

01149 47



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DIVISION DE ESTUDIOS DE POSGRADO

APLICACION DEL ALGORITMO DE LA *p*-mediana A
PROGRAMAS DE DESARROLLO SOCIAL UTILIZANDO
UN SIMULADOR DE VUELO TRIDIMENSIONAL DEL
TERRITORIO NACIONAL COMO VISUALIZADOR DE
LOS RESULTADOS EN INTERNET

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERIA
P R E S E N T A
ELIO VEGA MUNGUA



DIRECTORA: DRA JUDITH ZUBIETA GARCIA

CIUDAD UNIVERSITARIA, MEXICO, D. F.

2002

TESIS CON
FALLA DE ORIGEN

Dedico este trabajo a:

Mi papá Juvenal por su ejemplo, y apoyo incondicional que siempre me ha brindado.

Mi mamá Sofia por su dedicación y amor incondicional.

A mis hermanos José, Lila y Nadia por su cariño y alegrías que hemos vivido

A mis tios y tias Hugo, José, Lila, Virginia, Leonor, Martha, Elsa, Cristina por su apoyo incondicional

A mis demás Familiares por su confianza

A mi amiga y asesora Dra. Judith Zubieta García por su paciencia, apoyo, confianza, conocimientos e inestimables consejos, a mi amigo el Dr. Alberto Alonso y Coria.

A mis amigos y amigas Carmen Ramos, Victor Godoy, José Luis Villarreal, Lizbeth Heras, Karina Jimenez, Ivonne Jimenez, Gabriela Barbosa, Claudia Cortes por su amistad, compañía y apoyo.

A la memoria de mis abuelitos José y Elena

Agradezco a la UNAM por que es la fuente de mis conocimientos A mis maestros y en especial a todas las personas del Departamento de Visualización de la DGSCA por el apoyo para realizar este trabajo.

Índice General

Introducción	V
1 El problema de las localidades aisladas	3
1.1 Características de las localidades pequeñas	4
2 Modelos para la localización de instalaciones	6
2.1 Modelo simple de interacción espacial	7
2.1.1 Una aplicación del modelo simple de interacción espacial	9
3 Algoritmo de localización <i>p-mediana</i>	14
3.1 El algoritmo	17
3.1.1 Paso A: Inicialización	18
3.1.2 Paso B: Construcción de los árboles	18
3.1.3 Paso C: Redefinición de las raíces para cada árbol	19
3.1.4 Paso D: Intercambio de raíces	19
3.1.5 Paso E: Atracción de una raíz por otro árbol	20
3.1.6 Paso F: Unión de dos raíces en un solo árbol	20
3.2 Análisis de resultados	21
3.3 Estudio del caso del estado de Hidalgo	21
3.3.1 Criterios	22
3.3.2 Ponderadores	24
3.4 Estudio del caso del estado de México	29
3.5 Estudio del caso del estado de Querétaro	31
4 Desarrollo de un sistema para desplegar resultados	35
4.1 Proceso preliminar	36
4.2 Representación digital	39
4.3 Despliegue gráfico	42
4.3.1 ¿Que es Java3D?	42
4.3.2 El API de Java3D	43
4.3.3 Pasos para escribir programas en Java3D	46
4.3.4 El Sistema de Coordenadas en el Universo Virtual	51

ÍNDICE GENERAL

4.4	Descripción de las clases del sistema	56
4.4.1	Clase SimuladorDeVuelo	56
4.4.2	Clase crearPanels	57
4.4.3	Clase crearGrafoDeEscena	57
4.4.4	Clase ConstructorUniversoVirt	59
4.4.5	Clase Escenario	59
4.4.6	Clase ObjSimVuelo	60
4.4.7	Clase DibujaMapa	61
4.4.8	Clase VentMapaMexico	63
4.4.9	Clase Ionlat.Axy	64
4.4.10	Clase LectorMallaARC	64
4.4.11	Clase IdentificadorDeMovimiento	65
4.4.12	Clase TexturasDeObjetos	65
5	Conclusiones	67
	Bibliografía	68
	Anexo	72

Introducción

La ubicación y localización de instalaciones ha sido un tema que lleva décadas de estudio, pero es en siglo XX cuando ha tenido su mayor auge al ser parte de la nueva área, la Investigación de Operaciones, permitiendo vincular áreas interdisciplinarias. Economistas, urbanistas, arquitectos, ingenieros de diferentes ramas y matemáticos han redescubierto un mismo interés en la ubicación de instalaciones. Pero cada uno le da un enfoque distinto a los demás, por lo que surgen distintas opiniones sobre un mismo problema.

En este trabajo nos dedicaremos a explicar analíticamente un modelo matemático para la ubicación y localización de instalaciones proveedoras de servicios, para optimizar la prestación de éstos, así como su implementación en un programa de computadora.

El criterio que se utilizará con el fin de encontrar la mejor solución, es el de minimizar la función que determina la distancia a recorrer, o lo que es lo mismo: minimizar la distancia máxima de viaje.

Para nuestro particular caso de estudio, este criterio ayudará a desarrollar mecanismos para ampliar la provisión de servicios que brinda cierto tipo de infraestructura pública - por ejemplo, en lo referente a educación y salud- sin tener que realizar grandes inversiones en todas y cada una de las localidades donde habita la población cuyo acceso a este tipo de servicios ha sido tradicionalmente reducido. La forma en que se suministrarán dichos servicios, es por medio de comunidades que cuenten con dicho conjunto de servicios básicos, a este conjunto de comunidades les damos el nombre de "núcleos proveedores de servicios".

El objetivo principal de este proyecto es entonces el diseño de un algoritmo matemático que permita determinar, con rapidez y precisión, qué localidades -dentro de los límites territoriales de cada entidad federativa de la República Mexicana- poseen las características necesarias para ser consideradas como "núcleos proveedores de servicios" en potencia.

Una vez establecido el conjunto de aquéllas que cumplen con el perfil previamente definido como deseable, es posible aplicar un algoritmo que facilite la identificación de los puntos o localidades óptimas para garantizar que cualquiera que sea la inversión en infraestructura que se haga, aumentará la población usuaria y con ello los beneficiarios de dicho servicio.

El algoritmo tendrá un carácter general, es decir, podrá ser utilizado para cualquier entidad federativa, siempre y cuando la información que lo alimente cumpla con el perfil establecido para tal efecto. Con esto se garantiza que el impacto en el ámbito local,

regional y nacional, de aquellos fondos que se tengan disponibles o que puedan gestionarse para este fin, de una manera eficaz y efectiva. Este algoritmo es el de la *p-mediana*.

Capítulo 1

El problema de las localidades aisladas.

Como resultado de los cambios económicos y sociales, la dinámica del poblamiento de México ha tenido cambios importantes en la segunda mitad de siglo XX ya que en un lapso breve han sucedido dos importantes sucesos demográficos que han consistido en la estabilización del número de localidades con menos de 100 mil habitantes y cuando se produjo un significativo crecimiento de los asentamientos urbanos. Es en las décadas de los setenta y ochenta cuando parece haberse creado cerca de 50 mil asentamientos humanos.

La distribución de la población avanzó en un sentido inverso a la evolución de las localidades, consolidándose de ese modo el carácter eminentemente urbano de la población de México, ya que en 1970, las ciudades de 15 mil y más habitantes, vivían 18 millones de personas, es decir, más del doble que en diez años atrás; la tendencia se consolidó en los quince años posteriores, de forma que en 1995, en 347 ciudades vivían poco más de 58 millones de personas. Así, la población que reside en las ciudades pasó de representar el 46 por ciento de la población en 1970 a 58 por ciento en 1995, mientras que el número de localidades urbanas se incrementó de 161 a 347, representando 0.25 y 0.18 por ciento del total de asentamientos de 1970 a 1995, respectivamente.

Los asentamientos con menos de 2500 habitantes fueron los protagonistas del impresionante aumento de las localidades del país, pues prácticamente se duplicaron en los últimos 25 años. En 1970, 95 mil localidades tenían menos de 2,500 habitantes, en 1990, 154 y en 1995, 198 mil respectivamente. En cambio, la población que reside en dichas localidades aumentó sólo en cinco millones, al pasar de 18 millones de personas en 1970 a 24 millones en 1995. En consecuencia, la densidad media de las localidades pequeñas fue en franco descenso al pasar de 208 a 121 habitantes, respectivamente, ver Cuadro 1.1.

Diversos factores de índole económica, demográfica e incluso cultural influyen en la tendencia a multiplicar el número de asentamientos, tanto en la conformación de grandes metrópolis como en su dispersión en el medio rural. Este último es hacia el medio al que

Cuadro 1.1.

Tamaño de la localidad	Número de localidades y población por tamaño de localidad, 1970-1995			
	1970	1980	1990	1995
	Num. de loc. Población	Num. de loc. Población	Num. de loc. Población	Num. de loc. Población
Nacional	97,580 - 48,225,238	125,300 - 66,846,833	156,602 - 81,249,645	201,138 - 91,158,290
1 a 99 hab.	33,630 - 1,471,154	78,806 - 1,888,982	108,307 - 2,190,339	131,305 - 2,638,091
100 a 499 hab.	23,035 - 6,389,007	31,054 - 7,344,871	32,244 - 7,760,320	33,426 - 7,965,923
500 a 999 hab.	7,473 - 5,190,166	8,473 - 5,856,009	8,515 - 5,922,495	8,578 - 6,982,776
1,000 a 2,499 hab.	4,232 - 6,366,283	4,936 - 7,227,342	4,930 - 7,416,770	3,043 - 7,588,037
2,500 a 4,999 hab.	1,201 - 4,129,872	1,147 - 4,092,168	1,364 - 4,547,366	1,457 - 4,996,974
5,000 a 9,999 hab.	339 - 3,764,305	513 - 3,527,104	609 - 4,226,394	665 - 4,644,778
10,000 a 19,999 hab.	248 - 3,409,846	248 - 3,407,480	298 - 4,046,017	332 - 4,567,636
20,000 a 49,999 hab.	114 - 3,405,818	119 - 3,596,371	167 - 3,075,138	204 - 6,146,276
50,000 a 99,999 hab.	34 - 2,356,569	33 - 2,337,699	55 - 3,854,850	61 - 4,340,532
100,000 a 499,999 hab.	30 - 5,707,130	52 - 11,352,926	77 - 18,233,313	80 - 19,000,266
500,000 hab. y más	4 - 5,535,113	19 - 15,985,981	21 - 17,836,493	28 - 23,287,053

Fuentes: Secretaría de Industria y Comercio. IX Censo General de Población, 1970. Resumen General
INEGI X Censo General de Población y Vivienda, 1980. Resumen General
INEGI XI Censo General de Población y Vivienda, 1990. Resumen General
INEGI Censo de Población y Vivienda, 1995. Resultados Definitivos Tabulados Básicos

se plantea dirigir este trabajo para tratar de ayudar en el abatimiento de su rezago

1.1 Características de las localidades pequeñas

La situación geográfica adversa es una de las características principales que sobresalen en estas localidades lo que deriva en condiciones de mayor rezago económico, social y demográfico. Su ubicación geográfica tiene una importancia decisiva para determinar la estructura de oportunidades a la cual tiene acceso la población. En el caso particular de quienes viven en localidades con menos de 2500 habitantes, la cercanía o alejamiento de las ciudades, así como la disponibilidad de vías y medios de comunicación, determinan diferentes oportunidades de obtener un empleo bien remunerado, una vivienda adecuada a las necesidades de las familias y acceder a servicios básicos como educación y salud, así como otros bienes y servicios esenciales cuya carencia compromete su presente y futuro.

En ese sentido, el diseño de estrategias y programas públicos de apoyo a la población con mayores rezagos requiere de estudios que dimensionen adecuadamente las necesidades de la población y consideren su situación geográfica, pues además de lo oneroso e inviable que resultaría pretender llevar todos los servicios a todas las localidades del país, las necesidades de la población tiene diferencias significativas, derivadas de entre otros aspectos, de la cultura, las costumbres y su ubicación geográfica. Asimismo, debe considerarse que el conocimiento del tamaño de la localidad, si esta situada cerca de núcleos urbanos o alejados de ellos, tiene gran trascendencia para la pleneación del desarrollo regional sustentable y para la evaluación de los costos directos e indirectos que su atención supone.

Una forma de conocer la diversidad de las situaciones que nos podemos encontrar en microregiones es utilizando un Sistema de Información Geográfica ya que de esta manera de las 195 mil localidades con menos de 2500 habitantes que reporta el conteo de 1995, 71 mil están situadas en el área de influencia de las ciudades, y en ellas viven 10.5 millones habitantes, de forma que el mundo urbano de México esta integrado por las 14,277 loca-

TESIS CON
FALLA DE ORIGEN

lidades con 15 mil y más habitantes que integran las 347 ciudades del país. Con ello, el volumen de población urbana asciende a 68.5 millones de personas, y no únicamente por los 58 millones de personas que viven en localidades con más de 15 mil habitantes.

A su vez, del total de localidades con menos de 2500 habitantes que están fuera del área de influencia de las ciudades, 55 mil se encuentran cerca de una carretera, y alrededor de 69 mil está aisladas, es decir alejadas de las ciudades y carreteras. En consecuencia, los asentamientos ubicados fuera del área de influencia de las ciudades asciende a 123 mil y en ellos viven 7.2 millones de personas, de los cuales 6.6 millones residen en localidades aisladas.

Otro aspecto que ayuda a dimensionar la índole de las dificultades que enfrenta la acción pública para atender a la población que reside en localidades pequeñas, es el hecho de que conforme los poblamientos se alejan de las ciudades su densidad disminuye. Así, mientras en las localidades situadas en el área de influencia de las ciudades tienen un promedio de dos mil habitantes, las ubicadas cerca de carreteras alcanzan un promedio de 700 personas y las localidades aisladas de 120. De esta forma, el aprovechamiento de las economías de aglomeración es factible en los asentamientos más próximos a las ciudades, pero sumamente difícil en los que están alejados de ellas.

TESIS CON
FALLA DE ORIGEN

Capítulo 2

Modelos para la localización de instalaciones.

Como se mencionó al principio de este trabajo, la ubicación y localización de instalaciones ha sido un tema que lleva décadas de estudio, pero es en este siglo cuando ha tenido su mayor auge al ser parte de una nueva área dentro de la Investigación de Operaciones. Economistas, urbanistas, arquitectos, ingenieros de diferentes ramas y matemáticos han vuelto a tener interés en el tema de la ubicación de instalaciones: sin embargo, cada uno puede darle un uso distinto. Donde resulta interesante es cuando surgen distintas opiniones alrededor de un mismo problema.

Una de las dificultades más serias que presenta este proyecto es de orden conceptual y metodológico. La correcta definición de las características que conformarán el perfil deseado que una localidad debe presentar para ser considerada "núcleo potencial". Deberá contener una valoración real de distintas variables sociodemográficas y económicas de cada una de las localidades. Entre éstas, deberán considerarse:

- Su población total y su estructura por edades
- Su actividad económica predominante
- Las características de las viviendas habitadas
- El índice de marginación
- Las telecomunicaciones
- La infraestructura básica y los servicios ya disponibles en la localidad; la accesibilidad y las distancias entre las localidades y los centros de mayor población que ofrezcan servicios de mejor calidad

Sin embargo, otro problema, de orden operativo, es la información. Se requieren datos desagregados por localidad y con un mismo referente de tiempo. Los resultados

TESIS CON
FALLA DE ORIGEN

provenientes de diversas áreas del INEGI¹ son invaluable para garantizar la pertinencia y la oportunidad del proyecto, empezando por la propia definición de localidad

Otra dificultad está asociada a la definición de los criterios que deberán observarse en el proceso de localización óptima, es decir, a la determinación de ciertas prioridades en las que invariablemente deberán incorporarse aquéllas provenientes de los programas de trabajo de las autoridades estatales. Por ejemplo, si se desea privilegiar a las localidades más marginadas, deberá considerarse:

- El total de viviendas a beneficiar y la población que será potencialmente atendida.
- Las características, la ubicación geográfica y la accesibilidad de la localidad (tipo de infraestructura disponible, existencia de caminos, cercanía con localidades comunicadas, etc.)
- La alternativa de que haya poblaciones o municipios dentro de un mismo estado que no resulten beneficiados directamente

Un gran universo de alternativas fue analizado para definir las características de un núcleo potencial. Por un lado, las características que revelan que una localidad presenta una demanda insatisfecha de servicios que, mediante un programa como el que aquí se propone, pudieran ser cubiertos; por otro lado, las características de su entorno cultural, económico y geográfico que puedan garantizar, con el menor riesgo posible, que las localidades aledañas gravitarán a su alrededor para la satisfacción de sus demandas.

De esta manera, se tiene identificado el perfil deseable para los núcleos potenciales proveedores de servicios así como la herramienta de *Investigación de Operaciones* que resulta más conveniente para la construcción del modelo matemático de optimización; pero antes de explicar a fondo lo anterior, primero se discutirá el siguiente algoritmo.

2.1 Modelo simple de interacción espacial

Uno de los primeros problemas que se pueden encontrar es el de definir los conjuntos de localidades que presentan una mayor relación. Es decir, encontrar las poblaciones que, de acuerdo con la intensidad y frecuencia de sus relaciones (flujos de población, bienes, información) definen los sistemas de asentamiento.

Como ejemplo podemos tomar del Cuadro 2.1 en donde se observa un conjunto de localidades del estado de Hidalgo, que se encuentren dispersas. Como sucede en la realidad, las localidades tendrán diferentes características

¹Instituto Nacional de Estadística Geografía e Informática

Entre éstas, las demográficas (por ejemplo, el tamaño de la población o la estructura poblacional) y las económicas serán las más relevantes para la planeación regional aunque la consideración de otras características sociales, políticas y culturales perfilarían mejor la imagen del conjunto de asentamientos

Cuadro 2.1: Comunidades en el estado de Hidalgo

Nombre	Pob.	Pob. < 3	SSA	UMF	HOS	IMSOL	TEGFO	DYCONS	TEL	PRIM	SEC	BACH
La Mesa de												
Linantilla	562	462	0	0	0	0	0	0	0	1	0	0
Xiquila	1679	255	0	0	0	1	0	1	1	1	1	0
Tepeolotl	299	61	0	0	0	0	0	0	0	2	0	0
Atlatco	438	77	1	0	0	0	0	1	0	1	0	0
Coxhuaco 2°	323	269	0	0	0	0	0	0	0	1	0	0
Coxhuaco 1°	83	59	0	0	0	0	0	0	0	1	0	0

Las localidades están interconectadas por una red de comunicaciones, la que permitirá que las distintas poblaciones establezcan relaciones entre sí, ya que es por esta red, que podrán fluir los bienes, servicios y la población entre las diferentes poblaciones.

Dado que los asentamientos tienen características distintas, los flujos variarán en cuanto a su tipo. Por otro lado, los flujos no se distribuirán homogéneamente; las características de cada asentamiento definirán su capacidad para atraer o generar flujos. De esta manera, algunas localidades recibirán flujos importantes y articularán el funcionamiento de la región, y otras serán sobre todas emisoras de flujos y por lo tanto dependientes funcionales de las primeras.

La localización de los servicios podrá ajustarse o no a esa estructura de acuerdo a los objetivos de planeación regional, pero no se puede ignorar por que en función de ella es como se definirán las acciones de planificación

Para esta planificación de los servicios, los flujos más relevantes son los de población. Desgraciadamente el obtener información exacta y fidedigna es muy difícil y en extremo costoso, por el tiempo y el dinero que hay que invertir para recolectarla.

Otra alternativa es usar otros flujos como indicadores -tendidos telefónicos, pero como en nuestro caso de estudio las poblaciones son rurales altamente marginadas, este servicio no representa un indicador viable.

Una alternativa para explotar los flujos entre los asentamientos de manera rápida y económica es utilizar un modelo simple de interacción espacial. Este tipo de modelo se apoya en los conceptos básicos de la Teoría de Lugar Central. Conforme a ésta, los flujos entre los asentamientos estarán relacionados positivamente con la centralidad de cada lugar y negativamente con la distancia que exista entre ellos.

Según la definición de centralidad ya presentada (capacidad exportadora de servicios de cada localidad), cuanto mayor sea la centralidad de un lugar, mayor será su capacidad de atraer a personas que buscan satisfacer sus necesidades. Pero si la centralidad anima la existencia de flujos y relaciones, la distancia que separa a los asentamientos la inhibe. Este

**TESIS CON
FALLA DE ORIGEN**

es, en términos esquemáticos, el razonamiento que sustenta a los modelos de interacción espacial.

Un modelo simple de interacción espacial puede ser útil, en primer momento, para explotar las interrelaciones funcionales entre un conjunto de asentamientos. El modelo se define formalmente de la siguiente manera:

$$I_{i,j} = \frac{P_i P_j}{d_{ij}} \quad (2.1)$$

En donde $I_{i,j}$ es la interacción entre dos asentamientos llamados "i" y "j", cuyas poblaciones son P_i y P_j que están separadas por una distancia igual a d_{ij} .

Este modelo simple de interacción espacial es una antología de la teoría gravitacional de Newton (por ello con frecuencia se le llama también modelo gravitacional) y expresa la intensidad de la interacción entre dos asentamientos gravitacional) $I_{i,j}$ estará directamente relacionada con el tamaño de sus masas (P_i y P_j) e inversamente relacionada con la distancia que los separa d_{ij} .

2.1.1 Una aplicación del modelo simple de interacción espacial

Para ejemplificar la aplicación de un modelo simple de interacción espacial, se considerarán las localidades del Cuadro 2.1. Los datos que se requieren son:

- Población de las localidades
- La distancia que existe entre todas las localidades

Resulta claro que los requerimientos de información para aplicar un modelo simple de interacción espacial son mínimos. Por lo mismo, los resultados deben ser interpretados con cuidado, como se verá mas adelante.

Para ilustrar la aplicación del modelo se presentarán los cálculos de la interacciones entre los poblaciones Xiquila, Atalco y Tepeolol

- 1 Población de la localidad Xiquila = 1679
- 2 Población de la localidad Tepeolol = 299
- 3 Población de la localidad Atalco = 438
- 4 Distancia de Xiquila a Tepeolol = 3.579280km
- 5 Distancia de Xiquila a Atalco = 0.997235km
6. Distancia de Atalco a Tepeolol = 2.834763km

**TESIS CON
FALLA DE ORIGEN**

Con la información anterior se puede aplicar el modelo de la siguiente forma:

La interacción entre las localidades Xiquila y Tepeolol (I_{ij} , en este caso $i = \text{Xiquila}$ y $j = \text{Tepeolol}$) es igual a la población de Xiquila ($P_i = 1679$ multiplicada por la población de Tepeolol ($P_j = 299$) y dividida entre la distancia que las separa ($d_{ij} = 3\ 579\ 280$). Al sustituir los valores en el modelo se tiene lo siguiente:

$$\text{Interacción entre Xiquila y Tepeolol} = \frac{1679 \times 299}{3.579280} = 140257$$

De la misma manera se definen las interacciones entre Xiquila y Atalco y entre Atalco y Tepeolol. Por lo tanto:

$$\begin{aligned} \text{Interacción entre Xiquila y Atalco} &= \frac{1679 \times 438}{0.997235} = 737441 \\ \text{Interacción entre Atalco y Tepeolol} &= \frac{438 \times 299}{2.834763} = 46198 \end{aligned}$$

Al aplicar el modelo para las demás localidades se tendría el panorama completo de las interacciones y se podría ordenar la información en forma de matriz; se produce así una matriz de interacciones

Los flujos generados a partir de un modelo simple de interacción espacial no pueden ser interpretados por su valor absoluto, sino solamente en términos relativos. Esto quiere decir que los valores numéricos que aparecen en el Cuadro 2.2 (que son los valores absolutos de los flujos entre las distintas localidades) tienen que ser transformados en porcentajes. Es decir, relativizados, la razón de esto es que los modelos simples de interacción espacial expresan la probabilidad de que se establezcan flujos entre las localidades. Por ello, mientras más población exista en las localidades (mientras mayor sea la masa de cada localidad) más probable será que se generen contactos entre las poblaciones de las dos localidades. Por otro lado, mientras más distancia exista entre dos localidades menos probable será que las poblaciones se interrelacionen

Cuadro 2.2: Matriz de interacciones generadas por un modelo simple de interacción espacial.

Nombre	La Mesa de Limantitla	Xiquila	Tepeolol	Atalco	Coxhuaco 2°	Coxhuaco 1°
La Mesa de Limantitla	0	9030596	37284	147025	20621	49871
Xiquila	9030596	0	140257	737441	124690	225003
Tepeolol	37284	140257	0	46198	10001	24999
Atalco	20621	737441	46198	0	44463	47863
Coxhuaco 2°	20621	124690	10001	44463	0	11573
Coxhuaco 1°	49871	225003	24999	47863	11873	0

TESIS CON
FALLA DE ORIGEN

Por lo tanto, si el modelo calcula probabilidades de interrelación, los resultados de su aplicación deben ser expresados en términos de probabilidad. La solución es muy sencilla gracias a las ventajas que ofrecen las matrices para manipular datos y realizar cálculos numéricos. Lo que se requiere para relativizar los valores de la matriz es calcular el valor total de los flujos generados entre los asentamientos considerados. Para esto se realiza lo siguiente. Ver Cuadro 2.3

Cuadro 2.3: Matriz de interacciones, totales por columna, por renglón y valor total de la matriz.

Nombre	La Mesa de Limantitla	Xiquila	Tepeolol	Atlatco	Coxhuaco 2°	Coxhuaco 1°	
La Mesa de Limantitla	0	9030596	37284	147025	20621	49871	9285397
Xiquila	9030596	0	140257	737441	124690	225003	10257987
Tepeolol	37284	140257	0	46198	10001	24999	258739
Atlatco	147025	737441	46198	0	44463	47863	1022990
Coxhuaco 2°	20621	124690	10001	44463	0	11873	211648
Coxhuaco 1°	49871	225003	24999	47863	11873	0	359609
	9285397	10257987	258739	1022990	211648	359609	21396370

1. Se suman los valores de la matriz en sentido horizontal y se obtiene un valor total para cada renglón (por ejemplo, para la columna de Xiquila el valor total es: $9030596 + 0 + 140257 + 737441 + 124690 + 225003 = 10257987$)
2. De igual manera, se suman los valores totales de la matriz en sentido vertical y se obtiene un valor total para cada columna (por ejemplo, para la columna de La Mesa de Limantitla el valor total es: $0 + 9030596 + 37284 + 20621 + 20621 + 49871 = 9158993$).
3. Se suman los valores totales de cada renglón y se obtiene el valor total de todos los renglones de la matriz ($9285397 + 10257987 + 258739 + 1022990 + 211648 + 359609 = 21396370$)
4. Se suman los valores totales de cada columna y se obtiene el valor total de todas las columnas de la matriz ($9285397 + 10257987 + 258739 + 1022990 + 211648 + 359609 = 21396370$)
5. Para probar que los cálculos han sido realizados correctamente se compara la suma de todas las columnas y de todos los renglones (totales generados en el paso 3 y 4), los cuales deben ser iguales ($21396370 = 21396370$). A este valor (21396370) se le llamará "el valor total de la matriz" Ver Cuadro 2.3

TESIS CON
FALLA DE ORIGEN

- 6 Se deben expresar los flujos en términos de probabilidad, por lo que se divide el valor que aparece en la matriz entre el valor total de la matriz (21396370) y se multiplican por 100, Cuadro 2.4. Por lo tanto, el valor total de la matriz relativizada siempre será igual a 100 y representará el total (100%) de los flujos que se generan entre los asentamientos analizados (por ejemplo, el valor total de la matriz se relativizó de la siguiente manera $\frac{2139637}{2139637} \times 100 = 100$)

La información del Cuadro 2.4 se interpreta como la probabilidad de que una localidad tenga contactos con otra. Así, la probabilidad de interacción entre las localidades Tepeolol y Atalco es de 0.21%, mientras entre Xiquila y La Mesa de Limantitla es de 42.20%. Esto quiere decir que de cada 100 personas, bienes y vehículos que se muevan en la región, 0.21 se moverán entre Tepeolol y Atalco y casi 42.20 entre Xiquila y La Mesa de Limantitla.

Cuadro 2.4: Matriz de interacciones, relativizada en función del valor total de la matriz.

Nombre	La Mesa de Limantitla	Xiquila	Tepeolol	Atalco	Coxhuaco 2°	Coxhuaco 1°	
La Mesa de Limantitla	0	42.206206	0.17425385	0.68714927	0.096376161	0.23308159	43.397067
Xiquila	42.206206	0	0.65551774	3.4465706	0.5827624	1.0515943	47.942618
Tepeolol	0.17425385	0.65551774	0	0.21591513	0.046741583	0.11683758	1.2092659
Atalco	0.68714927	3.4465706	0.21591513	0	0.20780628	0.22369682	4.7811381
Coxhuaco 2°	0.096376161	0.5827624	0.046741583	0.20780628	0	0.055490721	0.98917714
Coxhuaco 1°	0.23308159	1.0515943	0.11683758	0.22369682	0.055490721	0	1.680701
	43.397067	47.942618	1.2092659	4.7811381	0.98917714	1.680701	100

Habría cuando menos otras dos maneras de relativizar los valores de la matriz. Si lo que interesa conocer es la probabilidad de que una localidad envíe flujos a otra, lo adecuado sería relativizar los valores de cada renglón respecto al valor total del renglón. El procedimiento es similar al anterior, sólo que en vez de relativizar los valores respecto al valor total de la matriz, se relativizan respecto al de cada renglón. En el Cuadro 2.5 se observa una matriz de este tipo.

TESIS CON
FALLA DE ORIGEN

Cuadro 25: Matriz de interacciones, relativizada en función del valor total de cada renglón.

Nombre	La Mesa de Limantitla	Xiquila	Tepeolol	Atlatco	Coahuaco 2°	Coahuaco 1°	
La Mesa de Limantitla	0	97.255895	0.40153372	1.5834003	0.2220799	0.53709066	100
Xiquila	88.034777	0	1.3672956	7.8894461	1.2155406	2.193442	100
Tepeolol	14.409888	54.207908	0	17.855059	3.8652851	9.66186	100
Atlatco	14.372086	72.086824	4.5159777	0	4.3463768	4.6787359	100
Coahuaco 2°	9.743064	58.913857	4.7252986	21.007994	0	5.6097861	100
Coahuaco 1°	13.86784	62.56879	6.951717	13.309734	3.3016415	0	100
	9285397	10257987	258739	1022990	211648	359609	21396370

Para ejemplificar la manera como se interpretan los resultados de la matriz del Cuadro 25 se tomará el caso de la localidad Xiquila. De acuerdo con la matriz, la probabilidad de que la localidad Xiquila envíe flujos a la localidad Atlatco es de 7.88%, mientras que la probabilidad de que Atlatco envíe flujos a Xiquila es de 72.08%. Esto indica que es probable que Xiquila esté mucho más relacionada con todas las demás poblaciones que éstas entre sí, ya que Xiquila aglomera más del 90% de las relaciones. La otra manera de manipular la matriz sería relativizar los valores respecto al total de cada columna, y la matriz resultante expresaría la probabilidad que tiene cada localidad de recibir flujos de los demás. Según la matriz que se presenta en el Cuadro 26, Xiquila es la comunidad que recibirá más flujos que las demás.

Cuadro 26: Matriz de interacciones, relativizada en función del valor total de cada columna.

Nombre	La Mesa de Limantitla	Xiquila	Tepeolol	Atlatco	Coahuaco 2°	Coahuaco 1°	
La Mesa de Limantitla	0	88.034777	14.409888	14.372086	9.743064	13.86784	9285397
Xiquila	97.255895	0	54.207908	72.086824	58.913857	62.56879	10257987
Tepeolol	0.40153372	1.3672956	0	4.5159777	4.7252986	6.951717	258739
Atlatco	1.5834003	7.8894461	17.855059	0	21.007994	13.309734	1022990
Coahuaco 2°	0.2220799	1.2155406	3.8652851	4.3463768	0	3.3016415	211648
Coahuaco 1°	0.53709066	2.193442	9.66186	4.6787359	5.6097861	0	359609
	100	100	100	100	100	100	100

El modelo de interacción usado en los ejercicios anteriores es uno de los más sencillos que existe sirve para ilustrar la idea general que respalda todos los demás modelos. La idea de que la probabilidad de contactos entre dos unidades espaciales está positivamente relacionada con la masa de cada una y negativamente relacionada con la distancia que las separa es la base de los modelos de interacción espacial.

TESIS CON
FALLA DE ORIGEN

Capítulo 3

Algoritmo de localización *p-mediana*.

El algoritmo de ubicación llamado "*p-mediana*", es un modelo que ha sido utilizado en una gran variedad de aplicaciones, una de ellas es la localización de localidades que proveerán - por ejemplo, estos servicios pueden ser de salud, educación, polo económico, etc. Una parte muy importante que influye en la búsqueda de soluciones al modelo es la parte heurística desarrollada por [Maranzana, 1964] y [Teitz, 1968]

Por lo mismo, el número de soluciones puede ser muy grande, aunque debe reconocerse que éste varía en gran medida en función de la solución inicial que se proponga. Hay que resaltar que antes de utilizar computadoras para resolver los problemas que este tipo de algoritmos aborda, por lo general no era posible incluir una treintena de localidades; en cambio hoy en día se puede incluir un número casi ilimitado de ellas

Para juzgar la conveniencia de usar computadoras en este tipo de situaciones, se debe tomar en cuenta tanto el desempeño del propio equipo de cómputo como la precisión de los resultados. Por ello, en una primera versión se creó un programa en C adecuado para estaciones de trabajo UNIX, mismo que posteriormente fue reescrito en Java para ser utilizado en computadoras personales

Con base a los capítulos anteriores, en este capítulo se presenta la propuesta para atender las necesidades básicas de la población que habita en localidades con menos de 2,500 habitantes, utilizando este modelo de la "*p-mediana*", con el objetivo de fortalecer la justicia distributiva y reducir las desigualdades regionales en donde se aplique este sistema, ubicando la población más desprotegida que por lo general se encuentra en las localidades pequeñas y es en donde se registran los mayores rezagos demográficos, las coberturas más bajas de servicios básicos, la mayor incidencia de la pobreza extrema y la marginación.

Debe destacarse el hecho de que la distribución territorial de las localidades pequeñas determinan en parte las oportunidades efectivas que tiene la población para acceder a servicios básicos que no se ofrecen en su localidad de residencia; quienes viven cerca de las ciudades o de centros urbanos, recorriendo distancias cortas pueden acceder a los servicios básicos como educación, vivienda, telefonía así como a bienes y servicios

TESIS CON
FALLA DE ORIGEN

Con la finalidad de identificar las localidades situadas cerca de las ciudades o de pequeños centros urbanos, las localidades fuera del área de influencia de las ciudades pero cerca de una carretera y las localidades que se encuentran en un aislamiento, es decir fuera del área de influencia de las ciudades y centros urbanos, así como alejadas de las vías de comunicación, se establecieron los siguientes criterios:

- El área de influencia de las ciudades, es decir los asentamientos con 15 mil y más habitantes, se definió con un radio de 5 kilómetros a partir del límite de sus AGEB ¹. Las localidades ahí ubicadas fueron consideradas parte del área de influencia de las ciudades.
- Para estimar las áreas de influencia de los centros urbanos regionales, es decir, asentamientos con un rango de población de 2,500 a 14,999 habitantes, se fijó un radio de 2.5 kilómetros desde el límite de sus AGEB y las localidades ahí ubicadas se consideraron como parte del área de influencia de esos centros urbanos.

Una vez definidas las áreas de influencia urbana, se integró el universo de las localidades cercanas a una vía de comunicación, por lo que se buscó a todas aquellas que estuvieran a 3 kilómetros de una carretera o camino.

El resto de las localidades se consideraron aisladas por que su distancia a las vías de comunicación y los centros urbanos es lo suficientemente grande como para que una persona la recorra caminando.

Como se ha mencionado, el objetivo de aplicar el algoritmo de la "*p*-mediana" es contar con un soporte para fortalecer la justicia distributiva, mediante el mejoramiento de la población con mayores rezagos. Por eso, el universo de trabajo lo constituye el conjunto de localidades con menos de 2,500 habitantes, dispersas a lo largo de las carreteras y caminos revestidos, así como las que se encuentran en situación de aislamiento.

Esta propuesta parte del reconocimiento de que es difícil y oneroso llevar todos los servicios a todas las localidades, además que muchos de los asentamientos son poblaciones de incierta consolidación, razón por la cual no parece prudente emprender obras de infraestructura y equipamiento urbano que suponen cuantiosas inversiones públicas en localidades sumamente pequeñas e inestables. En su lugar, se explora la alternativa en la cual dicha población pueda tener acceso a los servicios básicos, no obstante que en su localidad de residencia se carezca de ellos.

Otro objetivo es sugerir una estrategia que propicie la maximación del gasto social para la atención de la población más rezagada. Para ello, se propone la focalización de acciones, identificando aquellas localidades pequeñas que por su ubicación estratégica y el equipamiento de que ya disponen, con un pequeño incremento en su acervo de servicios básicos, puedan atender las necesidades de las personas que residen en las localidades pequeñas situadas cerca de ellas, es decir dentro de su área de influencia.

¹ Áreas Geoestadísticas Básicas

Para cumplir con estos dos objetivos, se establecieron los siguientes criterios para identificar a las localidades que se consideran como Centros Potenciales Proveedores de Servicios:

- 1 Con la finalidad de identificar asentamientos con cierta consolidación y potencial de desarrollo, se eligieron aquellas localidades que tenían una población entre 100 y 2,499 habitantes.
2. A fin de asegurar accesibilidad del asentamiento a lo largo del año, el Centro Potencial Proveedor de Servicios debía estar ubicado a una distancia máxima de 3 kilómetros de una carretera o camino.
3. Que la localidad elegida tuviera cierta infraestructura y equipamiento básico de un servicio de salud, un servicio de educación (escuela primaria), algún medio de comunicación (al menos servicio de telefonía rural) y una tienda DICONSA ². Debe mencionarse que la existencia de la tienda DICONSA es una garantía de que dicha localidad cuenta con un abasto regular y es accesible por medios terrestres y se tiene cierto consentimiento comunitario de que esa localidad es un centro de abasto.

Los criterios de elegibilidad para las localidades pequeñas en situación de aislamiento fueron los mismos que los requeridos para los cercanos a vías de comunicación, salvo el criterio de distancia de carreteras, considerándose como universo de trabajo sólo las localidades situadas a más de 5 kilómetros de un centro urbano y a más de 3 kilómetros de una vía de comunicación.

A continuación se describirá el algoritmo, el cual comprende la iteración de seis pasos.

Antes de explicar los pasos, hay que recalcar que en una primera etapa se define un peso para cada comunidad, este valor está determinado por su tamaño y su cercanía a una carretera, con este ponderador se califica la capacidad que una comunidad tiene para ser considerada potencialmente proveedora.

Para estar en condiciones de calificar o ponderar las distancias lineales entre las localidades potencialmente proveedoras y las beneficiarias, en el cuadro 3.1, se asignaron los siguientes ponderadores los cuales como se mencionó están en función de su población y su accesibilidad a un camino.

Cuadro 3.1: Ponderadores

	Pob. ≤ 500	Pob. > 500 y Pob. < 1,250	Pob. > 1,250 y Pob. < 2,500
Dist. a camino < 500 mts	3	1	1
Dist. a camino > 500 mts y Dist. a camino < 1500	4	3	2
Dist. a camino > 1500 mts	5	5	3

Continuando con la descripción del algoritmo, los dos primeros pasos se construyen los árboles iniciales y son idénticos a los planteados por [Maranzana, 1964]; el tercer paso

²Distribuidora y Comercializadora Nacional S. A.

**TESIS CON
FALLA DE ORIGEN**

consiste en realizar un pequeño intercambio de raíces. Con estos tres primeros pasos es posible obtener una o varias soluciones óptimas; en los siguientes tres pasos, que en ocasiones pueden ser lentos, sólo se trata de mejorar la solución.

En efecto los tres últimos pasos son independientes entre sí. Por ello pueden utilizarse a discreción del usuario e incluso el diseño del paso seis se puede adecuar libremente, en función de las condiciones o características específicas del problema que se esté abordando.

3.1 El algoritmo

Sea N el conjunto de todas las localidades a considerar en una entidad federativa, incluyendo tanto a los núcleos potenciales como las localidades que recibirán servicios de los primeros. Establezcamos las siguientes definiciones.

Sea V el conjunto de localidades (tanto proveedoras como beneficiarias).

Sea A el conjunto de aristas que comunican a las localidades proveedoras con sus beneficiarias.

A cada localidad x_i (perteneciente a V) se le asocia un peso w_i y a cada par de localidades x_i y x_j se le asocia una arista $ar_{i,j}(x_i, x_j)$ (perteneciente a A) y una distancia $d(x_i, x_j)$, la cual es simétrica y representa la distancia más corta entre estas dos localidades.

El algoritmo consiste en construir p árboles mínimos, T^1, T^2, \dots, T^p , con sus respectivas raíces r^1, r^2, \dots, r^p , las cuales representan a aquellas localidades que se convertirán en Centros Proveedores de Servicios. Se les llama "árboles mínimos" porque cada uno de ellos minimiza la distancia de un conjunto de localidades que, en este caso, son sus hojas con respecto a la raíz.

Supongamos que:

- $V = (x_1, x_2, \dots, x_N)$ son todas las localidades en una entidad federativa dada.
- $T^K = (y_1, y_2, \dots, y_{I_k})$ es el conjunto de localidades del árbol T^K , en donde el elemento r^k es la raíz.
- $P = (r^1, r^2, \dots, r^p)$ es el conjunto raíces, que representan a las localidades que proveerán servicios.
- $I_K = (1, 2, \dots, n)$ indica el número de localidades que pertenecen al árbol T^K correspondiente.
- $J^K(r^k) = \sum_{i \in I^K} j = 1, \dots, I_k = w_i \cdot d(x_j, r^k)$ que es el tránsito mínimo del árbol T^K ; este valor es la contribución de este árbol al definir antes.
- $\min d(x_i, P) = \min d(x_j, r^i)$ es la distancia mínima entre una raíz y una localidad.
- $D^q = \max d(x_i, x_j)$ es la máxima distancia entre cualquier par separado de localidades, dentro del conjunto correspondiente al árbol T^k .

TESIS CON
FALLA DE ORIGEN

3.1.1 Paso A: Inicialización

A1

Con base a los criterios que se establecieron al principio de este capítulo, se obtiene una base de datos que debe contener a todas las localidades que se desean aplicar el algoritmo.

Se escoge un número p de localidades que fungirán como Centros Proveedores de Servicios en la solución inicial del algoritmo. Hay que mencionar que para poder discriminar de las bases de datos aquellas localidades que se considerarán como potencialmente proveedoras de servicios, se deben de cumplir las siguientes características:

- Población entre 100 y hasta 2,499 habitantes
- Distancia máxima de 3 kilómetros o menos de una carretera.
- Disponibilidad de una tienda DICONSA, telefonía rural, centro de salud y al menos una escuela primaria

A2

Después de seleccionar cuáles serán las p localidades que tentativamente conforman la solución inicial, se construye el conjunto $P = (r^1, r^2, \dots, r^p)$ que permite inicializar los árboles respectivos al agregar a cada uno su raíz correspondiente $T^1 = (r^1), T^2 = (r^2), \dots, T^p = (r^p)$

3.1.2 Paso B: Construcción de los árboles

B1

Dado el conjunto de localidades raíz, se define un índice que indica el número de elementos en cada árbol, este índice es un contador que en un principio esta con valor cero para cada árbol:

$$I^1 = 1, I^2 = 1, \dots, I^p = 1$$

B2

Para cada localidad beneficiaria o , lo que es lo mismo, para el conjunto de localidades que recibirá servicios $(V - P)$ se calcula la distancia con respecto a cada una de las raíces $d(x_i, r^j)$ donde $i = 1, \dots, V - P$; $j = 1, \dots, p$

B3

Una vez calculadas las distancias mínimas $\min d(x_i, P)$. Entonces se agregará la localidad al árbol $T^j = T^j + \{x_i\}$ con distancia mínima (d_{\min})

B4

Aumentar el contador de localidades que integran dicho árbol $I^j = I^j + 1$.

B5

Ir al Paso C

3.1.3 Paso C: Redefinición de las raíces para cada árbol

C1

Considerar a todos los elementos de cada árbol $y_i \in T^K, i = 1, \dots, I^K$ y encontrar para cada árbol una localidad que produzca el tránsito mínimo en ese árbol, el cual se define con la siguiente expresión:

$$J^K = \min(J^K[y_i]), i \in I^K$$

Si $y_i \neq \tau^K$, entonces:

$$P = P - \tau^K + y_i \text{ (reemplaza a la raíz).}$$

Regresar al Paso A

C2

De otra forma, si $y_i = \tau^K$ para todos los árboles, ir al Paso D.

3.1.4 Paso D: Intercambio de raíces

Este paso tiene la finalidad de eliminar un árbol, reubicando a las localidades asociadas a él en otro árbol, el cual debe ser el más cercano a cada una de ellas. También se realiza este paso con el fin de dividir un árbol en dos, cuando esta división conlleve algún ahorro en los costos totales

D1

Calcular para cada árbol T^K -el cual debe contener más de una localidad (sin incluir a la raíz)- la reducción en los costos generada al considerar una raíz adicional.

D2

Definir $S = (s_i, s_j) \subset T^K$ como el conjunto de dos vértices contenidos en el árbol T^K que cumplan con:

$J^k(X) = \{\sum_i w(y_i)d(y_i, X)$ en donde $i \in I^k\}$. Siendo X^k el vértice que se escoja del conjunto S y que cumpla con $J^k(X^k) = \min J^k(X)$

$J^k(S) = \sum_i w_i d(y_i, S), i \in I^k$ siendo estas dos localidades las que obtengan el mínimo, por consiguiente la reducción en el costo se obtiene en el árbol T^K es $W^K = J^K(\tau^K) - J^K(S)$

Suponiendo que la mayor reducción en los costos ocurre en el árbol T^s

$$W^s = \max W^k \text{ y } X^s = (y_{s1}, y_{s2}) \subset T^s$$

D3

Calcular para cada árbol $T^K \neq T^s$ el aumento en el costo que resulta al eliminar su raíz

Sea $Q = P - (\tau^K, \tau^s) + X^s$ sea un nuevo subconjunto de raíces en donde

$$L^K = \left\{ \sum_i w_i \cdot d(y_i, Q), i \in I^K \right\}$$

El aumento mayor en el costo puede ocurrir en el árbol T^m siempre y cuando:

$$L^m = \min(L^k - J^k(r^K))$$

D4

El intercambio de las raíces entre los árboles T^s y T^m sucede si $W^s > L^m$. En caso de que se dé esta situación, el árbol T^m se elimina y el T^s se divide en dos, redefiniendo así el conjunto $P = P - (r^m - r^s)$ y se regresará al Paso A con este nuevo conjunto de árboles y raíces.

Si no se elimina ni divide ningún árbol, entonces pasar al Paso E.

3.1.5 Paso E: Atracción de una raíz por otro árbol

Para todos los árboles T^K y T^s que tengan más de un vértice y que cumplan con la siguiente condición:

$$d(r^K, r^s) \leq D^K + 2D^s$$

Considérese a todas las localidades $y_i \in T^s$ y trate de encontrarse un par en $y_i \in T^K$ que cumpla con la siguiente condición:

$$d(y_{s(i)}, y_i) \leq d(y_{s(i)}, r^s)$$

Si se encuentra un par que cumpla esta condición, entonces deberá verificarse que:

$$J^K(x_i) + w(x_i)d(y_{s(i)}, x_i) - w(y_{s(i)})d(y_{s(i)}, r^s) < J^K(r^K)$$

Si se cumple la anterior condición, entonces x_i será la nueva raíz del árbol T^i y $y_{s(i)}$ será absorbida por el árbol T^i .

Se regresa al Paso A, si no sucede lo anterior, pasar al Paso F.

3.1.6 Paso F: Unión de dos raíces en un solo árbol

Considérese toda localidad que pertenezca a los árboles T^K y T^s que cumpla con la siguiente condición:

$$\min \{w(r^K) \cdot d(r^K, r^s), w(r^s) \cdot d(r^K, r^s)\} < J^K(r^K) + J^s(r^s)$$

Esta condición considera la unión de los árboles T^K y T^S , que puede resolverse mediante la enumeración completa del problema de la 2-mediana para encontrar dos raíces. Si éstas son iguales a r^K y r^S entonces reemplazar las raíces viejas por las nuevas y regresar al Paso A

Si se trata de las mismas, es decir si $\{t^R, t^S\} = \{r^R, r^S\}$ entonces ésta es la mejor solución que se puede encontrar.

Opcionalmente se puede realizar un paso adicional si se quieren comparar tres árboles y sus respectivas raíces. El procedimiento es análogo al descrito aquí

3.2 Análisis de resultados

Después de ejecutar los primeros cuatro pasos se obtienen tres valores con los cuales nos permite determinar cuando la solución es óptima, el **costo máximo** (la suma de las distancias) es el valor más grande que se obtiene del conjunto de árboles a los que se les ha incorporado una nueva localidad proveedora, por eso cuando se dice que el costo máximo sucedió en determinado árbol, significa que es el árbol en donde se recorre mayor distancia.

Y por el contrario, la **transición mínima** (suma de distancias) es el valor más pequeño que se obtiene del conjunto de árboles

El **mayor incremento en el costo** indica que cuando se ha eliminado alguna raíz, la distancia en algunos árboles se incrementa, este valor indica en qué árbol se produce el mayor incremento

Para consultar el listado de los resultados, ir al Anexo I de esta tesis.

3.3 Estudio del caso del estado de Hidalgo.

El algoritmo de la p -mediana se aplicó en el estado de Hidalgo utilizando distintos criterios con el propósito de establecer cuales eran los mejores. A partir de estos ejercicios se pudo confirmar la hipótesis a partir de las cuales se formuló la propuesta de que un pequeño incremento en los acervos de los servicios de algunas localidades situadas estratégicamente podrían ser un medio adecuado para atender los núcleos de población rural geográficamente dispersos.

Al inicio, el universo que compone a todas las poblaciones que componen a Hidalgo se dividió en dos, uno en donde se establecían aquellas localidades con 5 mil y más habitantes y otro con localidades con menos de 5 mil habitantes, de este último se escogieron aquellas localidades consideradas que tienen alta y muy alta marginación. Hay que aclarar que en un principio se incorporó este índice que mide la proporción de la población analfabeta de una localidad, el número de viviendas en malas condiciones y de la población ocupada en el sector primario, además de otras variables como el tipo de piso de las viviendas, drenaje, agua entubada o el acceso al fluido eléctrico. Este criterio se eliminó cuando se

realizó un estudio de campo y se visitaron cuatro localidades de la Huasteca Hidalguense identificadas como Centros Potenciales de Provisión de Servicios Estas fueron: Los Otates, Jaltocán, Xiquila y Toctitlán, así como algunas de las localidades que se ubican en sus respectivas áreas de influencia.

Esta visita tuvo los siguientes objetivos:

- Verificar que la información en las bases de datos sobre la existencia de servicios básicos, tiendas DICONSA, telefonía rural, carreteras pavimentadas y terracería recubierta y distancias fueran consistentes con lo observado
- Observar las condiciones espaciales y de pertenencia de las áreas de influencia de los Centros Proveedores de Servicios propuestos

Por lo general, se observó que la información socioeconómica de las bases de datos son confiables, en cuanto al acervo de servicios de educación y salud, como a la existencia de tiendas DICONSA, servicio de telefonía rural. La constatación de que existía este tipo de servicios también validó la pertinencia y eficacia de los criterios que se utilizaron para definir el universo de trabajo, es decir las características de las localidades elegibles como Centros Potenciales Desarrollo Social, es de esta experiencia que evidenció la idea que la alta y muy alta marginación no era una medida conveniente para incorporarse como atributo de un Centro Proveedor, debido a que el universo de trabajo de campo es el medio rural el índice de marginación de la población dedicada a la agricultura no aportaba mucho. Asimismo, aún cuando el índice de marginación por lo regular sí muestra los rezagos importantes en las condiciones de vida y permite establecer órdenes de prioridades de atención, sostenerlo como criterio de inclusión significaba excluir a localidades de marginación media que al incrementar su acervo de servicios, podría convertirse en un Centro Proveedor.

Otro ajuste derivado del trabajo de campo es la redefinición del rango-tamaño de las localidades que deben considerarse para definir el universo de aquellas que serán beneficiarias y las atendidas; ya que al constatar que el volumen de población de un asentamiento le está correlacionado una cierta dotación de servicios básicos y equipamientos urbanos debido a que la densidad de las localidades propicia la economía de la región. En particular, el trabajo de campo reveló que la madurez de los asentamientos de entre 2,500 y 4,999 habitantes es mayor de la pensada, pues las condiciones de vida que ofrece a sus habitantes, aún muy alejadas de las ciudades, son sensiblemente diferentes a las localidades con menos de 2,500 pobladores. Con ello, el universo de trabajo se redujo. en el límite superior el rango fue situado en 2,499 habitantes y en el inferior a 500, en un principio el límite inferior era 100

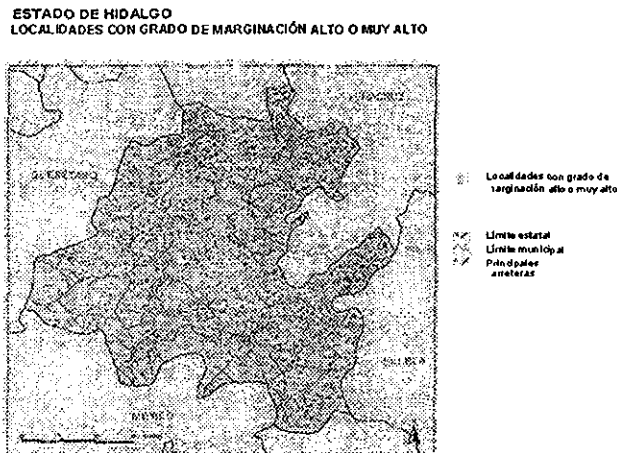
3.3.1 Criterios

Los criterios para seleccionar las localidades proveedoras en el estado de Hidalgo, en principio fueron definidos en términos de las siguientes características:

- Poblaciones con alta o muy alta marginación.
- Comunidades con una población mayor a 500 y menor a 5 mil habitantes.
- Con una cercanía a una carretera menor a 1 km.
- Que cuenten con servicio de DICONSA
- Con servicio de comunicación (teléfono o telégrafo)
- Con al menos una escuela

En la figura 3 1 se muestra a las localidades del estado de Hidalgo que tienen la más alta marginación

Figura 3 1:



A las comunidades que es tengan alta o muy alta marginación, con una población menor a 5 mil y mayor a 100 habitantes, pero que no cuenten con los servicios antes mencionados, pero que están a menos de 2 5 km de las comunidades proveedoras, se les considera cómo beneficiarias. Por lo que en un primer ejercicio se consideraron a 127 en donde 22 comunidades son las que proveerán servicios y 105 son las beneficiarias.

**TESIS CON
FALLA DE ORIGEN**

3.3.2 Ponderadores

En este primer ejercicio para estar en condiciones de "calificar" o ponderar las distancias lineales entre las localidades potencialmente proveedoras y las potencialmente beneficiarias, se asignaron los siguientes ponderadores para las localidades con más de 2,500 habitantes

1. Para las comunidades proveedoras, las cuales cumplen con el siguiente perfil tener una población = 100 y = 2,500, tengan DICONSA, una primaria y al menos un servicio de salud.
2. Para las comunidades beneficiarias que cumplan con población = 2,500 y población = 1,250 y distancia a ciudad = 100 mts
3. Para las comunidades beneficiarias que cumplan con población = 1,250 y población = 500 y distancia a ciudad = 100 mts
4. Para las comunidades beneficiarias que cumplan con población = 500 y distancia a ciudad = 100 mts.
5. Para las comunidades beneficiarias que cumplan con población = 2,500 y población = 1,250 y distancia a ciudad = 1 km y distancia a carretera = 100 mts
6. Para las comunidades beneficiarias que cumplan con población = 1,250 y población = 500 y distancia a ciudad = 1 km y distancia a carretera = 100 mts
7. Para las comunidades beneficiarias que cumplan con población = 500 y distancia a ciudad = 1 km y distancia a carretera = 100 mts.
8. Para las comunidades beneficiarias que cumplan con población = 2,500 y población = 1250 y distancia a carretera = 1 km
9. Para las comunidades beneficiarias que cumplan con población = 1,250 y población = 500 y distancia a carretera = 1 km
10. Para las comunidades beneficiarias que cumplan con población = 500 y distancia a carretera = 1 km

Posteriormente, al observar que cuando se aplicaba el algoritmo en dos universos contenidos en un mismo estado, se modificaron los criterios para designar por sus características a una localidad como Centro Proveedor de Servicios, estos ponderadores fueron los siguientes:

- 1 para las comunidades proveedoras y también para las comunidades beneficiarias con más fácil acceso, esto es, que estén sobre un camino o carretera y a menos de 1 km de cualquiera de estos.

TESIS CON
FALLA DE ORIGEN

- 3 para aquellas comunidades con acceso regular, esto es, que estén sobre un camino y este se encuentre a una distancia $1 = d = 1.5$ km
- 5 para las comunidades de difícil acceso, esto es, no se encuentra sobre camino (terracería) y se encuentren a una distancia de $1.5 = d = 2.5$ km

Después de correr el algoritmo, el número de comunidades proveedoras se redujo a 16, pero con este resultado se observó que se dejaba sin atención un gran área geográfica, por lo que se cambió el criterio de inclusión tanto para las beneficiarias como las proveedoras. Se excluyó el criterio de distancia a carretera y sólo se consideraron a todas aquellas comunidades que tienen una población entre 100 y 2,500 personas, así como una distancia mayor a 5 km de una población.

En la figura 3 2 se muestra a las 16 localidades del estado de Hidalgo, como se puede observar la dispersión de las localidades es muy grande.

Figura 3 2:



En los Cuadros 3 2 y 3 3 se muestran los resultados de las localidades proveedoras y sus beneficiarias de este ejercicio.

En las figuras 3 3 y 3 4 se muestra información socio-económica local de dos localidades proveedoras de servicios en el estado de Hidalgo

Una vez que al separar en dos conjuntos las localidades y se corría el algoritmo de la *p-mediana* de forma separada para cada uno, se observó que cuando una localidad beneficiaria buscaba cual raíz era la más cercana para formar parte de dicho árbol, al

TESIS CON
FALLA DE ORIGEN

Cuadro 3.2: Comunidades en el estado de Hidalgo

Arbol		Clave	Nombre localidad	Dist a carretera	Población
1	1	130230081	Santa Cruz	1710	1828
1		130230232	Tlapani	3897	74
1		130230114	Achichipil	2913	164
1		130230173	Chachaixpa	3621	125
1		130230036	Cuatcomete	1600	487
1		130230070	Palzoquico	1177	426
1		130230016	Axihuiyo	614	225
1	2	130230022	Cecocapa	1260	565
2		130230189	Topetate	622	261
2		130230018	Calmecate	990	469
2		130230064	Los Otates	117	918
2		130230041	Chalahuyapa	369	923
2		130230079	San Antonio	1470	280
2		130230032	Huitzschahuati	3080	520
3	3	130230093	Opexicitla	116	1340
3		130230071	Panacaxtlan	1860	1044
3		130230227	Las palomitas	1204	0
3		130230042	Chilico	224	2521
3		130230134	Zapotitla	189	305
3		130230191	La Curva Tlaltzintla	90	59
3		130230210	Palzoquiapa	190	247
3		130230066	Oxtomal 1o	3	1653
3		130230226	Oxtomal 2o	51	45
3		130230053	Hitzquilititla	1443	368
3		130230008	Aguacatitla	1907	600
4	4	130180025	Palo Semita	543	502
4		130180035	Santa Ana de Allende	1	1513
4		130180048	Zacate Grande	95	605
4		130180100	La Reforma de Palo semita	150	282
4		130180106	Barrio Arriba Ojo de Agua	2064	123
4		130180076	El Hualul	498	59
4		130180032	Rancho Nuevo	58	60
4		130180019	La Loma	2042	72
4		130180101	Zacatito	372	45
4		130180073	El Chinillar	2782	16
4		130180022	El Cocote	326	160
4		130180037	Saucillo Agua Nueva	2084	391
5	5	130230103	Xiquila	1445	1679
5		130230002	Atalco	14	438
5		130230194	Coxhuaco 1o	27	323
5		130230082	Coxhuaco 2o	169	83
5		130230137	Topol	160	299
5		130180063	La Mesa de Lamentitla	1729	562
6	6	130310011	Los Duranos	394	524
6		130310037	La Joya de Santa Maria	1482	45
7	7	130110036	Atlaltipa Huitzoatlaco	107	297
7		130110054	El suspiro	1668	0
7		130110045	Tlahica	1121	101
7		130110063	Zapopitl	1632	0
7		130110011	Coyolapa	1468	314
7		130110023	Palo Gordo	126	201
7		130110038	Plan Huateca	496	194
7		130110003	Atencuapa	2246	369
7		130110016	Huitzoatlaco	408	501
7		130110025	Pochoica	277	36
7		130110028	La reforma	309	28
8	8	130230023	San Francisco	2109	1145
8		130230049	Ahuahitla	1209	157
8		130230021	Pichicampa	4084	439
8		130230050	Congreso	1257	133
8		130230033	Tepeanaxac	2020	123
8		130230009	Comala	1416	91
9	9	130730047	Toctitlan	1978	630
9		130730109	Cuahuilacate	410	50
9		130730039	Tocontla	2068	109
9		130730039	Tlahuelongo	2176	125
9		130730036	San salvador	989	399
9		130730043	Tepeyac	1456	121
9		130730005	Amantlaol	3742	630
10	10	130620002	Acocxatlan	1737	1046
10		130620150	Xoyoxco	3703	0
10		130620087	Zacatlamaya	1979	30
10		130620047	Minala	143	32
10		130620002	Acayuca	2040	100

TESIS CON FALLA DE ORIGEN

Cuadro 3 3: Continuación de las comunidades en el estado de Hidalgo

Arbol		Clave	Nombre localidad	Dist a carretera	Población
11	11	130140003	Ahuacatlan	563	1195
11		130140016	Chiquila	1639	352
11		130140047	Cuaneetla	45	0
11		130140046	Tecomatla	210	0
11		130140041	Ixcatlan	9	0
11		130140096	Molhuasco	47	0
11		130140042	Atezo	19	20
11		130140104	Zacatepec	131	0
11		130140045	Tlalca	99	0
11		130140092	Cuasala	85	0
11		130140048	Tecointepe	278	0
11		130140039	Contia	547	0
11		130140051	Cohuaxmila	59	0
11		130140094	Ploochititla	239	0
11		130140082	Tepeatlaco	1136	0
12	12	130340013	Chalma	186	485
12		130340020	Ixtlahuaco	37	1054
12		130340009	Contepe	138	122
12		130340030	Tlaconsintla	142	95
12		130340020	Tenango	126	292
13	13	130370129	Tlaxco el viejo	708	124
13		130370069	San Antonio Tlaxco	1012	201
13		130370054	Tlaxco	1807	594
13		130370019	Yateco	208	0
13		130370017	Huayateco	65	222
13		130370110	Huiloco	84	0
13		130370102	Teasco	482	80
14	14	130060033	San Pablo Oxtotipan	619	517
14		130170015	San Juan el Sabino	794	866
15	15	130030026	Zacacuatlá	223	1727
16	16	130100010	Enchillos	1704	797
16		130100058	Rancho Guadalupe	3705	0
16		130100016	El Milagro San Simon	650	87
16		130100023	Totoapa el Grande	138	545
16		130100025	Totopita (La herradura)	1653	135

momento de graficar los resultados en el mapa, se observó que entre la localidad raíz y la beneficiaria se encontraban núcleos urbanos importantes, por lo que lógicamente, en la vida real si a una persona se le dice que en lugar de ir a el lugar más cercano y se le indica otro que no lo es, nunca va a ir

Actualmente cuando se utiliza el algoritmo de la *p*-mediana, se hace uso de la base de datos completa de la zona a estudiar, incluyendo tanto a ciudades como a poblaciones que cuentan con una o dos viviendas. Hay que aclarar, que de esta base de datos primero se obtienen todas las localidades que tengan menos de 2,500 habitantes y estén a menos de 1.5 kilómetros de una carretera o camino.

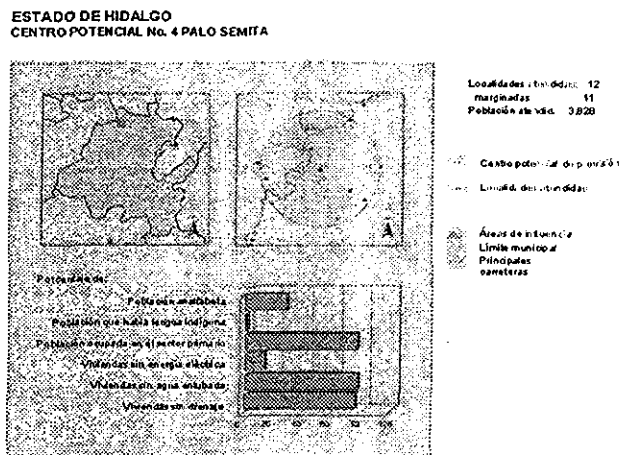
Es por esta razón que se modificaron los ponderadores que se muestran en el cuadro 3 1 y los criterios que se utilizaron para calificar cuáles localidades son las potencialmente proveedoras de tal manera que los nuevos criterios fueron los siguientes:

- Ubicación a más de 5 km. de una ciudad o de una localidad con más de 2,500 habitantes.
- Contar con presencia de DICONSA.
- Tener servicio de telefonía rural; al menos una escuela primaria; al menos algún servicio de salud, ya fuese éste de la Secretaría de Salud o de IMSS ³-Solidaridad

³Instituto Mexicano del Seguro Social



Figura 3 3:

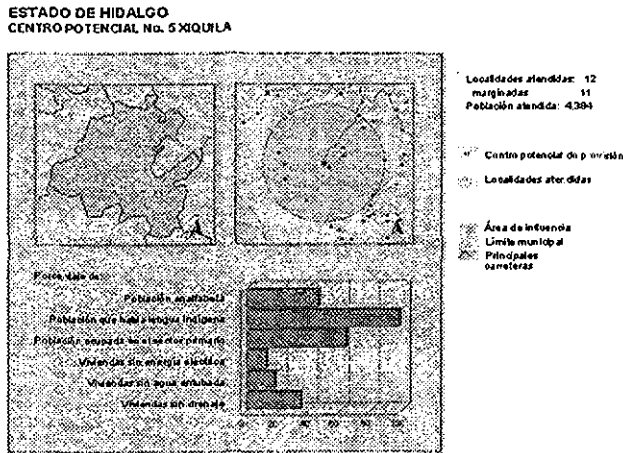


El total de comunidades en el estado de Hidalgo que se consideraron para efecto de este estudio fue de 4,463; en donde 147 localidades son las que proveerán servicios y 4,316 las que los recibirán.

La población total que se atendería con esta solución del modelo es de 2,107,835 personas.

TESIS CON
FALLA DE ORIGEN

Figura 3.4:



3.4 Estudio del caso del estado de México.

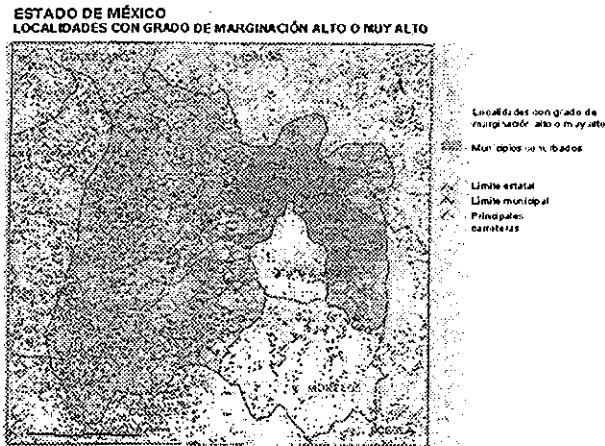
Para el caso del estado de México el total de localidades que se contemplaron en el estudio fue de 219, en donde 45 localidades inicialmente se contemplaron como centros potencialmente proveedores y 174 localidades fueron las beneficiarias, cada uno de estas localidades atiende entre 0 y 13 localidades además de la propia localidad proveedora; la población total que esta solución del modelo atiende es de 164,714 habitantes. En la figura 3.5 se muestra el universo total de las localidades que se contemplaron en este estudio.

En la segunda iteración se construyeron 34 árboles y cada uno de ellos atiende entre 0 y 14 localidades, además de la propia raíz al finalizar esta iteración y ya no se elimina ninguna comunidad proveedora; por lo que este resultado es óptimo. La población total que se atendería con esta solución del modelo es de 164,714 pobladores. En la figura 3.6 se indica la ubicación de los 34 localidades que fungirán como Centros Proveedores de Servicios.

Por la cercanía geográfica y características del perfil, 11 de las 45 localidades inicialmente proveedoras fueron reasignadas en esta segunda iteración del modelo y los árboles fueron reordenados. El total de localidades que se contemplaron en el estudio es pues de 219, en donde 34 localidades son los Centros Proveedores de Servicios y 185 son las beneficiarias.

A continuación, en la figura 3.7 y en la figura 3.8 se muestra información socio

Figura 3 5:

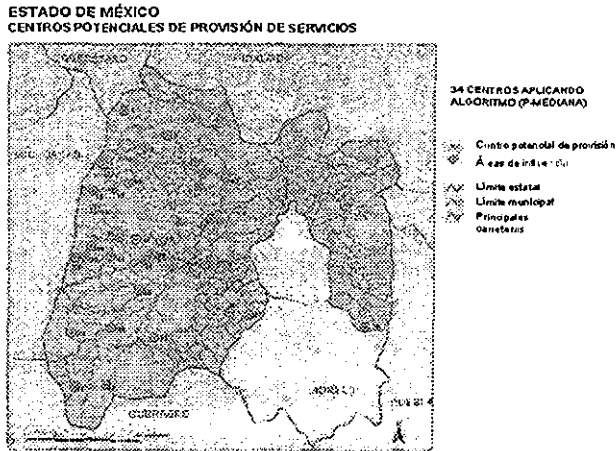


económica local del Centro Proveedor de Servicios en el estado de México, San Bartolo y la figura 3.8 de San Agustín Poteje

El valor máximo de la suma de las distancias a recorrer en todo el conjunto de árboles asciende a 1.379711 y se encuentra asociado al árbol 26, lo que significa que es en este árbol en donde se recorre la mayor distancia. Cuando se elimina alguna raíz normalmente se incrementa la distancia a recorrer en algunos árboles considerados. El mayor incremento a la distancia a recorrer ascendió a 7.408006 y estuvo asociado al árbol 27.

TESIS CON
FALLA DE ORIGEN

Figura 3 6:



3.5 Estudio del caso del estado de Querétaro.

Con los criterios y los ponderadores ya establecidos se aplicó el algoritmo de la (*p*-mediana) al estado de Querétaro en donde 38 comunidades son las que proveerán servicios. En la figura 3.9 se muestra a las localidades del estado de Querétaro que tienen la más alta marginación y en la figura 3.10 se muestra la ubicación de los Centros Proveedores de Servicios.

Información socio-económica local del Centro Proveedor de Servicios en el estado de Querétaro, San Nicolás de la Torre.

TESIS CON
FALLA DE ORIGEN

Figura 3 7:

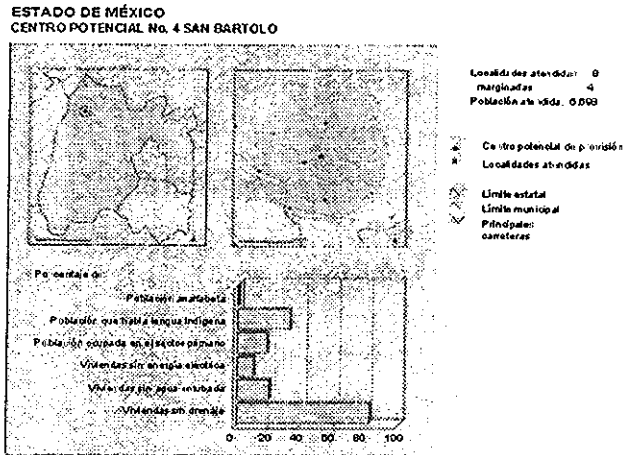
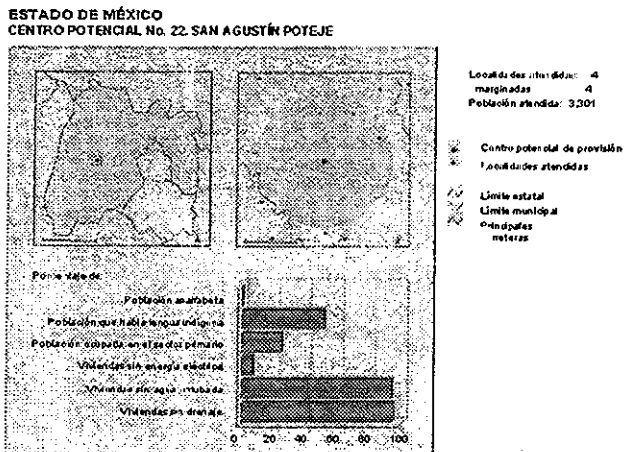


Figura 3 8:



TESIS CON
FALLA DE ORIGEN

Figura 3 9:

Querétaro: Grado de marginación alto y muy alto, por localidad, 1995

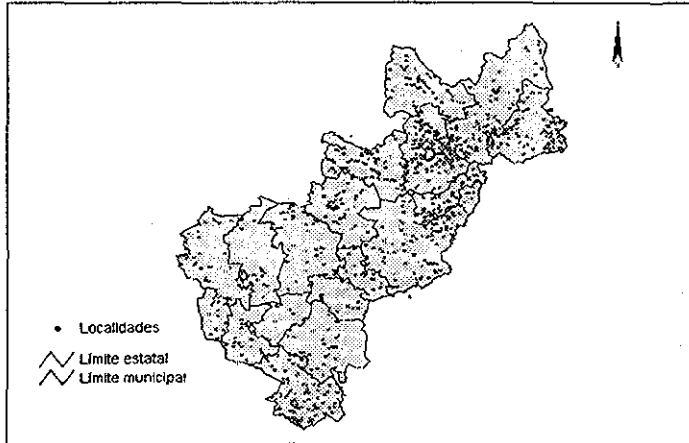
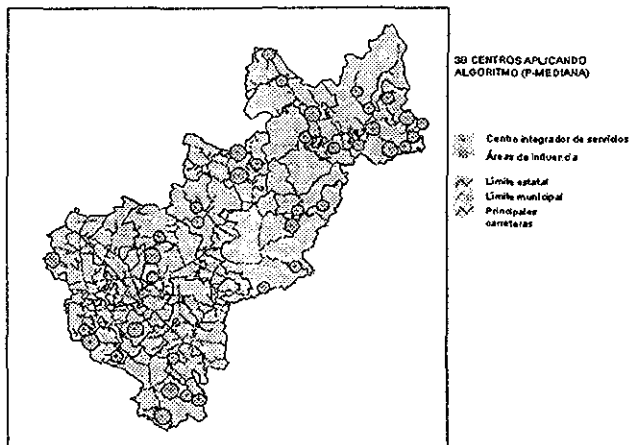


Figura 3 10:

Estado de Querétaro
Centros Integradores de servicios cercanos a carreteras



TESIS CON
FALLA DE ORIGEN

Figura 3 11:

Querétaro. Centro cercano a carretera, San Nicolás de la Torre, Amealco de Bonfil

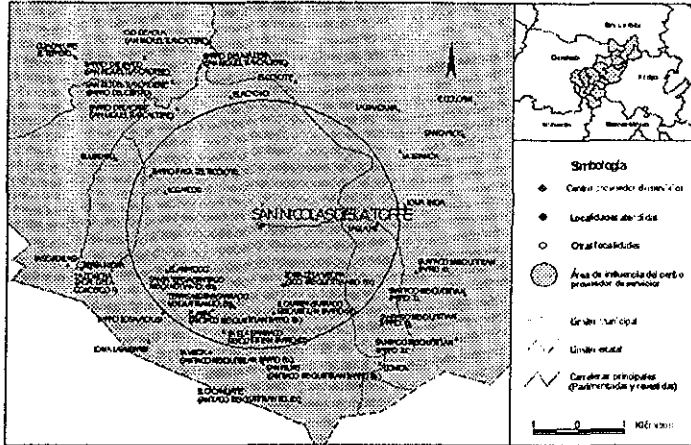
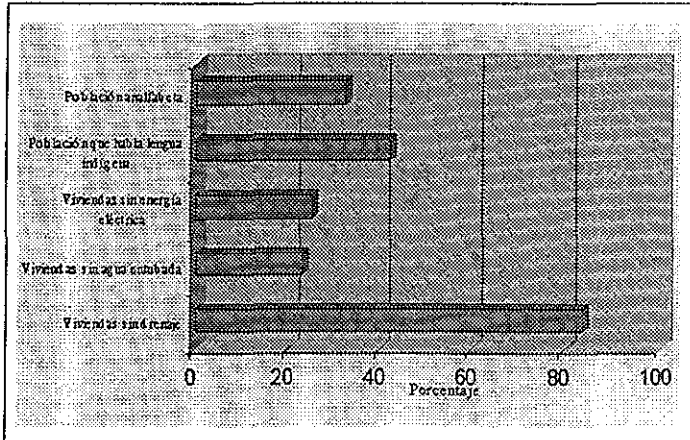


Figura 3 12:

Principales indicadores del Centro Integrador de Servicios San Nicolás de la Torre



TESIS CON FALLA DE ORIGEN

Capítulo 4

Desarrollo de un sistema para desplegar resultados.

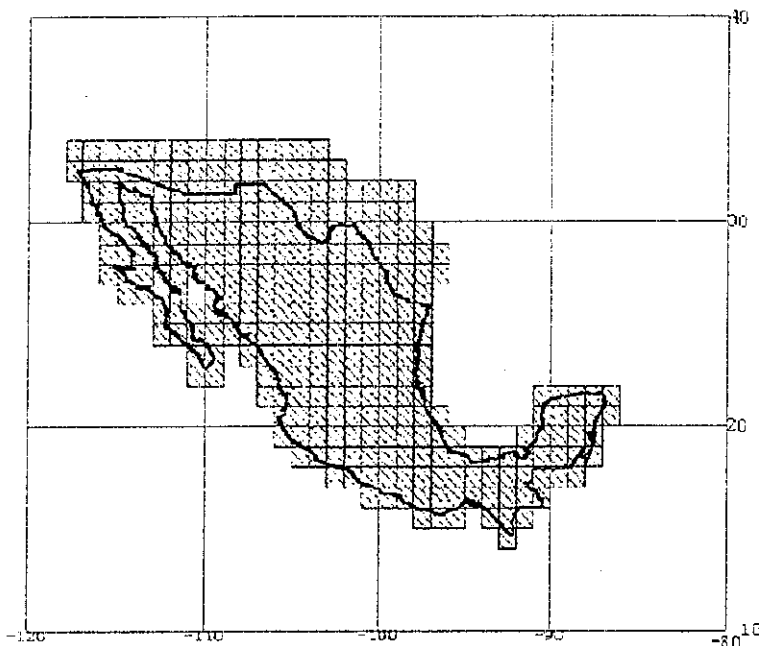
En este capítulo se explica cómo se desarrolló el programa para manejar información topográfica de una manera gráfica. Para lograr esto lo primero que se hizo fue obtener la información topográfica de la República Mexicana.

Los datos originales están constituidos por arreglos de alturas sobre el nivel del mar que corresponden a puntos en coordenadas geográficas obtenidos cada tres segundos de arco y que cubren todo el territorio nacional. Estos datos se obtuvieron de un disco compacto [GEMA, 1995] publicado por el *INEGI* que contiene 255 archivos que contienen un grado cuadrado cada uno y que se elaboraron a partir de las cartas topográficas escala 1 : 250,000, también del *INEGI*. En la figura 4.1 se muestran los 255 cuadrantes geográficos en que se dividió a la República para representar la topografía en forma digital.

A este tipo de archivos se les conoce como Modelos de Elevación Digital (MED), que como se menciona en el párrafo anterior, contiene las elevaciones de un terreno de una zona del globo terráqueo representadas en una malla dividida en intervalos. Estos intervalos siempre deben estar referenciados con algún sistema de coordenadas geográficas. Los más usuales son: el basado en latitud-longitud y el Universal Transversa de Mercator (*UTM*). Entre más densa sea la malla, se podrá obtener un mayor detalle por que sólo se cuenta con el valor de z (valor de elevación). Para calcular la posición que ocupa cada valor en la Tierra, se debe inferir desde la esquina inferior izquierda del archivo, que como se verá más tarde se obtiene del nombre de cada uno de los archivos. Los MED pueden contener información civil tales como vías de comunicación y edificios, por eso hay que aclarar que no son imágenes obtenidas de un escáner ni mucho menos mapas de *bits*, tampoco contiene contornos de elevaciones. Únicamente contiene el valor promedio de la elevación (media ponderada).

En cada archivo se tiene la información de un grado de latitud por un grado de longitud en una matriz de 1,442,401 alturas del terreno. Las coordenadas geográficas de cada elevación se infieren por su posición dentro de la matriz.

Figura 4 1: División de la República en 255 cuadrantes geográficos.



Los datos se recuperaron del disco compacto mediante un programa escrito *ad hoc* que los decodifica, los ordena y rota de tal manera que las columnas representan direcciones norte-sur y los renglones este-oeste. Con estas transformaciones elementales se corrige el problema que tiene el programa que provee el INEGI, en el que las imágenes que se obtienen son imágenes espejo del terreno real.

4.1 Proceso preliminar

Para sistematizar la identificación de archivos, se creó un arreglo de caracteres en donde los nombres de los archivos que contienen los datos topográficos indican la latitud y la longitud separadas por "n" y "o" que indican norte y oeste respectivamente.

Así, los nombres de los archivos indican la latitud y la longitud a partir de las cuales se tiene la secuencia de puntos en coordenadas geográficas a los que corresponde la altura sobre el nivel del mar. Estos se puede considerar que son valores asociados a una malla regular, la figura 4 2 ilustra los nodos (alturas) que conforman la malla.

Para crear la malla se deben especificar las dimensiones iniciales de un arreglo, que

TESIS CON
FALLA DE ORIGEN

para comenzar, debe ser, como mínimo, de $1,201 \times 1,201$ elementos, para reservar el espacio de memoria suficiente para almacenar un grado cuadrado. Posteriormente se escogen la longitud y la latitud de la zona que se quiere estudiar y se comienza la lectura de los datos.

Para evitar la limitación de representar solamente un grado cuadrado, se creó un algoritmo que durante la lectura reasigna la dimensión de los arreglos para poder acceder a archivos de regiones vecinas y así poder construir regiones mayores que las correspondientes a un grado cuadrado o bien regiones que se encuentran en la intersección de meridianos y paralelos que delimitan grados cuadrados contiguos.

El problema de la reasignación de memoria y por tanto de las dimensiones de los arreglos es uno cuando las regiones a unir colindan en el sentido horizontal (longitud) y otro cuando lo hacen en el sentido vertical (latitud). La malla es entonces de $n \times m$ elementos, en donde: $(n, m) \geq 1,201$ y n corresponde a la longitud y m la latitud.

Esto último se ilustra en la figura 4.3 en donde se muestran diferentes zonas contenidas en diferentes archivos. Como se puede observar la zona global está constituida por el alineamiento adecuado de las subzonas que la componen.

La manera de resolver este problema es la siguiente: Se crean dos ciclos, en donde el interior representa la longitud y el exterior la latitud. Ambos ciclos van desde 0 hasta la dimensiones máximas de longitud y latitud. Estas dos dimensiones pueden ser diferentes y se calculan dependiendo de las dimensiones de la zona que se quiere recuperar.

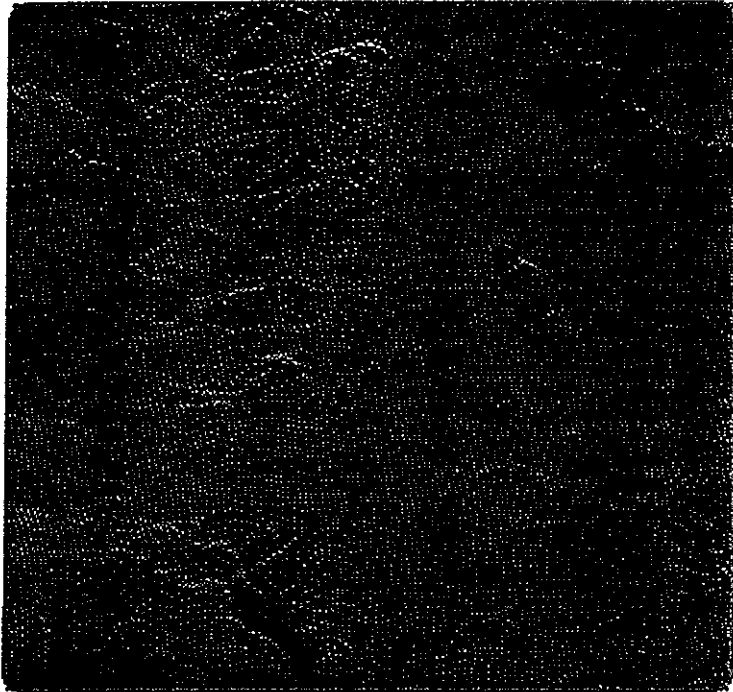
Para unir dos mallas en longitud, el problema no es complicado, puesto que basta leer el primer archivo y, a continuación, leer el que le sigue, por ejemplo, si se lee el archivo 19n100o.dat, a continuación se debe leer el archivo 19n99o.dat. El problema surge cuando se quieren unir dos mallas contiguas en latitud, ya que primero hay que leer 1,201 campos del primer archivo, e inmediatamente después leer del archivo en el que continúa esa región hacia el norte o hacia el sur, con otros 1,201 campos, para posteriormente regresar a leer otros 1,201 campos del primer archivo y repetir el proceso hasta agotar la extensión en latitud. Si se quiere ver una zona que involucre más de tres archivos, obviamente primero hay que leer 1,201 campos del primero, después 1,201 del segundo y por último 1,201 del tercero, hasta terminar el proceso de integración de la imagen.

Después de arreglar el orden en que se accederá a los archivos, se diseñó un algoritmo para que a cada altura le asignara su localización geográfica mundial, esto es, a todos y cada uno de los datos topográficos se les dió una localización tanto en longitud como en latitud en grados, minutos y segundos, de esta forma, es posible obtener después zonas más pequeñas.

Así, al tener identificados todos los datos, únicamente se necesita escoger el área de interés y representarlos gráficamente. En la figura 4.4 se muestra la imagen completa que resulta de unir todos los archivos de la figura 3.10. Esto además facilita, como es natural, la manipulación de la imagen.

Para obtener subconjuntos de la malla creada con el procedimiento anterior, se usa una

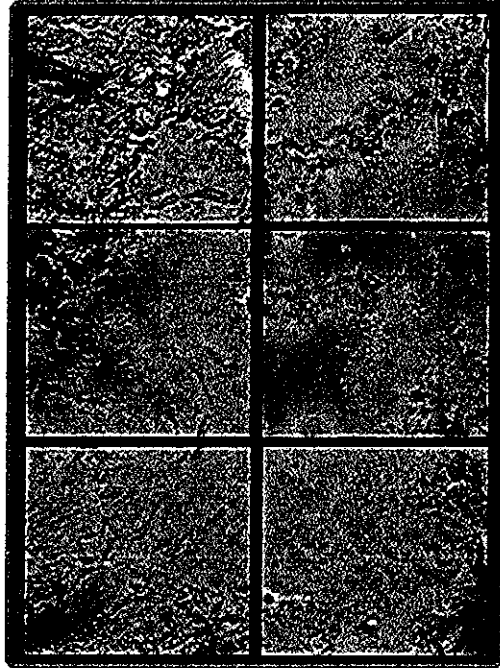
Figura 4.2: Imagen que muestra las alturas representadas por los puntos de colores, con que se construye la malla. La región que se observa comprende al Pico de Orizaba y a la Caldera de los Humeros en Veracruz.



rutina que cambia el número de elementos de la malla a partir de submuestras de datos. Esto es, extrae un elemento n del conjunto a lo largo de cada dimensión, en donde n es el valor obtenido por medio de un parámetro que se proporciona. Esta técnica preserva el aspecto de la relación de los datos de entrada, ya que la salida es de la misma dimensión que la de entrada, pero el número de elementos en cada dimensión se reduce. Para ilustrar esto, en la figura 4.5 se muestra cómo es reducido un conjunto. Cada elemento es representado por un punto y únicamente los puntos encerrados en círculos serán los que se envíen como salida a los subsiguientes módulos del programa.

La orientación en que están las imágenes es la usual: El norte hacia arriba y el este hacia la derecha. Las elevaciones almacenadas en un registro dado corresponden a puntos sobre el mismo meridiano, ordenados de sur a norte: así, cada registro representa un perfil del terreno, orientado de sur a norte. Los registros se encuentran ordenados de oeste a

Figura 4.3: Imágenes que se usan para integrar una región muy extensa a partir de regiones menores.



este. Todas las elevaciones se representan con enteros binarios de 16 bits, justificados a la derecha con el bit de orden más alto indicando el signo. Los valores permitidos están en el intervalo $[-32767, +32767]$ metros sobre el nivel del mar. Los valores desconocidos se indican con todos los bits prendidos. Los valores negativos se ignoran.

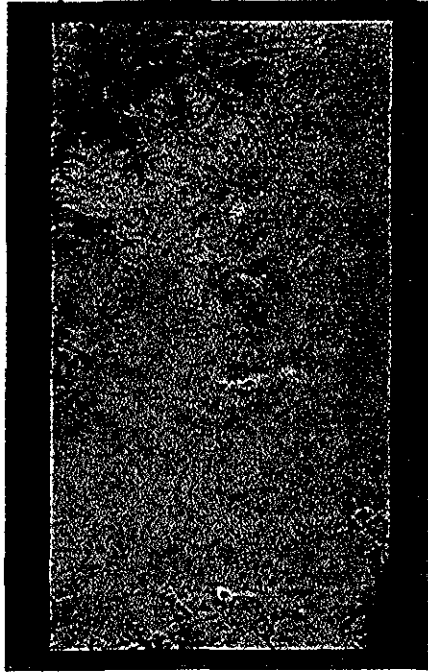
4.2 Representación digital

Debido a que la Tierra es aproximadamente esférica, cualquier representación de ésta en un plano, genera una distorsión, que en zonas muy pequeñas es insignificante, pero cuando se trata de regiones muy vastas como obtiene una gran distorsión. Es por eso, que durante varios siglos, se ha trabajado para representar la superficie curva de la Tierra en mapas planos por medio de varios esquemas, conocidos con el nombre de proyecciones para mapas. Cada una de estas proyecciones tiene ventajas y desventajas y su utilización depende de los propósitos que se persiguen.

Para convertir las coordenadas geográficas de las alturas del terreno a una forma:

TESIS CON
FALLA DE ORIGEN

Figura 4.4: Imagen completa a partir de todos los archivos ya unidos



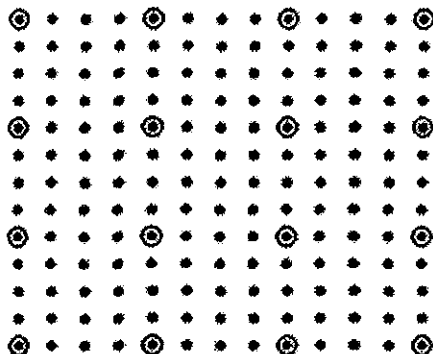
adecuada, se utilizó la representación Universal Transversa de Mercator (*UTM*). Lleva el nombre de transversa porque el orden de los datos dentro de la malla es en forma ascendente de sur a norte y de este a oeste a lo largo del ecuador. Con esta representación es posible realizar mediciones de distancias, calcular áreas, etcétera.

La proyección *UTM* es un modelo matemático que se utiliza para inferir posiciones en la superficie curva de la Tierra sobre una superficie plana. Se forma proyectando las coordenadas de la Tierra sobre un cilindro tangente a la Tierra en un meridiano de longitud. Es una proyección conforme, lo que implica que se mantiene la forma de áreas pequeñas. Los únicos parámetros necesarios para definir esta proyección son la longitud del meridiano central y la latitud de referencia, aunque también pueden especificarse un factor de escala, el falso este, y el falso norte. A medida que uno se aleja del meridiano central, la distorsión aumenta. Esta proyección se utiliza para minimizar la distorsión que se produce en mapas de áreas que se extienden de norte a sur.

La representación gráfica de los datos requiere que se mapee un grado cuadrado a una área cuadrangular. En virtud de que los datos están originalmente en coordenadas geográficas, las distancias entre dos puntos sobre dos meridianos dados son menores en el

TESIS CON
FALLA DE ORIGEN

Figura 4 5: Obtención de subconjuntos de datos.



hemisferio norte y mayores en el hemisferio sur.

La conversión de coordenadas se hace sobre cada grado cuadrado por separado, utilizando las siguientes expresiones:

$$x = \frac{111276.1806 \cos \phi \Delta \lambda}{(1 - 0.0067686579 \text{sen}^2 \phi)^{1/2}} + \frac{5.649 \cos^3 \phi (\Delta \lambda)^3 (1 - \tan^2 \phi + 0.006815 \cos^2 \phi)}{(1 - 0.0067686579 \text{sen}^2 \phi)^{1/2}} \quad (4.1)$$

$$y = 6332500.489(0.0175424666A - 0.0025601 \text{sen}A + 0.0000027 \text{sen}4A) \quad (4.2)$$

Donde:

$$A = \phi + (0.0087861165 \tan \phi \cos^2 \phi (\Delta \lambda)^2 (1 - 0.0067686579 \text{sen}^2 \phi))$$

En donde λ y ϕ son las coordenadas geográficas del punto a transformar, medidas en grados y λ_0 es la longitud del meridiano central de la región. Con estas expresiones se obtiene finalmente $x_e = fa + x$ si el punto está al este del meridiano central o $x_o = fa - x$ si se encuentra al oeste. En estas últimas expresiones $fa = 500,000$ es la falsa abscisa del meridiano central. Para una discusión más detallada de estas transformaciones se puede consultar por ejemplo la referencia [Bribiesca, 1993] que se cita al final.

Después de la transformación de coordenadas de los elementos que se desean visualizar se construye una malla de puntos (x, y) que corresponde a la localización de las alturas que se leyeron previamente.

**TESIS CON
FALLA DE ORIGEN**

Los coeficientes que se encuentran en la ecuación 4.1 y la ecuación 4.2 corresponden a constantes establecidas por el cambio de coordenadas. Utilizando las ecuaciones 4.1 y 4.2 cualquier punto definido por sus coordenadas geográficas es transformado a coordenadas *UTM*.

4.3 Despliegue gráfico

El siguiente paso, que se tiene que dar después de leer todos los datos, es crear la malla y transformarla en una superficie para finalmente desplegarla en una imagen, esto se logra utilizando *Java3D*

El Java3D API es una interface para escribir programas que desplieguen gráficas tridimensionales además de interactuar con ellas. Java3D es una extensión del *Java 2 JDK*. El API contiene un conjunto de constructores de alto nivel para crear y manipular geometrías en 3D, así como de las técnicas necesarias para dibujar la geometría en una imagen. Java3D contiene las funciones necesarias para crear imágenes, visualizaciones, animaciones y aplicaciones con objetos gráficos interactivos.

4.3.1 ¿Que es Java3D?

El Java3D API es un conjunto de clases que sirven como interface para dibujar gráficas sofisticadas en tercera dimensión utilizando sonido, si se desea.

El programador puede utilizar constructores de alto nivel para crear y manipular las geometrías que definen a los objetos tridimensionales. Estos objetos geométricos se localizan dentro de un universo virtual que sirve para dibujar los objetos en la pantalla de la computadora, generando una imagen. El diseño de este API se puede adecuar para que se puedan crear espacios virtuales con una gran variedad de tamaños por lo que se pueden definir dimensiones galácticas o subatómicas.

Este API realiza los detalles de dibujo (*render*) de una forma automática, además de que toma ventaja del procesamiento en paralelo, por eso el dibujo en Java3D se puede optimizar. Otra posibilidad es la de optimizar automáticamente el despliegue de los objetos que conforman a la escena.

Un programa en Java3D crea instancias de los objetos que componen a la escena, utilizando una estructura conocida como Grafo de Escena. El grafo de escena es el método que se utiliza para ordenar los objetos que forman parte de un escenario, y se representa por medio una estructura de árbol en donde se muestra cual es el contenido del universo virtual y cómo será dibujado.

Un programa escrito en Java3D se puede ejecutar como una aplicación o un *applet*, siempre y cuando se tenga un navegador capaz de desplegar programas en Java3D.

4.3.2 El API de Java3D

Todo programa en Java3D es un conjunto de objetos ordenados en base a una serie de jerarquías basada en la herencia. Esta colección de objetos describen a un espacio virtual que es la guía para dibujarlos en una imagen posteriormente. Este API contiene aproximadamente 100 clases contenidas dentro del paquete *javax.media.j3d*. A estas clases por lo general se les conoce como el núcleo de Java3D.

Aunque existen cientos de variables y métodos dentro de las clases de Java3D, en el universo virtual se pueden crear animaciones utilizando unas pocas clases. Además de las clases de Java3D se utilizarán las clases del paquete *com.sun.j3d.utils* conocidas como las utilerías de Java3D, ya que en estas se encuentran las clases de más bajo nivel, por lo que son necesarias para cualquier programa en Java3D.

Estas utilerías tienen cuatro categorías: contenedores, constructores de escena, clases geométricas y utilerías. Estas utilerías reducen significativamente el número de líneas de código, ya que además se hace uso de las clases contenidas en el *java.awt* y *javax.vecmath*. El paquete *java.awt* define el *Abstract Windowing Toolkit* (AWT) que sirve para crear la ventana en donde se despliega la imagen. El paquete *javax.vecmath* define una clase matemática para realizar operaciones con puntos, vectores, matrices y otras funciones matemáticas.

El término de objeto visual se utiliza para referir a un objeto en el grafo de la escena (un cubo o una esfera por ejemplo). Por el contrario, el término objeto, se utiliza únicamente para cuando se hacen instancias de una clase. El término contenedor se utiliza para aquellos objetos visuales que pertenecen a un grafo de escena.

Construir una gráfica de escena

El universo virtual se crea utilizando una grafo de escena. Este grafo de escena se construye gracias a la correcta utilización de las clases de Java3D, ya que se van incorporando poco a poco las geometrías, los sonidos, las luces, la ubicación, la orientación y la apariencia visual de los objetos. Una definición básica de un grafo de escena, es que es una estructura de datos, representada por medio de nodos y arcos. Un nodo es un dato y un arco es la relación que existe entre los datos. Los nodos en un grafo de escena define a las instancias que se utilizan en las clases de Java3D. Los arcos representan a las dos posibles relaciones que existen entre las instancias de Java3D.

La relación más común es la de padre-hijo. Un nodo grupo puede tener una gran cantidad de hijos pero únicamente un padre. Un nodo hoja puede tener un padre pero ningún hijo. La otra relación es la referencia, esta asocia a un objeto *NodoComponente* con un nodo de la gráfica de escena. Un objeto *NodoComponente* define los atributos de la geometría así como los atributos utilizados para dibujar a los objetos visuales.

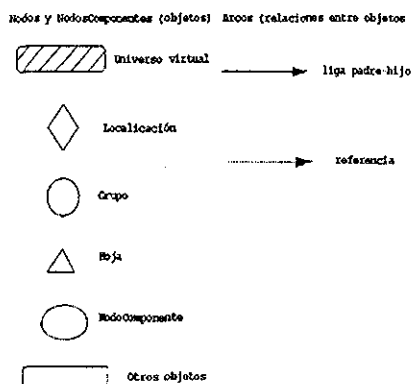
Un grafo de escena se va construyendo como un grafo acíclico-directo (*directed-acyclic graph*, *DAG*) compuesto de nodos y arcos. En un grafo directo los arcos tienen una sola dirección, por lo que un grafo acíclico-directo tiene un solo sentido esto significa que los

ciclos no están permitidos. Se comienza con un nodo y la dirección del grafo no puede regresar al mismo nodo.

Existe una sola ruta desde la raíz del DAG hacia cada una de las hojas, a esta ruta se le conoce con el nombre de, ruta del grafo de escena. El significado de cada ruta radica en que indica el estado en que se encuentra cada elemento de la escena, representado por medio de una hoja, este estado se refiere a la ubicación, orientación y tamaño de cada objeto. Cuando se plasman los objetos en la imagen, Java3D puede tomar ventaja de esta jerarquía, lo que permite que el despliegue de los objetos sea más ordenado. Por eso hay que señalar, que el programador no tiene control sobre el orden en que se dibujarán los objetos.

La representación gráfica de el grafo de escena sirve para diseñar tanto el programa como la documentación, un ejemplo de esto es la figura 4.6 en donde se muestran los símbolos que se utilizan para construir cualquier grafo de escena.

Figura 4.6: Símbolos que representan los objetos de un grafo de escena.



Cada uno de los símbolos que están a la izquierda de la figura 4.6 representan a un objeto que será usado en el grafo de escena. Los dos primeros símbolos representan a dos clases muy específicas: *VirtualUniverse* y *Locale*. Los tres siguientes son los que representan a las clases *Group*, *Leaf* y *NodeComponent* utilizados para indicar las subclases de un objeto determinado. El último se para representar a cualquier otra clase de objeto.

La flecha sólida representa la relación padre-hijo entre dos objetos, la flecha punteada indica la referencia con otro objeto. Pero con estas referencias en muy fácil construir una gráfica de escena ilegal, el único peligro que se corre, es que el programa puede compilar pero no dibujar nada.

Cuando un programa en Java3D define una escena ilegal, el sistema mismo detecta el problema reportando una excepción, el programa puede seguir corriendo sin desplegar

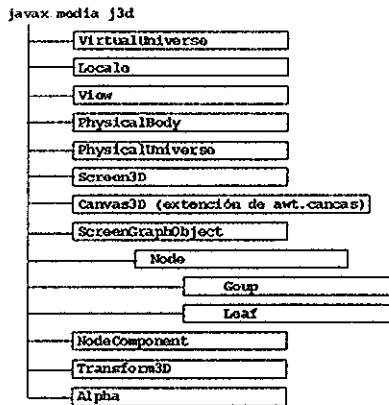
TESIS CON
 FALLA DE ORIGEN

nada, por lo que es necesario detenerlo.

Clasificación de la herencia en las clases alto nivel

Podemos hacer un resumen de los tres niveles del árbol de clases que aparece en la figura 4.7. Las clases *VirtualUniverse*, *Locale*, *Group*, y *Leaf* aparecen en un mismo nivel, el resto del grafo de escena está compuesto de objetos *SceneGraphObjects*. Esta clase *SceneGraphObjects* es la superclase que contiene a todas las clases del tipo *Core* y *Utility*.

Figura 4.7: Resumen de la clasificación de la herencia en las clases de alto nivel de Java3D



De *SceneGraphObject* derivan dos clases más *Node* y *NodeComponent*, la primera contiene la mayor parte de los objetos que componen al grafo de escena, un objeto *Node* puede ser tanto un nodo *Group* o un objeto *Leaf*.

Clase *Node*

Es una superclase abstracta que contiene a las clases *Group* y *Leaf*. La clase *Node* define a algunos de los métodos más importantes, así como a sus clases y son precisamente estas las que componen al grafo de escena.

Clase *Group*

Es una superclase que se utiliza para indicar tanto la ubicación como la orientación de los objetos visuales en el espacio virtual. Dos de las clases que le pertenecen son la *BranchGroup* y la *TransformGroup*. En la representación gráfica del grafo de escena el círculo representa a la clase *Group* y tiene las letras BG como siglas.

Clase *Leaf*

Es una superclase utilizada para indicar tanto la forma, el sonido y el comportamiento de los objetos visuales en el universo virtual. Algunas de las subclasses de *Leaf* son

TESIS CON
 FALLA DE ORIGEN

Shape3D, *Light*, *Behavior* y *Sound*. Estos objetos no pueden tener hijos pero pueden referenciar a otros *NodeComponents*

Clase *NodeComponent*

Es una superclase que especifica la geometría, la apariencia, la textura y las propiedades de los materiales que se utilizarán en el nodo *Shape3D NodeComponents* no forma parte del grafo de escena pero es referenciado por el. Un objeto *NodeComponent* puede referenciarse por más de un objeto *Shape3D*.

4.3.3 Pasos para escribir programas en Java3D

Las subclases de *SceneGraphObject* son bloques que se van conjuntando dentro del grafo de escena. Los siete pasos básicos para desarrollar un programa en Java3D son:

- Crear un objeto *Canvas3D*
- Crear un objeto *VirtualUniverse*
- Crear un objeto *Locale* y agregarlo al objeto *VirtualUniverse*
- Construir un grafo de escena

Crear un objeto *View*

Crear un objeto *ViewPlatform*

Crear un objeto *PhysicalBody*

Crear un objeto *PhysicalEnvironment*

Añadir los objetos *ViewPlatform*, *PhysicalBody*, *PhysicalEnvironment* y *Canvas3D* al objeto *View*

- Construir el contenido de una rama(s) grupo.
- Compilar la(s) rama(s) grupo
- Insertar las subgráficas dentro de *Locale*

Estos pasos ignoran algunos detalles pero se ilustra el concepto de forma general los elementos que todo programa en Java3D debe incluir. Ya que si se siguen estos pasos, la ventaja que se tiene, es que en un futuro, la estructura de los programas será muy similar, además de que al utilizar la clase *VirtualUniverse* reduce significativamente el tiempo y el esfuerzo que se emplea para programar; sin embargo, no permite incluir varias vistas del universo virtual

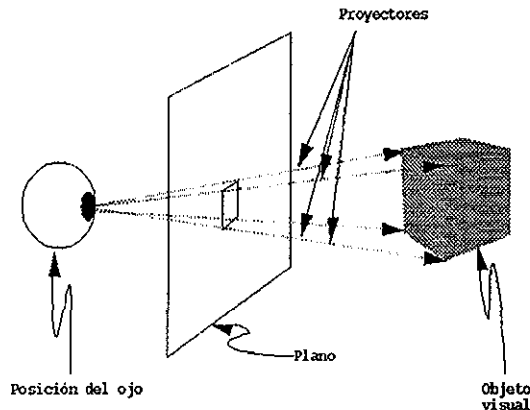
El objeto *SimpleUniverse* crea una vista completa del grafo de escena. Esta vista del grafo de escena incluye a una plano, que consiste en un rectángulo en donde se proyecta el

contenido en forma de imagen. El objeto *Canvas3D* es el vínculo para que dicha imagen se dibuje en una ventana de la pantalla de la computadora.

En la figura 4.8 se muestra la relación que tiene este plano con la posición del ojo (virtual) y el universo virtual, la posición del ojo se encuentra detrás del plano y los objetos en frente de este, se van dibujando en él. El dibujo se realiza proyectando los objetos en el plano, para ilustrar esta idea, en la figura 4.8 se muestran cuatro proyectores.

Por conveniencia, el plano se coloca en el centro del universo virtual, la orientación que se tiene es la siguiente; el eje de las positivas de la *Z* esta orientado en dirección a el usuario (es la profundidad), el eje de las *X* es horizontal y sus valores positivos están a la derecha; el eje de las *Y* es vertical al centro del plano y sus valores positivos están orientados hacia arriba, por consecuencia el punto $(0,0,0)$ es el centro del plano.

Figura 4.8: Ejemplo conceptual del dibujo del contenido del universo virtual en el plano con respecto a la posición del ojo u observador



Un programa en Java3D mueve la parte posterior del punto de vista con respecto a la ubicación del origen. El objeto *SimpleUniverse* pertenece a la clase *ViewingPlatform*, la que a su vez contiene al método *setNominalViewingTransform* que es el que define la posición del ojo, por conveniencia esta corresponde a $(0,0,2.41)$ y se encuentra observando en dirección al eje negativo de las *Z*, partiendo del origen.

Después de crear los objetos *Canvas3D* y *SimpleUniverse*, el siguiente paso es la creación del contenido del grafo de escena. Tanto su estructura como su contenido por lo general varía de un programa a otro, lo que significa que no se puede crear una clase que defina el contenido de cualquier universo virtual.

A continuación se crea el contenido de un grafo de escena, se incorpora en el universo utilizando el método *addBranchGraph* contenido en el *SimpleUniverse*, el método *addBranchGraph* hace una instancia a *BranchGroup*, que a su vez se añade como hijo.

TESIS CON
FALLA DE ORIGEN

utilizando un objeto *Locale* creado por el *SimpleUniverse*.

Algunos conceptos de Java3D

Antes de entrar de lleno a la creación de un grafo de escena, es necesario establecer dos conceptos, estos son "vivo" y "compilado". En el momento que se inserta un grafo de escena en *Locale*, se dice que esta vivo y en consecuencia, cada uno de los objetos que esta en el grafo de escena vuelve a la vida, esto nos permite realizar lo siguiente

Hacer que los objetos vivos puedan dibujarse inmediatamente, sin embargo sus parámetros no se pueden alterar, ya que pierden la capacidad de modificar su "comportamiento"

Por otro lado, los objetos del *BranchGroup* se pueden compilar, esto significa que Java3D ordena el contenido del objeto *BranchGroup* de tal manera que se establece todos y cada uno de sus ancestros haciendo más eficiente el dibujo de la escena. Se recomienda mucho utilizar esta técnica como un paso previo al de dibujarlos. Es mejor compilar únicamente los objetos del tipo *BranchGroup* e insertarlos posteriormente en los de tipo *Locale*

Hay que hacer la observación, de que esto no "comienza el dibujo" de la escena. El dibujo en Java3D se puede definir como un ciclo infinito que va dibujando el contenido del grafo de escena, esto se realiza en el momento que se tiene una instancia del tipo *View*. Una vez que se ha comenzado el dibujo en Java3D se siguen las siguientes operaciones

Proceso conceptual de dibujar una escena en Java 3D

```
mientras(sea_verdad)
{
    Proceso de entrada
    SI(peticion de salida) romper
    Ejecutar comportamiento (animacion )
    Recorrer el grafo de escena y dibujar los objetos
}
Limpiar y salir
```

Un programa en Java 3D por lo general comienza definiendo una clase que se extiende de *Applet*, los programas en Java3D se pueden escribir para que se ejecuten como aplicaciones, pero al utilizar la clase *Applet*, ayuda a que el despliegue en ventanas sea más fácil y rápido.

La clase principal en un programa de Java3D define un método en donde ir construyendo el contenido del grafo de escena, en nuestro caso se proporciona este método, el cual tiene el nombre de *SimuladorDeVuelo()*. Como se mencionó, esta clase deriva de *Applet*, esto no impide que se pueda ejecutar como aplicación ya se hace uso de la clase *MainFrame*, con esta clase se puede hacer uso de las ventanas del AWT ¹ que permiten a

¹Abstract Windowing Toolkit

un *Applet* ejecutarse como aplicación, también se puede indicar el tamaño de la ventana en el momento de construir la clase

Existe una serie de clases de Java3D que son indispensables para crear cualquier aplicación que utilice este API.

Clase *BranchGroup*

Los objetos definidos con esta clase se utilizan para definir a los grafos de escena. Las instancias de *BranchGroup* son las raíces, a partir de donde se generan las demás subgráficas, es por esta razón que este tipo de objetos son los únicos que pueden ser hijos de los objetos *Locale*, aunque pueden tener cuantos hijos se desee. Estos hijos así mismo pueden ser objetos del tipo *Group* o *Leaf*.

Las instancias que se realizan con estos objetos sirven para definir la raíz del grafo de escena, siendo el punto de partida de todas las ramas. Estos objetos son los únicos que pueden insertarse en los objetos tipo *Locale*.

Clase *Canvas3D*

Esta clase se deriva de la clase *Canvas* perteneciente al *Abstract Windowing Toolkit* (AWT). Un objeto *Canvas3D* se puede tomar como el punto de observación para ver la ramificación del grafo de escena.

Construye e inicializa un nuevo objeto *Canvas3D* para que Java3D lo reconozca como válido permitiendo hacer el dibujo en la pantalla de la computadora. Recibe un objeto del tipo *GraphicsConfiguration* que es una extensión de la clase AWT *Canvas*.

Clase *Transform3D*

Este tipo de objetos representan las transformaciones que se aplican a los objetos geométricos en el espacio 3D, tales como, translación y rotación. Por lo general estos objetos solamente se utilizan para crear objetos de tipo *TransformGroup*. Un objeto *Transform3D* se puede construir al combinar varios objetos *Transform3D*. Entonces un objeto *TransformGroup* se puede construir al utilizar más de un objeto *Transform3D*. Su principal función es construir un objeto que representa a la matriz identidad, esto le permite aplicar distintas transformaciones en un objeto, translación, rotación y escala. Cuando se trabaja la rotación, los ángulos se deben expresar en radianes; una rotación completa es 2π , una manera fácil para definir los ángulos, es utilizar la constante *Math.PI*, o bien escribir el valor directamente en radianes. Algunas aproximaciones básicas son: 45° es 0.785, 90° son 1.57 y 180° es 3.14.

Clase *TransformGroup*

Es una subclase de *Group*, las instancias tipo *TransformGroup* se utilizan para la creación de los grafos de escena definiendo a los conjuntos de objetos que tienen nodos padre-hijo. Los objetos *TransformGroup* tienen el control de las transformaciones geométricas. Una transformación se realiza por medio de un objeto *Transform3D*, su principal función es indicar las transformaciones en el grafo de escena. Las transformaciones se almacenan en un objeto *Transform3D*, que posteriormente se transmiten a un objeto del tipo *TransformGroup* en el instante mismo en que éste se crea, o utilizando el método *setTransform()*.

Clase *Vector3f*

Es una clase matemática del paquete *javax.vecmath* que sirve para crear un vector de dimensión tres, ya que son los tres valores necesarios para indicar las transformaciones de las geometrías. Los objetos *Vector3f* no se utilizan de forma directa para la construcción del grafo de escena. Únicamente se utilizan para indicar traslaciones, normales, etc., estos elementos almacenan las coordenadas (x,y,z) de las transformaciones.

El grafo de escena que define a un programa en Java3D se puede utilizar para dibujar los objetos de la escena en forma inmediata, sin embargo esto no es muy eficiente, por lo que se necesita utilizar un método eficiente para recorrer la organización del universo virtual, en Java3D existen dos formas para lograrlo, la primera es compilar cada rama del grafo de escena. La otra es insertar directamente cada rama en el universo virtual, la que a continuación se comentará es la primera opción.

El objeto *BranchGroup* tiene un método que realiza esta compilación, que en otras palabras es recorrer el contenido de cada rama del grafo de escena para ir incorporando los elementos a la escena, esto permite obtener un despliegue más eficiente.

Otra de las posibles optimizaciones, es la combinación a lo largo del grafo de escena, de los *TransformGroups* en uno solo. Por ejemplo, si se tienen dos objetos *TransformGroup* ubicados en una misma rama, estos dos se pueden representar como uno solo.

Una vez que una rama se vuelve "viva" o "compilada", el sistema de dibujo de Java3D convierte esta rama en una estructura interna más eficiente con lo que se acelera el dibujo, otra ventaja que se tiene, es que muchas de las transformaciones se realizan internamente, como por ejemplo, el combinar una transformación con objetos gráficos en un grafo de escena, nos permite obtener una escena en donde el dibujo en la ventana tiene movimiento.

Existen ciertos casos, en donde se necesitan modificar algunos parámetros en el grafo de escena una vez que se han dibujado los objetos, y así se puede aplicar movimiento a los objetos que se están dibujando en la ventana de una forma muy fácil, para que esto suceda, se cuenta con una serie de parámetros a los que se les da el nombre de *capabilities*. Por lo tanto, un objeto *SceneGraphObject* tiene una serie de bits que determinan que "capacidad" está habilitada en un objeto, para que se pueda modificar después que se ha dibujado los objetos.

Para hacer que los objetos que se dibujan en la ventana tengan movimiento, se debe utilizar la clase *Behavior*, que permite interactuar con los objetos visuales. Este "comportamiento" puede variar para cada objeto y el programador puede definir con antelación un número predeterminado de "comportamientos" y una vez que se especifica a que objeto se le va a aplicar, en forma automática, el propio Java3D realiza las adecuadas transformaciones de ubicación, orientación, color y demás atributos.

Hay que señalar que la animación es distinta a la interacción, ya que la primera se ejecuta como respuesta al paso del tiempo, y la segunda esta condicionada a lo que el usuario haga.

Cada objeto visual que esta en el universo virtual puede tener su propio comportamiento, o por el contrario depender de más de uno. Para indicar estos "comportamientos",

el programador debe de crear distintos objetos, para cada uno de estos, añadirlos al grafo de escena correspondiente e inmediatamente después añadir el objeto visual para que establezcan las referencias apropiadas a cada objeto.

Hay que indicar, que en caso de que un universo virtual contenga muchos objetos con movimiento se necesita más poder de cómputo para realizar todos los cálculos, de otra manera se degradará el dibujo.

Para tratar de resolver este problema, Java3D permite crear una frontera espacial para indicar en donde dicho comportamiento se llevará a cabo. A esta frontera o limite se le conoce como región de acción. Este comportamiento no se lleva a cabo o se activa hasta que un *ViewPlatform activation volume* intersecta la región que define un objeto tipo *Behavior*, estas regiones permiten a Java3D ser más eficiente en el momento de que el universo virtual contiene mucho movimiento.

A continuación se enumeran los pasos que se deben seguir para que la escena tenga movimiento.

1. Crear un *TransformGroup*
2. Especificar la región de acción
3. Se emparenta objeto tipo *Behavior* como hijo del *TransformGroup*

Región de acción

Como se mencionó, cada objeto *behavior* tiene su radio de acción, por lo que se utiliza el método *setSchedulingBounds* perteneciente a la clase *Behavior*. Existen varias alternativas para indicar este campo de acción, el más sencillo, es crear un objeto *BoundingSphere*, pero hay que señalar la existencia de objetos *BoundingBox* y *BoundingPolitype*.

El método *setScheduling* delimita la región que los objetos del tipo *Behavior* utilizan como área de influencia, misma que a su vez está delimitada por una frontera que se puede ser definida.

Clase *BoundingSphere*

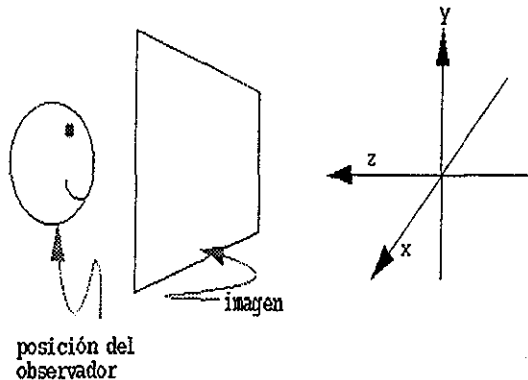
Creará una esfera en que define el campo de acción. Los parámetros que se utilizan, son para indicar el origen y el radio; la posición de esta clase está predeterminada en el origen (0,0,0). El radio es muy importante ya que debe ser lo bastante grande para contener al objeto visual.

4.3.4 El Sistema de Coordenadas en el Universo Virtual

Una instancia de la clase *VirtualUniverse* marca la raíz del grafo de escena en cualquier programa de Java3D, a este espacio virtual también lo podemos definir como el espacio tridimensional en donde residen todos los objetos. Un objeto *Locale* en un universo virtual, es el que establece el sistema de coordenadas cartesiano para ese universo, se pueden tener varios objetos *Locale* en un mismo universo.

La función principal de los objetos *Locale*, es servir de punto de referencia para los objetos que están contenidos en el universo virtual, este sistema de coordenadas, tiene una orientación de la mano derecha (*right-handed*). La dirección del eje de las X es positivo hacia la derecha, el de las Y es hacia arriba y el de la Z esta orientado hacia el usuario (profundidad), las unidades que se utilizan son metros. La figura 4.9 ilustra cual es la orientación del universo virtual con respecto al observador

Figura 4.9: Orientación del universo virtual con respecto al observador



Definiciones básicas de los objetos visuales

Un nodo del tipo *Shape3D* en un grafo de escena, sirve para indicar que hay un objeto visual *Shape3D* es uno de las subclases de la clase *Leaf*, por lo que este tipo de objetos únicamente pueden ser hojas en el árbol del grafo de escena. Otra cosa que hay que tomar en cuenta, es que este tipo de objetos no tiene información acerca la forma o el color del objeto. Esa información se guarda en los objetos *NodeComponent* que después son referenciados por los objetos *Shape3D*. Un objeto *Shape3D* también puede referenciar a otro del tipo *Geometry* y *Appearance*.

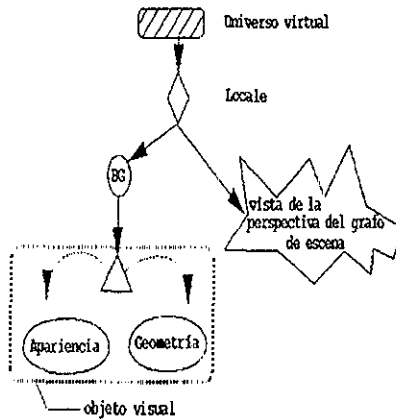
En los grafos de escena, el símbolo que se utilizo para identificar un objeto es el triángulo, a partir de ahora, un objeto visual se representará con el mismo triángulo que será un objeto del tipo *Shape3D* y dos óvalos que son los *NodeComponents*, todos ellos contenidos dentro de un rectángulo punteado. Como ejemplo se muestra en la siguiente figura 4.10.

Un objeto visual se puede definir utilizando un objeto *Shape3D* y un nodo *Geometry*, opcionalmente el objeto *Shape3D* también se puede referenciar utilizando un nodo *Appearance* sin problema alguno. Los constructores para *Shape3D* muestran que se pueden crear con más de un nodo de referencia.

Un objeto *Shape3D* no esta "vivo" ni "compilado", ya que hace referencia a objetos

TESIS CON
FALLA DE ORIGEN

Figura 4.10: Símbolos de objetos

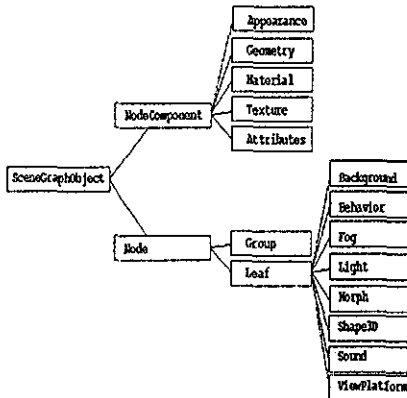


de tipo *Geometry* y/o *Appearance* que son del tipo *NodeComponents*, es por eso que sus componentes pueden alterarse con dichos métodos para así utilizar objetos *Shape3D* “vivos” o “compilados” si se han establecido que propiedades tiene el objeto

Componentes tipo Nodo

Este tipo de objetos contienen las especificaciones exactas de los atributos que tiene cada objeto. Muchas de las subclases *NodeComponent* definen algunos atributos visuales, la figura 4.11 muestra parte de la jerarquía que prevalece en la clase *NodeComponent*, así como sus descendientes

Figura 4.11: Jerarquía en la clase *NodeComponent*



TESIS CON
FALLA DE ORIGEN

Definiendo las clases de un Objeto Visual

El mismo objeto visual, puede aparecer sin ningún problema en más de una ocasión en un universo virtual, esto hace que se tenga más cuidado en el momento de crearlo. Hay que decir, que existen varias formas para crear una clase que defina a un objeto visual.

El siguiente segmento de código ilustra en forma general, el esqueleto de una clase llamada `ObjetoVisual`, en donde se encuentra en forma esquemática, la organización de un objeto visual. Los métodos se encuentran vacíos en el código. Este ejemplo no se utiliza en ningún otro programa, ya que es demasiado general.

```

1. public class ObjetoVisual extends Shape3D{
2.     private Geometry voGeometria;
3.     private Appearance voApariencia;
4.     Se crea un objeto del tipo Shape3D con una
5.     geometr'\{i}a y una apariencia
6.     La geometr'\{i}a se construye en el m\etodo creaGeometria
7.     La apariencia se crea en el m\etodo creaApariencia
8.     public ObjetoVisual() {
9.         voGeometria = creaGeometria();
10.        voApariencia = creaApariencia();
11.        this.setGeometry(voGeometria);
12.        this.setAppearance(voApariencia);
13.    }
14.    private Geometry creaGeometria(){
15.        // c\odigo para crear la geometr'\{i}a
16.    }
17.
18.    private Appearance creaApariencia(){
19.        // c\odigo para crear la apariencia
20.    }
21.} // fin de la clase ObjetoVisual

```

En el momento que se utiliza un *Shape3D*, se facilita mucho la creación de un objeto visual, de esta forma, el constructor puede utilizarse para crear un objeto nuevo e insertarlo como hijo de algún *Group*, y todo en una sola línea de código, por ejemplo, en la siguiente línea de código, *objRaiz* es una instancia de *Group*, creando de esta forma un objeto visual e incorporándolo al mismo tiempo como hijo de *objRaiz* en el grafo de escena.

```
objRaiz addChild(new ObjetoVisual());
```

El constructor `ObjetoVisual` crea un objeto que a su vez crea un objeto *Shape3D*, este sirve para referenciar a los *NodeComponents* creados a su vez por los métodos `creaGeometria()` y `creaApariencia()`. El método `creaGeometria()` crea un componente-nodo-geometría para utilizarlo en el objeto visual. El método `creaApariencia()` es el responsable de crear un nodo-componente que define la apariencia del objeto.

Otra posible organización es definir una clase contenedora que no derive del API de Java3D. En este diseño, la clase que contiene al objeto visual puede tener un nodo del tipo *Group* o uno del tipo *Shape3D* como raíz de la sub gráfica. Esta clase tendrá que definir algunos métodos que regresen una referencia a la raíz. Esta técnica es más tediosa pero más fácil de entender, por lo que en este trabajo se optó por usarla para construir la maya que formará la topografía.

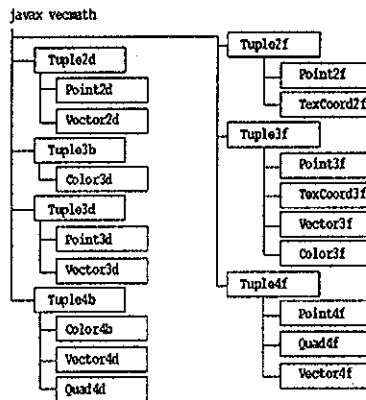
Una tercera organización que se utilizó es la clase *Box* definida en el paquete de Java3D conocido como *com.sun.j3d.utils.geometry*; esta clase es extensión de *Primitive*, la que a su vez pertenece a *Group*. El diseño detallado de esta clase no se discute en este trabajo, pero el código fuente se encuentra disponible en el Java3D API.

Clases Matemáticas

Para crear objetos visuales utilizamos las clases y subclases contenidas en *Geometry*. Muchas de estas subclases sirven para definir puntos, líneas o polígonos rellenos, pero en este momento vamos a discutir varias clases matemáticas (*Point**, *Color**, *Vector**, *TextCoord**, *Tuple**), que necesitamos para indicar y manipular la relación que necesitamos establecer en cada uno de los vértices de las geometrías y que forzosamente se necesitan conocer.

Utilizamos un asterisco al final de cada una de las clases, con el fin de indicar que tienen varias representaciones, por ejemplo *Tuple** indica que existen las siguientes clases: *Tuple2f*, *Tuple2d*, *Tuple3d*, *Tuple3f*, *Tuple3d*, *Tuple4b*, *Tuple4f*, *Tuple4d*. Para cada caso el primer número indica cuantos elementos tiene en cada tupla, y la letra indica el tipo, 'f' significa real, 'd' indica que es de doble precisión y real y 'b' para byte. Por lo que *Tuple3f* utiliza tres valores reales de simple precisión.

Figura 4.12: Jerarquía de las clases Matemáticas

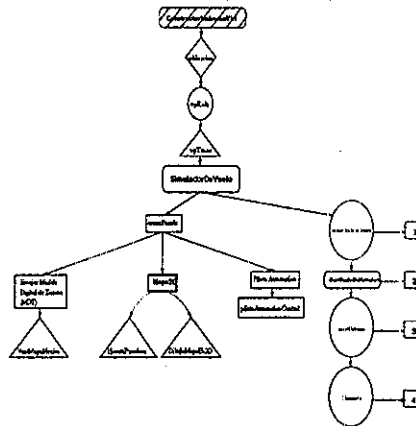


TESIS CON
FALLA DE ORIGEN

4.4 Descripción de las clases del sistema

En la figura 4 13 se muestra el grafo acíclico-directo del sistema en donde se observa la jerarquía de las principales clases que componen al sistema, por lo que en esta sección se explicarán la funciones que realiza cada una y cómo se relacionan entre ellas. La clase principal es la `SimuladorDeVuelo.java` ya que es en donde se conjunta todo el despliegue de la aplicación

Figura 4 13: Grafo acíclico-directo del sistema



4.4.1 Clase SimuladorDeVuelo

Paquete: `SimVuelo` `SimuladorDeVuelo`

```
public class SimuladorDeVuelo extends Applet implements ActionListener
```

Constructores: `SimuladorDeVuelo(Frame objetoFrame)`

`SimuladorDeVuelo()`

Crea un objeto `SimuladorDeVuelo` que permite desplegar todas las ventanas del sistema por lo que recibe como argumentos objetos `Frame` o no.

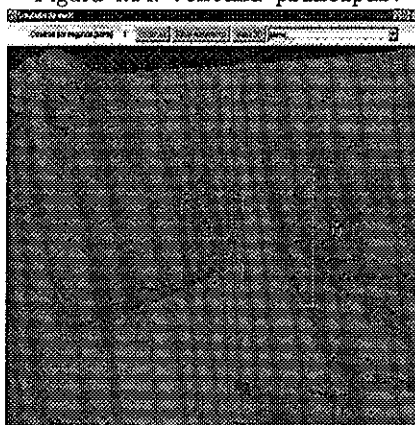
`SimuladorDeVuelo` permite que las ventanas a desplegarse en un *applet* o como una aplicación. Esta clase es la encargada de llamar a todas las clases del sistema comenzando por la clase `panelPrincipal` que es la encargada de crear en *Panel* o ventana principal; a continuación esta, clase se pasa como argumento a la clase `canvas3D` que sirve para añadir los objetos 3D en la ventana principal.

A continuación se crea un objeto llamado `escena` utilizando la clase `crearGrafoDeEscena` que se encarga de crear el grafo de escena con lo que se van incorporando todos los objetos tridimensionales a la escena utilizando las clases propias de `Java3D`, después se crea un

TESIS CON
FALLA DE ORIGEN

objeto llamado *notifica* utilizando la clase *identificadorDeMovimiento* que como su nombre lo indica, su función es obtener los movimientos del ratón para animar la vista que se observa en la ventana principal y simular un vuelo sobre los objetos. Después se crea el objeto *universo* con la clase *constUniverso* pasándole como argumento el objeto *canvas3D* y por último se crea el objeto que va a incorporar la topografía en forma de una superficie tridimensional así como los demás objetos que componen la escena, este objeto es *escenario* y utiliza la clase *Escenario*. En la figura 4.14 se muestra como se observa.

Figura 4.14: Ventana principal.



4.4.2 Clase crearPanels

Paquete: `SimVuelo.SimuladorDeVuelo.crearPanels`

Constructor: `class crearPanels()`

Clase que permite incorporar en la clase *SimuladorDeVuelo* añadir los botones *Escoger MDI*, *Quitar MDI*, *Incluir MDI* y *Salir*, ver figura 4.15. La función de cada uno de estos es escoger, sobre un mapa de la República Mexicana, la topografía que se desea desplegar en la ventana principal

4.4.3 Clase crearGrafoDeEscena

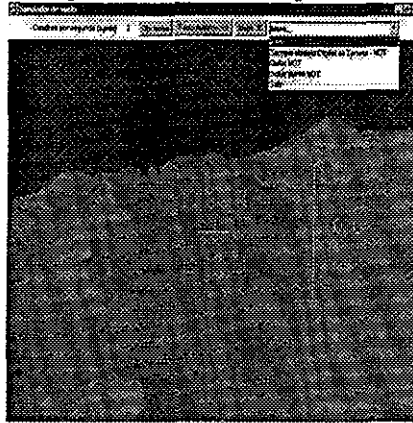
Paquete: `SimVuelo.SimuladorDeVuelo.crearGrafoDeEscena`

Constructor: `public BranchGroup crearGrafoDeEscena()`

Esta clase se encarga de inicializar al objeto *objRoot* de tipo *BranchGroup* para permitir modificar los objetos tridimensionales en la escena, esto significa que se puede interactuar con los objetos que se ven en escena permitiendo que se animen. También se encarga

TESIS CON
FALLA DE ORIGEN

Figura 4.15: Menú de opciones.



de añadir a la escena los vértices a seguir en forma de vuelo predeterminado o piloto automático, utiliza los métodos `inicializaPilotoAutom1(objRoot)` y `inicializaPilotoAutom2(objRoot)`.

```
private void inicializaPilotoAutom2( BranchGroup bg ) {
    puntosParaSeguir = new PuntosParaSeguir();

    puntosParaSeguir addPoint( 4050.0f, 4030.0f, 0.0f );

    puntosParaSeguir addPoint( -5050.0f, 4795.0f, 0.0f );

    puntosParaSeguir attachMarkers( bg );
}
```

Las instrucciones anteriores muestran como se van incorporando todos los vértices a la escena.

Otro objeto que se utiliza en esta clase es

```
Behavior perf = new
CuadrosPorSeg( perfData );
```

encargada de desplegar el número de cuadros por segundo que se despliegan en la ventana principal, dependiendo de la velocidad con que se mueve el ratón.

TESIS CON
FALLA DE ORIGEN

4.4.4 Clase ConstructorUniversoVirt

Paquete:SimVuelo ConstructorUniversoVirt

Constructor: `public ConstructorUniversoVirt(Canvas3D c)`

Esta es una de la clases más importantes, ya que es aquí en donde se establecen todos los parámetros del Universo Virtual que contendrá a los métodos *VirtualUniverse*, *Locale*, *TransformGroup* y *View* con los cuales se construyen las dimensiones y la ubicación de los objetos en el espacio 3D

Aquí también se establecen las propiedades de las transformaciones que se aplican a los objetos:

```
vpIrans = new IransformGroup(t);
vpIrans setCapability(vpIrans .ALLOW_TRANSFORM_READ);
vpIrans setCapability(vpIrans .ALLOW_TRANSFORM_WRITE);
vpIrans setCapability(Group .ALLOW_CHILDREN_READ);
vpIrans setCapability(Group .ALLOW_CHILDREN_WRITE);
vpIrans setCapability(Group .ALLOW_CHILDREN_EXTEND);
```

Los parámetros *ALLOW_TRANSFORM_READ* y *ALLOW_TRANSFORM_WRITE* permiten el acceso de la información que modifica las transformaciones del objeto y los parámetros *ALLOW_CHILDREN_READ*, *ALLOW_CHILDREN_WRITE* y *ALLOW_CHILDRE* contienen las propiedades para permitir que un hijo se añadido al nodo *Group* antes de que se ejecute en el modo “compilado” o “vivo”

4.4.5 Clase Escenario

Paquete:SimVuelo Escenario

Constructor:

```
public class Escenariopublic Escenario() {
    this addChild( getTextureScenery() );
    this compile();
} extends BranchGroup
```

Esta clase es la encargada de incorporar la topografía al Universo Virtual en forma de una malla en donde cada vértice es una altura sobre el nivel del mar por lo que se crea un superficie tridimensional con la forma del terreno.

La forma en que se obtienen los datos de la altimetría, es utilizando la siguiente instrucción:

```
ObjSimVuelo
topograf = new ObjSimVuelo("Datos/topografia");
```

en donde se indica que se va a acceder a un archivo cuyo nombre es topografía rapido que se crea en la clase ObjSimVuelo.

4.4.6 Clase ObjSimVuelo

Paquete: SimVuelo ObjSimVuelo

```
public class ObjSimVuelo extends BranchGroup
```

Constructor: ObjSimVuelo(String nombreDelarchivo, float x, float y, float z, float roll, float pitch, float yaw)

Esta clase es la encargada de leer la información de un archivo que contiene la información topográfica en formato ASCII para convertirla posteriormente en un formato de tipo binario con el propósito de que cuando se vuelva a utilizar el archivo y no haya sufrido alteración alguna; como por ejemplo escoger otra zona, la lectura de la información sea mucho más rápida y eficiente que en archivo de tipo ASCII

Se obtienen varias variables que vienen en el encabezado del archivo y es la información que describe las dimensiones de la malla (LongitudX, LongitudY), la distancia que existe de un vértice a otro se almacena en la variable dimenCelda y algo muy importante las coordenadas en valores UTM de la posición en el Globo Terráqueo de la altura localizada en la esquina inferior izquierda (si tomamos como referencia el que si se observa la malla como una imagen proyectada en un monitor)

```
private LectorMallaARC mdt;
private int longitudX;
private int longitudY;
private float dimenCelda;
private float ignorado;
private float min;
private float max;
private double esquinaX;
private double esquinaY;
private lonlatAxy convUtm = new lonlatAxy();
```

La forma en que se obtienen cada uno de los valores es:

```
mdt = new LectorMallaARC(nombrearchivo);
longitudX = mdt.ObtenerDimenDelRaster().width;
longitudY = mdt.ObtenerDimenDelRaster().height;
ignorado = (float) mdt.ObtenerValorIgnorado();
min = (float) mdt.ObtenerMinimo();
max = (float) mdt.ObtenerMaximo();
dimenCelda = (float) mdt.ObtenerDimCelda();
double esquinax11 = (float) mdt.ObtenerEsquinaX11();
double esquinay11 = (float) mdt.ObtenerEsquinaY11();
```

También se utilizan varios vectores, que sirven para almacenar las coordenadas de cada vértice en un conjunto de tres o cuatro elementos para ir construyendo superficies en

forma de rectángulos o cuadrados para reconstruir de la manera más realista la superficie que representa a la topografía.

A cada vértice se le asigna también una normal, esto para poder calcular la luz que incide sobre este punto y así calcular la iluminación de la escena. Para este cálculo también se le asigna un color a cada vértice que se refleja en la tonalidad y el aspecto de la superficie, en este caso es un color café oscuro, lo que da un parecido al color de la tierra.²

```
private Color3f[] colores;
private Vector cordenadIriangulos = new Vector();
private Vector cordenadCuadrados = new Vector();
private Vector coloresIriangulos = new Vector();
private Vector coloresCuadrados = new Vector();
private Vector normalesIriangulos = new Vector();
private Vector normalesCuadrados = new Vector();
private Vector triangularObjetos = new Vector();
```

4.4.7 Clase DibujaMapa

Paquete: SimVuelo Mapa DibujaMapa

Constructor: class DibujaMapa()

Esta clase es la encargada de dibujar en forma bidimensional los mapas y los resultados del algoritmo de la *p-mediana*, esto se hace dibujando a las localidades en forma de puntos sobre el mapa que define el contorno del estado que se está aplicando el algoritmo de la *p-mediana*. Es importante mencionar que esta clase puede dibujar de manera independiente las localidades o los polígonos que definen la forma de los estados; todos los resultados están georeferenciados en coordenadas UTM para contar con una certidumbre de la ubicación de toda la información de la zona geográfica que se está estudiando.

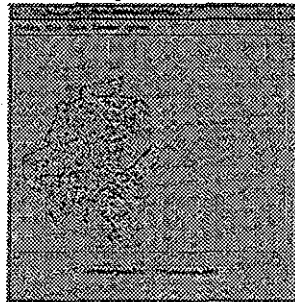
Para dibujar los polígonos de los estados se implementó un método que lee archivos tipo *lin* de *ARC-INFO* con el propósito de utilizar un formato conocido. En cuanto al formato para alimentar con los datos socio-económicos al método de la *p-mediana*, este consiste en un archivo de tipo ASCII en donde se definen los siguientes datos para cada localidad: Clave oficial proporcionada por el INEGI para identificar a cada población del país, la población (de 1995) que habita en la localidad, su distancia a un camino o carretera, su índice de marginación, grado de marginación, el número de clínicas de seguridad social, número de unidades de medicina familiar, el número de hospitales, el número de clínicas de IMSS-Solidaridad, el número de telégrafos, el número de tiendas DICONSA, el número de teléfonos rurales, el número de primarias, secundarias y bachilleratos, por último se proporcionan las coordenadas en UTM de la ubicación geográfica de la localidad.

²Para mayor información revisar las citas [Foley, 1983] y [van Dam, 1990] que vienen en la bibliografía al final de esta tesis.

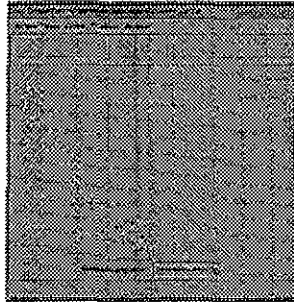
Figura 4.16: Ventanas de dibujo de mapas en 2D.



Cuando se han dibujado los resultados, ya sea de los contornos de los estados, ver figura 4.16 y los de la *p-mediana*, ver figura 4.17 y figura 4.18, se tienen varias opciones para obtener información de los resultados, la primera es al pasar el ratón por la zona en donde se dibujan los resultados para obtener en forma automática en la parte inferior derecha de la ventana, las coordenadas en UTM de donde está el ratón, otra opción es presionando el botón izquierdo del ratón sobre los puntos rojos que se utilizan para indicar la ubicación de un Centro Proveedor de Servicios para que se desplieguen en una nueva ventana las claves de aquellas comunidades a las que prestará servicios

Figura 4.17: Resultados de la *p-mediana* en el estado de Hidalgo.

También se pueden salvar los resultados de cada ejercicio para que se puedan graficar en sistemas como *ARC-INFO* para que la información se reutilice y no se limite a solo este sistema

Figura 4.18: Resultados de la *p-mediana* en el estado de Veracruz.

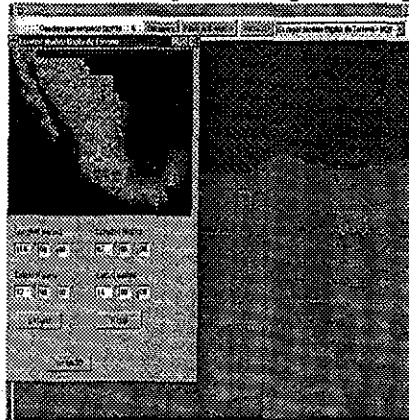
4.4.8 Clase VentMapaMexico

Paquete: SimVuelo.VentImag.VentMapaMexico

Constructor: `public class VentMapaMexico();`

En la figura 4.19 se observa en una ventana, la imagen de la República Mexicana que representa la topografía de todo el país, además en la parte inferior se observan una serie de casillas en donde se muestran las coordenadas geográficas (en coordenadas de latitud y longitud) de los límites superior e inferior que abarca la imagen

Figura 4.19: Ventanas para escoger la topografía.



La forma en que se puede escoger la topografía es de dos maneras, la primera es utilizando las cuatro casillas en donde por medio del teclado se ingresan las coordenadas en grados minutos y segundos, que son los que definirán los cuatro puntos cardinales de la zona a escoger; la segunda opción es utilizando el ratón en forma interactiva sobre la

imagen, basta con presionar el botón izquierdo y arrastrarlo para que se valla dibujando un rectángulo, el cual representa la zona que se desea escoger.

Una vez que se ha seleccionado el área geográfica deseada, basta con presionar en botón de Ejecutar para que se cree el archivo en código ASCII con los valores de las alturas sobre el nivel del mar

4.4.9 Clase lonlatAxy

Paquete: SimVuelo lonlatAxy

Constructor: void lonlatAxy(double LON, double LAT, int Z0)

Clase que se utiliza para convertir en forma automática coordenadas geográficas (longitud, latitud) a coordenadas UTM (x,y).

Figura 4.20: Ejemplo de conversión de coordenadas geográficas (longitud, latitud) a coordenadas UTM (x,y).



Como ejemplo, en la figura 4.20 se proporcionan las coordenadas UTM y geográficas de la cuenca de México

4.4.10 Clase LectorMallaARC

Paquete: SimVuelo LectorMallaARC

TESIS CON
FALLA DE ORIGEN

Constructor: `LectorMallaARC(String nombrearchivo);`

Esta clase se encarga de acceder al archivo que contiene los datos de la altimetría de la zona que se escogió; esta clase guarda en un arreglo matricial los datos de las alturas que se encuentran en formato entero.

4.4.11 Clase IdentificadorDeMovimiento

Paquete: `SimVuelo` `IdentificadorDeMovimiento`

Constructor: `public IdentificadorDeMovimiento()`

Esta clase sirve para indicar la “dirección” de hacia donde se dirige la ventana en donde se despliega la superficie tridimensional que representa la topografía a

```
double \angulo = java.lang.Math.atan( direccion.x / direccion.z );

if ( direccion.z > 0 ) \angulo = Math.PI + \angulo;
if ( direccion.x > 0 && direccion.z < 0 ) \angulo = Math.PI*2 + \angulo;

pantInstrum.conjEncabez( (double)Math.round(angulo/Math.PI*180.0) );
```

Las anteriores líneas de código se utilizan para calcular la orientación de la “vista” que el ratón o las flechas del teclado va estableciendo como recorrido del simulador de vuelo

4.4.12 Clase TexturasDeObjetos

Paquete: `SimVuelo` `TexturasDeObjetos`

Constructores:

```
public TexturasDeObjetos( String filename ) {
    super( filename );
    addItexture();
}

public TexturasDeObjetos( String filename, float x, float y, float z,
                          float roll, float pitch, float yaw ) {
    super( filename, x, y, z, roll, pitch, yaw );
    addItexture();
    app.setCapability( Appearance.ALLOW_IEXTURE_WRITE );
    enableItextureMapping( false );
}
```

En esta clase se encarga de incorporar una textura a la superficie que representa la topografía ³

```
Vector4f s = new Vector4f(0.001f, 0.0f, 0.0f, 0.0f);
Vector4f t = new Vector4f(0.0f, 0.0f, 0.001f, 0.0f);
TexCoordGeneration texGen = new TexCoordGeneration(
    TexCoordGeneration.OBJECT_LINEAR,
    TexCoordGeneration.TEXTURE_COORDINATE_2,
    s, t );

tex = new TextureLoader("Datos/Texturas/textura.jpg", c );
```

En las línea de código se indican los parámetros que Java3D requiere para que la imagen que se va a utilizar como textura tengan los atributos correctos para que se logre una buena apariencia al realizar el dibujo en la ventana.

³Para mayor información revisar las citas [Foley, 1983] y [van Dam,1990] que vienen en la bibliografía al final de esta tesis

Capítulo 5

Conclusiones

En esta tesis se presentan los resultados del desarrollo de un sistema en el que se involucran aspectos de la computación que rara vez están íntimamente relacionados: Graficación por computadora e Investigación de Operaciones.

Este trabajo fué pensado originalmente con fines de estudio para proponer una solución al gran problema de la marginación social que existe en nuestro país, particularmente en un momento en el que los presupuestos federal y estatal han sido reducidos drásticamente. Los Centros Proveedores de Servicios constituyen una propuesta que sugiere alternativas de asignación de gasto social de forma austera, impulsando el desarrollo de zonas rurales aisladas.

Respecto a los servicios que es necesario aumentar para que los centros proveedores cumplan su función de atender las necesidades de la población que vive cerca de ellos, hay que resaltar que son diferentes entre cada uno, en algunos casos sobresale la carencia del servicio de salud, aun cuando se tiene una mejor cobertura del servicio de educación, existiendo varios centros proveedores donde éste es el único servicio más o menos presente en las localidades atendidas. También se observó que los Centros Proveedores de Servicios cercanos a carreteras y sus localidades atendidas conforman un universo heterogéneo. Otra forma de considerar la heterogeneidad es analizando los centros según su grado de marginación y el de las localidades que atendería.

Cabe destacar que el análisis de los resultados, sugiere la necesidad de redoblar esfuerzos en la construcción de caminos, así como de revestir los existentes para que puedan ser transitados todo el año. Sin duda ello reducirá las penalidades y esfuerzos que actualmente sufre y realiza la población para trasladarse a las localidades con servicios. Además debe señalarse lo conveniente del uso de recursos que para esos fines tiene asignado tanto el gobierno de la entidad como el federal.

Es importante mencionar dos cuestiones en cuanto a los resultados generales del estudio. La primera se refiere a que en la selección de los centros potenciales de servicios, es decir antes de la aplicación del algoritmo de la *p-mediana*, no se consideró el grado de marginación de las localidades y sin embargo más de la mitad de los asentamientos

ubicados en las áreas de influencia de los centros proveedores aislados tienen esa condición. Esa evidencia describe cierta eficiencia de la metodología, los criterios y la aplicación del algoritmo de la *p-mediana* para cumplir con el objetivo de identificar a la población más desaventajada a partir de la distribución territorial de los asentamientos, el acervo de servicios básicos, así como la distancia respecto de las ciudades y centros urbanos-regionales.

La segunda observación se relaciona con la oportunidad efectiva de cristalizar o hacer realidad esta propuesta, es decir, de aumentar el acervo de servicios de las localidades elegidas como centros proveedores. Esa posibilidad, obviamente, depende de la disponibilidad de recursos públicos, pero no sólo de eso. El cambio de paradigma de la política social, el nuevo marco institucional y la emergencia de una activa sociedad civil, crea oportunidades valiosas para la ejecución de iniciativas como la presentada.

Otra de las evaluaciones de los resultados, se llevó a cabo con relación a los riesgos a que están expuestos los asentamientos humanos, así como, la normatividad aplicada por las autoridades federales para el cuidado de los recursos naturales y el medio ambiente.

Desde luego, la aplicación irrestricta de este tipo de modelos a la realidad social no podrá solucionar todos los problemas de marginación que tradicionalmente han enfrentado estas localidades, ya que estos modelos difícilmente pueden incorporar valores culturales que, pudieran contravenir las propuestas que se sugieren. También hay que considerar que las distancias utilizadas en el modelo son lineales, mientras que la topografía define recorridos distintos tanto en distancia como ruta a seguir. Este será un trabajo deberá corregirse, haciendo hincapié en que no es trivial obtener una solución a esta limitante.

El sistema que se presenta cumple con las características necesarias para el fin que fue originalmente pensado ya que se puede representar los resultados de las simulaciones que se generan al utilizar el algoritmo de la "*p-mediana*" en forma tridimensional o bidimensional. Asimismo el sistema genera información geo-referenciada para aquellas localidades de la República Mexicana que resultan seleccionadas, por lo que se pueden observar los obstáculos naturales que afectan la solución provista por el algoritmo, al recorrer "volando" las rutas que se proponen; que serán las que la población de dichas localidades se recorran a pie la mayoría de las veces.

Otra dificultad que se presentó, es el despliegue gráfico a través del Internet, ya que la información topográfica ocupa grandes espacios tanto en disco como en memoria, y en el momento que se desea visualizar en el navegador la región que se está estudiando, el envío de la geometría en el *WEB* lleva demasiado tiempo por lo que se corta la comunicación con el servidor, para resolver este problema se debe de analizar la forma de implementar un algoritmo que comprima y descomprima la información geométrica de tal forma que no se pierda detalle y conserve el orden de los datos.

Por último hay que mencionar que las aplicaciones potenciales de este tipo de sistemas son muy variadas. Entre ellas se pueden mencionar estudios de planeación urbana, de transporte, ecológicos y sistemas de información geográfica.

Bibliografia

Bibliografía

Ávila, JL

Dispersión de la población: Retos y Propuestas de Intervención
El Mercado de Valores, Prospectiva del Sistema Urbano Mexicano I.
marzo 2000, Nacional Financiera, México, 2000. pp 34-36

Barrilleux, Jon

3D user interfaces with Java3D
Manning, 2001

Bribiesca, Ernesto

La Topografía del Valle de México Representada en forma Digital
Reportes de Desarrollo
Volumen 3, No 7, Marzo, 1993.
IIMAS, UNAM

Brown, Kirk y Daniel Petersen

Ready to run Java3D
John Wiley & Sons Inc, 1999.

Consejo Nacional de población y Consejo Estatal de Población de Guanajuato. **Centros Proveedores de Servicios: Una Estrategia para Atender la Dispersión de la Población en Guanajuato**

Isbn 970-628-466-4
México, 2000

Domingues, Nelio

Annals of Operations Research.
J. C. Blazer AG, science Publishers, 1994.

Flanagan, David
Java Examples in a Nutshell
O'Reilly, 1997.

Flanagan, David.
Java in a Nutshell
O'Reilly, 1997.

Foley, James, Andries van Dam, Steven K Feigner y John F Huges.
Computer Graphics Principles and Practice.
Addison-Wesley Systems Programming Series
Washington D.C, 1990

Foley, James y Andries van Dam
Fundamentals of Interactive Computer Graphics
Addison-Wesley The Systems Programming Series
Washington D C, 1983.

Instituto Nacional de Estadística, Geografía e Informática.
Geomodelos de Altimetría del Territorio Nacional
GEMA - Disco compacto.
México, 1995.

Liwu, Li
Java data structures and programing.
Springer, 1998

Maranzana, F.
On the location of supply points to minimize transport costs.
Operations Research Quarterly, vol 15, pp 261-27, 1964.

Teitz y Bart
**Heuristic Methods for Estimating the Generalized Vertex Median
of a Weighted Graph**.
Operations Research, 16(5), 955-961, 1968.

Vega, Elio.

Sistema de visualización de datos topográficos del territorio nacional

Tesis de Licenciatura de matemáticas Aplicadas y Computación

ENEP Acatlán, México, 1999

Zubieta, J.

Centros Rurales Proveedores de Servicios

[en] Tamayo, J. (editor), El Desarrollo Regional y Urbano en México:

Interpretaciones, Experiencias y Nuevas Tecnologías

Edit Porrúa, México. (en imprenta)

Zubieta, J y E. Vega

Alternativas de Inversión Pública para Desarrollo Rural

3er Seminario Internacional de Ingeniería de Sistemas

Academia Mexicana de Ingeniería y la Facultad de Ingeniería de la UNAM

Ixtapa Zihuatanejo, Gro , noviembre, 2000

Zubieta, J

Centro Rurales Proveedores de Servicios

Reunión de Especialista El Desarrollo Regional y Urbano en México:

Interpretaciones, Experiencias y Nuevas Tecnologías

CISDER Consultores, OCDE/OECD y el Colegio de Ingenieros Civiles de México, A. C

México D. F. , octubre 2000.

Zubieta, J

Una Alternativa de Inversión en Gasto Social en Localidades Rurales Dispersas.

Seminario Población y Desarrollo, Programa LEAD-El Colegio de México

IISUNAM y Conapo

México D. F. , octubre 2000.

Zubieta, J, E. Vega, A. Alonso

Aplicación del Algoritmo de la *p*-mediana a Programas de Desarrollo Social.

X Congreso Latino Iberoamericano de Investigación de Operaciones.

X CLAIO, México D. F. , septiembre 2000

Anexo 1

Anexo 1

Listado de la salida del programa al aplicar el algoritmo de la *p-mediana* en el estado de Hidalgo¹:

- El arbol1 cuya raíz tiene la clave 130460044 tiene 21 elementos y son los siguientes: 1) 130460029 2) 130460037 3) 130460015 4) 130460089 5) 130460129 6) 130460096 7) 130460094 8) 130460095 9) 130460092 10) 130460107 11) 130460091 12) 130460087 13) 130460067 14) 130460145 15) 130460088 16) 130460086 17) 130460007 18) 130460018 19) 130460137 20) 130460122 21) 130460032
- El arbol2 cuya raíz tiene la clave 130490029 tiene 10 elementos y son los siguientes: 1) 130490080 2) 130490045 3) 130490037 4) 130490026 5) 130490024 6) 130490005 7) 130490006 8) 130490059 9) 130490017 10) 130490057
- El arbol3 cuya raíz tiene la clave 130460043 tiene 45 elementos y son los siguientes: 1) 130460143 2) 130460009 3) 130460150 4) 130460038 5) 130460039 6) 130460046 7) 130460078 8) 130460014 9) 130460142 10) 130460035 11) 130460049 12) 130460012 13) 130460074 14) 130460146 15) 130460085 16) 130460117 17) 130460084 18) 130460112 19) 130460073 20) 130460020 21) 130460066 22) 130460071 23) 130460081 24) 130460125 25) 130460032 26) 130460030 27) 130460034 28) 130460016 29) 130460048 30) 130460050 31) 130460109 32) 130280205 33) 130460017 34) 130460033 35) 130460036 36) 130460144 37) 130460033 38) 130460042 39) 130460031 40) 130460063 41) 130460099 42) 130460059 43) 130460135 44) 130460024 45) 130460061
- El arbol4 cuya raíz tiene la clave 130490016 tiene 36 elementos y son los siguientes: 1) 130490077 2) 130490023 3) 130490009 4) 130490069 5) 130490013 6) 130490061 7) 130490058 8) 130490042 9) 130490004 10) 130490070 11) 130490078 12) 130490007 13) 130490010 14) 130490015 15) 130490055 16) 130490003 17) 130490008 18) 130490081 19) 130490011 20) 130490083 21) 130180081 22) 130490046 23) 130490038 24) 130490079 25) 130490073 26) 130490022 27) 130180050 28) 130180047 29) 130180074 30) 130180115 31) 130180059 32) 130180042 33) 130180103 34) 130180011 35) 130180079 36) 130180049
- El arbol5 cuya raíz tiene la clave 130280081 tiene 59 elementos y son los siguientes: 1) 130280020 2) 130280035 3) 130280110 4) 130280106 5) 130280073 6) 130460098 7) 130280010 8) 130280232 9) 130280123 10) 130280114 11) 130280003 12) 130280028 13) 130460115 14) 130460097 15) 130460071 16) 130460060 17) 130460045 18) 130280073 19) 130280036 20) 130280070 21) 130460105 22) 130460036 23) 130460114 24) 130280016 25) 130460103 26) 130460124 27) 130460102 28) 130280016 29) 130460101 30) 130280014 31) 130460061 32) 130460105 33) 130280022 34) 130280029 35) 130280025 36) 130460068 37) 130280113 38) 130280031 39) 130320024 40) 130280028 41) 130280009 42) 130280059 43) 130460057 44) 130460123 45) 130280083 46) 130320031 47) 130280014 48) 130460023 49) 130320033 50) 130460147 51) 130320006 52) 130320004 53) 130320013 54) 130320006 55) 130320006 56) 130320003 57) 130320032 58) 130320026 59) 130320022
- El arbol6 cuya raíz tiene la clave 130490001 tiene 51 elementos y son los siguientes: 1) 130490052 2) 130490002 3) 130490040 4) 130490019 5) 130490067 6) 130490021 7) 130490039 8) 130490041 9) 130490084 10) 130490032 11) 130490071 12) 130490076 13) 130490075 14) 130490043 15) 130490025 16) 130490018 17) 130490050 18) 130490044 19) 130450062 20) 130490048 21) 130490012 22) 130490014 23) 130490056 24) 130490034 25) 130490065 26) 130490085 27) 130490035 28) 130490064 29) 130490031 30) 130490047 31) 130490027 32) 130490033 33) 130490020 34) 130180058 35) 130490054 36) 130490082 37) 130490030 38) 130180072 39) 130180104 40) 130180090 41) 130490036 42) 130490074 43) 130180010 44) 130180057 45) 130180004 46) 130180099 47) 130180105 48) 130490072 49) 130490028 50) 130180039 51) 130180089
- El arbol7 cuya raíz tiene la clave 130340014 tiene 18 elementos y son los siguientes: 1) 130340033 2) 130340032 3) 130340006 4) 130730026 5) 130730107 6) 130620018 7) 130620019 8) 130340048 9) 130180068 10) 130620130 11) 130620127 12) 130340027 13) 130620030 14) 130620026 15) 130340047 16) 130340050 17) 130620093 18) 130620019
- El arbol8 cuya raíz tiene la clave 130280064 tiene 20 elementos y son los siguientes: 1) 130280233 2) 130280221 3) 130280074 4) 130280078 5) 130280018 6) 130280055 7) 130280040 8) 130280220 9) 130280004 10) 130280228 11) 130280082 12) 130280041 13) 130280186 14) 130280033 15) 130280111 16) 130280079 17) 130250040 18) 130280017 19) 130280052 20) 130250047
- El arbol9 cuya raíz tiene la clave 130280093 tiene 32 elementos y son los siguientes: 1) 130280118 2) 130280050 3) 130280027 4) 130280087 5) 130280026 6) 130280071 7) 130280227 8) 130280001 9) 130320034 10) 130320002 11) 130280042 12) 130280030 13) 130280020 14) 130280134 15) 130280191 16) 130280216 17) 130280104 18) 130280210 19) 130250066 20) 130280235 21) 130280226 22) 130280061 23) 130280112 24) 130280135 25) 130280192 26) 130280053 27) 130280008 28) 130280225 29) 130280011 30) 130280013 31) 130280229 32) 130280076
- El arbol10 cuya raíz tiene la clave 130730040 tiene 12 elementos y son los siguientes: 1) 130460062 2) 130460025 3) 130460130 4) 130460156 5) 130460027 6) 130460004 7) 130730037 8) 130730019 9) 130730023 10) 130730021 11) 130340045 12) 130340024
- El arbol11 cuya raíz tiene la clave 130180035 tiene 22 elementos y son los siguientes: 1) 130180029 2) 130180107 3) 130180067 4) 130180027 5) 130180003 6) 130180036 7) 130180038 8) 130180093 9) 130180048 10) 130180100 11) 130180110 12) 130180012 13) 130180016 14) 130180023 15) 130180025 16) 130180076 17) 130180075 18) 130180032 19) 130180019 20) 130180073 21) 130180022 22) 130180037
- El arbol12 cuya raíz tiene la clave 130460055 tiene 41 elementos y son los siguientes: 1) 130460028 2) 130460139 3) 130460138 4) 130460070 5) 130460110 6) 130460151 7) 130460075 8) 130460001 9) 130460154 10) 130460155 11) 130460148 12) 130460131 13) 130460128 14) 130460005 15) 130460153 16) 130460152 17) 130460149 18) 130460003 19) 130460077 20) 130460064 21) 130460008 22) 130460026 23) 130460026 24) 130460076 25) 130460058 26) 130460047 27) 130280075 28) 130460136 29) 130460134 30) 130460054 31) 130460065 32) 130280130 33) 130280237 34) 130280217 35) 130280128 36) 130730018 37) 130280023 38) 130280236 39) 130280127 40) 130730011 41) 130730029
- El arbol13 cuya raíz tiene la clave 130180034 tiene 42 elementos y son los siguientes: 1) 130180016 2) 130180087 3) 130180114 4) 130180030 5) 130180054 6) 130180083 7) 130180084 8) 130180033 9) 130180116 10) 130180005 11) 130180006 12) 130180053 13) 130180001 14) 130180096 15) 130180080 16) 130180015 17) 130180031 18) 130180077 19) 130180070 20) 130180063 21) 130180117 22) 130180062 23) 130180113 24) 130180088 25) 130180064 26) 130180109 27) 130180065 28) 130180009 29) 130180020 30) 130180028 31) 130180062 32) 130180061 33) 130180051 34) 130180021 35) 130180112 36) 130180069 37) 130620093 38) 130620021 39) 130620017 40) 130180026 41) 130620104 42) 130180040
- El arbol14 cuya raíz tiene la clave 130280080 tiene 14 elementos y son los siguientes: 1) 130280199 2) 130280198 3) 130280234 4) 130280067 5) 130280044 6) 130280121 7) 130280005 8) 130280009 9) 130280008 10) 130280045 11) 130280021 12) 130280001 13) 130280054 14) 130110012 14) 130110016
- El arbol15 cuya raíz tiene la clave 130620025 tiene 34 elementos y son los siguientes: 1) 130180044 2) 130180045 3) 130180108 4) 130180111 5) 130180043 6) 130180012 7) 130180093 8) 130180075 9) 130620069 10) 130180055 11) 130620145 12) 130180008 13) 130180013

¹Las claves son las que INEGI estableció para identificar a cada localidad del país

TESIS CON
FALLA DE ORIGEN

14) 130628098 15) 130620034 16) 130620094 17) 130180002 18) 130180071 19) 130620010 20) 130620012 21) 130620149 22) 130620148 23) 130180096 24) 130180017 25) 130620141 26) 130620099 27) 130620107 28) 130620139 29) 130620020 30) 130620140 31) 130620116 32) 130620117 33) 130620101 34) 130620027

El arbol61 cuya raíz tiene la clave 130400011 tiene 57 elementos y son los siguientes: 1) 130400046 2) 130400068 3) 130400071 4) 130400054 5) 130400019 6) 130400044 7) 130400028 8) 130400936 9) 130400050 10) 130400053 11) 130400026 12) 130400027 13) 130400014 14) 130400041 15) 130400025 16) 130400038 17) 130400039 18) 130400034 19) 130400065 20) 130400016 21) 130400090 22) 130400048 23) 130400067 24) 130400087 25) 130400024 26) 130400066 27) 130400052 28) 130400001 29) 130400940 30) 130400042 31) 130400089 32) 130400006 33) 130400096 34) 130400003 35) 130400040 36) 130400020 37) 130400002 38) 130400091 39) 130400032 40) 130400018 41) 130180024 42) 130400055 43) 130400070 44) 130400022 45) 130400060 46) 130400063 47) 130400099 48) 130400088 49) 130400015 50) 130400094 51) 130400079 52) 130400080 53) 130400098 54) 130400035 55) 130400100 56) 130400064 57) 130400031

El arbol71 cuya raíz tiene la clave 130280057 tiene 75 elementos y son los siguientes: 1) 130320023 2) 130280203 3) 130320037 4) 130320010 5) 130320001 6) 130280025 7) 130320035 8) 130280133 9) 130280182 10) 130280091 11) 130320036 12) 130280197 13) 130280037 14) 130280101 15) 130280204 16) 130280151 17) 130280200 18) 130280212 19) 130280089 20) 130280040 21) 130280121 22) 130280024 23) 130280220 24) 130280211 25) 130280207 26) 130280012 27) 130280068 28) 130280084 29) 130280131 30) 130280090 31) 130280215 32) 130280066 33) 130280085 34) 130280058 35) 130280149 36) 130280157 37) 130280176 38) 130280143 39) 130280146 40) 130280155 41) 130280218 42) 130280148 43) 130280153 44) 130280098 45) 130280156 46) 130280185 47) 130280150 48) 130280175 49) 130280110 50) 130280144 51) 130280092 52) 130280231 53) 130280213 54) 130280061 55) 130280219 56) 130280152 57) 130280208 58) 130280015 59) 130280024 60) 130280184 61) 130280066 62) 130280179 63) 130280223 64) 130280188 65) 130280168 66) 130280201 67) 130280062 68) 130280131 69) 130280209 70) 130280196 71) 130730067 72) 130280216 73) 130730098 74) 130280099 75) 130730102

El arbol18 cuya raíz tiene la clave 130280103 tiene 25 elementos y son los siguientes: 1) 130280222 2) 130280034 3) 130280096 4) 130280183 5) 130280002 6) 130280194 7) 130280186 8) 130280032 9) 130280180 10) 130280137 11) 130280125 12) 130280202 13) 130280063 14) 130280007 15) 130280069 16) 130280193 17) 130280046 18) 130260041 19) 130110005 20) 130260057 21) 130260005 22) 130260021 23) 130260028 24) 130110017 25) 130260007 26) 130260010

El arbol19 cuya raíz tiene la clave 130110035 tiene 15 elementos y son los siguientes: 1) 130260063 2) 130280102 3) 130260065 4) 130110051 5) 130110034 6) 130110030 7) 130110036 8) 130260016 9) 130110054 10) 130110045 11) 130110027 12) 130110063 13) 130110023 14) 130110011 15) 130110038

El arbol20 cuya raíz tiene la clave 130730021 tiene 34 elementos y son los siguientes: 1) 130620131 2) 130400035 3) 130620124 4) 130620043 5) 130730112 6) 130340025 7) 130340044 8) 130620045 9) 130340044 10) 130340012 11) 130730049 12) 130730020 13) 130340043 14) 130730042 15) 130730054 16) 130730041 17) 130340039 18) 130730027 19) 130730012 20) 130340057 21) 130730104 22) 130730006 23) 130340034 24) 130730105 25) 130730095 26) 130730030 27) 130340008 28) 130730101 29) 130730065 30) 130340019 31) 130730010 32) 130730051 33) 130730022 34) 130730032

El arbol21 cuya raíz tiene la clave 130470001 tiene 8 elementos y son los siguientes: 1) 130470011 2) 130470016 3) 130470009 4) 130470023 5) 130470038 6) 130310013 7) 130470032 8) 130310014

El arbol22 cuya raíz tiene la clave 130730038 tiene 16 elementos y son los siguientes: 1) 130730024 2) 130730055 3) 130730034 4) 130730050 5) 130730096 6) 130730046 7) 130730057 8) 130730014 9) 130730003 10) 130730093 11) 130730027 12) 130730109 13) 130730111 14) 130730119 15) 130730118 16) 130730061

El arbol23 cuya raíz tiene la clave 130310009 tiene 32 elementos y son los siguientes: 1) 130310015 2) 130310037 3) 130310016 4) 130310011 5) 130400030 6) 130400013 7) 130400005 8) 130400051 9) 130400081 10) 130400047 11) 130310038 12) 130310035 13) 130400045 14) 130400058 15) 130400097 16) 130400093 17) 130310010 18) 130310046 19) 130310027 20) 130310053 21) 130310064 22) 130310065 23) 130400021 24) 130310001 25) 130310035 26) 130400097 27) 130310063 28) 130310042 29) 130310024 30) 130310059 31) 130310049 32) 130310056

El arbol24 cuya raíz tiene la clave 130620005 tiene 25 elementos y son los siguientes: 1) 130620006 2) 130620014 3) 130620051 4) 130620132 5) 130620123 6) 130620024 7) 130620115 8) 130620135 9) 130620097 10) 130620106 11) 130620142 12) 130620015 13) 130620076 14) 130620112 15) 130620111 16) 130620093 17) 130710100 18) 130620102 19) 130620113 20) 130710097 21) 130620011 22) 130620013 23) 130620082 24) 130620014 25) 130620138

El arbol25 cuya raíz tiene la clave 130620004 tiene 22 elementos y son los siguientes: 1) 130620028 2) 130340003 3) 130620128 4) 130620029 5) 130340056 6) 130620055 7) 130620058 8) 130620069 9) 130620134 10) 130620146 11) 130620096 12) 130620007 13) 130620147 14) 130620084 15) 130620129 16) 130620137 17) 130620001 18) 130620022 19) 130620023 20) 130620122 21) 130620150 22) 130400007

El arbol26 cuya raíz tiene la clave 130260038 tiene 49 elementos y son los siguientes: 1) 130260029 2) 130260033 3) 130260049 4) 130260018 5) 130260037 6) 130260048 7) 130260015 8) 130260019 9) 130260027 10) 130260055 11) 130260043 12) 130260031 13) 130260044 14) 130260012 15) 130260053 16) 130260041 17) 130260020 18) 130260020 19) 130260030 20) 130260030 21) 130260030 22) 130260030 23) 130260030 24) 130260030 25) 130260030 26) 130260030 27) 130260030 28) 130260030 29) 130260030 30) 130260030 31) 130260030 32) 130260030 33) 130260030 34) 130260030 35) 130260030 36) 130260030 37) 130260030 38) 130260030 39) 130260030 40) 130260030 41) 130260030 42) 130260030 43) 130260030 44) 130260030 45) 130260030 46) 130260030 47) 130260030 48) 130260030 49) 130260030

El arbol27 cuya raíz tiene la clave 130400037 tiene 46 elementos y son los siguientes: 1) 130180098 2) 130180097 3) 130180014 4) 130400059 5) 130400056 6) 130400062 7) 130400051 8) 130400075 9) 130400078 10) 130400074 11) 130400082 12) 130400007 13) 130400084 14) 130400043 15) 130710126 16) 130400012 17) 130710043 18) 130710114 19) 130400073 20) 130710089 21) 130400033 22) 130710085 23) 130400076 24) 130400095 25) 130710124 26) 130710007 27) 130710066 28) 130710109 29) 130710092 30) 130710105 31) 130710122 32) 130710014 33) 130710115 34) 130710121 35) 130710046 36) 130710127 37) 130310021 38) 130710004 39) 130400057 40) 130710106 41) 130710022 42) 130710110 43) 130710104 44) 130710116 45) 130710102 46) 130710090

El arbol28 cuya raíz tiene la clave 130260013 tiene 21 elementos y son los siguientes: 1) 130260008 2) 130260014 3) 130260056 4) 130730044 5) 130730098 6) 130730113 7) 130260004 8) 130260049 9) 130730007 10) 130730045 11) 130260050 12) 130260033 13) 130730016 14) 130260009 15) 130260023 16) 130260018 17) 130260026 18) 130260046 19) 130260024 20) 130260017 21) 130260001

El arbol29 cuya raíz tiene la clave 130110001 tiene 28 elementos y son los siguientes: 1) 130110006 2) 130110063 3) 130110016 4) 130250035 5) 130250046 6) 130110020 7) 130250025 8) 130110025 9) 130110021 10) 130110026 11) 130250008 12) 130110050 13) 130110064 14) 130110027 15) 130110055 16) 130250067 17) 130110041 18) 130110062 19) 130110060 20) 130250033 21) 130110040 22) 130110033 23) 130110097 24) 130110058 25) 130250067 26) 130110061 27) 130250013 28) 130110031

El arbol30 cuya raíz tiene la clave 130730005 tiene 12 elementos y son los siguientes: 1) 130730039 2) 130730059 3) 130730036 4) 130730047 5) 130730043 6) 130730060 7) 130730009 8) 130730108 9) 130730025 10) 130260039 11) 130730002 12) 130730015

El arbol31 cuya raíz tiene la clave 130470008 tiene 26 elementos y son los siguientes: 1) 130470027 2) 130470010 3) 130470014 4) 130470065 5) 130470034 6) 130470020 7) 130470018 8) 130470039 9) 130470019 10) 130470042 11) 130470022 12) 130470006 13) 130470007 14) 130470013 15) 130470004 16) 130470040 17) 130470021 18) 130840058 19) 130470030 20) 130470035 21) 130470026 22) 130470093 23) 130840113 24) 130840153 25) 130840002 26) 130840068

El arbol32 cuya raíz tiene la clave 130340015 tiene 21 elementos y son los siguientes: 1) 130340029 2) 130730117 3) 130730110 4) 130340058 5) 130620087 6) 130340005 7) 130300117 8) 130620002 9) 130730013 10) 130340022 11) 130730108 12) 130340041 13) 130340002 14) 130420047 15) 130620016 16) 130730100 17) 130730063 18) 130340010 19) 130730099 20) 130730115 21) 130340031

El arbol33 cuya raíz tiene la clave 130800023 tiene 18 elementos y son los siguientes: 1) 130110008 2) 130110029 3) 130110019 4) 130110057 5) 130110031 6) 130110009 7) 130110037 8) 130110004 9) 130260027 10) 130110002 11) 130110039 12) 130800008 13) 130800028 14) 130800021 15) 130800001 16) 130800012 17) 130800011 18) 130800015

El arbol34 cuya raíz tiene la clave 130710012 tiene 21 elementos y son los siguientes: 1) 130710078 2) 130710015 3) 130710030 4) 130710130 5) 130710037 6) 130620008 7) 130710134 8) 130710118 9) 130710020 10) 130710048 11) 130420011 12) 130710095 13) 130420044 14) 130710129 15) 130710132 16) 130710049 17) 130710135 18) 130710026 19) 130710117 20) 130710052 21) 130420009

El arbol35 cuya raíz tiene la clave 130110032 tiene 15 elementos y son los siguientes: 1) 130110022 2) 130780033 3) 130800018 4) 130800031 5) 130800022 6) 130800014 7) 130800035 8) 130800038 9) 130800010 10) 130800007 11) 130800026 12) 130800024 13) 130800024 14) 130800020 15) 130800006

TESIS CON
FALLA DE ORIGEN

El arbol36 cuya raíz tiene la clave 130140007 tiene 17 elementos y son los siguientes: 1) 130260043 2) 130260051 3) 130260036 4) 130260011 5) 130260040 6) 130260032 7) 130260016 8) 130260052 9) 130140018 10) 130140084 11) 130140095 12) 130140013 13) 130140089 14) 130140096 15) 130140017 16) 130140070 17) 130140004

El arbol37 cuya raíz tiene la clave 130140009 tiene 36 elementos y son los siguientes: 1) 130730004 2) 130140074 3) 130260025 4) 130730028 5) 130140076 6) 130260054 7) 130140031 8) 130140015 9) 130140068 10) 130140062 11) 130140098 12) 130140022 13) 130140033 14) 130140102 15) 130140025 16) 130140099 17) 130140093 18) 130140026 19) 130140002 20) 130140078 21) 130140080 22) 130140027 23) 130140092 24) 130140028 25) 130140064 26) 130140052 27) 130140073 28) 130140001 29) 130140081 30) 130140069 31) 130140063 32) 130140065 33) 130140005 34) 130140105 35) 130140011 36) 130140106

El arbol38 cuya raíz tiene la clave 130710001 tiene 21 elementos y son los siguientes: 1) 130710131 2) 130710112 3) 130710036 4) 130710086 5) 130710093 6) 130710069 7) 130710079 8) 130710021 9) 130710076 10) 130710035 11) 130710067 12) 130710120 13) 130710094 14) 130710088 15) 130710065 16) 130710098 17) 130710034 18) 130710029 19) 130710010 20) 130710125 21) 130710072

El arbol39 cuya raíz tiene la clave 130800027 tiene 10 elementos y son los siguientes: 1) 130140014 2) 130800009 3) 130800016 4) 130800003 5) 130140010 6) 130800013 7) 130800017 8) 130800025 9) 130140008 10) 130800002

El arbol40 cuya raíz tiene la clave 130710002 tiene 11 elementos y son los siguientes: 1) 130710005 2) 130710066 3) 130710054 4) 130710058 5) 130710033 6) 130420025 7) 130710063 8) 130710077 9) 130710103 10) 130710019 11) 130710099

El arbol41 cuya raíz tiene la clave 130840019 tiene 47 elementos y son los siguientes: 1) 130310003 2) 130310048 3) 130310043 4) 130310033 5) 130310032 6) 130470002 7) 130310005 8) 130310032 9) 130310061 10) 130310020 11) 130470033 12) 130840131 13) 130310028 14) 130470024 15) 130310031 16) 130310008 17) 130470041 18) 130470116 19) 130310025 20) 130310019 21) 130470012 22) 130470015 23) 130430007 24) 130430032 25) 130840026 26) 130430049 27) 130840013 28) 130840022 29) 130430045 30) 130470025 31) 130840033 32) 130840020 33) 130840147 34) 130430047 35) 130840096 36) 130430026 37) 130840011 38) 130840048 39) 130840119 40) 130840111 41) 130430014 42) 130840122 43) 130430044 44) 130840156 45) 130430028 46) 130840010 47) 130430016

El arbol42 cuya raíz tiene la clave 130780017 tiene 14 elementos y son los siguientes: 1) 130250034 2) 130250002 3) 130250054 4) 130780025 5) 130780014 6) 130780038 7) 130780032 8) 130780047 9) 130780023 10) 130780007 11) 130780005 12) 130780010 13) 130780024 14) 130780011

El arbol43 cuya raíz tiene la clave 130140003 tiene 30 elementos y son los siguientes: 1) 130730033 2) 130730062 4) 130140016 5) 130140090 6) 130140019 7) 130140058 8) 130140058 9) 130440016 10) 130140091 11) 130140047 12) 130340007 13) 130140066 14) 130140037 15) 130140043 16) 130140046 17) 130140041 18) 130140096 19) 130140042 20) 130140104 21) 130140045 22) 130140092 23) 130140048 24) 130140039 25) 130140051 26) 130140094 27) 130140082 28) 130140100 29) 130140103 30) 130140063

El arbol44 cuya raíz tiene la clave 130340020 tiene 27 elementos y son los siguientes: 1) 130730116 2) 130730031 3) 130340055 4) 130340018 5) 130340051 6) 130730064 7) 130340040 8) 130340052 9) 130340021 10) 130340036 11) 130420038 12) 130420048 13) 130420046 14) 130340099 15) 130340013 16) 130420021 17) 130420013 18) 130420035 19) 130340030 20) 130420021 21) 130420036 22) 130420019 23) 130340023 24) 130340053 25) 130420019 26) 130340026 27) 130340001

El arbol45 cuya raíz tiene la clave 130310023 tiene 27 elementos y son los siguientes: 1) 130310060 2) 130310018 3) 130310041 4) 130710033 5) 130310017 6) 130710025 7) 130310022 8) 130710111 9) 130310007 10) 130710136 11) 130310029 12) 130340002 13) 130310044 14) 130310034 15) 130310045 16) 130710032 17) 130430020 18) 130430011 19) 130310030 20) 130710056 21) 130710096 22) 130710011 23) 130710123 24) 130710108 25) 130710041 26) 130710023 27) 130710087

El arbol46 cuya raíz tiene la clave 130790023 tiene 29 elementos y son los siguientes: 1) 130140055 2) 130140097 3) 130140075 4) 130140066 5) 130140065 6) 130140085 7) 130140085 8) 130140021 9) 130790026 10) 130680080 11) 130790017 12) 130790019 13) 130790099 14) 130140061 15) 130790019 16) 130680091 17) 130680094 18) 130680030 19) 130680049 20) 130680015 21) 130680034 22) 130680068 23) 130680021 24) 130680032 25) 130680027 26) 130680040 27) 130680014 28) 130680089 29) 130680010

El arbol47 cuya raíz tiene la clave 130330004 tiene 28 elementos y son los siguientes: 1) 130420017 2) 130330016 3) 130330093 4) 130710027 5) 130710058 6) 130330007 7) 130710059 8) 130710060 9) 130710070 10) 130710064 12) 130200011 13) 130710080 14) 130710061 15) 130710101 16) 130710003 17) 130710074 18) 130710083 19) 130710040 20) 130710099 21) 130710016 22) 130710024 23) 130710062 24) 130710113 25) 130710017 26) 130710082 27) 130710057 28) 130710081

El arbol48 cuya raíz tiene la clave 130780018 tiene 6 elementos y son los siguientes: 1) 130800019 2) 130780013 3) 130800029 4) 130800034 5) 130780021 6) 130780020

El arbol49 cuya raíz tiene la clave 130790020 tiene 7 elementos y son los siguientes: 1) 130680045 2) 130790006 3) 130680041 4) 130680008 5) 130790018 6) 130680017 7) 130680024

El arbol50 cuya raíz tiene la clave 130780001 tiene 20 elementos y son los siguientes: 1) 130780026 2) 130780042 3) 130780048 4) 130780049 5) 130780006 6) 130780040 7) 130780022 8) 130780031 9) 130780019 10) 130780016 11) 130780029 12) 130790019 13) 130780034 14) 130780028 15) 130790041 16) 130780004 17) 130780035 18) 130790002 19) 130790027 20) 130780030

El arbol51 cuya raíz tiene la clave 130680002 tiene 26 elementos y son los siguientes: 1) 130680038 2) 130800030 3) 130680051 4) 130680052 5) 130680037 6) 130680007 7) 130680013 8) 130680012 9) 130680085 10) 130680010 11) 130680053 12) 130680059 13) 130680029 14) 130680011 15) 130680044 16) 130680007 17) 130680023 18) 130680020 19) 130680084 20) 130680045 21) 130680050 22) 130680088 23) 130680078 24) 130680035 25) 130680047 26) 130680005

El arbol52 cuya raíz tiene la clave 130330002 tiene 20 elementos y son los siguientes: 1) 130420022 2) 130420043 3) 130420023 4) 130420045 5) 130330011 6) 130420034 7) 130420042 8) 130420040 9) 130420037 10) 130330018 11) 130420024 12) 130330005 13) 130420015 14) 130420031 15) 130420033 16) 130420030 17) 130420005 18) 130420029 19) 130330017 20) 130330001

El arbol53 cuya raíz tiene la clave 130840036 tiene 37 elementos y son los siguientes: 1) 130840069 2) 130840165 3) 130840031 4) 130840052 5) 130840152 6) 130840161 7) 130840154 8) 130840162 9) 130430018 10) 130840047 11) 130840110 12) 130840121 13) 130840066 14) 130430041 15) 130840135 16) 130840144 17) 130840065 18) 130840028 19) 130430042 20) 130840006 21) 130430029 22) 130840124 23) 130430031 24) 130840095 25) 130840173 26) 130840149 27) 130840023 28) 130840127 29) 130840040 30) 130840123 31) 130840158 32) 130840105 33) 130840174 34) 130430021 35) 130840164 36) 130840071 37) 130430004

El arbol54 cuya raíz tiene la clave 130790001 tiene 33 elementos y son los siguientes: 1) 130790024 2) 130420018 3) 130790013 4) 130790007 5) 130420027 6) 130790014 7) 130790011 8) 130790003 9) 130420051 10) 130420001 11) 130790022 12) 130790005 13) 130420050 14) 130790002 15) 130790035 16) 130790030 17) 130790033 18) 130790021 19) 130420006 20) 130420004 21) 130420014 22) 130790034 23) 130790029 24) 130790016 25) 130790012 26) 130790028 27) 130790025 28) 130420003 29) 130420008 30) 130790008 31) 130790004 32) 130420026 33) 130420016

El arbol55 cuya raíz tiene la clave 130290001 tiene 3 elementos y son los siguientes: 1) 130200003 2) 130200017 3) 130200018

El arbol56 cuya raíz tiene la clave 130430023 tiene 48 elementos y son los siguientes: 1) 130430017 2) 130430046 3) 130430030 4) 130430013 5) 130430033 6) 130430040 7) 130430003 8) 130710013 9) 130430038 10) 130430008 11) 130430043 12) 130430015 13) 130430023 14) 130430027 15) 130710018 16) 130150014 17) 130430001 18) 130430050 19) 130430006 20) 130710084 21) 130430035 22) 130430019 23) 130710038 24) 130430034 25) 130150002 26) 130150072 27) 130430009 28) 130150080 29) 130150018 30) 130150045 31) 130150069 32) 130150079 33) 130150090 34) 130430022 35) 130430010 36) 130150070 37) 130430004 38) 130150045 39) 130150004 40) 130150029 41) 130710075 42) 130150010 43) 130150078 44) 130150077 45) 130150030 46) 130150010 47) 130150016 48) 130150043

El arbol57 cuya raíz tiene la clave 130680025 tiene 24 elementos y son los siguientes: 1) 130790001 2) 130680076 3) 130680046 4) 130680025 5) 130680023 6) 130680009 7) 130680003 8) 130420012 9) 130680022 10) 130680092 11) 130680001 12) 130680056 13) 130680004 14) 130680035 15) 130680081 16) 130680006 17) 130680026 18) 130790017 19) 130680082 20) 130680079 21) 130680036 22) 130810022 23) 130680026 24) 130810092

El arbol58 cuya raíz tiene la clave 130840062 tiene 23 elementos y son los siguientes: 1) 130840060 2) 130840037 3) 130840082 4) 130840092 5) 130840103 6) 130840128 7) 130840081 8) 130840032 9) 130840101 10) 130840046 11) 130840053 12) 130840117 13) 130840057 14) 130840094 15) 130840139 16) 130840088 17) 130840100 18) 130840169 19) 130840023 20) 130840067 21) 130840177 22) 130840175 23) 130590084

TESIS CON
FALLA DE ORIGEN

El arbol59 cuya raíz tiene la clave 130200006 tiene 22 elementos y son los siguientes: 1) 130200024 2) 130200008 3) 130200005 4) 130200002 5) 130200012 6) 130200023 7) 130200009 8) 130200014 9) 130200025 10) 130200059 11) 130200013 12) 130200054 13) 130710008 14) 130200020 15) 130370129 16) 130200019 17) 130370017 18) 130370110 19) 130370106 20) 130200007 21) 130200004 22) 130200021

El arbol60 cuya raíz tiene la clave 130840027 tiene 46 elementos y son los siguientes: 1) 130840041 2) 130840089 3) 130840079 4) 130840044 5) 130840098 6) 130840007 7) 130840098 8) 130840090 10) 130840178 11) 130840115 12) 130840001 13) 130840168 14) 130840063 15) 130840133 16) 130840141 17) 130840160 18) 130840024 19) 130840130 20) 130840039 21) 130840106 22) 130840132 23) 130840043 24) 130840045 25) 130840099 26) 130840109 27) 130840055 28) 130840005 29) 130840145 30) 130840114 31) 130840014 32) 130840077 33) 130840118 34) 130840171 35) 130840078 36) 130840179 37) 130840016 38) 130840112 39) 130840056 40) 130840003 41) 130840176 42) 130840167 43) 130840018 44) 130840136 45) 130840064 46) 130840126

El arbol61 cuya raíz tiene la clave 130810008 tiene 14 elementos y son los siguientes: 1) 130860019 2) 130680057 3) 130680031 4) 130680016 5) 130680075 6) 130690074 7) 130810001 8) 130810007 9) 130810033 10) 130810048 11) 130810049 12) 130810043 13) 130810010 14) 130810023

El arbol62 cuya raíz tiene la clave 130300027 tiene 23 elementos y son los siguientes: 1) 130430005 2) 130300011 3) 130300059 4) 130300022 5) 130300036 6) 130300131 7) 130300047 8) 130300055 9) 130300056 10) 130300126 11) 130300012 12) 130300024 13) 130300049 14) 130840042 15) 130840172 16) 130300057 17) 130300107 18) 130300012 19) 130840051 20) 130300111 21) 130840054 22) 130300138 23) 130840160

El arbol63 cuya raíz tiene la clave 130150027 tiene 15 elementos y son los siguientes: 1) 130710073 2) 130150076 3) 130300052 4) 130150074 5) 130150006 6) 130150032 7) 130150022 8) 130300117 9) 130150061 10) 130150041 11) 130150011 12) 130150008 13) 130150025 14) 130150007 15) 130300124

El arbol64 cuya raíz tiene la clave 130840004 tiene 22 elementos y son los siguientes: 1) 130840155 2) 130840070 3) 130840053 4) 130840073 5) 130840072 6) 130840008 7) 130840137 8) 130840120 9) 130840075 10) 130840107 11) 130840093 12) 130840021 13) 130840050 14) 130840148 15) 130840029 16) 130840142 17) 130840074 18) 130840146 19) 130840170 20) 130840054 21) 130840061 22) 130850010

El arbol65 cuya raíz tiene la clave 130810025 tiene 13 elementos y son los siguientes: 1) 130810041 2) 130810004 3) 130810038 4) 130810037 5) 130810020 6) 130810024 7) 130810021 8) 130810009 9) 130810019 10) 130810031 11) 130810009 12) 130810046 13) 130810026

El arbol66 cuya raíz tiene la clave 130370059 tiene 31 elementos y son los siguientes: 1) 130370002 2) 130810005 3) 130370055 4) 130810047 5) 130370026 6) 130810050 7) 130370055 8) 130810001 9) 130810053 10) 130370019 11) 130370029 12) 130810014 13) 130370058 14) 130370066 15) 130370061 16) 130370049 17) 130370132 18) 130370001 19) 130370039 20) 130370058 21) 130370037 22) 130370083 23) 130370055 24) 130370066 25) 130370023 26) 130370024 27) 130370022 28) 130360040 29) 130370045 30) 130370046 31) 130370004

El arbol67 cuya raíz tiene la clave 130370038 tiene 52 elementos y son los siguientes: 1) 130370102 2) 130370050 3) 130200910 4) 130710071 5) 130370093 6) 130370114 7) 130370031 8) 130370084 9) 130370041 10) 130370122 11) 130370093 12) 130370113 13) 130370131 14) 130370047 15) 130370094 16) 130370107 17) 130370064 18) 130370071 19) 130370027 20) 130370051 21) 130370070 22) 130370013 23) 130370115 24) 130370073 25) 130370077 26) 130370103 27) 130370053 28) 130370042 29) 130370006 30) 130370076 31) 130370075 32) 130370014 33) 130370035 34) 130370020 35) 130370121 36) 130370101 37) 130370123 38) 130370028 39) 130370099 40) 130370128 41) 130370122 42) 130370040 43) 130370126 44) 130370060 45) 130370089 46) 130370108 47) 130370065 48) 130370018 49) 130370085 50) 130370082 51) 130370030 52) 130370052

El arbol68 cuya raíz tiene la clave 130150001 tiene 29 elementos y son los siguientes: 1) 130150015 2) 130150003 3) 130710119 4) 130150093 5) 130150054 6) 130150089 7) 130710028 8) 130150064 9) 130150086 10) 130150024 11) 130150047 12) 130150060 13) 130150031 14) 130150092 15) 130150011 16) 130150035 17) 130300051 18) 130150012 19) 130150009 20) 130150087 21) 130150037 22) 130150038 23) 130300010 24) 130150094 25) 130150040 26) 130150028 27) 130150021 28) 130150039 29) 130150069

El arbol69 cuya raíz tiene la clave 130810003 tiene 29 elementos y son los siguientes: 1) 130810044 2) 130810045 3) 130810054 4) 130810056 5) 130810052 6) 130810015 7) 130810012 8) 130810018 9) 130810042 10) 130810040 11) 130810039 12) 130810017 13) 130810028 14) 130360015 15) 130810016 16) 130360024 17) 130360009 18) 130360034 19) 130360027 20) 130360019 21) 130360042 22) 130360007 23) 130360001 24) 130360042 25) 130360045 26) 130360012 27) 130360037 28) 130360025 29) 130360043

El arbol70 cuya raíz tiene la clave 130270003 tiene 37 elementos y son los siguientes: 1) 130270103 2) 130270088 3) 130270083 4) 130270045 5) 130270086 6) 130270021 7) 130270031 8) 130270073 9) 130270016 10) 130270085 11) 130270072 12) 130270008 13) 130270027 14) 130270054 15) 130270048 16) 130270013 17) 130270101 18) 130270042 19) 130270113 20) 130350100 21) 130270046 22) 130270012 23) 130270106 24) 130270044 25) 130270039 26) 130270009 27) 130270043 28) 130270029 29) 130350131 30) 130350127 31) 130270087 32) 130270014 33) 130350110 34) 130270007 35) 130270050 36) 130350072 37) 130350160

El arbol71 cuya raíz tiene la clave 130360006 tiene 4 elementos y son los siguientes: 1) 130810011 2) 130360020 3) 130360008 4) 130360004

El arbol72 cuya raíz tiene la clave 130300034 tiene 33 elementos y son los siguientes: 1) 130300116 2) 130300108 3) 130300110 4) 130300008 5) 130300109 6) 130300033 7) 130300007 8) 130300058 9) 130300064 10) 130300012 11) 130300018 12) 130300127 13) 130300042 14) 130300014 15) 130300039 16) 130300130 17) 130300021 18) 130300061 19) 130300031 20) 130300017 21) 130300133 22) 130300129 23) 130300041 24) 130300123 25) 130300043 26) 130300101 27) 130300053 28) 130300065 29) 130300023 30) 130300015 31) 130300029 32) 130300104 33) 130300137

El arbol73 cuya raíz tiene la clave 130270036 tiene 18 elementos y son los siguientes: 1) 130270020 2) 130270099 3) 130270053 4) 130270049 5) 130270104 6) 130270064 7) 130270051 8) 130270062 9) 130350132 10) 130270052 11) 130270093 12) 130270095 13) 130270093 14) 130350046 15) 130270098 16) 130270096 17) 130270047 18) 130270037

El arbol74 cuya raíz tiene la clave 130580004 tiene 49 elementos y son los siguientes: 1) 130840150 2) 130840076 3) 130840151 4) 130580011 5) 130840071 6) 130580016 7) 130580008 8) 130580039 9) 130580006 10) 130580021 11) 130580025 12) 130580027 13) 130580002 14) 130580018 15) 130580042 16) 130580041 17) 130580041 18) 130580001 19) 130580013 20) 130580036 21) 130580023 22) 130580037 23) 130580044 24) 130580034 25) 130580046 26) 130580045 27) 130580015 28) 130580003 29) 130580001 30) 130580019 31) 130580012 32) 130580020 33) 130580009 34) 130580024 35) 130600017 36) 130600016 37) 130600037 38) 130600049 39) 130580017 40) 130300025 41) 130300075 42) 130600031 43) 130580032 44) 130600042 45) 130300035 46) 106060039 47) 130600068 48) 130600021 49) 130600044

El arbol75 cuya raíz tiene la clave 130590028 tiene 14 elementos y son los siguientes: 1) 130590008 2) 130590055 3) 130590038 4) 130590054 5) 130590006 6) 130590032 7) 130590076 8) 130590022 9) 130580014 10) 130590074 11) 130590015 12) 130590005 13) 130590092 14) 130590089

El arbol76 cuya raíz tiene la clave 130150020 tiene 35 elementos y son los siguientes: 1) 130150065 2) 130150051 3) 130150023 4) 130150050 5) 130150052 6) 130150048 7) 130150103 8) 130150109 9) 130150049 10) 130150026 11) 130370104 12) 130150058 13) 130370068 14) 130150059 15) 130150107 16) 130150095 17) 130150035 18) 130150106 19) 130500023 20) 130370062 21) 130150057 22) 130150013 23) 130150083 24) 130150082 25) 130370067 26) 130150104 27) 130150101 28) 130150085 29) 130150108 30) 130150019 31) 130150112 32) 130150126 33) 130150088 34) 130150097 35) 130550020

El arbol77 cuya raíz tiene la clave 130300008 tiene 35 elementos y son los siguientes: 1) 130300063 2) 130150005 3) 130150067 4) 130150068 5) 130300105 6) 130300139 7) 130300009 8) 130300122 9) 130300100 10) 130300119 11) 130300115 12) 130300136 13) 130300062 14) 130300120 15) 130300135 16) 130300003 17) 130300112 18) 130300118 19) 130300025 20) 130300134 21) 130300113 22) 130300114 23) 130300097 24) 130300035 25) 130300019 26) 130300032 27) 130300004 28) 130300001 29) 130300099 30) 130300069 31) 130300102 32) 130300016 33) 130300087 34) 130300038 35) 130300121

El arbol78 cuya raíz tiene la clave 130590031 tiene 5 elementos y son los siguientes: 1) 130590050 2) 130590029 3) 130590039 4) 130590017 5) 130590077

El arbol79 cuya raíz tiene la clave 130530067 tiene 29 elementos y son los siguientes: 1) 130530125 2) 130530139 3) 130530115 4) 130530077 5) 130530039 6) 130530076 7) 130530154 8) 130530114 9) 130530147 10) 130530092 11) 130530150 12) 130530021 13) 130530085 14) 130530005 15) 130530122 16) 130530012 17) 130530084 18) 130530113 19) 130530167 20) 130530120 21) 130530134 22) 130530134 23) 130530004 24) 130270022 25) 130530157 26) 130530014 27) 130530068 28) 130530013 29) 130530149 30) 130530151

El arbol80 cuya raíz tiene la clave 130590019 tiene 40 elementos y son los siguientes: 1) 130590018 2) 130590036 3) 130590025 4) 130590049 5) 130590026 6) 130590024 7) 130590027 8) 130590090 9) 130590048 10) 130590083 11) 130590021 12) 130590037 13) 130590097

TESIS CON
FALLA DE ORIGEN

- 14) 130590010 15) 130590013 16) 130590014 17) 130590019 18) 130590100 19) 130590099 20) 130590016 21) 130590001 22) 130590086 23) 130590043 24) 130590098 25) 130590046 26) 130590023 27) 130590028 28) 130590088 29) 130590007 30) 130590011 31) 130590012 32) 130590035 33) 130590029 34) 130590035 35) 130590095 36) 130590052 37) 130590053 38) 130590067 39) 130590080 40) 130290026
- El arbol81 cuya raíz tiene la clave 130590030 tiene 5 elementos y son los siguientes: 1) 130590073 2) 130590004 3) 130590069 4) 130590072 5) 130590040
- El arbol82 cuya raíz tiene la clave 130600030 tiene 15 elementos y son los siguientes: 1) 130590002 2) 130600066 3) 130600018 4) 130600022 5) 130600032 6) 130600073 7) 130600009 8) 130600020 9) 130600047 10) 130600067 11) 130600071 12) 130600056 13) 130600013 14) 130600010 15) 130600005
- El arbol83 cuya raíz tiene la clave 130270001 tiene 20 elementos y son los siguientes: 1) 130270002 2) 130270015 3) 130270092 4) 130270091 5) 130270100 6) 130270111 7) 130270103 8) 130270018 9) 130270005 10) 130270110 11) 130270059 12) 130270034 13) 130270040 14) 130270010 15) 130270030 16) 130270058 17) 130270065 18) 130270038 19) 130270107 20) 130270090
- El arbol84 cuya raíz tiene la clave 130530001 tiene 40 elementos y son los siguientes: 1) 130530111 2) 130530049 3) 130530054 4) 130530042 5) 130530148 6) 130530069 7) 130530094 8) 130530019 9) 130530018 10) 130530017 11) 130530048 12) 130530073 13) 130530093 14) 130530057 15) 130530145 16) 130530061 17) 130530112 18) 130530058 19) 130530050 20) 130530034 21) 130530066 22) 130530041 23) 130530086 24) 130530165 25) 130530028 26) 130530030 27) 130530036 28) 130530081 29) 130530026 30) 130530023 31) 130530164 32) 130530066 33) 130530141 34) 130530125 35) 130530142 36) 130530060 37) 130530044 38) 130530149 39) 130530079 40) 130530090
- El arbol85 cuya raíz tiene la clave 130290004 tiene 5 elementos y son los siguientes: 1) 130290089 2) 130290013 3) 130290096 4) 130290048 5) 130290010
- El arbol86 cuya raíz tiene la clave 130290011 tiene 24 elementos y son los siguientes: 1) 130590034 2) 130590008 3) 130290028 4) 130290060 5) 130290110 6) 130290065 7) 130290103 8) 130590059 9) 130290088 10) 130290076 11) 130290019 12) 130290033 13) 130290008 14) 130290105 15) 130290108 16) 130290100 17) 130290109 18) 130290027 19) 130290042 20) 130290032 21) 130290044 22) 130290020 23) 130290034 24) 130290069
- El arbol87 cuya raíz tiene la clave 130290015 tiene 15 elementos y son los siguientes: 1) 130290002 2) 130590020 3) 130290040 4) 130290066 5) 130290070 6) 130290077 7) 130290079 8) 130290106 9) 130290111 10) 130290043 11) 130290039 12) 130290038 13) 130290011 14) 130290102 15) 130290047
- El arbol88 cuya raíz tiene la clave 130360002 tiene 39 elementos y son los siguientes: 1) 130360010 2) 130360018 3) 130360011 4) 130360005 5) 130370008 6) 130360026 7) 130360021 8) 130360017 9) 130360035 10) 130370072 11) 130360032 12) 130370009 13) 130360014 14) 130370088 15) 130360022 16) 130360013 17) 130360029 18) 130360039 19) 130360038 20) 130360031 21) 130370109 22) 130360036 23) 130360003 24) 130360016 25) 130360030 26) 130120074 27) 130360023 28) 130360041 29) 130120041 30) 130120069 31) 130120019 32) 130120016 33) 130120059 34) 130120021 35) 130120012 36) 130120024 37) 130120030 38) 130120067 39) 130120014
- El arbol89 cuya raíz tiene la clave 130270004 tiene 37 elementos y son los siguientes: 1) 130530159 2) 130530035 3) 130530152 4) 130530029 5) 130530070 6) 130530161 7) 130270004 8) 130530153 9) 130530053 10) 130530155 11) 130270011 12) 130270102 13) 130530075 14) 130530163 15) 130270055 16) 130270028 17) 130270070 18) 130530118 19) 130270079 20) 130270024 21) 130270059 22) 130270109 23) 130270077 24) 130270032 25) 130270089 26) 130270076 27) 130270026 28) 130270035 29) 130270025 30) 130270112 31) 130600027 32) 130600029 33) 130600060 34) 130600019 35) 130600058 36) 130600032 37) 130600049
- El arbol90 cuya raíz tiene la clave 130120003 tiene 40 elementos y son los siguientes: 1) 130370057 2) 130370098 3) 130370096 4) 130370087 5) 130370042 6) 130370010 7) 130370007 8) 130370027 9) 130370100 10) 130370090 11) 130370017 12) 130370044 13) 130370019 14) 130370011 15) 130370130 16) 130370018 17) 130120051 18) 130120051 19) 130120045 20) 130120055 21) 130120066 22) 130120067 23) 130120048 24) 130120050 25) 130120029 26) 130120065 27) 130120061 28) 130120072 29) 130120084 30) 130120066 31) 130120042 32) 130120008 33) 130120027 34) 130120036 35) 130120056 36) 130120035 37) 130120023 38) 130120064 39) 130120063
- El arbol91 cuya raíz tiene la clave 130530001 tiene 55 elementos y son los siguientes: 1) 130530158 2) 130530129 3) 130530071 4) 130530109 5) 130530080 6) 130530095 7) 130530128 8) 130530027 9) 130530025 10) 130530054 11) 130530133 12) 130530130 13) 130530037 14) 130530011 15) 130530038 16) 130530065 17) 130530138 18) 130530166 19) 130530056 20) 130530106 21) 130270006 22) 130530144 23) 130530126 24) 130530010 25) 130530162 26) 130530064 27) 130530074 28) 130530099 29) 130530103 30) 130530024 31) 130270023 32) 130530063 33) 130600024 34) 130530093 35) 130600016 36) 130530136 37) 130600045 38) 130530008 39) 130600006 40) 130530009 41) 130530043 42) 130600065 43) 130600043 44) 130600064 45) 130530156 46) 130600059 47) 130600036 48) 130600057 49) 130600047 50) 130600050 51) 130600054 52) 130600002 53) 130600043 54) 130600052 55) 130600035
- El arbol92 cuya raíz tiene la clave 130290018 tiene 17 elementos y son los siguientes: 1) 130290055 2) 130290017 3) 130290005 4) 130290083 5) 130290064 6) 130290009 7) 130290093 8) 130290066 9) 130290084 10) 130290097 11) 130290023 12) 130290001 13) 130290035 14) 130290022 15) 130290012 16) 130290037 17) 130290024
- El arbol93 cuya raíz tiene la clave 130540040 tiene 19 elementos y son los siguientes: 1) 130300066 2) 130300048 3) 130300047 4) 130300096 5) 130300028 6) 130300046 7) 130300002 8) 130300030 9) 130540036 10) 130300103 11) 130190016 12) 130540037 13) 130540039 14) 130300026 15) 130540038 16) 130190007 17) 130190010 18) 130540035 19) 130190011
- El arbol94 cuya raíz tiene la clave 130550001 tiene 23 elementos y son los siguientes: 1) 130370033 2) 130370015 3) 130550010 4) 130550005 5) 130550002 6) 130550022 7) 130370021 8) 130550019 9) 130370074 10) 130550009 11) 130550006 12) 130300035 13) 130300017 14) 130550016 15) 130550007 16) 130550008 17) 130300029 18) 130550018 19) 130550015 20) 130550014 21) 130300004 22) 130540050 23) 130030042
- El arbol95 cuya raíz tiene la clave 130550013 tiene 9 elementos y son los siguientes: 1) 130550021 2) 130550012 3) 130550017 4) 130550011 5) 130550003 6) 130550004 7) 130540005 8) 130540013 9) 130540006
- El arbol96 cuya raíz tiene la clave 130040001 tiene 31 elementos y son los siguientes: 1) 130040023 2) 130040016 3) 130040021 4) 130530031 5) 130040015 6) 130040020 7) 130040009 8) 130530016 9) 130040022 10) 130530007 11) 130040029 12) 130040013 13) 130040008 14) 130040031 15) 130530020 16) 130040017 17) 130040007 18) 130040019 19) 130040028 20) 130040018 21) 130530022 22) 130040005 23) 130040032 24) 130040010 25) 130040006 26) 130040006 27) 130040003 28) 130040026 29) 130040006 30) 130040002 31) 130040011
- El arbol97 cuya raíz tiene la clave 130600027 tiene 8 elementos y son los siguientes: 1) 130600059 2) 130600003 3) 130600075 4) 130600040 5) 130600060 6) 130600034 7) 130170029 8) 130170008
- El arbol98 cuya raíz tiene la clave 130600001 tiene 35 elementos y son los siguientes: 1) 130530107 2) 130530015 3) 130600030 4) 130530083 5) 130600038 6) 130530082 7) 130600053 8) 130600004 9) 130600014 10) 130600003 11) 130600002 12) 130600020 13) 130600067 14) 130600031 15) 130600012 16) 130600013 17) 130600023 18) 130600062 19) 130600011 20) 130600026 21) 130600008 22) 130600021 23) 130600050 24) 130600061 25) 130600040 26) 130600044 27) 130600063 28) 130600015 29) 130600041 30) 130600010 31) 130600034 32) 130600028 33) 130600035 34) 130600022 35) 130600037
- El arbol99 cuya raíz tiene la clave 130600033 tiene 35 elementos y son los siguientes: 1) 130600036 2) 130600012 3) 130600051 4) 130600014 5) 130600050 6) 130600029 7) 130600058 8) 130600052 9) 130600001 10) 130600041 11) 130600054 12) 130600019 13) 130600063 14) 130600011 15) 130600064 16) 130600004 17) 130600028 18) 130600006 19) 130600070 20) 130600038 21) 130600008 22) 130600002 23) 130600076 24) 130600025 25) 130600015 26) 130600025 27) 130600035 28) 130600072 29) 130600069 30) 130600024 31) 130600043 32) 130600074 33) 130600007 34) 130170015 35) 130170016
- El arbol100 cuya raíz tiene la clave 130190001 tiene 26 elementos y son los siguientes: 1) 130300013 2) 130190022 3) 130190004 4) 130190003 5) 130190002 6) 130190020 7) 130190019 8) 130190014 9) 130190009 10) 130190027 11) 130190005 12) 130190017 13) 130190010 14) 130190006 15) 130190024 16) 130190021 17) 130500017 18) 130500006 19) 130190025 20) 130190023 21) 130500011 22) 130500018 23) 130190012 24) 130500008 25) 130500003 26) 130500015
- El arbol101 cuya raíz tiene la clave 130290007 tiene 28 elementos y son los siguientes: 1) 130290003 2) 130290029 3) 130290013 4) 130290014 5) 130290053 6) 130290016 7) 130290051 8) 130290051 9) 130290049 10) 130440080 11) 130440124 12) 130440037 13) 130440084 14) 130440075 15) 130440065 16) 130440085 17) 130440057 18) 130440064 19) 130440083 20) 130440082 21) 130440025 22) 130440078 23) 130440029 24) 130440109 25) 130440120 26) 130440072 27) 130440065 28) 130440005

TESIS CON
FALLA DE ORIGEN

El arbol102 cuya raíz tiene la clave 130120011 tiene 32 elementos y son los siguientes: 1) 130120040 2) 130120032 3) 130120071 4) 130120020 5) 130120022 6) 130120044 7) 130120055 8) 130120017 9) 130120046 10) 130120052 11) 130120039 12) 130120076 13) 130120002 14) 130120004 15) 130120009 16) 130240022 17) 130120057 18) 130120075 19) 130120001 20) 130240023 21) 130120028 22) 130240078 23) 130120018 24) 130240005 25) 130240006 26) 130120006 27) 130120058 28) 130240068 29) 130120037 30) 130240012 31) 130120038 32) 130240035.

El arbol103 cuya raíz tiene la clave 130350017 tiene 15 elementos y son los siguientes: 1) 130400012 2) 130350012 3) 130350031 4) 130350005 5) 130350009 6) 130360048 7) 130350029 8) 130350032 9) 130350015 10) 130350004 11) 130350021 12) 130350027 13) 130350035 14) 130350020 15) 130350011

El arbol104 cuya raíz tiene la clave 130170001 tiene 11 elementos y son los siguientes: 1) 130170018 2) 130170023 3) 130170043 4) 130170049 5) 130170046 6) 130170009 7) 130170025 8) 130170002 9) 130640012 10) 130640004 11) 130640013

El arbol105 cuya raíz tiene la clave 130540001 tiene 41 elementos y son los siguientes: 1) 130540011 2) 130540023 3) 130540012 4) 130540007 5) 130540003 6) 130540010 7) 130540032 8) 130540046 9) 130540034 10) 130540047 11) 130540044 12) 130540029 13) 130540004 14) 130540008 15) 130540021 16) 130540042 17) 130030012 18) 130540020 19) 130030040 20) 130540014 21) 130540002 22) 130540030 23) 130540017 24) 130540015 25) 130540018 26) 130030005 27) 130540019 28) 130230040 29) 130230026 30) 130230012 31) 130540027 32) 130230013 33) 130540024 34) 130540021 35) 130230045 36) 130540048 37) 130540023 38) 130230043 39) 130540026 40) 130540028 41) 130230039

El arbol106 cuya raíz tiene la clave 130010011 tiene 14 elementos y son los siguientes: 1) 130400014 2) 130010007 3) 130350008 4) 130010032 5) 130010047 6) 130010043 7) 130010048 8) 130010059 9) 130350023 10) 130010018 11) 130010028 12) 130010050 13) 130010019 14) 130350018

El arbol107 cuya raíz tiene la clave 130440030 tiene 19 elementos y son los siguientes: 1) 130290030 2) 130290021 3) 130290090 4) 130290025 5) 130290073 6) 130440116 7) 130290072 8) 130440056 9) 130440119 10) 130440056 11) 130440038 12) 130440024 13) 130440062 14) 130440102 15) 130440039 16) 130440071 17) 130440098 18) 130440115 19) 130440025

El arbol108 cuya raíz tiene la clave 130090008 tiene 22 elementos y son los siguientes: 1) 130030024 2) 130030026 3) 130030013 4) 130030011 5) 130030007 6) 130030006 7) 130030037 8) 130030020 9) 130030001 10) 130090006 11) 130030015 12) 130030018 13) 130030019 14) 130090005 15) 130030025 16) 130090016 17) 130030052 18) 130090012 19) 130030028 20) 130030014 21) 130090015 22) 130090001

El arbol109 cuya raíz tiene la clave 130400022 tiene 17 elementos y son los siguientes: 1) 130240037 2) 130240010 3) 130100049 4) 130240018 5) 130010022 6) 130010022 7) 130010048 8) 130240076 9) 130240029 10) 130010033 11) 130240066 12) 130240077 13) 130240073 14) 130240016 15) 130240041 16) 130010020 17) 130240066

El arbol110 cuya raíz tiene la clave 130230005 tiene 43 elementos y son los siguientes: 1) 130410025 2) 130500012 3) 130410003 4) 130500007 5) 130230007 6) 130410006 7) 130500013 8) 130500009 9) 130230019 10) 130500004 11) 130500002 12) 130230013 13) 130230009 14) 130230041 15) 130410022 16) 130230004 17) 130230004 18) 130410024 19) 130230010 20) 130230001 21) 130230028 22) 130230003 23) 130230024 24) 130410028 25) 130230022 26) 130230036 27) 130410027 28) 130410031 29) 130230003 30) 130410029 31) 130230032 32) 130230046 33) 130410007 34) 130230008 35) 130230044 36) 130230037 37) 130230006 38) 130230015 39) 130230030 40) 130230014 41) 130230002 42) 130230042 43) 130230005

El arbol111 cuya raíz tiene la clave 130030023 tiene 20 elementos y son los siguientes: 1) 130030027 2) 130030002 3) 130030039 4) 130030021 5) 130120008 6) 130120043 7) 130030018 8) 130120023 9) 130030016 10) 130380022 11) 130030019 12) 130090004 13) 130380020 14) 130380008 15) 130090017 16) 130380027 17) 130090009 18) 130380021 19) 130090021 20) 130090018

El arbol112 cuya raíz tiene la clave 130170020 tiene 13 elementos y son los siguientes: 1) 130170021 2) 130170042 3) 130170005 4) 130170024 5) 130170007 6) 130170017 7) 130170022 8) 130170014 9) 130640023 10) 130640010 11) 130640002 12) 130640007 13) 130170044

El arbol113 cuya raíz tiene la clave 130190015 tiene 27 elementos y son los siguientes: 1) 130500005 2) 130190013 3) 130190008 4) 130500001 5) 130190026 6) 130410011 7) 130410001 8) 130410021 9) 130670004 10) 130670012 11) 130410035 12) 130670007 13) 130410009 14) 130410002 15) 130410034 16) 130670032 17) 130670030 18) 130670001 19) 130410010 20) 130410008 21) 130670021 22) 130410004 23) 130670029 24) 130410005 25) 130670035 26) 130650002 27) 130700014

El arbol114 cuya raíz tiene la clave 130440013 tiene 26 elementos y son los siguientes: 1) 130440087 2) 130440099 3) 130440031 4) 130440001 5) 130440007 6) 130440011 7) 130440041 8) 130170040 9) 130440079 10) 130440070 11) 130440046 12) 130440027 13) 130440126 14) 130440018 15) 130440047 16) 130440017 17) 130440014 18) 130440074 19) 130440020 20) 130440002 21) 130440048 22) 130440049 23) 130440100 24) 130440019 25) 130440004 26) 130440076

El arbol115 cuya raíz tiene la clave 130350001 tiene 27 elementos y son los siguientes: 1) 130660009 2) 130660051 3) 130350026 4) 130350028 5) 130350024 6) 130350019 7) 130350038 8) 130350022 9) 130350025 10) 130350013 11) 130350034 12) 130350036 13) 130350004 14) 130350014 15) 130350003 16) 130350007 17) 130770074 18) 130770008 19) 130770008 20) 130770018 21) 130770048 22) 130770011 23) 130770007 24) 130770089 25) 130770078 26) 130770079 27) 130770041

El arbol116 cuya raíz tiene la clave 130170003 tiene 15 elementos y son los siguientes: 1) 130230003 2) 130230036 3) 130230046 4) 130230082 5) 130170038 6) 130170037 7) 130170033 8) 130170047 9) 130170013 10) 130170048 11) 130440016 12) 130170041 13) 130170012 14) 130440063 15) 130440052

El arbol117 cuya raíz tiene la clave 130240027 tiene 37 elementos y son los siguientes: 1) 130240013 2) 130240002 3) 130240070 4) 130240031 5) 130240026 6) 130240030 7) 130240072 8) 130240079 9) 130240011 10) 130240024 11) 130240033 12) 130240009 13) 130240047 14) 130240069 15) 130240060 16) 130240066 17) 130240062 18) 130240052 19) 130240057 20) 130240058 21) 130240052 22) 130240061 23) 130240001 24) 130240055 25) 130240053 26) 130240051 27) 130240032 28) 130240058 29) 130240045 30) 130240003 31) 130240020 32) 130240043 33) 130240067 34) 130240075 35) 130240050 36) 130240036 37) 130240074

El arbol118 cuya raíz tiene la clave 130020026 tiene 62 elementos y son los siguientes: 1) 130020014 2) 130020004 3) 130020011 4) 130020032 5) 130020048 6) 130020050 7) 130020027 8) 130020046 9) 130020013 10) 130020008 11) 130020009 12) 130020047 13) 130020028 14) 130020052 15) 130020034 16) 130020010 17) 130020036 18) 130020030 19) 130020026 20) 130020015 21) 130020059 22) 130020004 23) 130020044 24) 130020012 25) 130020045 26) 130020001 27) 130020007 28) 130020020 29) 130020007 30) 130020040 31) 130020018 32) 130020020 33) 130020029 34) 130020038 35) 130020035 36) 130020050 37) 130020022 38) 130020051 39) 130020041 40) 130020041 41) 130020021 42) 130020016 43) 130020003 44) 130020042 45) 130020039 46) 130020041 47) 130020076 48) 130020049 49) 130020002 50) 130020014 51) 130020006 52) 130020005 53) 130160010 54) 130160014 55) 130160041 56) 130160011 57) 130160104 58) 130160020 59) 130160081 60) 130160024 61) 130160012 62) 130160007

El arbol119 cuya raíz tiene la clave 130380004 tiene 22 elementos y son los siguientes: 1) 130120068 2) 130120073 3) 130120015 4) 130380013 5) 130120031 6) 130120034 7) 130380023 8) 130380005 9) 130380010 10) 130380031 11) 130380013 12) 130380039 13) 130450004 14) 130380025 15) 130380033 16) 130380026 17) 130380041 18) 130380042 19) 130380001 20) 130380028 21) 130380038 22) 130380006

El arbol120 cuya raíz tiene la clave 130240021 tiene 11 elementos y son los siguientes: 1) 130240015 2) 130240017 3) 130240038 4) 130010012 5) 130240004 6) 130240042 7) 130240071 8) 130010033 9) 130240044 10) 130450040 11) 130240034

El arbol121 cuya raíz tiene la clave 130440010 tiene 27 elementos y son los siguientes: 1) 130440002 2) 130440015 3) 130440073 4) 130440006 5) 130440128 6) 130440009 7) 130440012 8) 130440008 9) 130440001 10) 130440097 12) 130440017 13) 130440045 14) 130440119 15) 130440129 16) 130440101 17) 130440110 18) 130440089 19) 130440082 20) 130440055 21) 130440111 22) 130440112 23) 130440050 24) 130440089 25) 130440023 26) 130440077 27) 130440092

El arbol122 cuya raíz tiene la clave 130010010 tiene 14 elementos y son los siguientes: 1) 130010057 2) 130010025 3) 130010003 4) 130010050 5) 130010058 6) 130010029 7) 130010016 8) 130770002 9) 130010023 10) 130010061 11) 130010024 12) 130010025 13) 130010005 14) 130010051

El arbol123 cuya raíz tiene la clave 130380007 tiene 20 elementos y son los siguientes: 1) 130380040 2) 130380017 3) 130380009 4) 130380016 5) 130380003 6) 130380014 7) 130380006 8) 130380015 9) 130380035 10) 130380032 11) 130480150 12) 130480148 13) 130480049 14) 130480205 15) 130480209 16) 130480204 17) 130480207 18) 130480001 19) 130510060 20) 130510063

El arbol124 cuya raíz tiene la clave 130450001 tiene 60 elementos y son los siguientes: 1) 130450029 2) 130450016 3) 130450009 4) 130240025 5) 130450003 6) 130450035 7) 130450021 8) 130450020 9) 130450007 10) 130450017 11) 130450008 12) 130450018 13) 130450032 14) 130450028 15) 130450016 16) 130450030 17) 130450031 18) 130450038 19) 130450006 20) 130590002 21) 130450038 22) 130450031 23) 130450011 24) 130590012 25) 130450023 26) 130450077 27) 130450092

El arbol125 cuya raíz tiene la clave 130390001 tiene 30 elementos y son los siguientes: 1) 130390028 2) 130390028 3) 130390036 4) 130390012 24) 130390030 25) 130450023 26) 130450027 27) 130390032 28) 130450012 29) 130390028 30) 130390036 31) 130450026 32) 130390016

**TESIS CON
FALLA DE ORIGEN**

- 33) 130390004 34) 130450002 35) 130450005 36) 130450019 37) 130450022 38) 130450010 39) 130390007 40) 130390023 41) 130450015 42) 130390001 43) 130390055 44) 130390059 45) 130390079 46) 130390088 47) 130390014 48) 130510064 49) 130390026 50) 130390009 51) 130510016 52) 130510062 53) 130510061 54) 130390058 55) 130510015 56) 130390003 57) 130220004 58) 130510021 59) 130220076 60) 130510003
- El arbol125 cuya raíz tiene la clave 130470033 tiene 11 elementos y son los siguientes: 1) 130170006 2) 130440054 3) 130170030 4) 130440085 5) 130440051 6) 130440108 7) 130170045 8) 130170006 9) 130440123 10) 130440091 11) 130440121
- El arbol126 cuya raíz tiene la clave 130520019 tiene 37 elementos y son los siguientes: 1) 130382012 2) 130390010 3) 130390027 4) 130380018 5) 130290014 6) 130390013 7) 130520046 8) 130280024 9) 130520051 10) 130520059 11) 130520014 12) 130480210 13) 130140208 14) 130520018 15) 130520006 16) 130480219 17) 130480217 18) 130480201 19) 130520014 20) 130480206 21) 130480203 22) 130480190 23) 130480158 24) 130480202 25) 130480188 26) 130480193 27) 130510066 28) 130480153 29) 130480213 30) 130510033 31) 130510057 32) 130510017 33) 130480154 34) 130510058 35) 130480214 36) 130480216 37) 130480215
- El arbol127 cuya raíz tiene la clave 130520013 tiene 45 elementos y son los siguientes: 1) 130030009 2) 130540022 3) 130090024 4) 130030038 5) 130090011 6) 130090026 7) 130090003 8) 130090022 9) 130090025 10) 130090020 11) 130540049 12) 130090023 13) 130030010 14) 130030043 15) 130090019 16) 130090007 17) 130090008 18) 130090028 19) 130620015 20) 130520054 21) 130520037 22) 130520041 23) 130520056 24) 130520039 25) 130520026 26) 130520033 27) 130520028 28) 130520016 29) 130520036 30) 130520017 31) 130520040 32) 130520032 33) 130520064 34) 130520091 35) 130520009 36) 130520063 37) 130520063 38) 130520065 39) 130520047 40) 130520011 41) 130520048 42) 130520060 43) 130520067 44) 130520044 45) 130520034
- El arbol128 cuya raíz tiene la clave 130640003 tiene 27 elementos y son los siguientes: 1) 130640011 2) 130670002 3) 130670037 4) 130640001 5) 130640019 6) 130640031 7) 130640020 8) 130670003 9) 130640006 10) 130640016 11) 130670028 12) 130640008 13) 130670022 14) 130670024 15) 130670015 16) 130640019 17) 130670033 18) 130640017 19) 130670019 20) 130670003 21) 130640009 22) 130670025 23) 130670026 24) 130670014 25) 130670046 26) 130670018 27) 130670007
- El arbol129 cuya raíz tiene la clave 130010001 tiene 102 elementos y son los siguientes: 1) 130010055 2) 130010027 3) 130010040 4) 130010041 5) 130010031 6) 130770065 7) 130010004 8) 130770068 9) 130770069 10) 130010052 11) 130010017 12) 130770042 13) 130770073 14) 130010015 15) 130770057 16) 130010030 17) 130010005 18) 130770055 19) 130010068 20) 130010092 21) 130770085 22) 130770084 23) 130010021 24) 130770066 25) 130770082 26) 130010056 27) 130770033 28) 130770010 29) 130010038 30) 130770062 31) 130770072 32) 130770092 33) 130770075 34) 130770021 35) 130770024 36) 130770080 37) 130770087 38) 130770090 39) 130770093 40) 130770086 41) 130770034 42) 130770061 43) 130770003 44) 130770008 45) 130770001 46) 130570093 47) 130770017 48) 130010076 49) 130010036 50) 130560035 51) 130770070 52) 130770064 53) 130770012 54) 130570016 55) 130770053 56) 130770071 57) 130560005 58) 130770016 59) 130560036 60) 130560040 61) 130570117 62) 130560037 63) 130570024 64) 130560010 65) 130160006 66) 130160195 67) 130160080 68) 130560039 69) 130160033 70) 130570060 71) 130560011 72) 130570022 73) 130560018 74) 130560018 75) 130560033 76) 130560021 77) 130160110 78) 130560006 79) 130560044 80) 130560032 81) 130560038 82) 130570048 83) 130560003 84) 130560001 85) 130160029 86) 130160001 87) 130560008 88) 130570086 89) 130160013 90) 130160035 91) 130560038 92) 130160036 93) 130560034 94) 130560004 95) 130560029 96) 130560030 97) 130570079 98) 130160039 99) 130560007 100) 130560014 101) 130560012 102) 130570048
- El arbol130 cuya raíz tiene la clave 130760028 tiene 7 elementos y son los siguientes: 1) 130170011 2) 130170027 3) 130170032 4) 130640005 5) 130760015 6) 130760073 7) 130760005
- El arbol131 cuya raíz tiene la clave 130760023 tiene 9 elementos y son los siguientes: 1) 130760057 2) 130760031 3) 130760009 4) 130760072 5) 130640014 6) 130760010 7) 130760075 8) 130760004 9) 130760001
- El arbol132 cuya raíz tiene la clave 130520012 tiene 34 elementos y son los siguientes: 1) 130050006 2) 130230047 3) 130050035 4) 130050008 5) 130050002 6) 130520010 7) 130520066 8) 130050004 9) 130050009 10) 130520022 11) 130520055 12) 130050012 13) 130520053 14) 130050007 15) 130050034 16) 130052009 17) 130520008 18) 130520029 19) 130050001 20) 130280014 21) 130520005 22) 130750006 23) 130520031 24) 130520026 25) 130750004 26) 130750011 27) 130520007 28) 130520022 29) 130750026 30) 130750025 31) 130750001 32) 130750007 33) 130750008 34) 130690036
- El arbol133 cuya raíz tiene la clave 130760032 tiene 11 elementos y son los siguientes: 1) 130760052 2) 130760016 3) 130760064 4) 130760070 5) 130630027 6) 130630048 7) 130630008 8) 130630028 9) 130630004 10) 130630029 11) 130630002
- El arbol134 cuya raíz tiene la clave 130760025 tiene 41 elementos y son los siguientes: 1) 130670026 2) 130760019 3) 130740004 4) 130760076 5) 130760069 6) 130760029 7) 130760079 8) 130760084 9) 130760053 10) 130760078 11) 130760008 12) 130100020 13) 130760001 14) 130760018 15) 130760080 16) 130100014 17) 130760077 18) 130760022 19) 130760063 20) 130760006 21) 130760071 22) 130760040 23) 130760067 24) 130760065 25) 130760058 26) 130760061 27) 130760083 28) 130630011 29) 130760081 30) 130630074 31) 130760024 32) 130760032 33) 130760012 34) 130760062 35) 130760021 36) 130760002 37) 130630018 38) 130130002 39) 130630018 40) 130130015 41) 130630006
- El arbol135 cuya raíz tiene la clave 130700008 tiene 52 elementos y son los siguientes: 1) 130650003 2) 130700005 3) 130700017 4) 130700015 5) 130700018 6) 130700001 7) 130700009 8) 130650009 9) 130700002 10) 130700019 11) 130700016 12) 130740013 13) 130740003 14) 130650001 15) 130740015 16) 130650003 17) 130740001 18) 130650013 19) 130100022 20) 130650012 21) 130740007 22) 130740002 23) 130100011 24) 130100017 25) 130100010 26) 130100018 27) 130100016 28) 130100001 29) 130100023 30) 130100006 31) 130100017 32) 130100013 33) 130100002 34) 130100019 35) 130100012 36) 130100009 37) 130100018 38) 130100019 39) 130100021 40) 130760011 41) 130130008 42) 130130001 43) 130130014 44) 130130011 45) 130130022 46) 130130005 47) 130130013 48) 130130019 49) 130130006 50) 130130007 51) 130630075 52) 130130009
- El arbol136 cuya raíz tiene la clave 130220001 tiene 104 elementos y son los siguientes: 1) 130220062 2) 130220069 3) 130510092 4) 130510001 5) 130220042 6) 130570036 7) 130220025 8) 130220025 9) 130220027 10) 130220010 11) 130510052 12) 130220066 13) 130220007 14) 130220015 15) 130570133 16) 130570117 17) 130220052 18) 130510008 19) 130220017 20) 130220065 21) 130570020 22) 130220037 23) 130570120 24) 130220011 25) 130220066 26) 130220068 27) 130570025 28) 130220067 29) 130570150 30) 130510019 31) 130220047 32) 130570039 33) 130510038 34) 130570157 35) 130220027 36) 130570014 37) 130220008 38) 130570002 39) 130220029 40) 130570131 41) 130220026 42) 130570034 43) 130570132 44) 130510096 45) 130220054 46) 130570016 47) 130220074 48) 130220059 49) 130570148 50) 130570094 51) 130570138 52) 130220016 53) 130570069 54) 130570072 55) 130570161 56) 130570035 57) 130570152 58) 130220020 59) 130570008 60) 130570073 61) 130570151 62) 130220019 63) 130570012 64) 130220023 65) 130570056 66) 130220058 67) 130220064 68) 130570146 69) 130570155 70) 130220056 71) 130220075 72) 130220022 73) 130220061 74) 130220002 75) 130220040 76) 130570066 77) 130220062 78) 130220014 79) 130570083 80) 130220072 81) 130220041 82) 130220060 83) 130570004 84) 130220005 85) 130220044 86) 130220075 87) 130220057 88) 130630065 89) 130630046 90) 130220021 91) 130570114 92) 130380067 93) 130570144 94) 130570053 95) 130380008 96) 130220070 97) 130380034 98) 130630103 99) 130380071 100) 130630011 101) 130220012 102) 130630091 103) 130630020 104) 130630081
- El arbol137 cuya raíz tiene la clave 130830025 tiene 84 elementos y son los siguientes: 1) 130480211 2) 130480195 3) 130510065 4) 130510031 5) 130480157 6) 130830002 7) 130830009 8) 130830012 9) 130830019 10) 130510009 11) 130830074 12) 130830018 13) 130830077 14) 130510007 15) 130830016 16) 130830022 17) 130830093 18) 130830017 19) 130830023 20) 130830032 21) 130830042 22) 130830108 23) 130830084 24) 130830001 25) 130430043 26) 130830076 27) 130830038 28) 130830027 29) 130830069 30) 130830097 31) 130830042 32) 130830073 33) 130830074 34) 130820013 35) 130820005 36) 130830003 37) 130830011 38) 130830006 39) 130660011 40) 130760010 41) 130330105 42) 130750027 43) 130750029 44) 130830082 45) 130660018 46) 130660003 47) 130820004 48) 130750023 49) 130660020 50) 130660002 51) 130750024 52) 130750010 53) 130660010 54) 130750009 55) 130750022 56) 130660008 57) 130660005 58) 130660009 59) 130690002 60) 130690001 61) 130660001 62) 130690013 63) 130690034 64) 130690027 65) 130660026 66) 130690010 67) 130690007 68) 130690029 69) 130690030 70) 130690009 71) 130690028 72) 130690005 73) 130690023 74) 130690027 75) 130690035 76) 130690024 77) 130690004 78) 130690020 79) 130690002 80) 130690001 81) 130690019 82) 130690013 83) 130690026 84) 130690008
- El arbol138 cuya raíz tiene la clave 130830021 tiene 102 elementos y son los siguientes: 1) 130570028 2) 130570015 3) 130570074 4) 130570098 5) 130570011 6) 130570136 7) 130570061 8) 130570005 9) 130570007 10) 130570023 11) 130570147 12) 130570068 13) 130570027 14) 130570137 15) 130570070 16) 130570067 17) 130570145 18) 130570121 19) 130570046 20) 130570143 21) 130570142 22) 130570105 23) 130570086 24) 130570160 25) 130570091 26) 130570122 27) 130570032 28) 130570003 29) 130570141 30) 130570153 31) 130570006 32) 130570144 33) 130570156 34) 130570026 35) 130570162 36) 130570108 37) 130570158 38) 130570080 39) 130570059 40) 130570149 41) 130570054 42) 130830083 43) 130570019 44) 130570031 45) 130570041 46) 130570033 47) 130570009 48) 130830014 49) 130570075 50) 130570057 51) 130830062 52) 130570044 53) 130830007 54) 130570049 55) 130830019 56) 130570110 57) 130570106 58) 130570062 59) 130570013 60) 130830106 61)



130630009 62) 130630085 63) 130630101 64) 130630022 65) 130630056 66) 130630089 67) 130670112 68) 130670011 69) 130630047 70) 130670030 71) 130630090 72) 130630048 73) 130630087 74) 130630045 75) 130670051 76) 130670010 77) 130630059 78) 130630028 79) 130670032 80) 130610096 81) 130630049 82) 130630017 83) 130630100 84) 130630099 85) 130670009 86) 130670031 87) 130610046 88) 130670002 89) 130630025 90) 130670018 91) 130670004 92) 130610008 93) 130670019 94) 130610020 95) 130610030 96) 130610004 97) 130610054 98) 130610033 99) 130610056 100) 130610083 101) 130610010 102) 130610052

El arbol139 cuya raíz tiene la clave 130630015 tiene 9 elementos y son los siguientes: 1) 130630016 2) 130630110 3) 130630001 4) 130630035 5) 130630092 6) 130660019 7) 130630061 8) 130660013 9) 130660004

El arbol140 cuya raíz tiene la clave 130630016 tiene 41 elementos y son los siguientes: 1) 130670074 2) 130630014 3) 130630041 4) 130630070 5) 130630025 6) 130630052 7) 130630024 8) 130630072 9) 130630047 10) 130630053 11) 130630054 12) 130630062 13) 130630020 14) 130630069 15) 130630073 16) 130630059 17) 130630077 18) 130630023 19) 130630071 20) 130630001 21) 130630021 22) 130630044 23) 130630060 24) 130630055 25) 130630019 26) 130130018 27) 130630026 28) 130630061 29) 130630017 30) 130630056 31) 130630064 32) 130630049 33) 130630013 34) 130630003 35) 130630010 36) 130630009 37) 130630067 38) 130630007 39) 130630012 40) 130630005 41) 130630068

El arbol141 cuya raíz tiene la clave 130630026 tiene 18 elementos y son los siguientes: 1) 130630111 2) 130630002 3) 130630056 4) 130630021 5) 130630050 6) 130670035 7) 130670021 8) 130670008 9) 130670022 10) 130670011 11) 130670005 12) 130670034 13) 130670001 14) 130670037 15) 130670006

El arbol142 cuya raíz tiene la clave 130680017 tiene 115 elementos y son los siguientes: 1) 130160074 2) 130160040 3) 130160002 4) 130160107 5) 130160005 6) 130160106 7) 130160109 8) 130160069 9) 130160065 10) 130160037 11) 130160038 12) 130160021 13) 130160043 14) 130160028 15) 130160100 16) 130160101 17) 130570159 18) 130160092 19) 130160102 20) 130160015 21) 130160053 22) 130160050 23) 130160070 24) 130160084 25) 130160034 26) 130160089 27) 130160060 28) 130160018 29) 130160099 30) 130160004 31) 130160056 32) 130570055 33) 130570063 34) 130570018 35) 130160096 36) 130160025 37) 130160027 38) 130160063 39) 130160091 40) 130160008 41) 130160090 42) 130160087 43) 130570021 44) 130160068 45) 130160083 46) 130160003 47) 130160044 48) 130160094 49) 130160058 50) 130160097 51) 130610009 52) 130160030 53) 130160096 54) 130160019 55) 130160098 56) 130160009 57) 130160088 58) 130160054 59) 130610084 60) 130610037 61) 130610011 62) 130160016 63) 130160005 64) 130160017 65) 130160073 66) 130080094 67) 130080114 68) 130080094 69) 130080004 70) 130070106 71) 130080048 72) 130080141 73) 130610041 74) 130070045 75) 130080036 76) 130610048 77) 130610016 78) 130080092 79) 130080021 80) 130610054 81) 130080023 82) 130610017 83) 130080022 84) 130610045 85) 130080069 87) 130070118 88) 130610057 89) 130610070 90) 130070096 91) 130680016 92) 130680047 93) 130080111 94) 130070085 95) 130610075 96) 130070064 97) 130080039 98) 130080021 99) 130080060 100) 130080003 101) 130080100 102) 130080106 103) 130070091 104) 130070016 105) 130070083 106) 130080213 107) 130070056 108) 130080063 109) 130070007 110) 130080041 111) 130070098 112) 130070034 113) 130080049 114) 130080015 115) 130080093

El arbol143 cuya raíz tiene la clave 130070051 tiene 20 elementos y son los siguientes: 1) 130160085 2) 130160093 3) 130070020 4) 130070008 5) 130070037 6) 130070135 7) 130070155 8) 130070155 9) 130070029 10) 130070087 11) 130070086 12) 130070062 13) 130070088 14) 130070148 15) 130070033 16) 130070069 17) 130070162 18) 130070067 19) 130070072 20) 130070005

El arbol144 cuya raíz tiene la clave 130070032 tiene 48 elementos y son los siguientes: 1) 130070022 2) 130070079 3) 130070084 4) 130070073 5) 130070031 6) 130070019 7) 130070041 8) 130070023 9) 130070006 10) 130070074 11) 130070048 12) 130070047 13) 130080052 14) 130070090 15) 130070125 16) 130070164 17) 130070138 18) 130070010 19) 130070012 20) 130070153 21) 130070108 22) 130070026 23) 130070011 24) 130070111 25) 130070021 26) 130070139 27) 130070128 28) 130070151 29) 130070102 30) 130070101 31) 130070099 32) 130070105 33) 130070163 34) 130070004 35) 130070042 36) 130070055 37) 130070156 38) 130070098 39) 130070116 40) 130070089 41) 130070158 42) 130070039 43) 130070043 44) 130070154 45) 130070035 46) 130070025 47) 130070057 48) 130070082

El arbol145 cuya raíz tiene la clave 130610007 tiene 44 elementos y son los siguientes: 1) 130610034 2) 130610081 3) 130610086 4) 130610058 5) 130610003 6) 130670039 7) 130610040 8) 130610074 9) 130610023 10) 130610090 11) 130610069 12) 130670033 13) 130610071 14) 130610082 15) 130670020 16) 130610051 17) 130610078 18) 130670007 19) 130610015 20) 130610094 21) 130610001 22) 130670010 23) 130610060 24) 130610013 25) 130610077 26) 130610099 27) 130610091 28) 130670032 29) 130610002 30) 130610073 31) 130610029 32) 130610050 33) 130670030 34) 130610039 35) 130610047 36) 130610035 37) 130610043 38) 130610014 39) 130610012 40) 130210013 41) 130610093 42) 130210006 43) 130210011 44) 130210002

El arbol146 cuya raíz tiene la clave 130080005 tiene 65 elementos y son los siguientes: 1) 130080028 2) 130080043 3) 130080044 4) 130080127 5) 130080026 6) 130080053 7) 130080061 8) 130080077 9) 130080052 10) 130080018 11) 130080051 12) 130080051 13) 130080032 14) 130080143 15) 130080009 16) 130080021 17) 130080006 18) 130080206 19) 130080171 20) 130080024 21) 130080021 22) 130080218 23) 130080217 24) 130080001 25) 130080002 26) 130070118 27) 130070001 28) 130080070 29) 130080008 30) 130080025 31) 130070094 32) 130070160 33) 130070141 33) 130070136 34) 130080116 35) 130080151 36) 130080215 37) 130070024 38) 130610096 39) 130080205 40) 130080067 41) 130080060 42) 130080200 43) 130080208 44) 130070146 45) 130080194 46) 130070049 47) 130080019 48) 130080203 49) 130080064 50) 130080034 51) 130080014 52) 130210005 53) 130210001 54) 130210019 55) 130210022 56) 130080059 57) 130070046 58) 130210027 59) 130210001 60) 130070137 61) 130210028 62) 130210015 63) 130070053 64) 130080046 65) 130080011

El arbol147 cuya raíz tiene la clave 130080020 tiene 9 elementos y son los siguientes: 1) 130210016 2) 130080199 3) 130080053 4) 130080013 5) 130210029 6) 130210010 7) 130080035 8) 130080007 9) 130080037

No se elimina ninguna localidad por lo que, este resultado es óptimo.

El costo máximo (la suma de las distancias) es el valor más grande que se obtiene del conjunto de árboles a los que se les ha incorporado una nueva localidad proveedora, por eso cuando se dice que el costo máximo sucedió en determinado árbol, significa que es el árbol en donde se recorre mayor distancia. Este costo asciende a 147625 3053794421 y ocurre en el árbol 81.

Y por el contrario, la transición mínima (suma de distancias) es el valor más pequeño que se obtiene del conjunto de árboles, en este caso es -59577.9457931158 y ocurre en el árbol 55.

El más grande en costo indica que cuando se ha eliminado alguna localidad proveedora, la distancia en algunos árboles se incrementa, este valor indica en qué árbol se produce el mayor incremento.

TESIS CON
FALLA DE ORIGEN