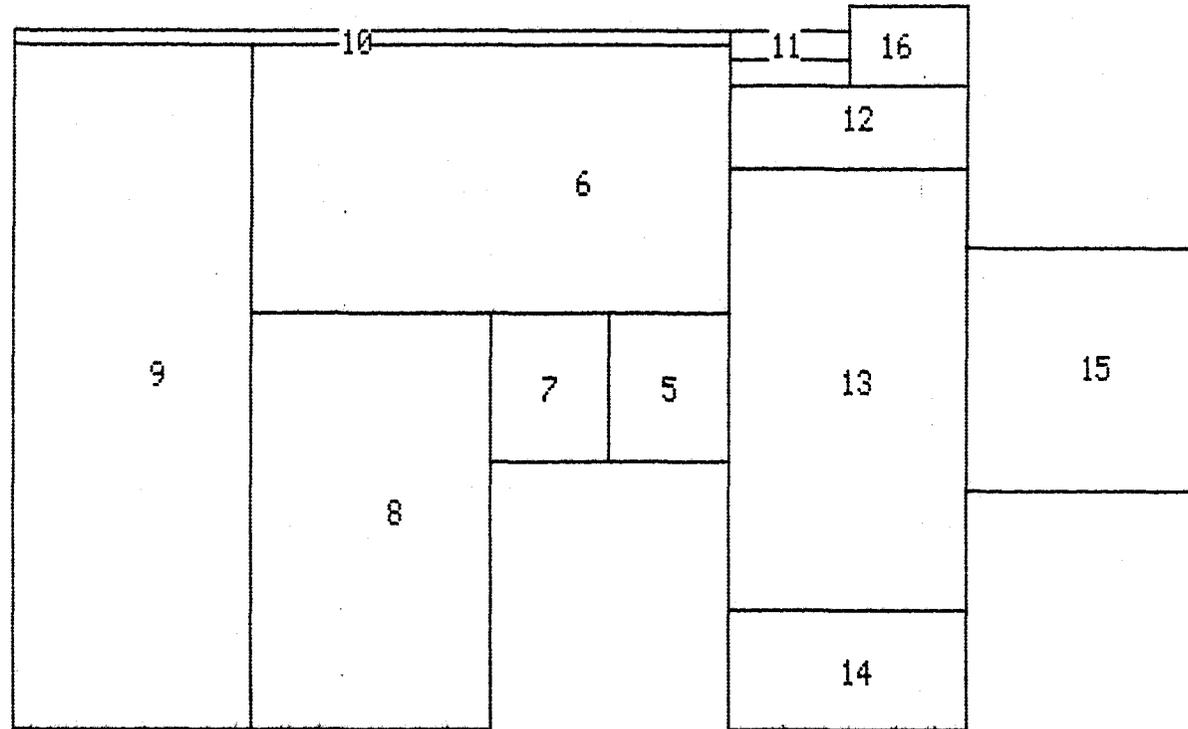
Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 3



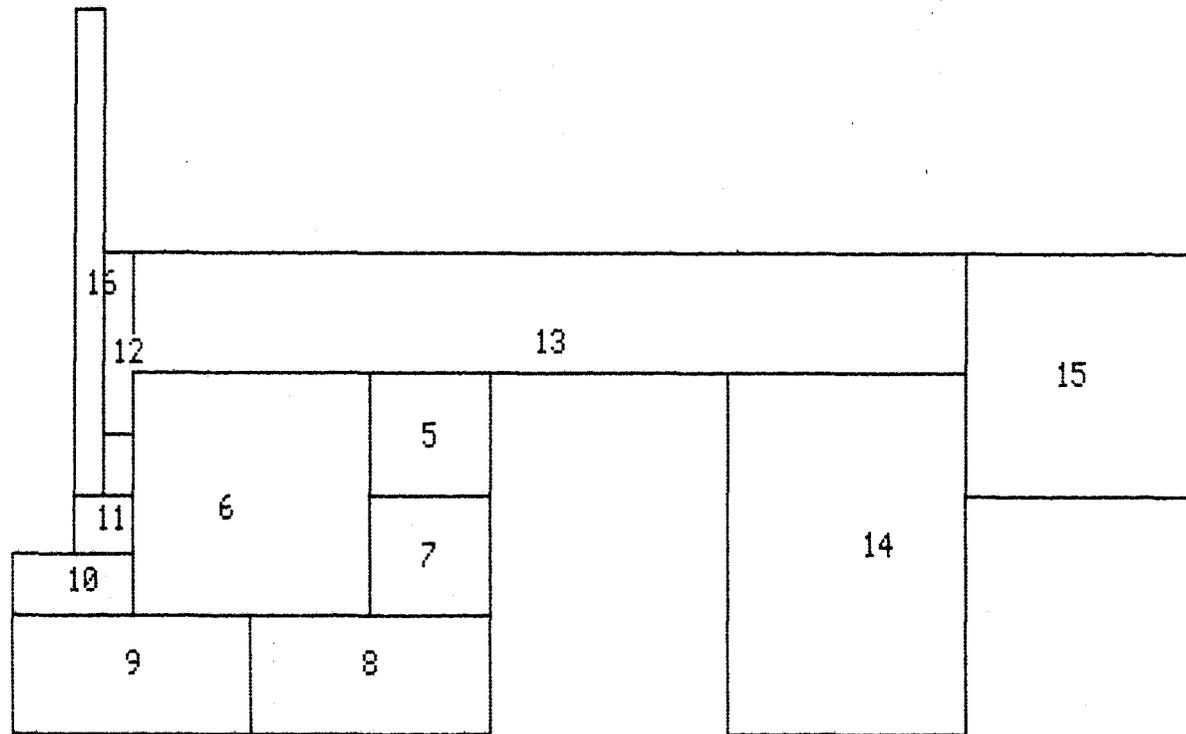
Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 3  
Introducir nueva arista (<0 salir, =0 real->pseudo): 9 - 6  
Introducir nueva arista (<0 salir, =0 real->pseudo): -10 - 6



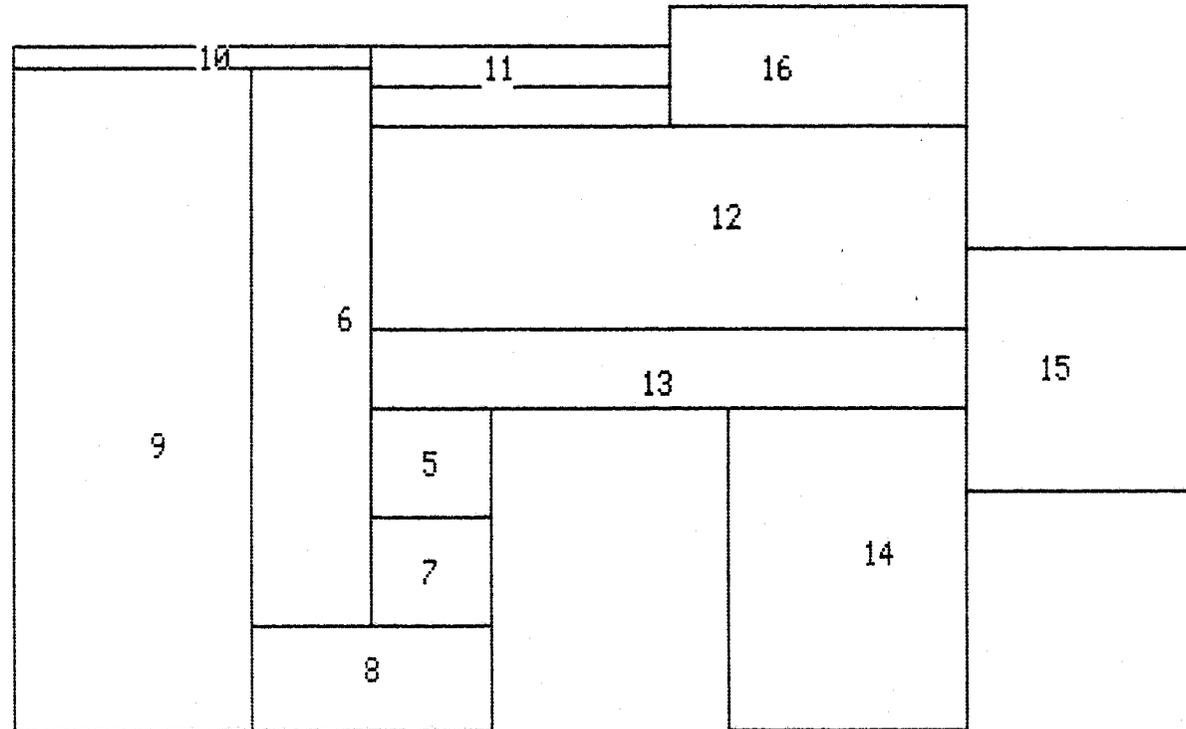
Introducir nueva arista (<0 salir. =0 real->pseudo): 9 - 6  
Introducir nueva arista (<0 salir. =0 real->pseudo): -10 - 6  
Introducir nueva arista (<0 salir. =0 real->pseudo): -8 - 6  
Introducir nueva arista (<0 salir. =0 real->pseudo): -6 - 13



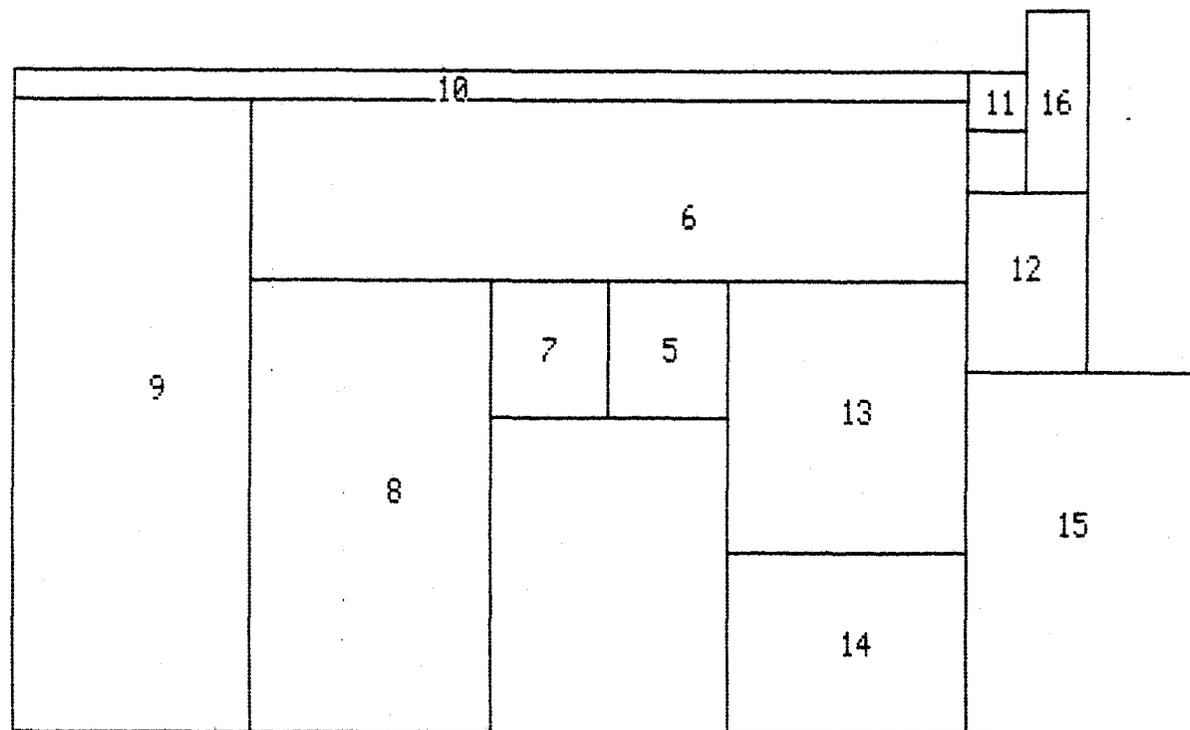
Introducir nueva arista (<0 salir, =0 real->pseudo): -10 - 6  
Introducir nueva arista (<0 salir, =0 real->pseudo): -8 - 6  
Introducir nueva arista (<0 salir, =0 real->pseudo): -6 - 13  
Introducir nueva arista (<0 salir, =0 real->pseudo): -9 - 3



```
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
```



```
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
```



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

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".

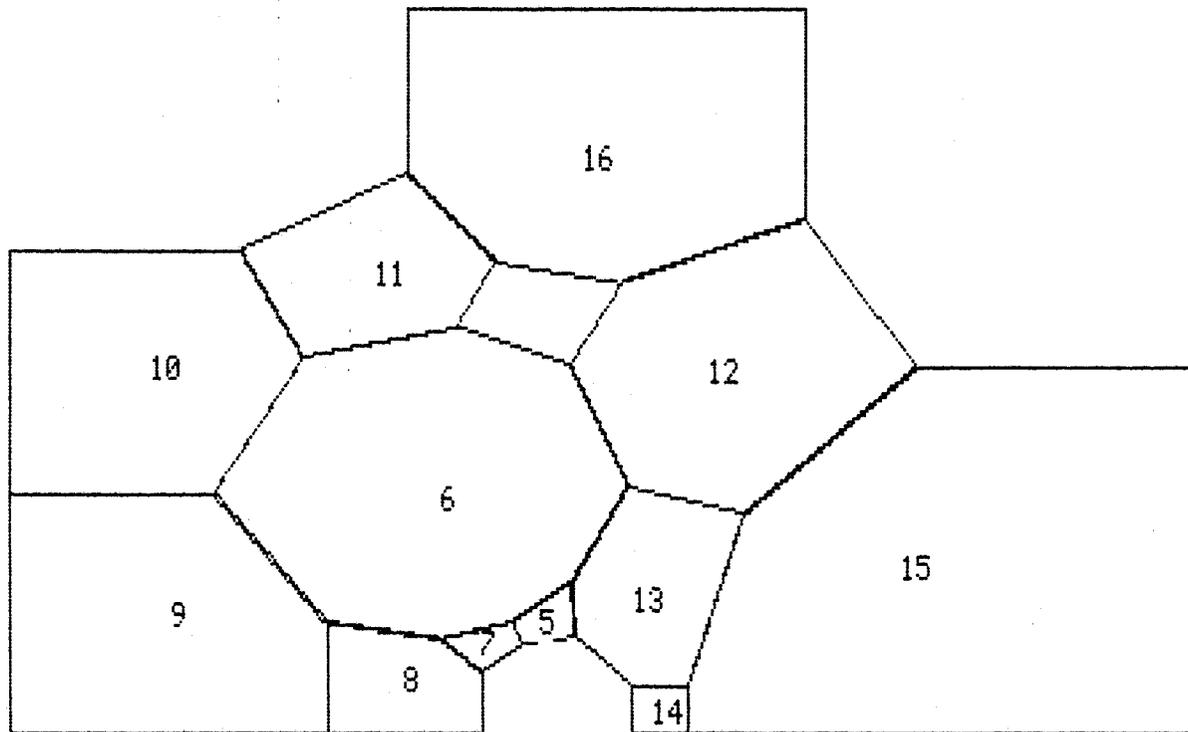
A continuación se representa un esquema adimensional de distribución en planta, en la que quedan señaladas aquellas particiones que poseen una orientación fija. (Figura en página siguiente).

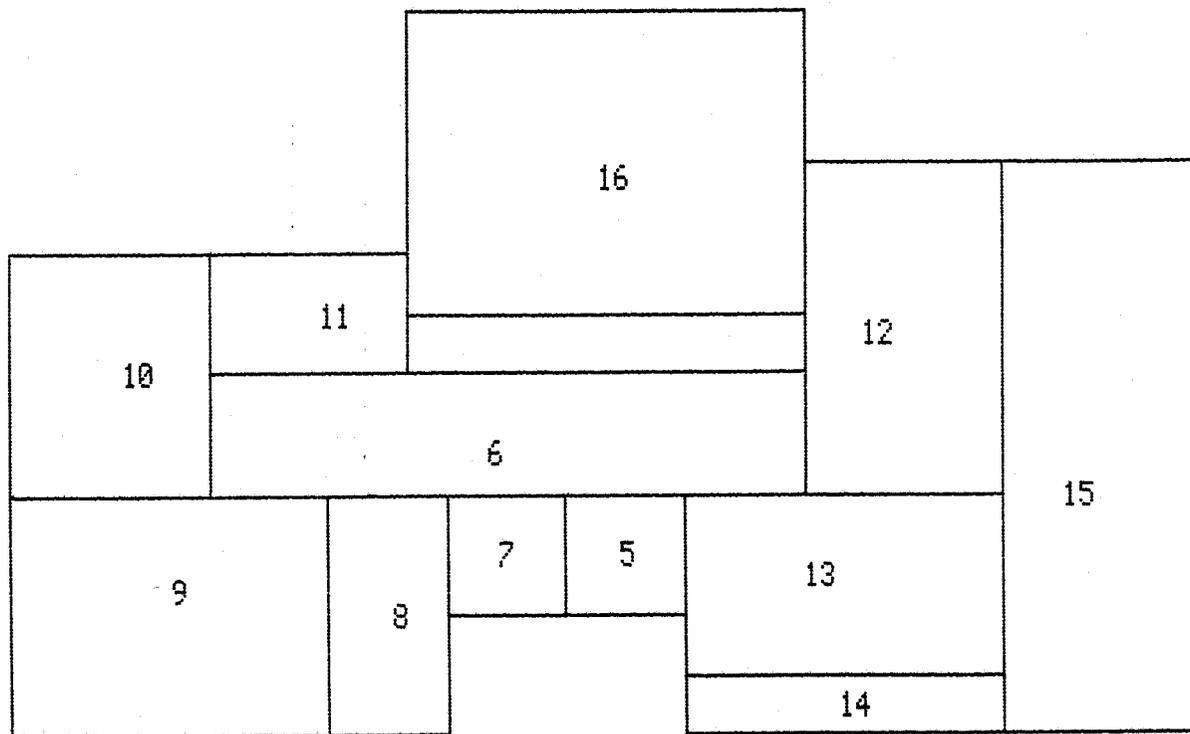
En este paso podemos, además, fijar la orientación de tantas particiones como se desee. En el ejemplo concreto que se trata, fijamos las orientaciones siguientes:

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

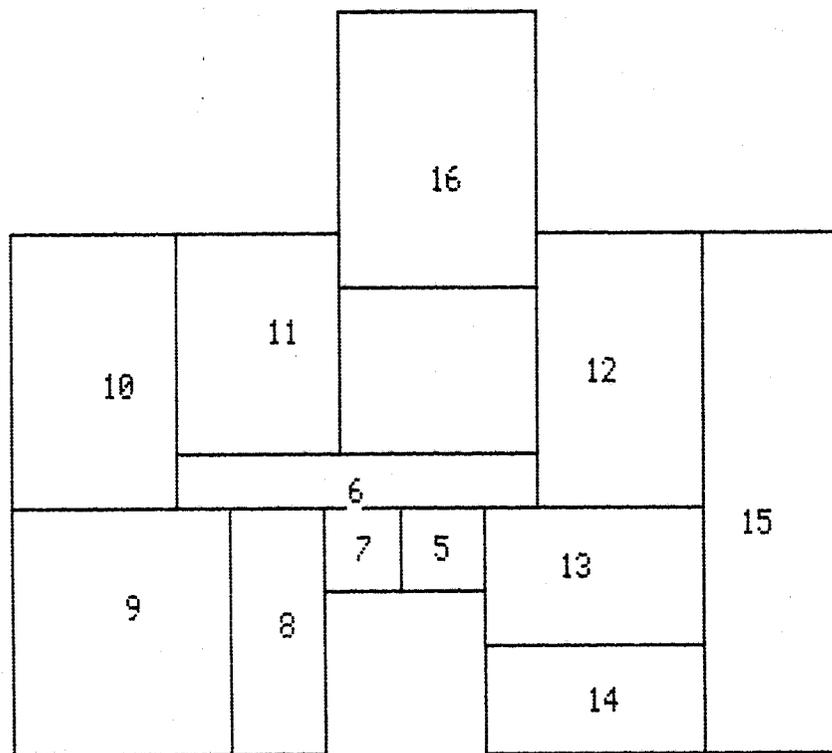
Se finaliza el ejemplo con la generación de todos los esquemas adimensionales que satisfacen los requisitos impuestos, y sus correspondientes esquemas optimizados dimensionalmente. La función objetivo elegida en este ejemplo ha sido:

$$\min: \sum_{i=1,n} X_i + \sum_{i=1,n} Y_i$$





SOLUCION 1



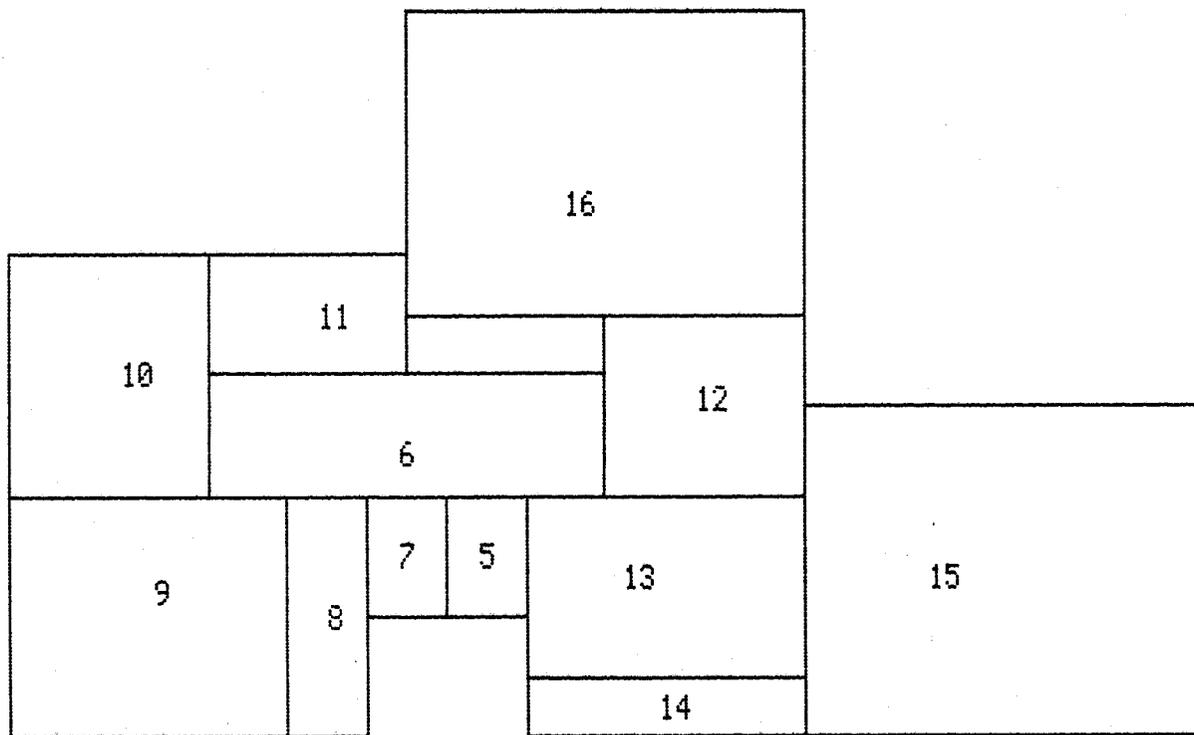
**SOLUCION 1**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 1

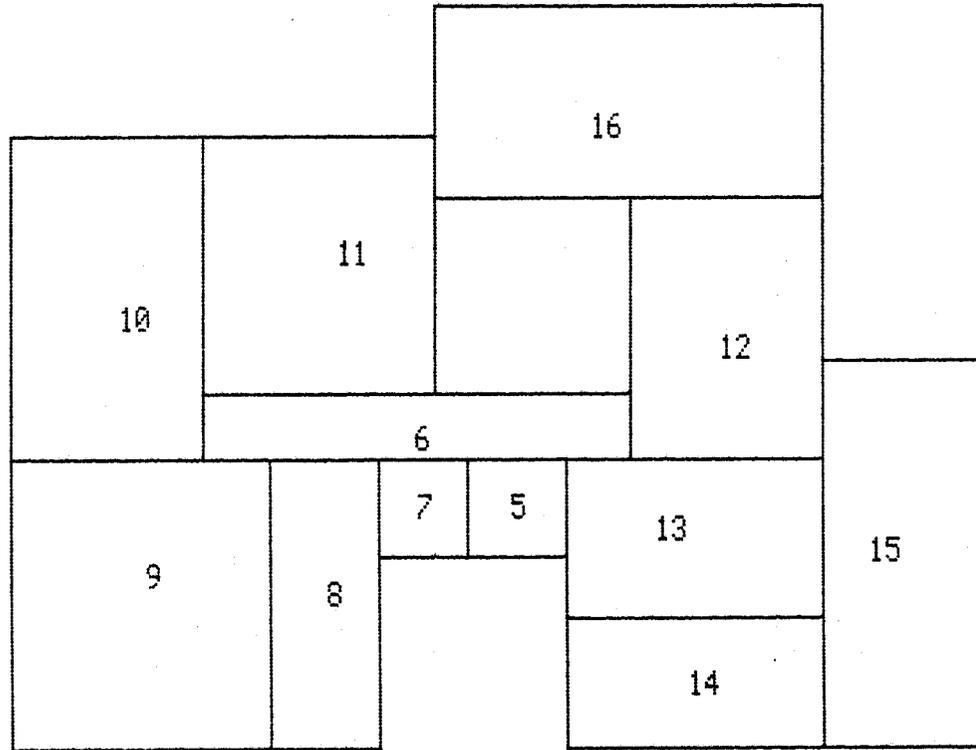
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	13.52	6.00	204.13	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	5.00	3.00	15.00	12.00
11 Dormitorio 3	3.00	4.00	3.00	12.00	0.00
12 Comedor	3.00	5.00	3.00	15.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	9.50	2.50	23.75	15.00
16 Estar	3.60	5.02	3.00	18.07	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	5.50	4.02	0.00	22.10	0.00
20 (vacío)	6.00	4.02	0.00	24.11	0.00

VALOR DE LA FUNCION OBJETIVO 144.9735

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 138.4162



SOLUCION 2



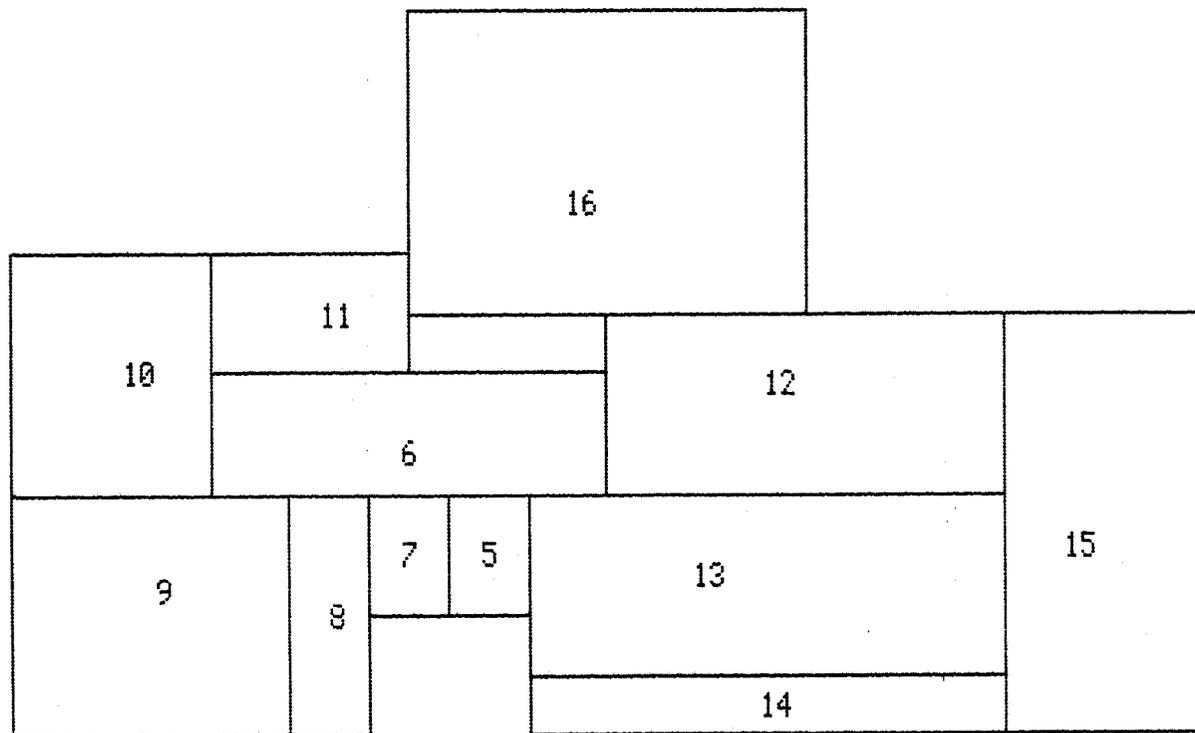
**SOLUCION 2**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 2

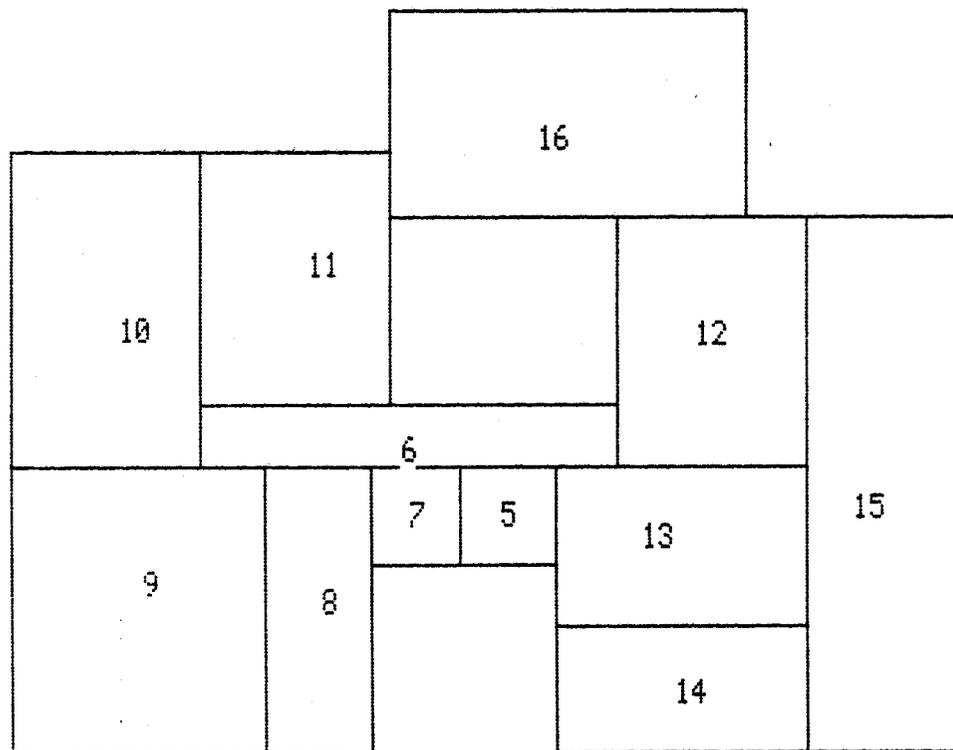
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.50	6.00	173.65	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	5.00	3.00	15.00	12.00
11 Dormitorio 3	3.60	4.00	3.00	14.40	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	6.00	3.00	3.00	18.00	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	5.50	0.00	13.75	0.00
20 (vacío)	6.60	2.00	0.00	13.20	0.00

VALOR DE LA FUNCION OBJETIVO 135.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 129



SOLUCION 3



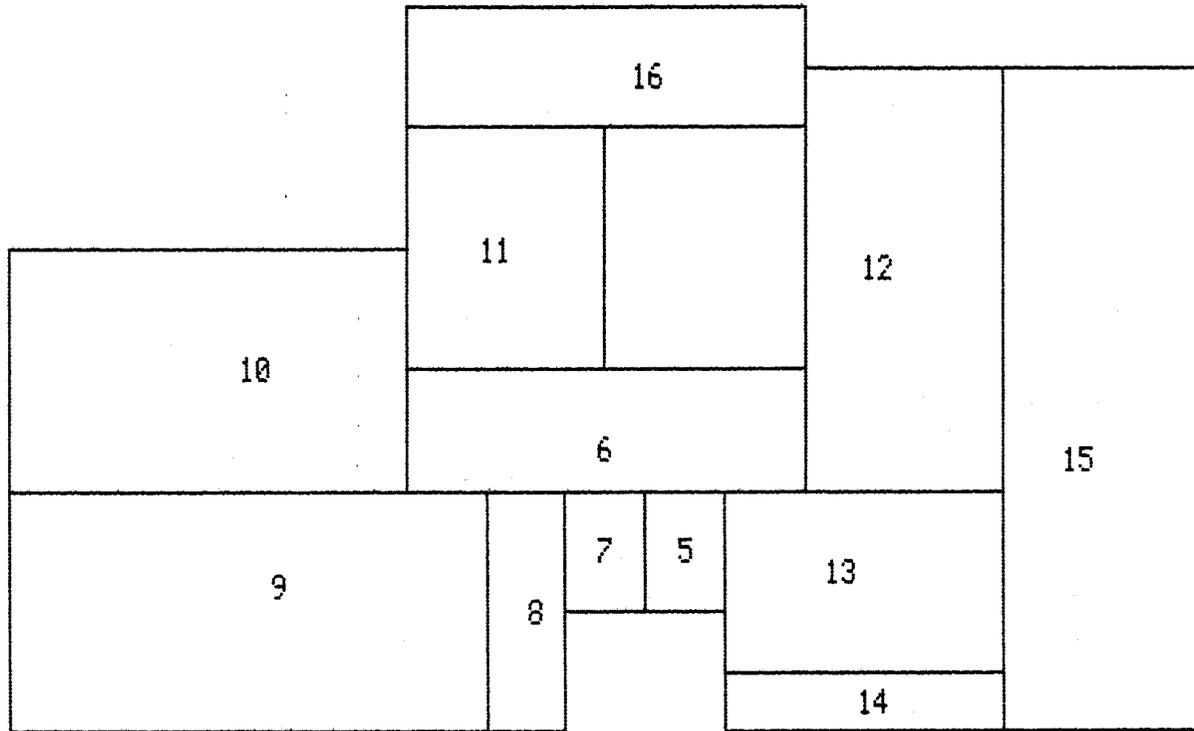
**SOLUCION 3**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 3

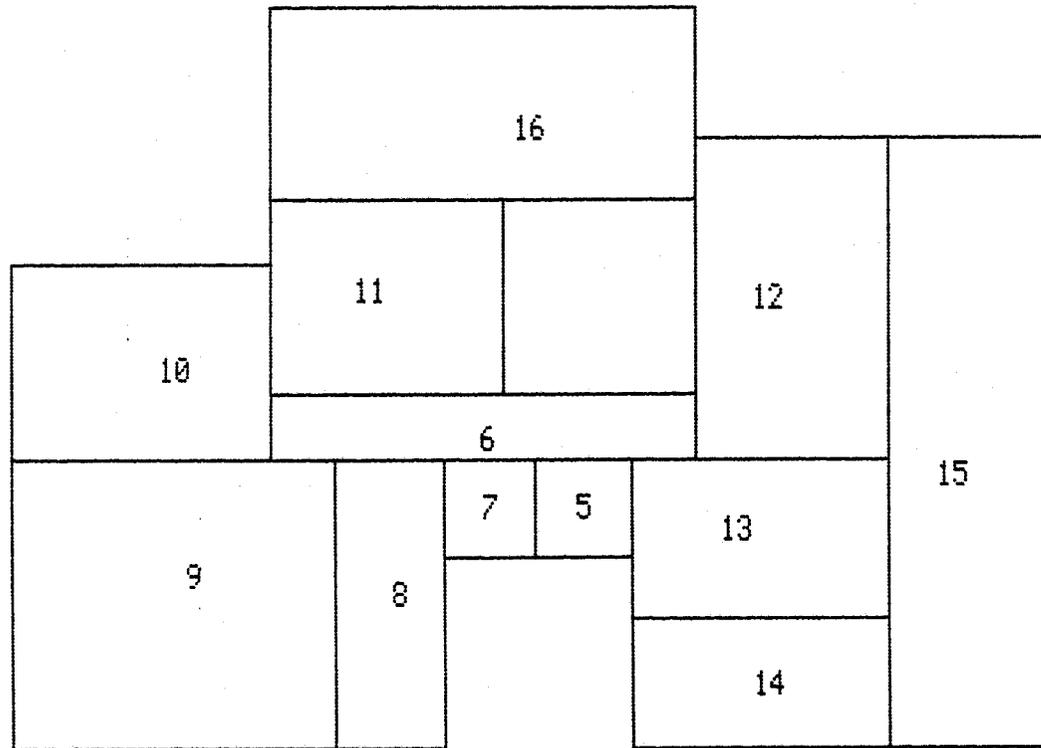
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.73	6.00	177.19	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	5.00	3.00	15.00	12.00
11 Dormitorio 3	3.00	4.00	3.00	12.00	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	8.50	2.50	21.25	15.00
16 Estar	5.60	3.23	3.00	18.11	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	3.50	3.23	0.00	11.32	0.00
20 (vacío)	6.00	2.23	0.00	13.41	0.00

VALOR DE LA FUNCION OBJETIVO 136.8373

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 132.9622



SOLUCION 4



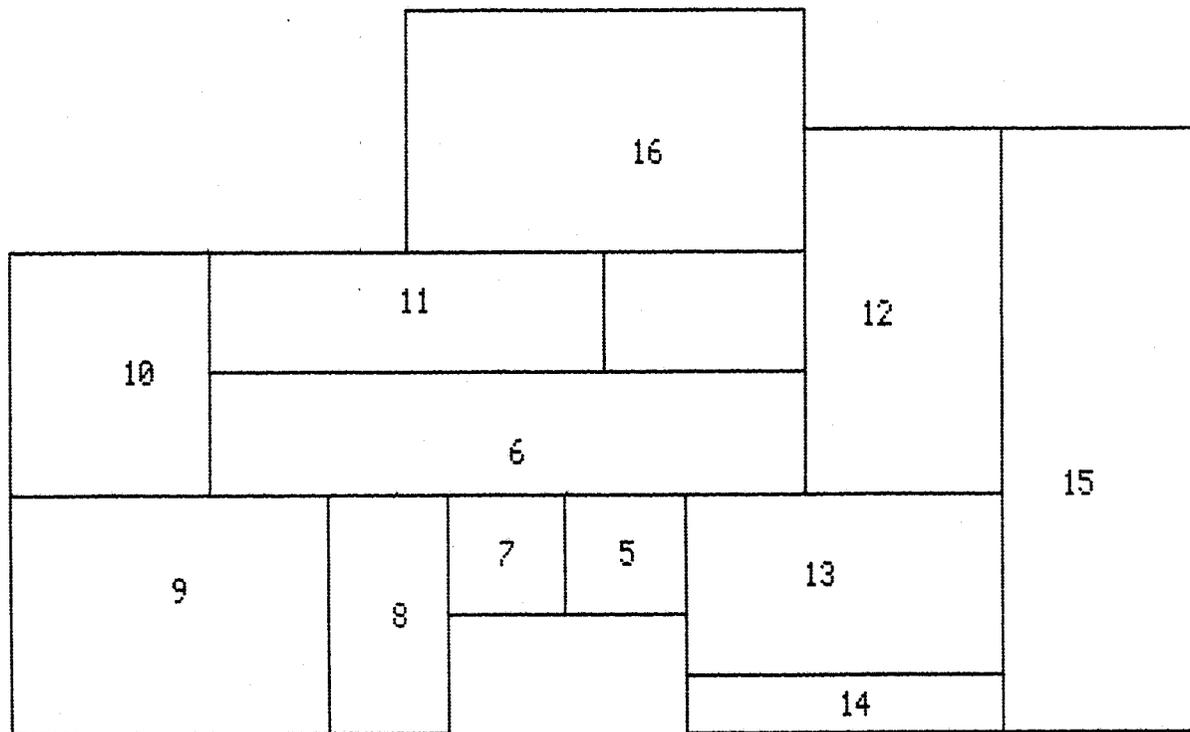
**SOLUCION 4**  
(dimensionada)

**DIMENSIONES DE LA SOLUCION 4**

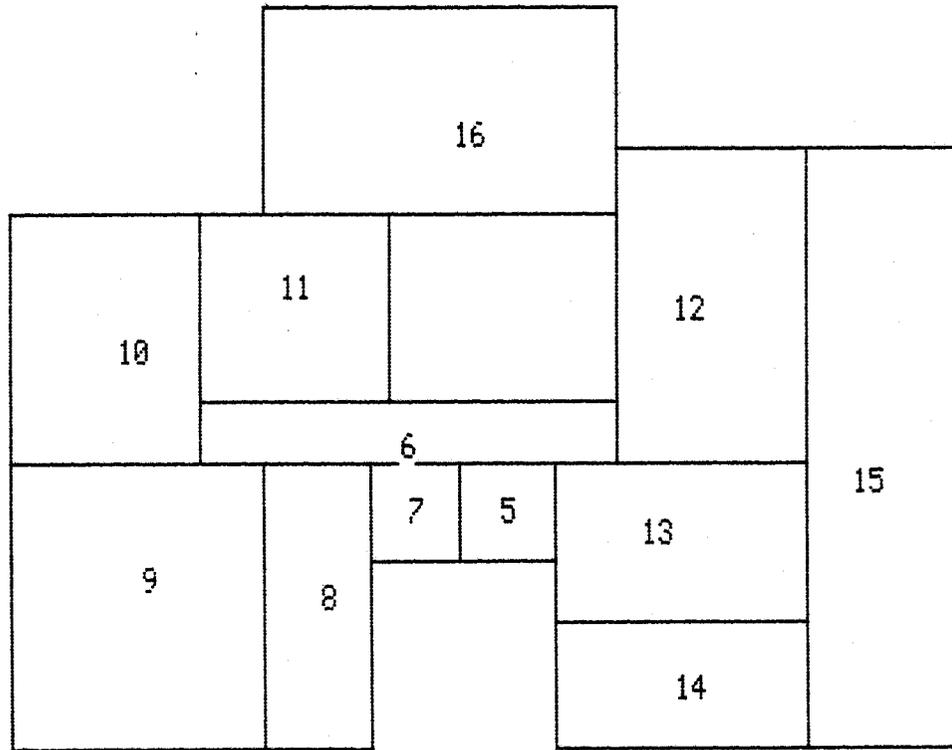
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	16.10	11.50	6.00	185.15	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	5.00	4.50	3.00	22.50	12.00
10 Dormitorio 2	4.00	3.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	5.00	3.00	15.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	9.50	2.50	23.75	15.00
16 Estar	6.60	3.00	3.00	19.80	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	5.50	2.00	0.00	11.00	0.00
20 (vacío)	4.00	4.00	0.00	16.00	0.00

VALOR DE LA FUNCION OBJETIVO 139.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 140.45



**SOLUCION 5**



### SOLUCION 5

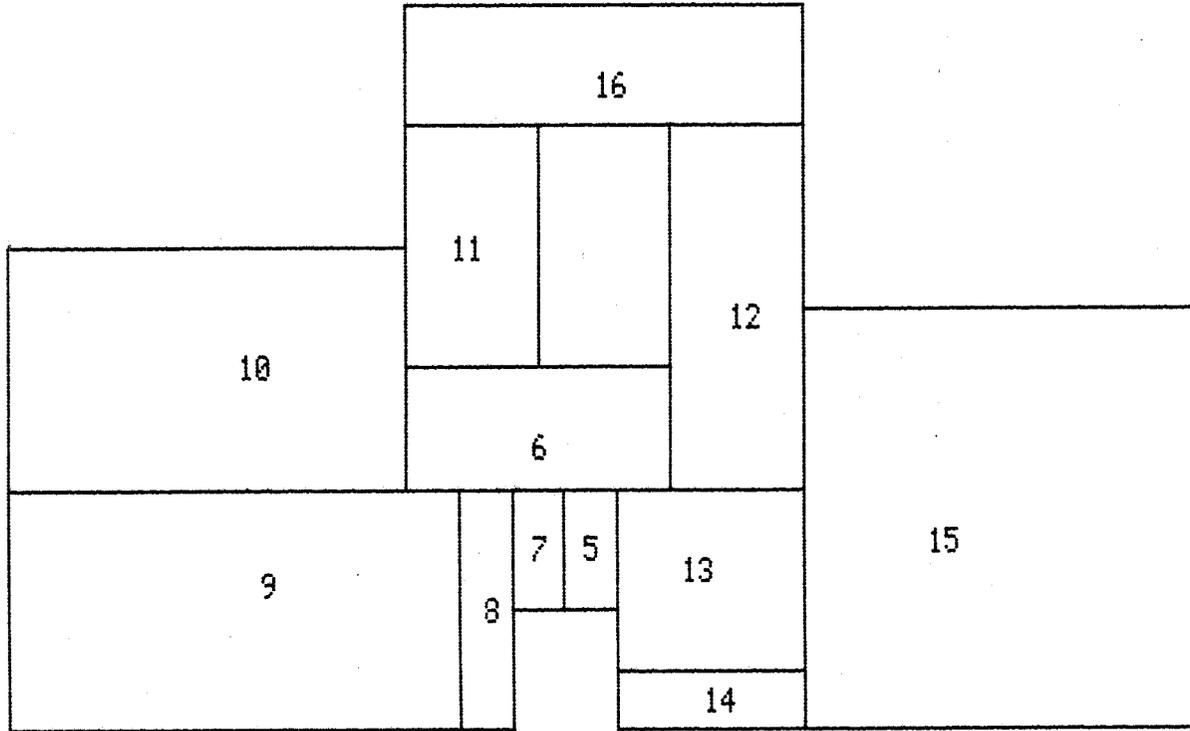
(dimensionada)

DIMENSIONES DE LA SOLUCION 5

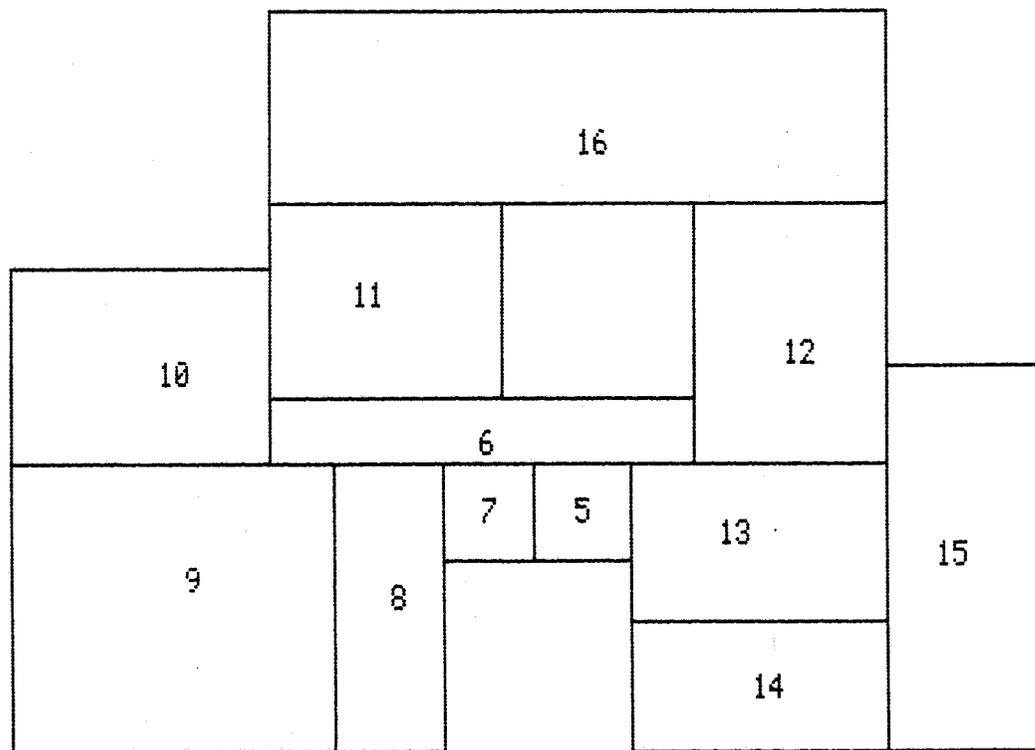
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.73	6.00	177.19	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	4.00	3.00	12.00	12.00
11 Dormitorio 3	3.00	3.00	3.00	9.00	0.00
12 Comedor	3.00	5.00	3.00	15.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	9.50	2.50	23.75	15.00
16 Estar	5.60	3.23	3.00	18.11	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	5.50	2.23	0.00	12.29	0.00
20 (vacío)	4.00	3.23	0.00	12.94	0.00

VALOR DE LA FUNCION OBJETIVO 136.8373

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 132.4622



SOLUCION 6



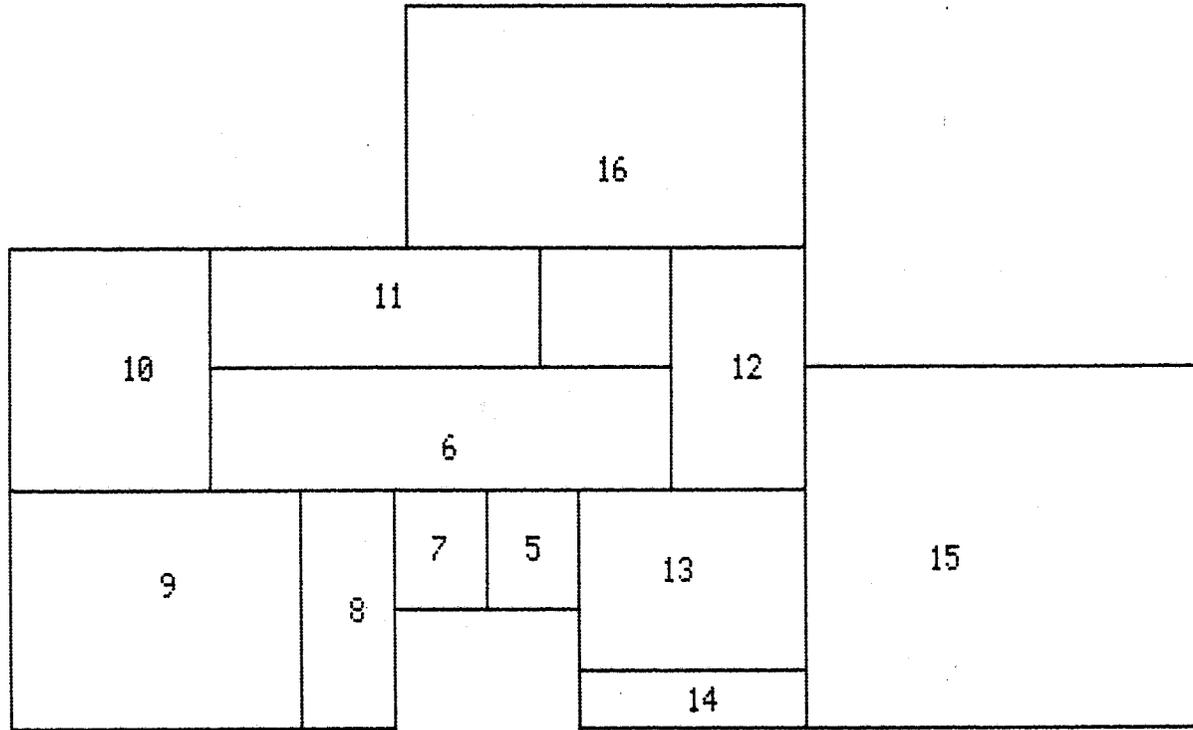
**SOLUCION 6**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 6

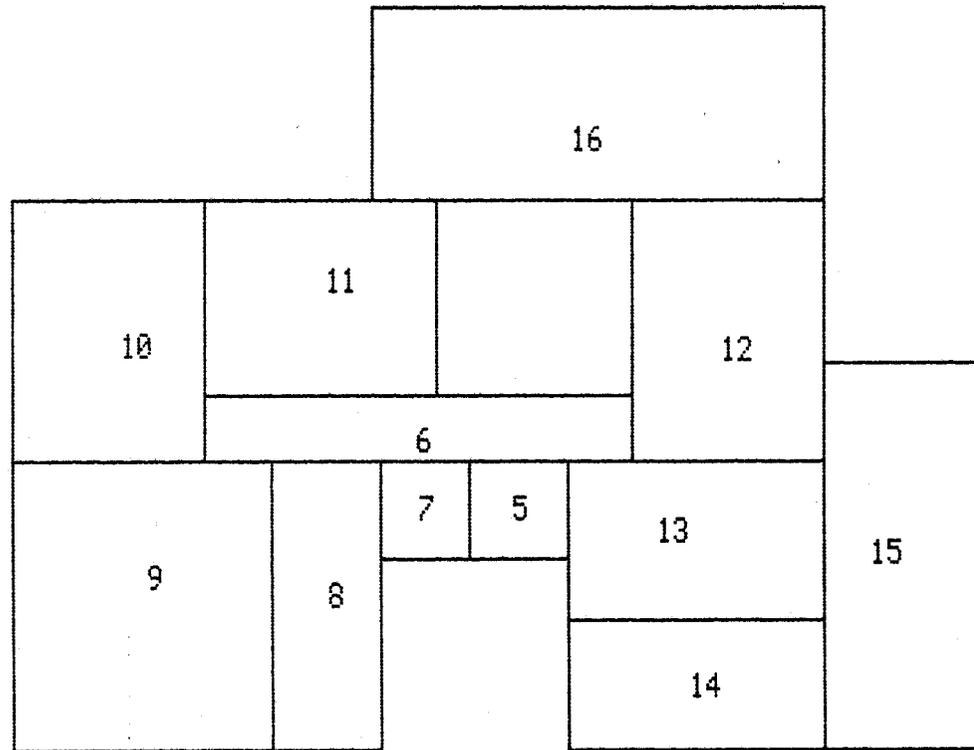
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	16.10	11.50	6.00	185.15	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	5.00	4.50	3.00	22.50	12.00
10 Dormitorio 2	4.00	3.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	9.60	3.00	3.00	28.80	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	5.50	0.00	13.75	0.00
20 (vacío)	4.00	4.00	0.00	16.00	0.00

VALOR DE LA FUNCION OBJETIVO 138.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 137.7



SOLUCION 7



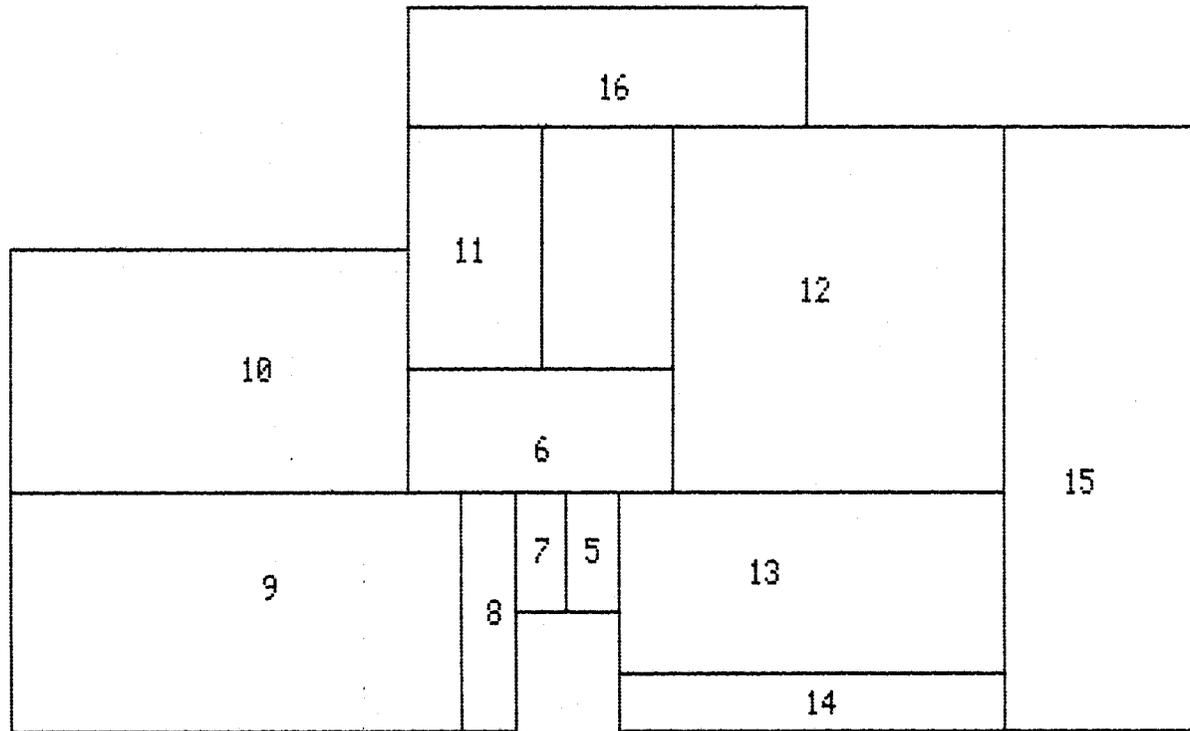
**SOLUCION 7**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 7

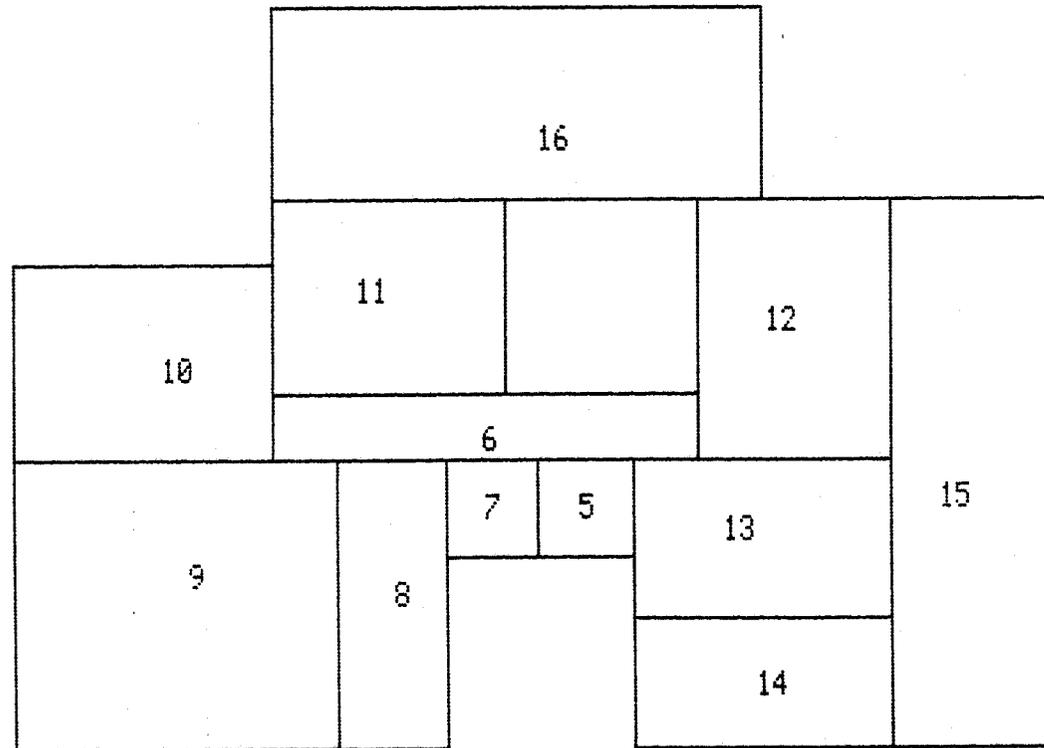
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.50	6.00	173.65	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	4.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	7.00	3.00	3.00	21.00	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	5.50	0.00	13.75	0.00
20 (vacío)	5.60	3.00	0.00	16.80	0.00

VALOR DE LA FUNCION OBJETIVO 134.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 125.4



SOLUCION 8



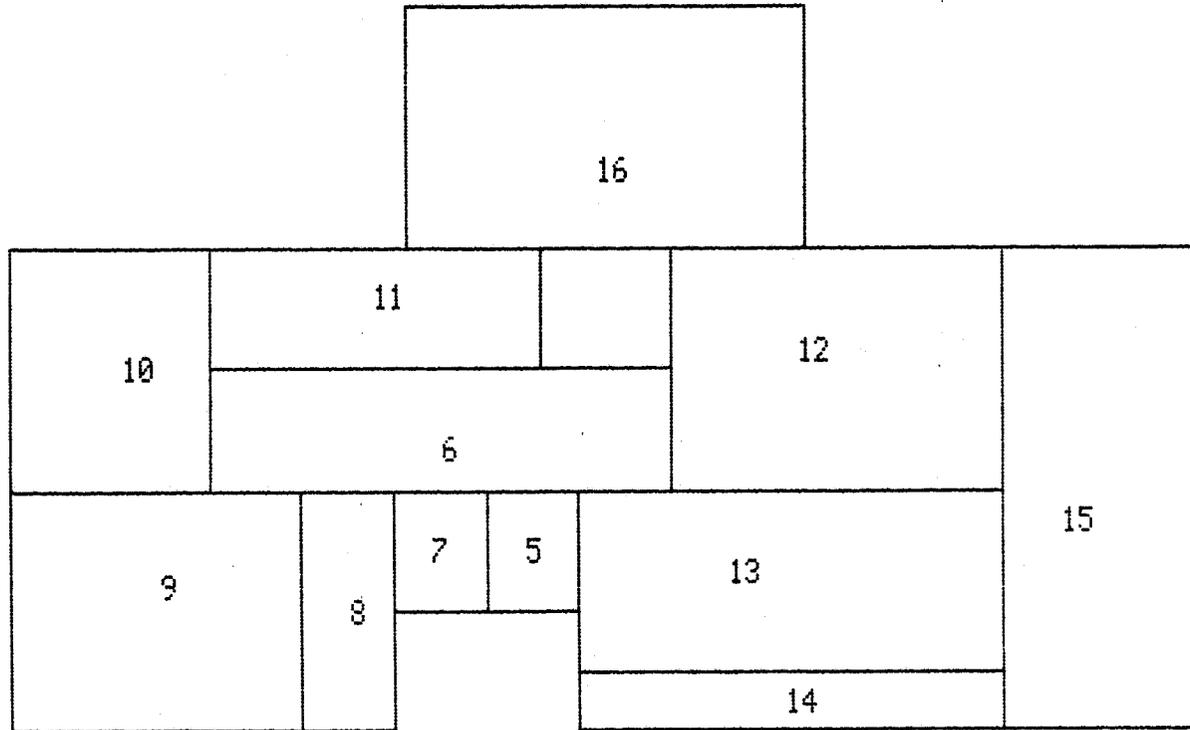
**SOLUCION 8**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 8

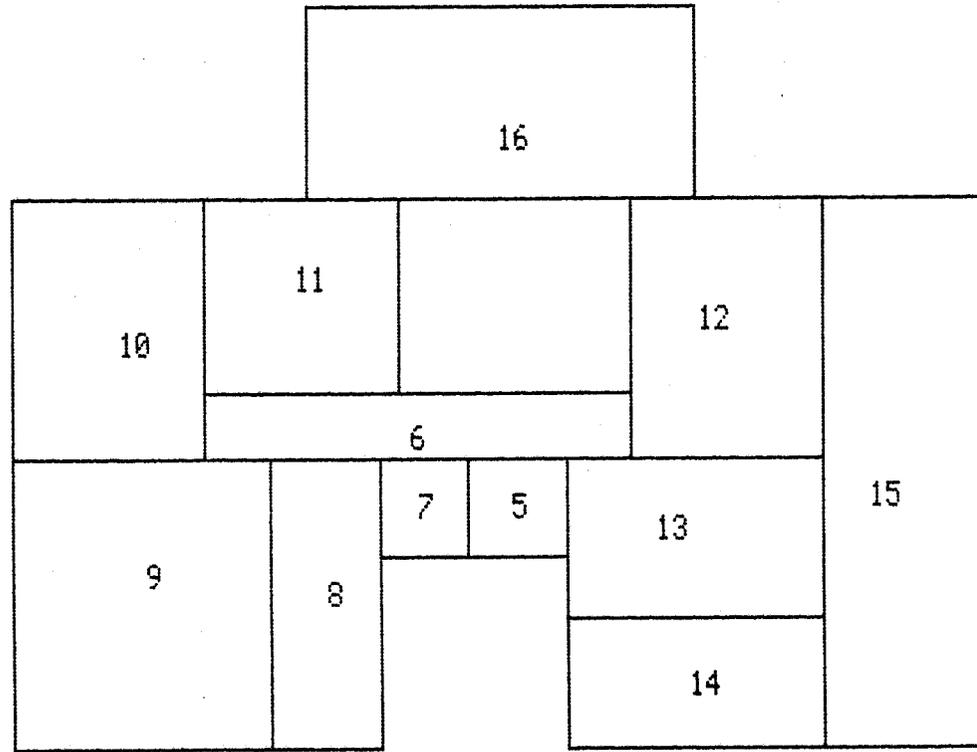
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	16.10	11.50	6.00	185.15	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	5.00	4.50	3.00	22.50	12.00
10 Dormitorio 2	4.00	3.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	8.50	2.50	21.25	15.00
16 Estar	7.60	3.00	3.00	22.80	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	4.50	3.00	0.00	13.50	0.00
20 (vacío)	4.00	4.00	0.00	16.00	0.00

VALOR DE LA FUNCION OBJETIVO 138.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 137.95



SOLUCION 9



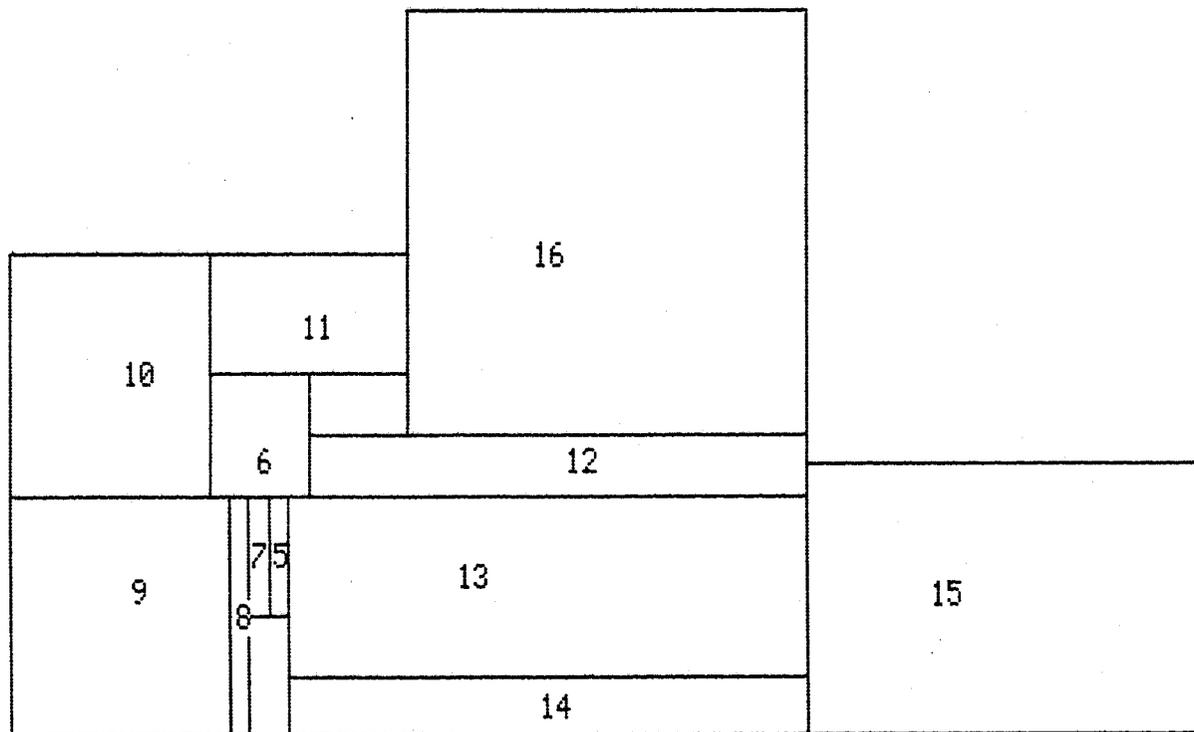
**SOLUCION 9**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 9

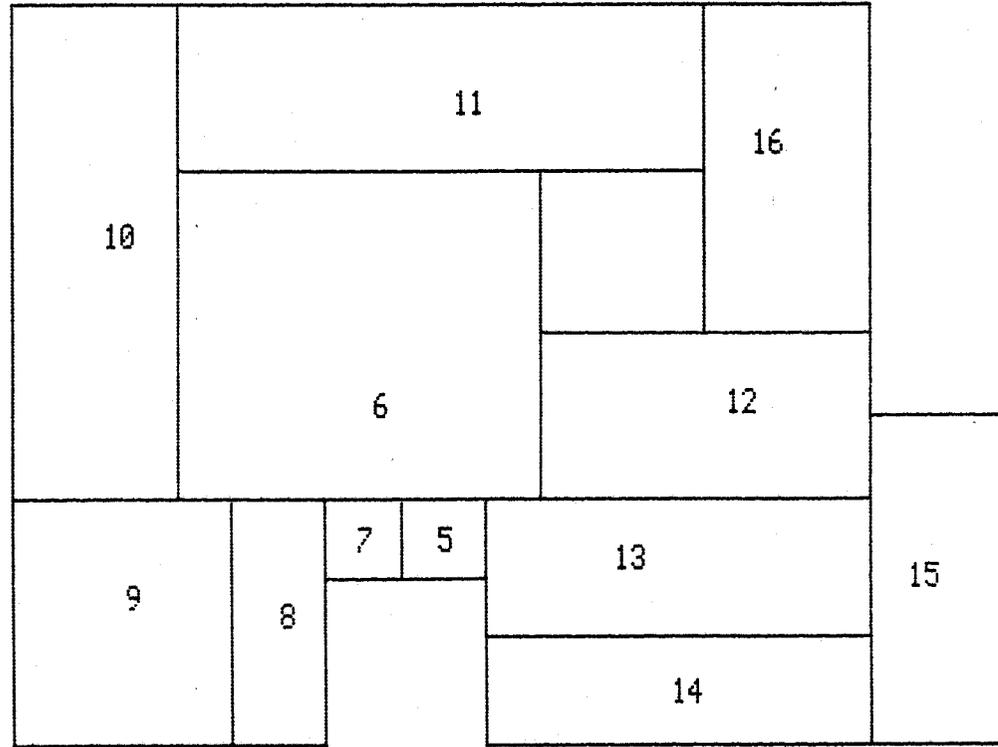
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.50	6.00	173.65	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	4.00	3.00	12.00	12.00
11 Dormitorio 3	3.00	3.00	3.00	9.00	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	4.00	2.00	2.00	8.00	0.00
15 Garage	2.50	8.50	2.50	21.25	15.00
16 Estar	6.00	3.00	3.00	18.00	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	4.50	3.00	0.00	13.50	0.00
20 (vacío)	4.60	3.00	0.00	13.80	0.00

VALOR DE LA FUNCION OBJETIVO 134.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 126.85



SOLUCION 10



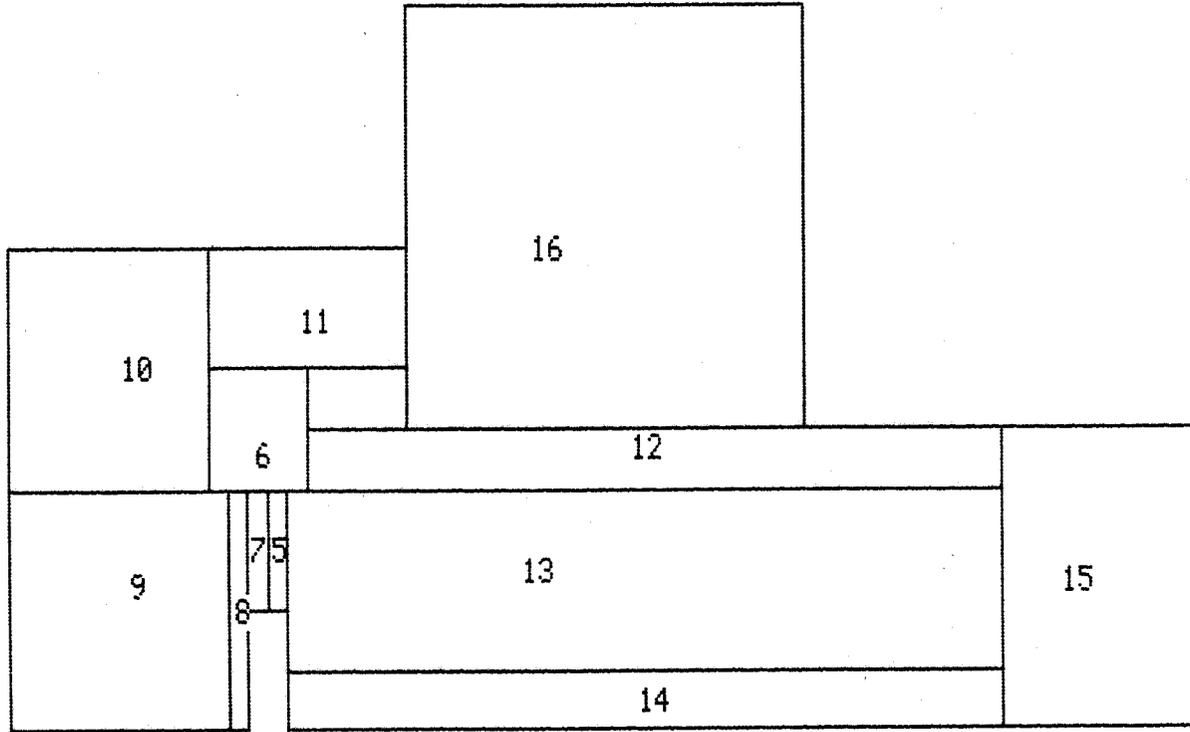
**SOLUCION 10**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 10

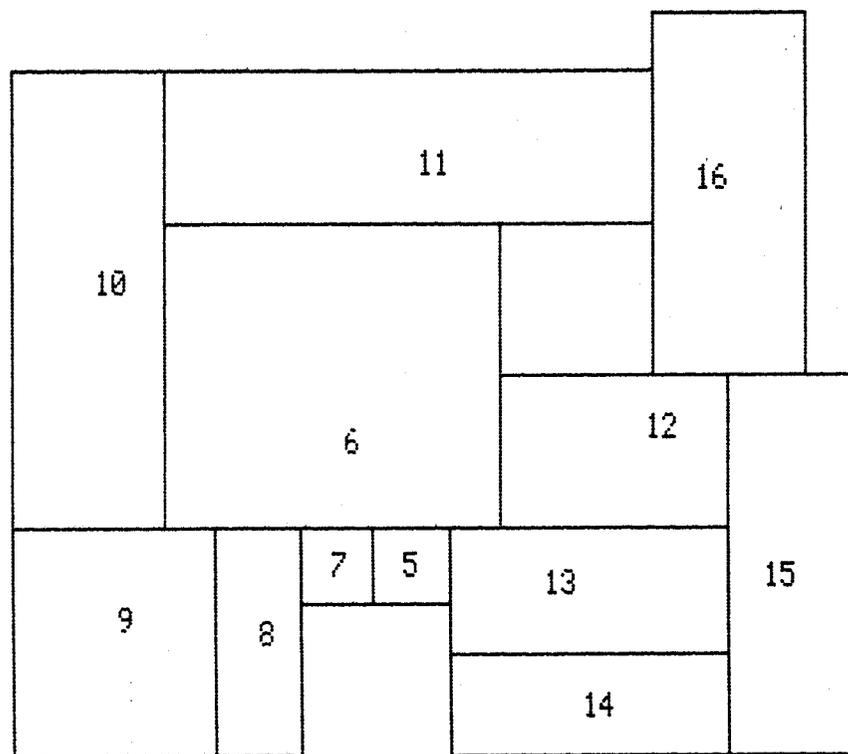
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	18.10	13.50	6.00	244.35	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	6.00	1.00	39.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	9.00	3.00	27.00	12.00
11 Dormitorio 3	9.60	3.00	3.00	28.80	0.00
12 Comedor	6.00	3.00	3.00	18.00	0.00
13 Cocina	7.00	2.50	2.50	17.50	10.00
14 Lavadero	7.00	2.00	2.00	14.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	3.00	6.00	3.00	18.00	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	7.50	0.00	18.75	0.00
20 (vacío)	12.60	0.00	0.00	0.00	0.00

VALOR DE LA FUNCION OBJETIVO 168.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 207.9



SOLUCION 11



# SOLUCION 11

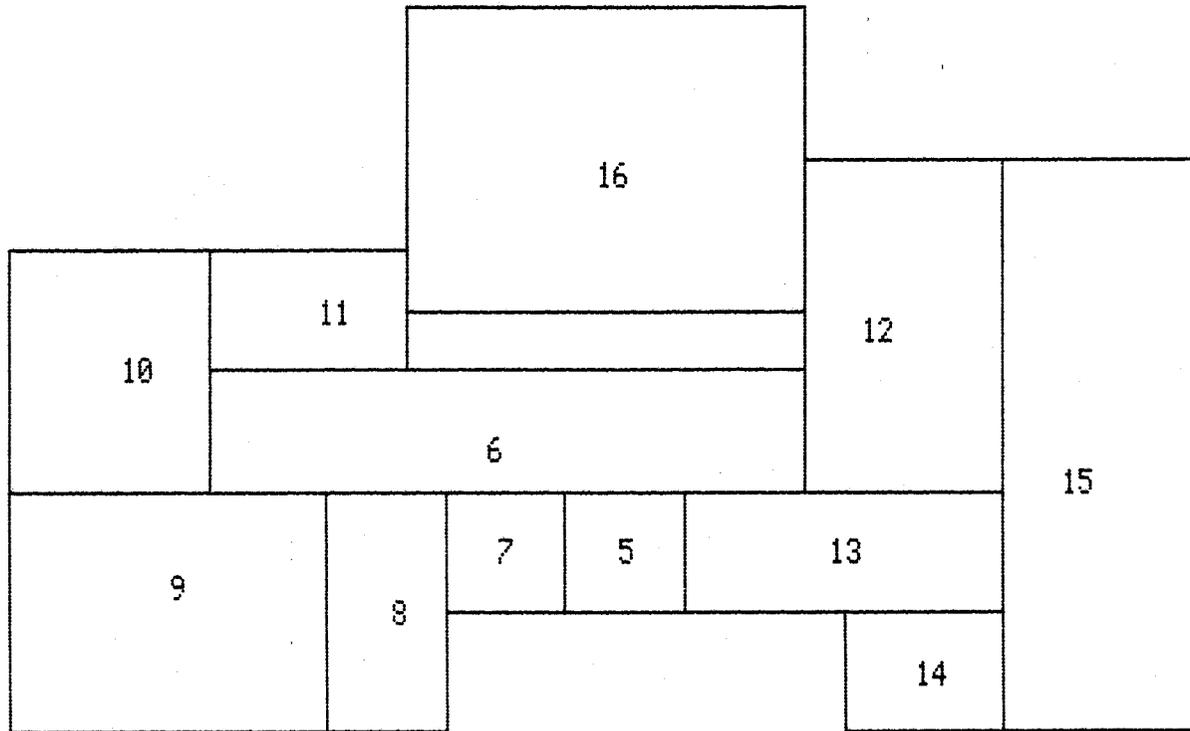
(dimensionada)

DIMENSIONES DE LA SOLUCION 11

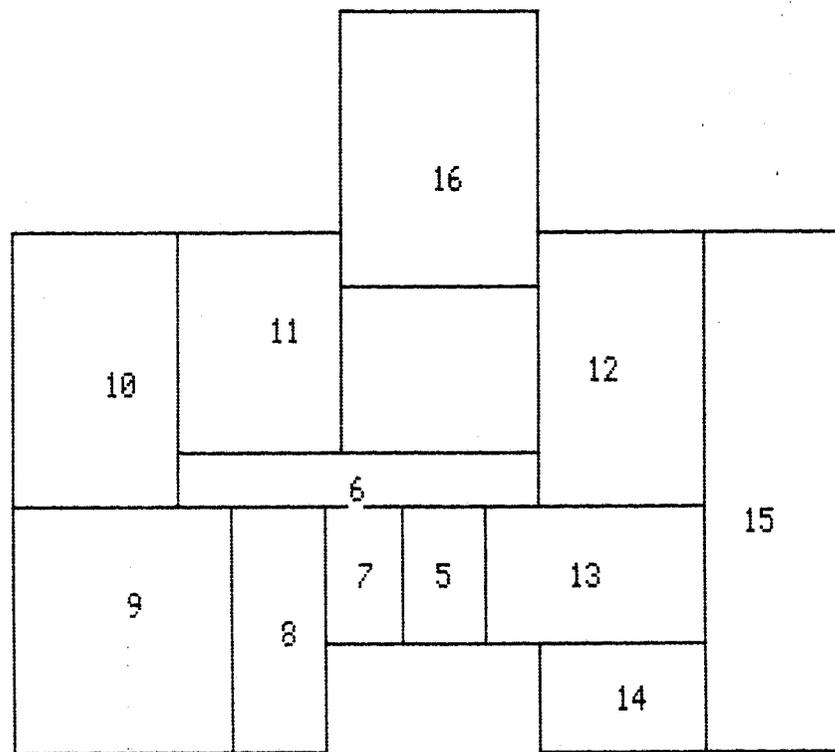
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	16.61	14.61	6.00	242.61	0.00
5 Vestibulo	1.50	1.50	1.50	2.25	0.00
6 Distribuidor	6.60	6.00	1.00	39.60	0.00
7 Aseo	1.40	1.50	1.40	2.10	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	9.00	3.00	27.00	12.00
11 Dormitorio 3	9.60	3.00	3.00	28.80	0.00
12 Comedor	7.00	3.00	3.00	21.00	0.00
13 Cocina	5.51	2.50	2.50	13.77	10.00
14 Lavadero	5.51	2.00	2.00	11.02	0.00
15 Garage	2.50	7.50	2.50	18.75	15.00
16 Estar	3.00	7.11	3.00	21.32	18.00
17 (vacío)	2.90	3.00	0.00	8.70	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	3.50	7.11	0.00	24.88	0.00
20 (vacío)	12.60	1.11	0.00	13.95	0.00

VALOR DE LA FUNCION OBJETIVO 170.8547

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 211.2599



SOLUCION 12



## SOLUCION 12

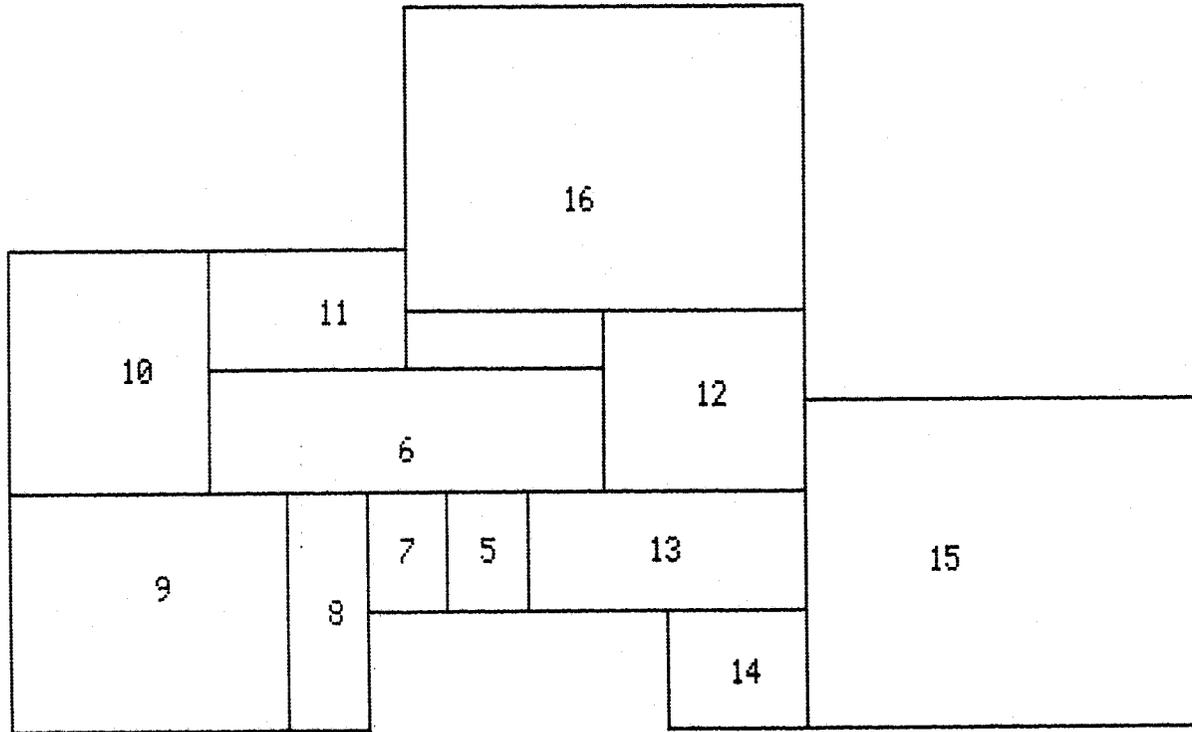
(dimensionada)

DIMENSIONES DE LA SOLUCION 12

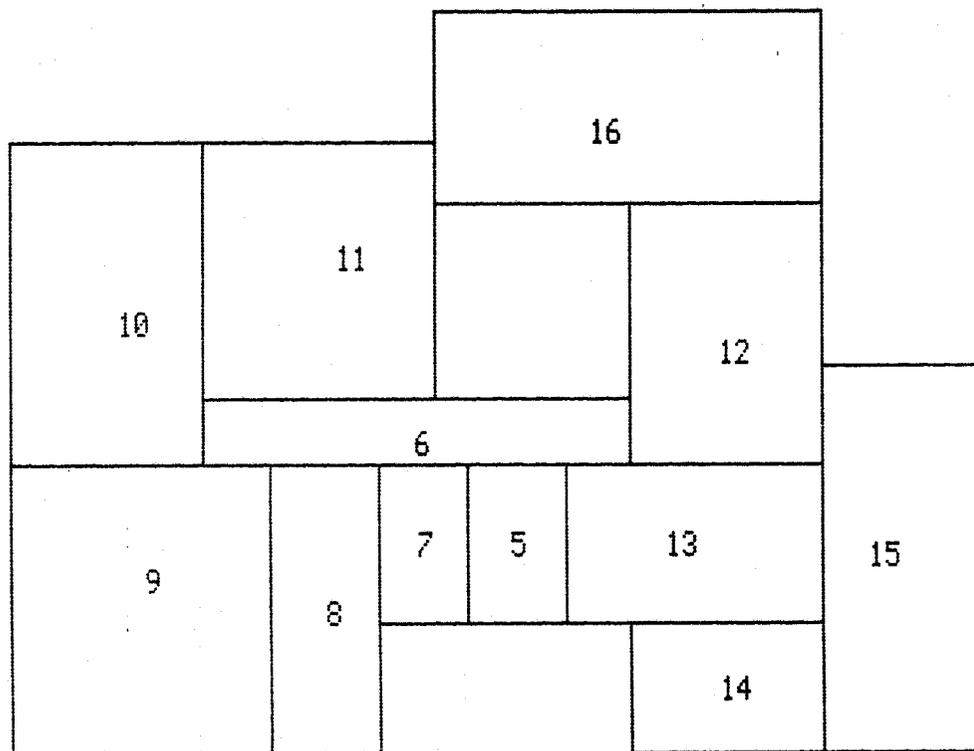
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	13.52	6.00	204.13	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	5.00	3.00	15.00	12.00
11 Dormitorio 3	3.00	4.00	3.00	12.00	0.00
12 Comedor	3.00	5.00	3.00	15.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	9.50	2.50	23.75	15.00
16 Estar	3.60	5.02	3.00	18.07	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	5.50	4.02	0.00	22.10	0.00
20 (vacío)	6.00	4.02	0.00	24.11	0.00

VALOR DE LA FUNCION OBJETIVO 145.9735

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 139.3162



SOLUCION 13



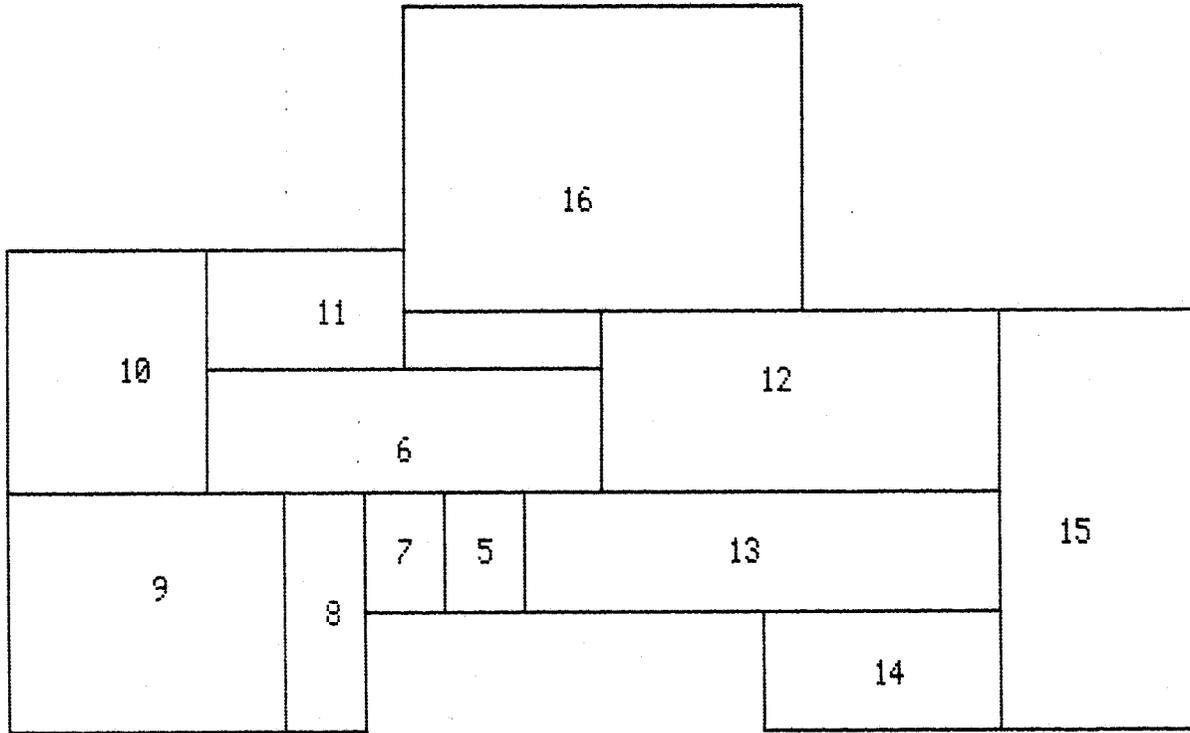
**SOLUCION 13**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 13

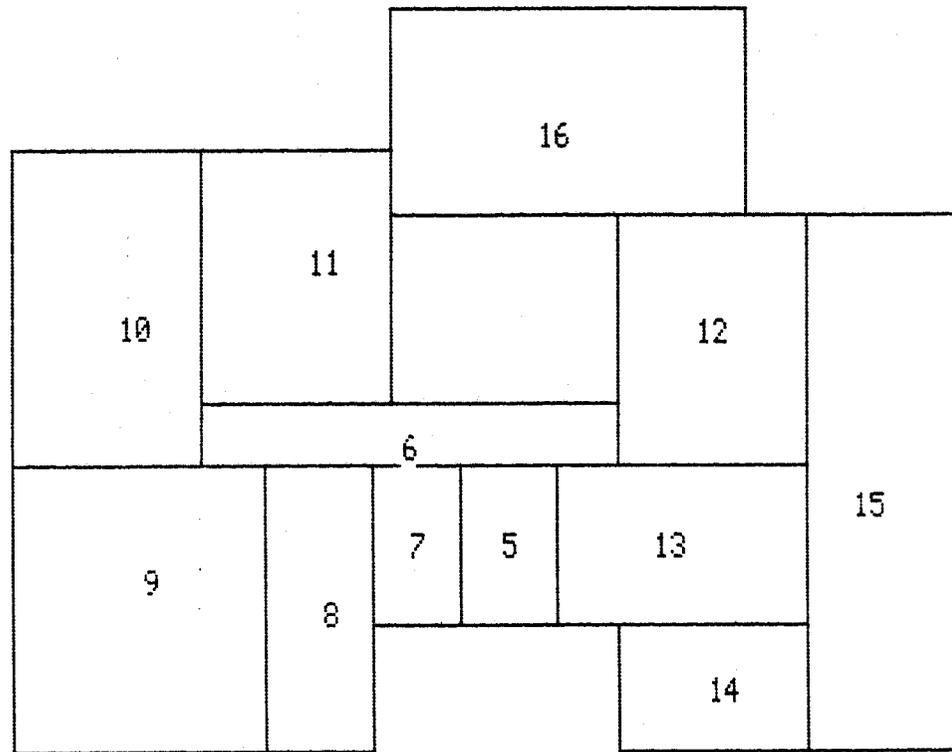
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.50	6.00	173.65	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	5.00	3.00	15.00	12.00
11 Dormitorio 3	3.60	4.00	3.00	14.40	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	6.00	3.00	3.00	18.00	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	5.50	0.00	13.75	0.00
20 (vacío)	6.60	2.00	0.00	13.20	0.00

VALOR DE LA FUNCION OBJETIVO 136.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 129.9



SOLUCION 14



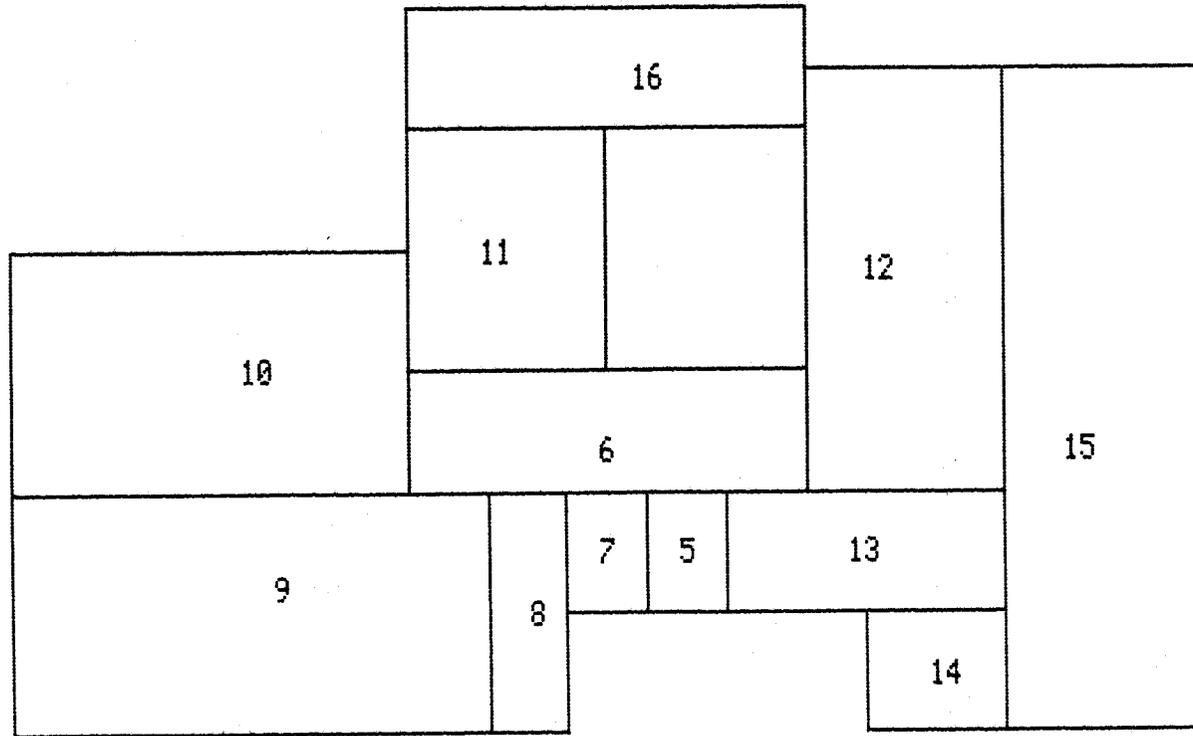
**SOLUCION 14**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 14

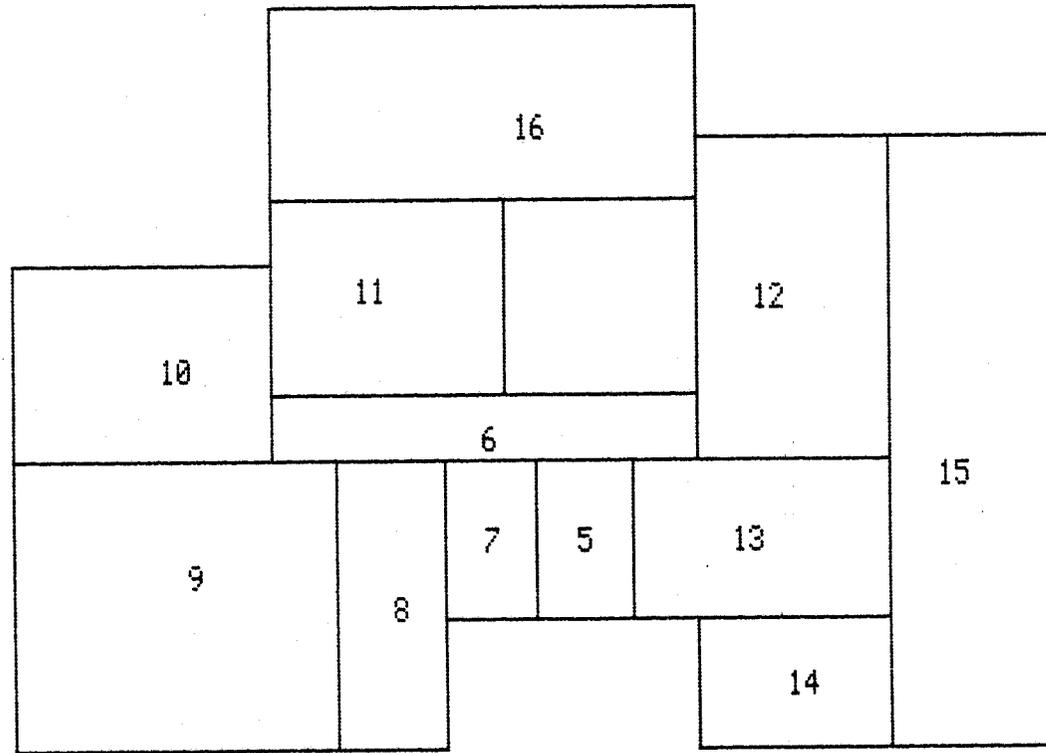
Local	Ancho	Largo	(min.)	Superf.	(min.)
4 Planta total	15.10	11.73	6.00	177.19	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	5.00	3.00	15.00	12.00
11 Dormitorio 3	3.00	4.00	3.00	12.00	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	8.50	2.50	21.25	15.00
16 Estar	5.60	3.23	3.00	18.11	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	3.50	3.23	0.00	11.32	0.00
20 (vacío)	6.00	2.23	0.00	13.41	0.00

VALOR DE LA FUNCION OBJETIVO 137.8373

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 133.8622



SOLUCION 15



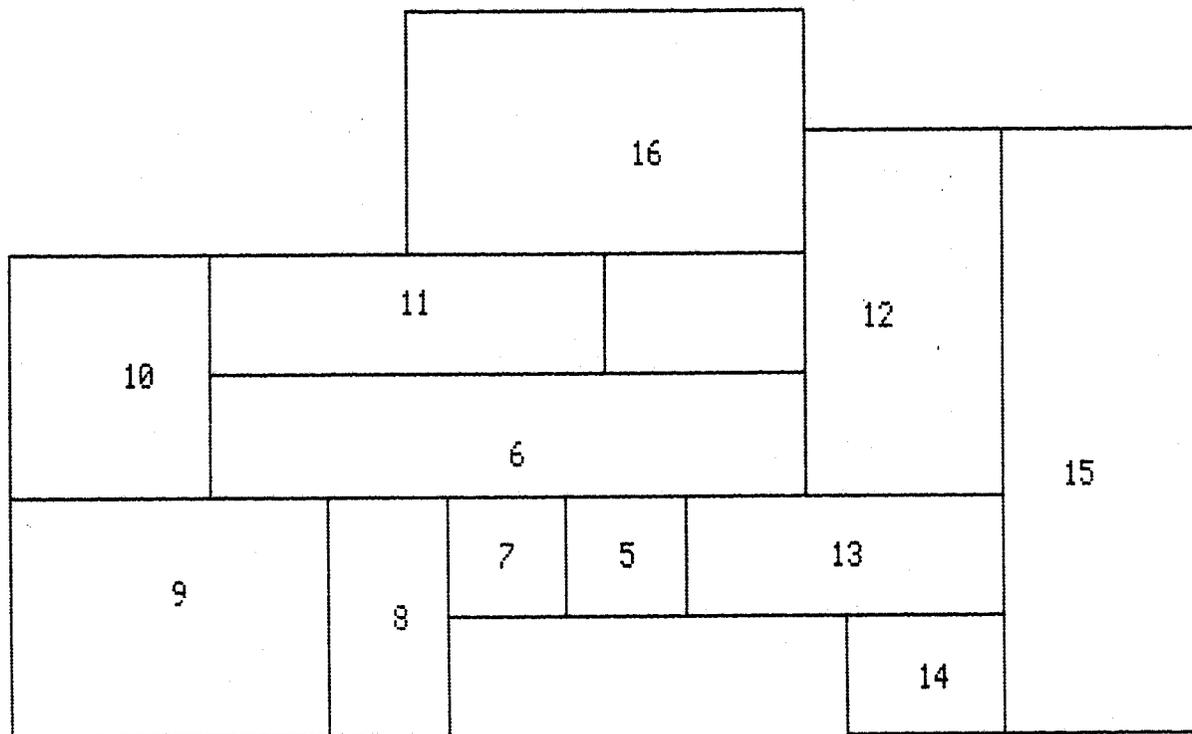
**SOLUCION 15**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 15

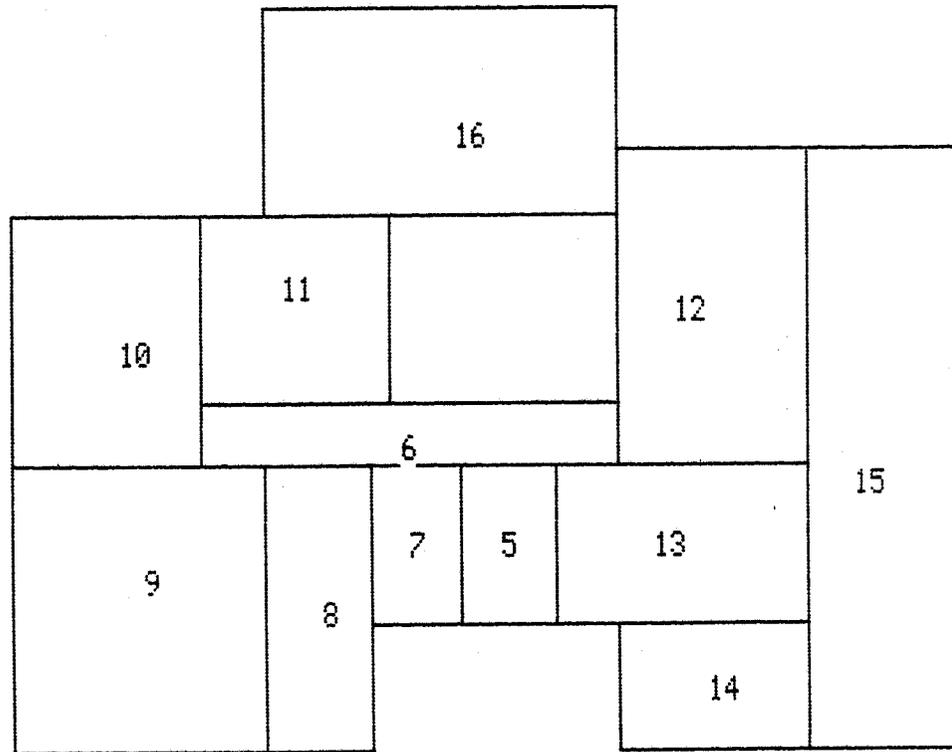
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	16.10	11.50	6.00	185.15	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	5.00	4.50	3.00	22.50	12.00
10 Dormitorio 2	4.00	3.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	5.00	3.00	15.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	9.50	2.50	23.75	15.00
16 Estar	6.60	3.00	3.00	19.80	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	5.50	2.00	0.00	11.00	0.00
20 (vacío)	4.00	4.00	0.00	16.00	0.00

VALOR DE LA FUNCION OBJETIVO 140.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 141.35



SOLUCION 16



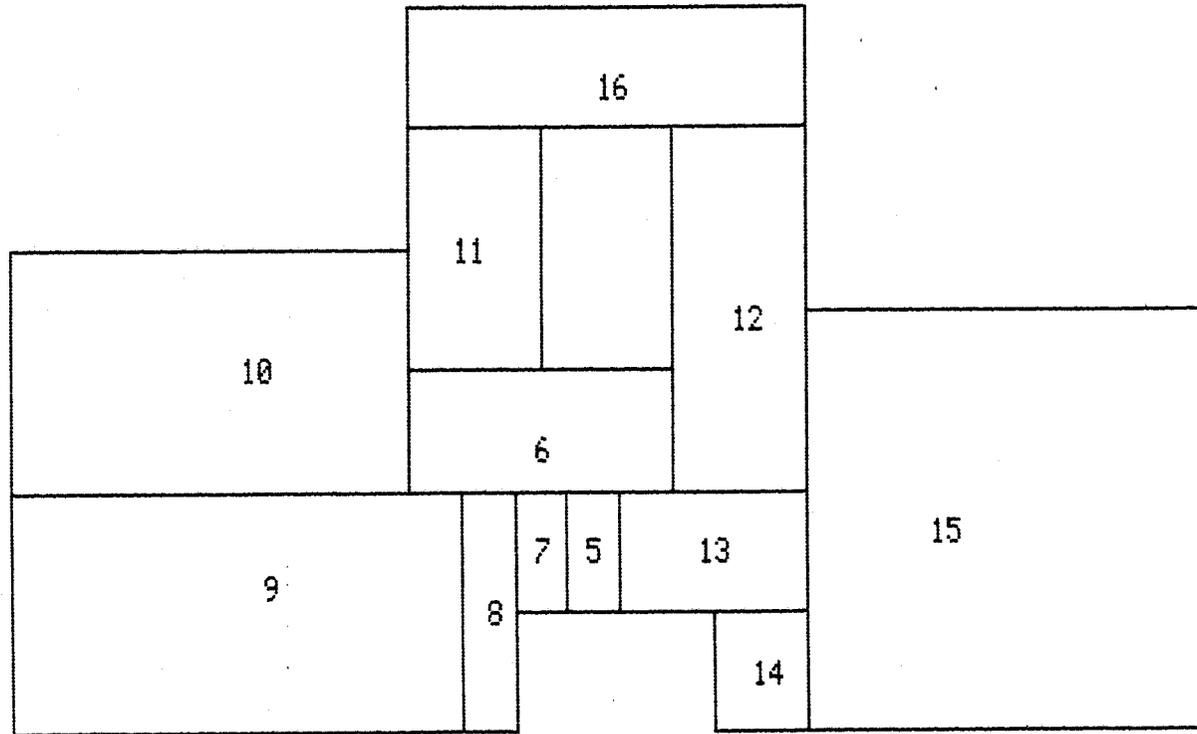
**SOLUCION 16**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 16

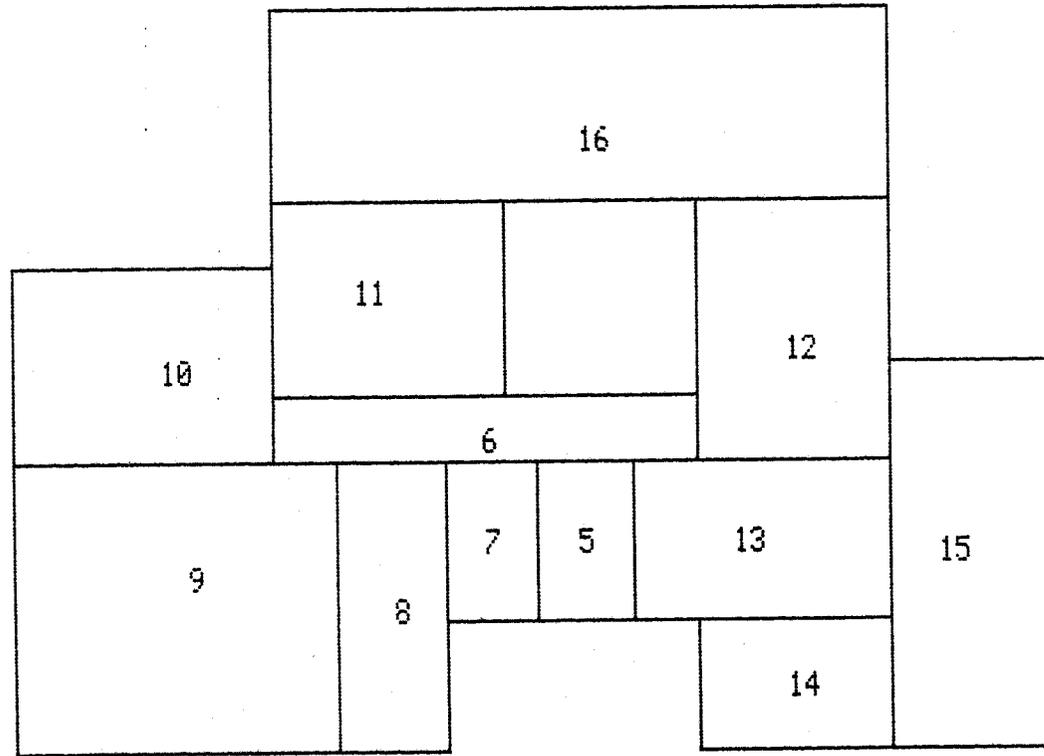
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.73	6.00	177.19	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	4.00	3.00	12.00	12.00
11 Dormitorio 3	3.00	3.00	3.00	9.00	0.00
12 Comedor	3.00	5.00	3.00	15.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	9.50	2.50	23.75	15.00
16 Estar	5.60	3.23	3.00	18.11	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	5.50	2.23	0.00	12.29	0.00
20 (vacío)	4.00	3.23	0.00	12.94	0.00

VALOR DE LA FUNCION OBJETIVO 137.8373

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 133.3622



SOLUCION 17



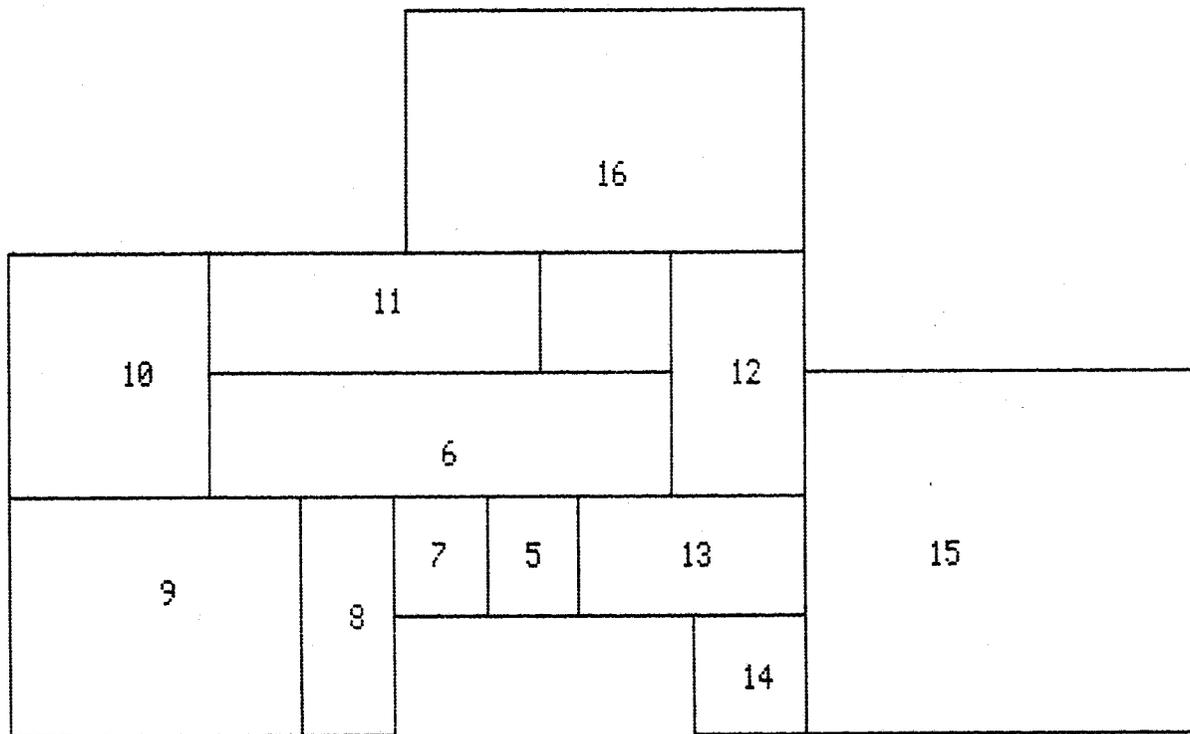
**SOLUCION 17**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 17

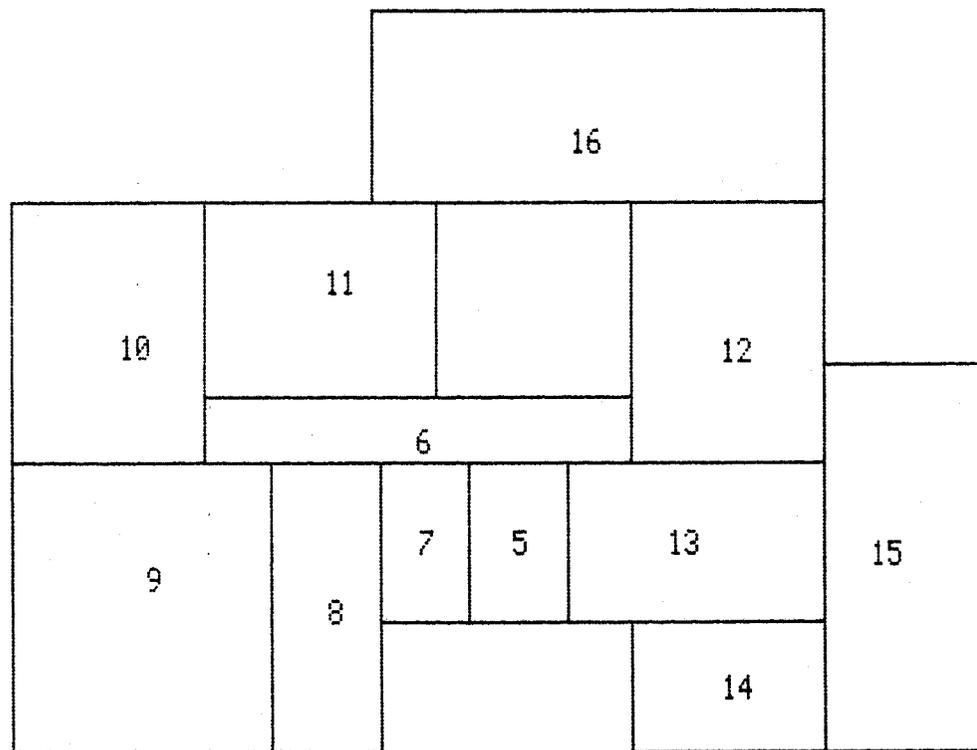
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	16.10	11.50	6.00	185.15	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	5.00	4.50	3.00	22.50	12.00
10 Dormitorio 2	4.00	3.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	9.60	3.00	3.00	28.80	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio (vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	5.50	0.00	13.75	0.00
20 (vacío)	4.00	4.00	0.00	16.00	0.00

VALOR DE LA FUNCION OBJETIVO 139.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 138.6



SOLUCION 18



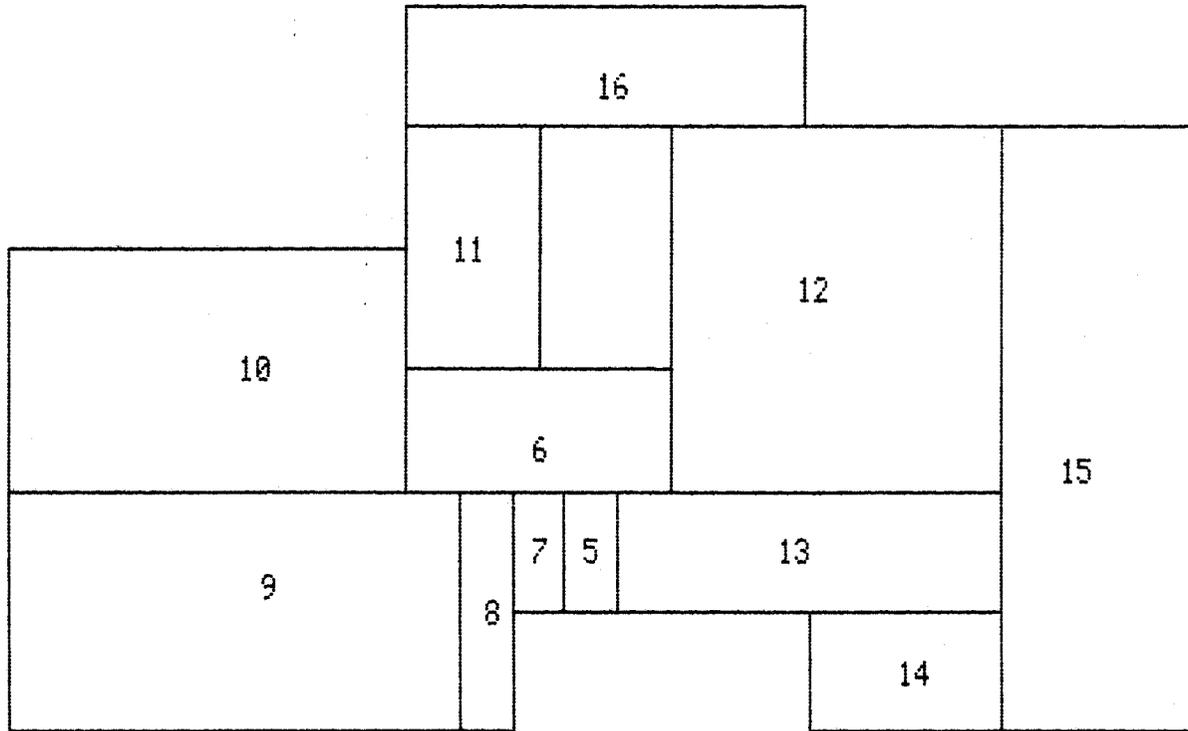
**SOLUCION 18**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 18

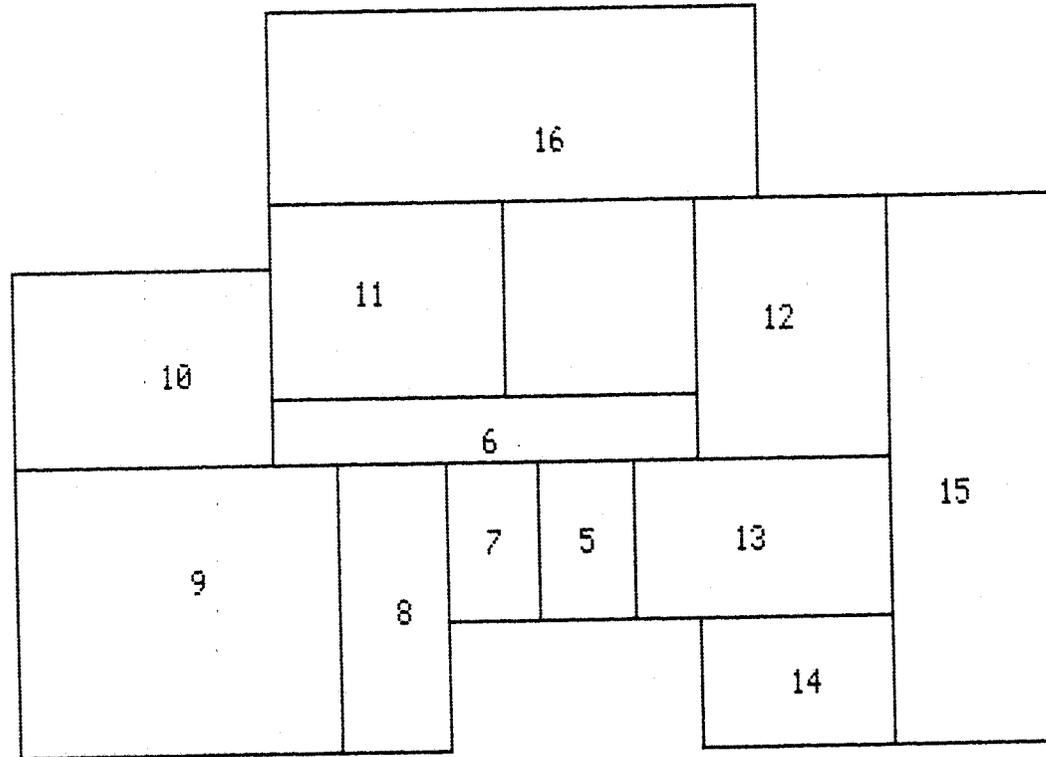
Local	Ancho	Largo	(min.)	Superf.	(min.)
4 Planta total	15.10	11.50	6.00	173.65	0.00
5 Vestíbulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	4.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	7.00	3.00	3.00	21.00	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	5.50	0.00	13.75	0.00
20 (vacío)	5.60	3.00	0.00	16.80	0.00

VALOR DE LA FUNCION OBJETIVO 135.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 126.3



SOLUCION 19



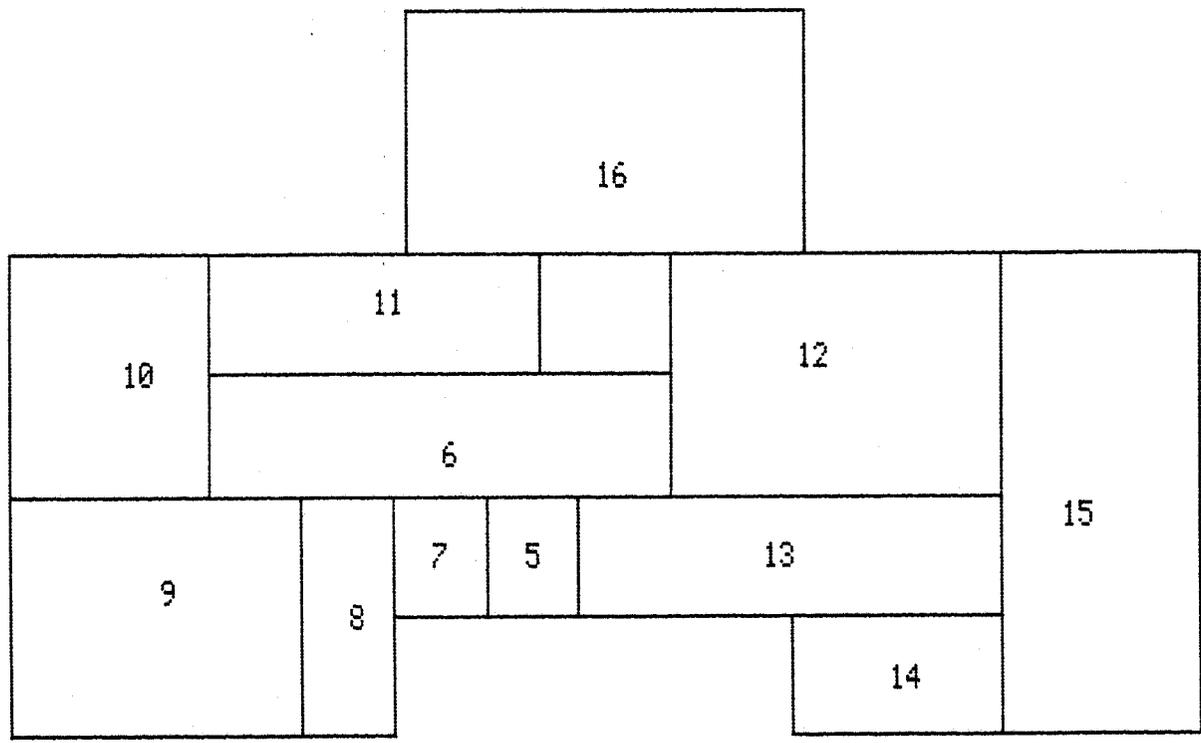
**SOLUCION 19**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 19

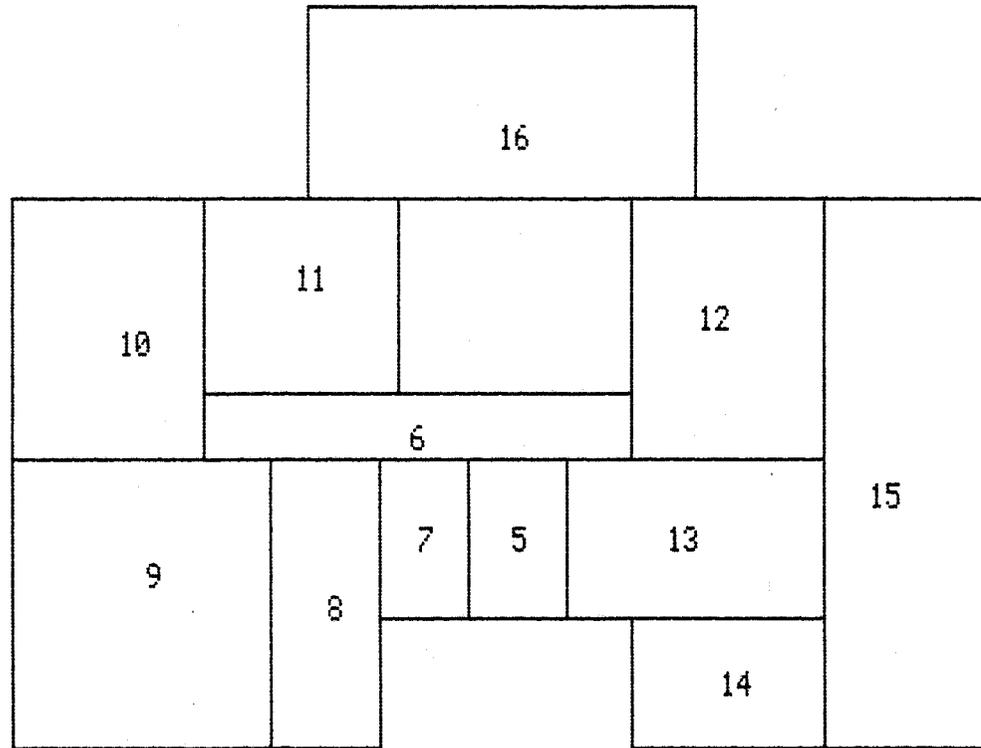
Local	Ancho	Largo	(min.)	Superf.	(min.)
4 Planta total	16.10	11.50	6.00	185.15	0.00
5 Vestíbulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	5.00	4.50	3.00	22.50	12.00
10 Dormitorio 2	4.00	3.00	3.00	12.00	12.00
11 Dormitorio 3	3.60	3.00	3.00	10.80	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	8.50	2.50	21.25	15.00
16 Estar	7.60	3.00	3.00	22.80	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	4.50	3.00	0.00	13.50	0.00
20 (vacío)	4.00	4.00	0.00	16.00	0.00

VALOR DE LA FUNCION OBJETIVO 139.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 138.85



SOLUCION 20



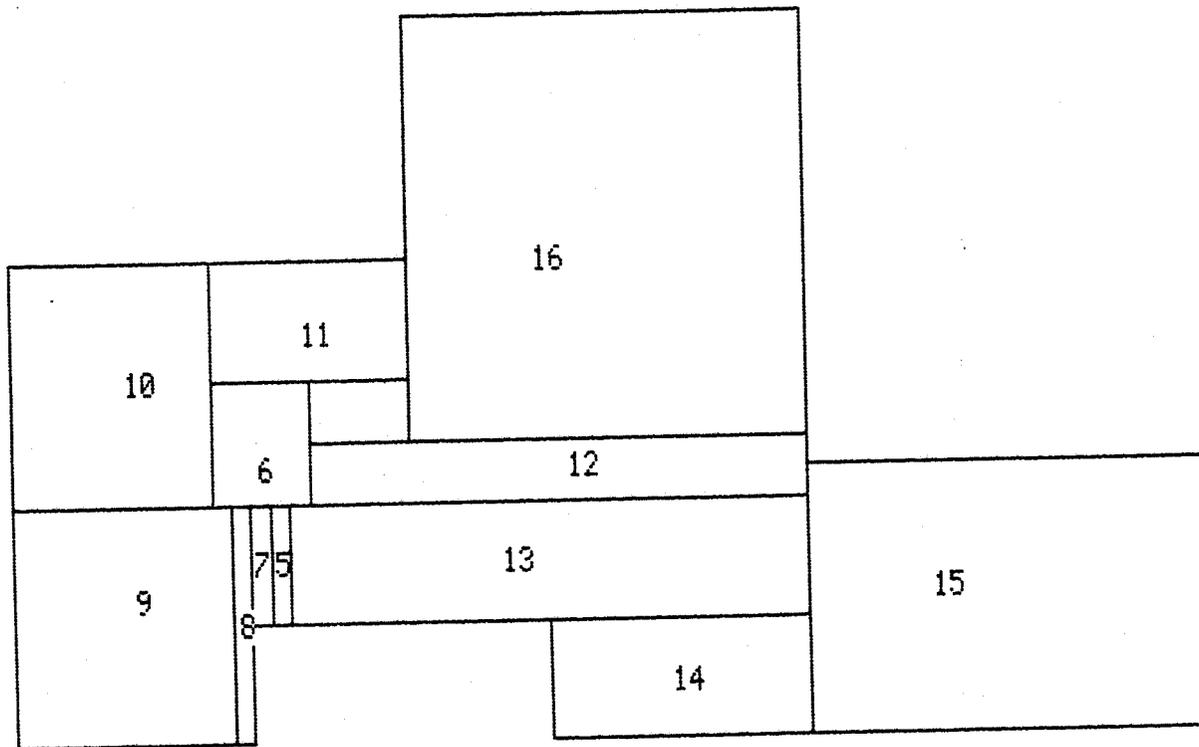
**SOLUCION 20**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 20

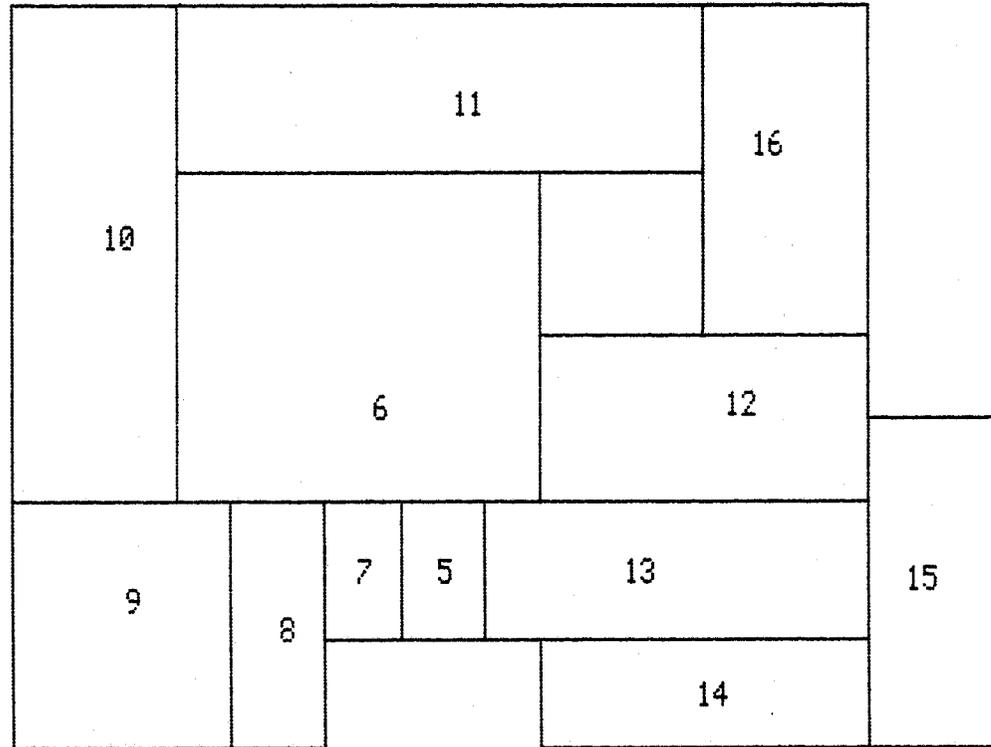
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	15.10	11.50	6.00	173.65	0.00
5 Vestíbulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	1.00	1.00	6.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	4.00	3.00	12.00	12.00
11 Dormitorio 3	3.00	3.00	3.00	9.00	0.00
12 Comedor	3.00	4.00	3.00	12.00	0.00
13 Cocina	4.00	2.50	2.50	10.00	10.00
14 Lavadero	3.00	2.00	2.00	6.00	0.00
15 Garage	2.50	8.50	2.50	21.25	15.00
16 Estar	6.00	3.00	3.00	18.00	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.60	3.00	3.00	10.80	0.00
19 (vacío)	4.50	3.00	0.00	13.50	0.00
20 (vacío)	4.60	3.00	0.00	13.80	0.00

VALOR DE LA FUNCION OBJETIVO 135.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 127.75



SOLUCION 21



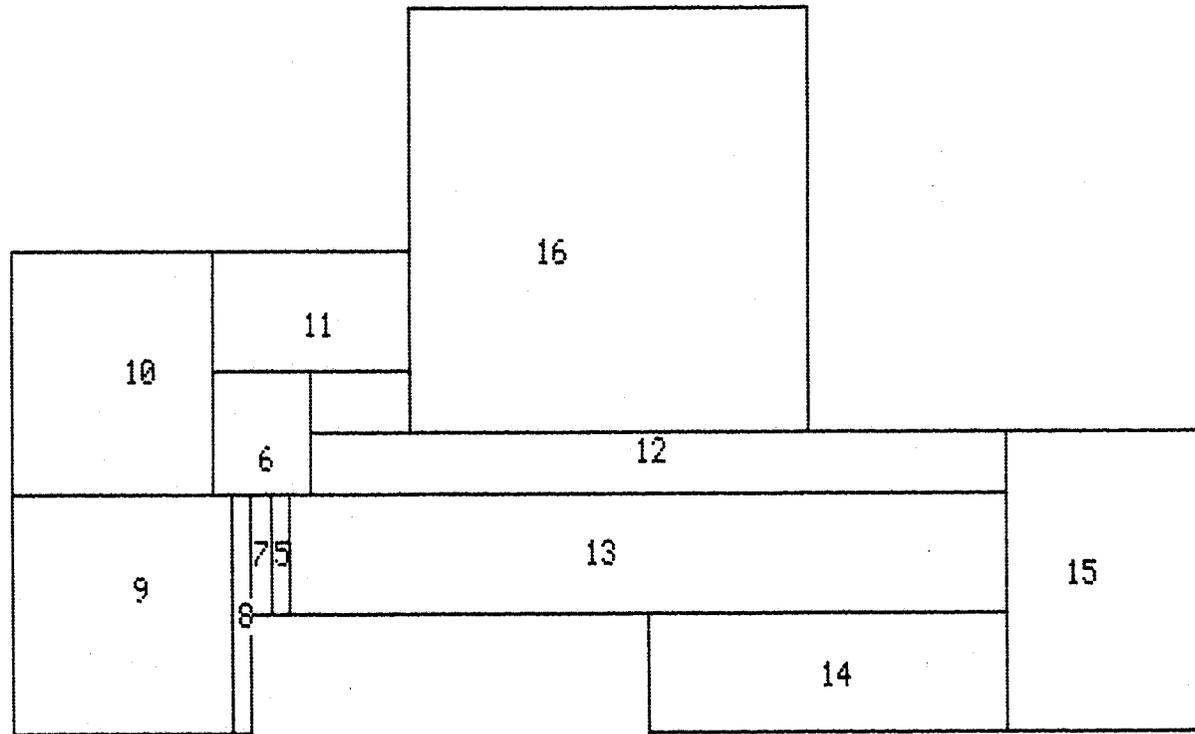
**SOLUCION 21**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 21

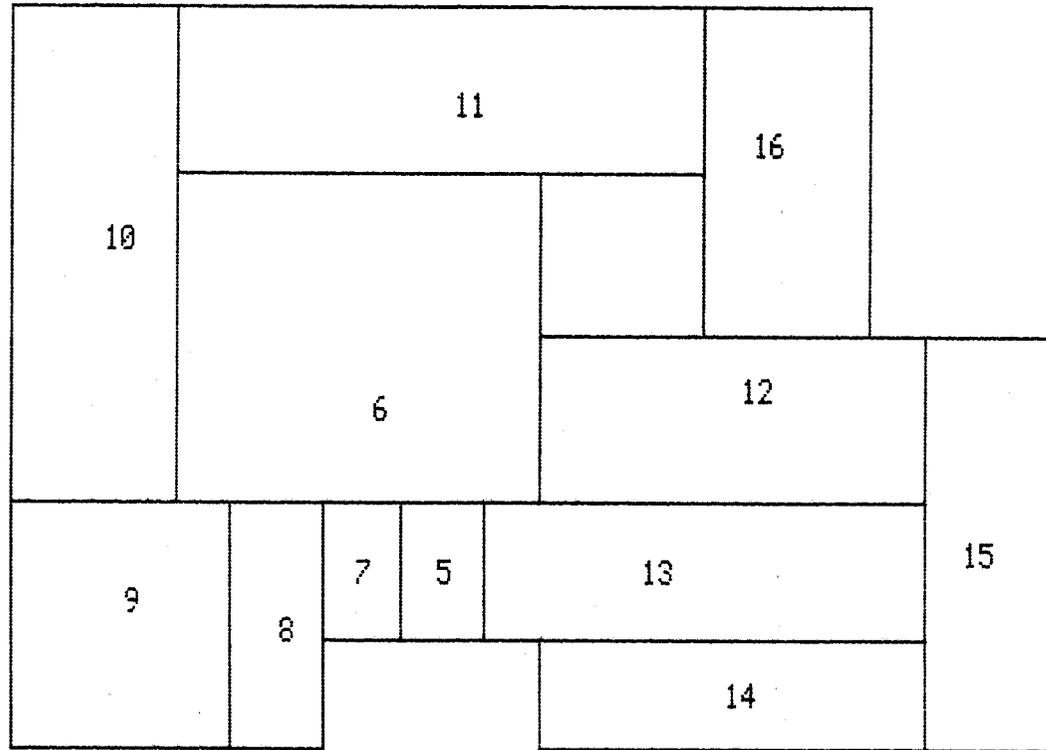
Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	18.10	13.50	6.00	244.35	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	6.00	1.00	39.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	9.00	3.00	27.00	12.00
11 Dormitorio 3	9.60	3.00	3.00	28.80	0.00
12 Comedor	6.00	3.00	3.00	18.00	0.00
13 Cocina	7.00	2.50	2.50	17.50	10.00
14 Lavadero	6.00	2.00	2.00	12.00	0.00
15 Garage	2.50	6.00	2.50	15.00	15.00
16 Estar	3.00	6.00	3.00	18.00	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	2.50	7.50	0.00	18.75	0.00
20 (vacío)	12.60	0.00	0.00	0.00	0.00

VALOR DE LA FUNCION OBJETIVO 169.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 208.8



SOLUCION 22



**SOLUCION 22**  
(dimensionada)

DIMENSIONES DE LA SOLUCION 22

Local	Ancho	Largo	(min.)	Superf. (min.)	
4 Planta total	19.10	13.50	6.00	257.85	0.00
5 Vestibulo	1.50	2.50	1.50	3.75	0.00
6 Distribuidor	6.60	6.00	1.00	39.60	0.00
7 Aseo	1.40	2.50	1.40	3.50	0.00
8 Baño	1.70	4.50	1.70	7.65	0.00
9 Dormitorio 1	4.00	4.50	3.00	18.00	12.00
10 Dormitorio 2	3.00	9.00	3.00	27.00	12.00
11 Dormitorio 3	9.60	3.00	3.00	28.80	0.00
12 Comedor	7.00	3.00	3.00	21.00	0.00
13 Cocina	8.00	2.50	2.50	20.00	10.00
14 Lavadero	7.00	2.00	2.00	14.00	0.00
15 Garage	2.50	7.50	2.50	18.75	15.00
16 Estar	3.00	6.00	3.00	18.00	18.00
17 (vacío)	3.90	2.00	0.00	7.80	0.00
18 Patio(vacío)	3.00	3.00	3.00	9.00	0.00
19 (vacío)	3.50	6.00	0.00	21.00	0.00
20 (vacío)	12.60	0.00	0.00	0.00	0.00

VALOR DE LA FUNCION OBJETIVO 174.9

SUPERFICIE TOTAL SIN ESPACIOS VACIOS 220.05

---

**ANEXO III. ALGUNOS CONCEPTOS  
DE LA  
TEORIA DE GRAFOS**

---

## 1. GRAFO

Un grafo  $G(V,A)$  se puede definir, en términos sencillos, como un par de conjuntos: el conjunto finito y no vacío de puntos o *vértices*  $V$ , y el de relaciones entre pares de vértices o *aristas*  $A$ , que puede ser vacío.

Si los vértices tienen asignada coordenadas en el plano tenemos un grafo *geométrico*, y en caso contrario será un grafo *abstracto*.

## 2. SUBGRAFO

Se dice que  $G'(A',V')$  es un *subgrafo* de  $G(A,V)$  si:

a)  $V' \subseteq V$  y  $A' \subseteq A$ ; y

b) toda arista de  $A'$  tiene sus extremos en  $V'$ .

## 3. VERTICES Y ARISTAS

Dos vértices  $u, v$  son *adyacentes* cuando están unidos por una arista  $a$ . En este caso se dice que la arista  $a$  *incide* sobre los vértices  $u$  y  $v$ , y éstos son los *extremos* de la arista  $a$ .

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

Un grafo será *simple* si no posee bucles ni aristas múltiples.

Entendemos por *valencia de un vértice* el número de aristas que inciden sobre él.

#### 4. RUTAS Y CIRCUITOS

Una *ruta* es una sucesión de aristas de un grafo, tales que cada arista (menos la primera y la  $n$ -ésima) tienen un vértice en común con la arista precedente, y el otro en común con la siguiente.

Se llama *circuito* o *ciclo* a aquella ruta cuyos vértices primero y último coincidan.

Se entiende por *longitud* de una ruta o circuito el número de aristas que lo forman.

#### 5. CONECTIVIDAD

Un grafo  $G(V, A)$  es *conexo* si para todo par de vértices distintos del conjunto  $V$  existe por lo menos una ruta que vaya de uno a otro. En caso contrario es *inconexo*.

Una *componente conexa* de un grafo es el subgrafo que contiene un vértice dado y todos los vértices que puedan enlazarse a él mediante una ruta.

Si de un grafo  $G$  eliminamos un vértice cualquiera  $v$  y todas las aristas que incidan sobre él, obtenemos un subgrafo  $G-v$  por *supresión de vértice*.

Si de un grafo  $G$  eliminamos cualquier arista  $a$ , obtenemos un subgrafo  $G-a$  por *supresión de arista*.

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

Un grafo  $G$  es *biconexo* si no posee vértices de corte. Así pues, el número mínimo de vértices a suprimir de un grafo biconexo para que resulte inconexo es dos.

Si  $G$  es un grafo conexo, una *componente biconexa* será el subgrafo biconexo que contenga el mayor número de vértices y aristas.

Un grafo biconexo  $G$  es *triconexo* si el número mínimo de vértices a suprimir para que resulte inconexo es tres.

Una *componente triconexa* de un grafo biconexo  $G$  será un subgrafo triconexo de éste con el máximo número posible de vértices y aristas.

En general, podemos decir que un grafo con al menos  $K+1$  vértices es *K-conexo* si para que resulte inconexo hay que suprimir  $K$  vértices -y las aristas que incidan sobre ellos- como mínimo.

## 6. PLANARIDAD

Un grafo es *planar* cuando puede ser representado en el plano de forma que no se intersecten geoméricamente dos aristas en puntos distintos a los vértices.

Si un grafo es planar y la inclusión de una arista entre dos

vértices cualesquiera -que no sea un bucle o una arista paralela- lo convierte en no planar, se tratará de un *grafo planar máximo*.

Entendemos por *cara* de un grafo planar la región del plano limitada por las aristas de un ciclo o circuito, tal que dos puntos arbitrarios de la citada región pueden unirse por un trayecto continuo que no atraviese ni vértices ni aristas.

La *valencia de una cara* será el número de aristas que la limitan.

Un grafo planar cuyas caras tienen todas valencia tres es una *triangulación*. Así pues, una triangulación es un grafo planar máximo.

## 7. DUALIDAD

El *dual*  $G^*$  de un grafo conexo y planar  $G$  es el grafo resultante de considerar por cada cara de  $G$  un vértice en  $G^*$  y por cada vértice de  $G$  una cara en  $G^*$ . De esta forma, una arista que relaciona dos vértices en  $G$  indicará que las dos caras correspondientes en  $G^*$  son adyacentes, y viceversa.

## 8. GRAFOS DIRIGIDOS

Un grafo será dirigido si todas y cada una de sus aristas -ahora llamadas *arcos*- son pares ordenados de vértices.

Un arco tiene, pues, un *vértice origen*  $v$  y un *vértice extremo*  $w$ .

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*.

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

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

## 9. ARBOLES

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

Todo subgrafo de un árbol es también un árbol.

Si  $G$  es un grafo y  $A$  es un árbol y a la vez un subgrafo de  $G$  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. Este vértice ha de ser tal que desde él puedan alcanzarse todos los vértices del árbol mediante rutas que partan del mismo.

Dado un grafo dirigido  $D$ , podemos establecer una partición de los arcos del mismo en dos clases distintas: un conjunto de arcos que nos definan un árbol extendido sobre  $D$ , y un conjunto de *frondas* que completen el número de arcos del grafo dirigido  $D$ . Si llamamos  $V$  al vértice origen de una fronda y  $W$  a su vértice extremo, y existe una ruta en el árbol extendido que va desde  $W$  hasta  $V$ , y esto ocurre para cada una de las frondas, a este grafo dirigido se le denomina *palmera*.

---

**BIBLIOGRAFIA**

---

Alexander, C. (1964)

*Notes on the Synthesis of Form.* Harvard University Press. Cambridge (Mass.). Trad: E.L. Revol. "Ensayo sobre la Síntesis de la Forma". 1969. Ed. Infinito. Buenos Aires.

Avdot'in, L.N. (1980)

"The use of computing techniques in the design of residential buildings". *Environment & Planning B*. Vol. 7, pp. 47-86.

Avondo-Bodino, G. (1979)

"Graph Theory and operations research". *Applications of Graph Theory*, (Wilson & Beineke ed.), Academic Press, Londres. pp. 223-253.

Baglivo, J.A. y Graver, J.E. (1983)

*Incidence and Symmetry in Design and Architecture.* Cambridge University Press. Cambridge.

Baybars, I. y Eastman, C.M. (1980)

"Enumerating architectural arrangements by generating their underlying graphs". *Environment & Planning B*. Vol. 7, pp. 289-310.

Berge, C. (1970)

*Graphes.* Dunod 1970. Bordas 1973. Paris.

Bloch, C.J. y Krishnamurti, R. (1978)

"The counting of rectangular dissections". *Environment & Planning B*. Vol 5, pp. 207-214.

Broadvent, G. (1974)

*Design in Architecture: Architecture and the Human Science.* John Wiley & Sons Ltd. Londres. Trad: "Diseño Arquitectónico: Arquitectura y Ciencias Humanas", ed. Gustavo Gili. Barcelona 1976.

- Brooks, R.L., Smith, C.A., Stone, A. y Tutte, W.T. (1940)  
 "The dissection of rectangles into squares". *Duke Mathematical Journal*. Vol. 7, pp. 312-341.
- Bruno, J., Steiglitz, K. y Weinberg, L. (1970)  
 "A new planarity test based on 3-connectivity". *IEEE Transactions on Circuit Theory*. Vol. 17, num. 2, pp. 197-206.
- Bryant, P.R. (1979)  
 "Graph Theory and electrical networks". *Applications of Graph Theory*, (Wilson & Beineke ed.), Academic Press, Londres. pp. 81-119.
- Cattermole, K.W. (1979)  
 "Graph Theory and communications networks". *Applications of Graph Theory*, (Wilson & Beineke ed.), Academic Press, Londres. pp. 17-57.
- Cea, J. (1971)  
*Optimisation theorie et algorithmes*. Dunod, col. Méthodes Mathématiques de l'informatique, vol. 2, Paris.
- Charalambous, C. (1977)  
 "Non linear least pth. optimization and non linear programming". *Mathematical Programming*. Vol. 12, pp. 195-225.
- Cousin, J. (1970)  
 "Topological organization of Architectural space". *Architectural Design*. Vol. 10, pp. 491-493.
- Doyle, J.K. y Graver, J.E. (1978)  
 "Mean distance in a directed graph". *Environment & Planning B*. Vol. 5, pp. 19-29.
- Doyle, J.K. y Graver, J.E. (1982)

"A summary of results on mean distance in shapes". *Environment & Planning B*. Vol. 9, pp. 177-179.

Earl, C.F. (1977)

"A note on generation of rectangular dissections". *Environment & Planning B*. Vol 4, pp. 241-246.

Earl, C.F. y March, L.J. (1979)

"Architectural applications of Graph Theory". *Applications of Graph Theory*, (Wilson & Beineke ed.), Academic Press, Londres. pp. 327-355.

Earl, C.F. (1980)

"Rectangular shapes". *Environment & Planning B*. Vol. 7, pp. 311-342.

Earl, C.F. (1981)

"Enumerating architectural arrangements: comment on recent paper by Baybars and Eastman". *Environment & Planning B*. Vol. 8, pp. 115-118.

Earl, C.F. y Johnson, J.H. (1981)

"Graph theory and Q-analysis". *Environment & Planning B*. Vol 8, pp. 367-391.

Eastman, C.M. (1970)

"Representations for space planning". *Communications of the ACM*. Vol. 13, num. 4, pp. 242-250.

Eastman, C. (1972)

"Preliminary report on a system for general space planning". *Communications of the ACM*. Vol 15, num. 2, pp. 76-87.

Eastman, C.M. (1973)

"Automated space planning". *Artificial Inteligence*. Vol 4, 41-64.

Fisher, G.J. (1966)

"Computer recognition and extraction of planar graphs from the incidence matrix". *IEEE Transactions on Circuit Theory*. Vol. 13, num. 2, pp. 154-163.

Flemming, U. (1978)

"Wall representations of rectangular dissections and their use in automated space allocation". *Environment & Planning B*. Vol. 5, pp. 215-232.

Flemming, U. (1980)

"Wall representations of rectangular dissections: additional results". *Environment & Planning B*. Vol. 7, pp. 247-251.

Gero, J.S. (1977)

"Note on 'Synthesis and optimization of small rectangular floor plans' of Mitchell, Steadman and Liggett". *Environment & Planning B*. Vol. 4, pp. 81-88.

Gilep, E., Murray, W., Wright, M. (1981)

*Practical optimization*. Academic Press. Londres.

Gilleard, J. (1978)

"LAYOUT - a hierarchical computer model for the production of architectural floor plans". *Environment & Planning B*. Vol. 5, pp. 233-241.

Grooms, H.R. (1972)

"Algorithm for matrix bandwidth reduction". *Journal of the Structural Division*. Enero, pp. 203-214.

Harray, F., March, L., Robinson, R.W. (1978)

"On enumerating certain design problems in terms of bicoloured graphs with no isolates". *Environment & Planning B*. Vol 5, pp. 31-43

- Hock, W. y Schittokowsky, K. (1981)  
*Test examples for nonlinear programming codes*. Board.  
Berlin. New York.
- Hopcroft, J y Tarjan, R. (1973)  
"Dividing a graph into triconnected components". *Siam J.  
Comput.*, Vol. 2, pp. 135-158.
- Hopcroft, J. y Tarjan, R. (1973)  
"Efficient algorithms for graph manipulation". *Communi-  
cations of the ACM*. Vol 16, pp. 372-378.
- Hopcroft, J y Tarjan, R. (1974)  
"Efficient Planary Testing". *Journal of the association  
for computing machinery*. Vol. 21, num. 4, pp. 549-568.
- Hutton, G y Roston, M. (1979)  
*Computer programs for the building industry*. Archi-  
tectural Press. Londres.
- Kaufmann, A. (1964)  
*La programmation dynamique*. Dunod. Paris. Trad.: La  
Programación Dinámica. 1967. Ed. Continental. México.
- Kaufmann, A. (1976)  
*Des point et des fleches... la theories des graphes*. Du-  
nod. Paris. Trad.: "Puntos y flechas. Teoría de los  
grafos". Ed. Marcombo. Barcelona.
- Koleichuk, V.F. (1980)  
"Programmed form generation in design". *Environment &  
Planning B*. Vol. 7, pp. 19-30.
- Korf, R.E. (1977)  
"A shape independent theory of space allocation". *Envi-  
ronment & Planning B*. Vol. 4, pp. 37-50.

Krejcirik, M. (1969)

"Computer-aided plant layout". *Computer-Aided Design*.  
Otoño, pp. 7-19.

Krishnamurti, R. y Roe, P.H. (1978)

"Algorithmic aspects of plan generation and enumeration". *Environment & Planning B*. Vol 5, pp. 157-177.

Krishnamurti, R. (1979)

"3-Rectangulations: an algorithm to generate box packings". *Environment & Planning B*. Vol 6, pp. 331-352.

Krishnamurti, R. (1980)

"The arithmetic of shapes". *Environment & Planning B*.  
Vol. 7, pp. 463-484.

Krishnamurti, R. (1981)

"The construction of shapes". *Environment & Planning B*.  
Vol. 8, pp. 5-40

Kuratowski, K. (1930)

"Sur le problème des courbes gauches en topologie".  
*Fundamenta Math.*, vol 15, pp. 271-283.

Kuratowski, K. (1961)

*Wstep do teorii mnogoscii topologii*. Panstowowe Wyda-  
wnictwo Naukowe. Varsovia. Trad.: "Introducción a la  
teoría de conjuntos y topología". 1966. Vicens-Vives.  
Barcelona. 1966.

Larrañeta, J. (1977)

*Programacion Lineal y Grafos*. Secretariado de Publica-  
ciones de la Universidad de Sevilla.

Lenart, M. (1980)

"The examination and construction of maximal configu-

rations". *Environment & Planning B*. Vol. 7, pp. 253-264.

Levin, P.H. (1964)

"Use the graphs to decide the optimum layout of buildings". *The Architect's Journal Information*. Vol. 7, pp. 809-815.

March, L. y Steadman, J.P. (1971)

*The Geometry of Environment*. Riba Publications Ltd. London.

March, L. y Earl, C.F. (1977)

"On counting Architectural plans". *Environment & Planning B*. Vol 4, pp. 57-80.

Mitchell, W.J., Steadman, J.P. y Liggett, R.S. (1976)

"Synthesis and optimisation of small rectangular floor plans". *Environment & Planning B*. Vol. 3, pp. 37-70.

Mitchell, W.J. (1977)

*Computer-Aided Architectural Design*. Mason / Charter Publisher. Nueva York.

Moucka, J. (1979)

"The use of propositional calculus in architectural design". *Environment & Planning B*. Vol 6, pp. 263-268.

Negroponte, N. (1970)

*The Architecture Machine*. M.I.T. Press. Cambridge (Mass.). Trad.: "La macchina per L'Architettura". Ed. Il Saggiatore. Milán, 1974.

Nielsen, F. (1979)

"Flowgraphs". *Applications of Graph Theory*, (Wilson & Beineke ed.), Academic Press, Londres. pp. 59-80.

Polak, E. (1971)

*Computational methods in optimization: a unified approach.* Academic Press. Londres.

Portlock, P.C. y Whitehead, B. (1974)

"Three dimensional layout planning". *Build Science*. Vol 9, pp. 45-53.

Prada, J.M. (1982)

"Computer design?". *ON*. Núm. 38, pp. 40-45.

Rao, S.S. (1978)

*Optimization theory and applications.* Wiley Eastern Limited. New Delhy.

Read, R. (1979)

"Algorithms in graph theory". *Applications of Graph Theory*, (Wilson & Beineke ed.), Academic Press, Londres. pp. 381-417.

Ritzman, L.P. (1972)

"The efficiency of computer algorithms for plant layout". *Magnament Science*. Vol 18, num. 5, part I, pp. 240-247.

Rodriguez, J.S. (1975)

"Node numbering optimization in structural analysis". *Journal of the Structural Division*. Febrero, pp. 361-376.

Scarano, R. (1979)

*Progettazione per ottimizzazione.* Ligouri Editore. Nápoles.

Seguí, J. (1972)

"Investigación sobre procesos de diseño. 1- Planteamiento general". Pub. del Seminario de Arquitectura y Automática en la E.T.S.A. de Madrid.

Seguí, J. y Gutiérrez Guitián, M.V. (1972)

"Ideas para el análisis y simulación de procesos artísticos". Pub. del Seminario de Arquitectura y Automática de la E.T.S.A.M.

Seguí, J. y Gutiérrez Guitián, M.V. (1974)

"Investigación en procesos de diseño. Modelo operativo de formalización". Boletín del centro de Cálculo de la Universidad de Madrid.

Sendra, J.J. (1984)

*Generación y optimización automática de esquemas de distribución de edificios en planta.* Tesis doctoral. E.T.S. de Arquitectura. Universidad de Sevilla.

Sendra, J.J. (1985)

"Generación y trazado de esquemas adimensionales de distribución de edificios en planta". *Revista internacional de métodos numéricos para el cálculo y diseño en ingeniería.* Vol. 1, num. 3, pp. 37-55.

Sendra, J.J. y Martín, J.D. (1985)

"Método para la optimización dimensional de esquemas distributivos en planta". Comunicación presentada al Primer Congreso Iberoamericano de métodos computacionales en Ingeniería. Madrid 1985.

Sevilla, C. (1972)

"Investigación sobre procesos de diseño: 2-Modelo de Organismo (I y II partes)". Pub. del Seminario de Arquitectura y Automática de la E.T.S.A.M.

Sevilla, C. (1981)

"Notas sobre la informatización y el diseño en Arquitectura". *Informes de la Construcción.* Num. 337, pp. 5-20.



Shavid, E. y Gali, D. (1974)

"A model for space allocation in complex buildings: A computer graphic approach". *Build International*. Vol 7, pp. 439-518.

Steadman, J.P. (1983)

*Architectural Morphology*. Pion Limited. Londres.

Stiny, G. (1980)

"Introduction to shape and shape grammars". *Environment & Planning B*. Vol. 7, pp. 343-351.

Stiny, G. y March, L. (1981)

"Design machines". *Environment & Planning B*. Vol. 8, pp. 245-255.

Tuner, J.C. (1970)

*Modern applied mathematics: probability-statistic-operational research*. The English University Press Ltd. Londres. Trad.: "Matemática moderna aplicada, probabilidades, estadística e investigación operativa". Alianza Editorial. Madrid. 1974.

Tutte, W.T. (1962)

"A census of planar maps". *Canadian Journal of Mathematics*. Vol 15, pp 249-271.

Tutte, W.T. (1963)

"How to draw a graph". *Proc. London Math. Soc.*, vol. 3, num. 13, pp. 743-768.

Wilson, R.J. y Beineke, L.W. (1979)

*Applications of graphs theory*. Academic Press. London.

Wright, R., Boyer, L.R.T. y Melin, J.W. (1971)

"Constraint processing in design". *Journal of the Structural Division*. Enero, pp. 481-494.