

03063



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

SISTEMA DE LOCALIZACIÓN DE RUTAS
MÍNIMAS EN UN MAPA PARA UNA PALM

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA DE LA COMPUTACIÓN
P R E S E N T A ;
JORGE OMAR MORALES OLVERA

DIRECTOR DE LA TESIS: DR. FERNANDO GAMBOA RODRÍGUEZ

MÉXICO, D. F.

2005.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

“La juventud tiene algo de sagrado porque en ella se ostenta inalterable la majestad de la vida. Es la estrofa que se levanta del polvo iluminado del sendero, bloque de energías que tiemblan de hartura y vigor de lozanía, audacia santa para las banderas altas. Su elemento esencial es la esperanza. Porque la juventud no consiste solamente en una etapa circunscrita por el tiempo, sino también en una actitud moral que se caracteriza por una viva confianza en la realización plena del bien y la verdad.”

Anacleto Gonzalez Flores

“Aquel que no busca eternamente embriagarse de juventud, eternamente se condena a una perpetua vejez...”

Anónimo

Tanto y a tantas personas a las que tengo que agradecer, y simplemente cuento con sólo unas líneas. Espero que ese agradecimiento a aquellas personas que no menciono aquí les llegue a través de las oraciones que a diario elevo al cielo rogando que sean bendecidos en toda su vida.

Doblo mi rodilla ante ti Señor como signo de profundo agradecimiento, y de eterno servicio, por haberme concedido la realización y culminación de este trabajo. A ti, Cristo Rey, salvador y modelo a seguir, dedico por completo este trabajo.

A mis padres, Maria Concepción Olvera y Santos Morales, incansables amigos y perpetuos compañeros de vida, que han sido mi apoyo en todo.

A ti, mamá, eterna luz en mi caminar, que has estado en los momentos en los que he tropezado. Gracias por esos 24 y “algo” años que has estado conmigo. Gracias por el inmenso apoyo, y, sobre todo, gracias por haberle respondido SI a Dios cuando te pidió que fueras tú la que me cuidara en este mundo.

A ti, papá, prudente consejero de mi actuar, que me indujiste a esta aventura, y que en el camino nunca dijiste “no”. Gracias por tu ejemplo de incansable guerrero en las batallas más estoicas. Gracias por tus consejos, siempre acertados para lograr este peldaño en mi vida académica.

A mis hermanos, Santos y Luis Enrique, que siempre confiaron en mí y nunca dejaron de apoyarme.

A Santos, el benjamín de la familia, que en este camino siempre tuve una palabra de aliento de su parte y en el momento justo. Gracias, porque con tu tenacidad me has logrado decir que las acciones dan fruto conforme la firmeza del esfuerzo realizado.

A Enrique, el que con su fortaleza siempre me mantuvo en la cima de mis aspiraciones. Gracias por esos momentos de alegría y de consejo que me has permitido vivir contigo. Gracias por tu incansable buena vibra, la cual ha provocado en mí un esfuerzo extra en toda ocasión.

A mis tíos, María Guadalupe y Fausto, y mis primos, David y Fausto, quienes me han apoyado cual segunda familia. Gracias, porque sin su ayuda este camino hubiera sido más difícil.

A la familia Ayala, ejemplo de bondad y tenacidad, que con su cariño brindado siempre me hicieron sentir como uno más de los suyos. En especial al papá, Miguel Ayala, y la mamá, Gloria Vieyra, que con su inigualable ejemplo, siempre me indujeron a seguir de pie. Y no me gustaría dejar de agradecer a los 13 hijos increíbles que, cada quién a su manera y con su carácter, me han brindado el calor y el cariño de un hogar. Miguel, Adelaida, Gloria, María, Ramón, Magdalena, Laura, Guadalupe, Manuel, Pedro, Juan, Emilia y Teresa.

A mis amigos, a todos los que Dios ha cruzado en mi camino, pero en especial a FRU (el buen tru cu tru, alias "El Gustavo"), que con su incansable ejemplo, ha hecho que me levante cuando he caído y que siga por el buen camino cuando me he desanimado. Y, por supuesto, gracias a Gloria, Miguel, Manuel, Adelaida (eterna confidente y hermana), Ramón, Rocío, Marijo, Viridiana, Irma, Felipe, Beto (ahijado), Rodrigo y a toda la "banda" que me conoce, pues cada uno ha influido en mí de una manera especial.

A mi asesor, el Dr. Fernando Gamboa, y mis sinodales, que me han brindado su ejemplo de trabajo, constancia y tenacidad. Gracias a ellos, pues fueron los que me han apoyado en la realización de este trabajo.

Al igual que al principio empecé agradeciendo al Rey de reyes, a Cristo Rey, no puedo omitir mi profundo agradecimiento y mi perpetua subordinación a su Santísima Madre, María Santísima, Reina de mi corazón, pues es ella la que me ha tenido en sus brazos en los momentos más difíciles de mi vida, y me ha conducido, cual Excelentísima Madre que quiere lo mejor para su hijo predilecto, hacia su Hijo Jesucristo, haciéndome ver el camino correcto.

Quisiera culminar con una particular comunión que tengo con Su Santidad Juan Pablo II, inigualable maestro de mi conciencia, respecto a nuestra Santísima Madre:

TOTUS TUUS

Jorge Omar Morales Olvera

CCR

VCR

Índice

Introducción.....	9
Capítulo 1. Sistema de Posicionamiento Global	15
1.1 SPG.....	15
1.2 Breve historia del SPG	16
1.3 Funcionamiento del SPG.....	18
1.3.1 Señal emitida.	19
1.4 Errores en el SPG	20
1.5 Recepción.....	22
1.5.1 Tipos de dispositivos.	22
1.5.2 Cálculo de posición.....	23
1.5.3 Determinación de la precisión	23
1.5.4 Posibilidades de incrementar la precisión.....	25
1.6 Aplicaciones de los SPG	26
1.7 ¿En qué nos ayudará estos dispositivos?.....	27
1.8 Conclusión.....	28
Capítulo 2. Ruta mínima entre dos puntos	30
2.1 Grafos	30
2.2 Problema de la ruta mínima.....	32
2.3 Problema del agente viajero	32
2.4 Problema en nuestra ciudad.....	33

2.5 ¿De qué manera se resuelve el problema?.....	35
2.6 Algoritmo de Dijkstra.....	39
2.7 Aportación del algoritmo de Dijkstra en este trabajo de tesis.....	42
Capítulo 3. Mapas en la Palm.....	44
3.1 Características de la Palm.....	44
3.2 La Palm para aplicaciones gráficas.....	46
3.3 Forma en que se dibujarán los mapas.....	46
3.4 Formato de los datos de un mapa.....	51
3.5 SVG más pequeño que RASTER.....	53
3.6 En qué nos ayudarán los formatos de los mapas.....	56
3.7 Conclusión.....	57
Capítulo 4. Visión General del Sistema.....	60
4.1 Diseño del sistema.....	60
4.2 Detalle de cada elemento.....	62
4.2.1 Tipos de Archivo de Mapas.....	62
4.2.2 Ventajas y desventajas de RASTER y Vectoriales.....	63
4.2.3 Filtro convertidor.....	64
4.2.4 Convertidor.....	64
4.2.5 Tipos de archivos de GMAP.....	66
4.3 Ventajas y desventajas en el trato de los datos.....	70
4.4 Proceso de los mapas.....	72
4.5 Conclusiones.....	73
Capítulo 5. Diseño del Sistema.....	76
5.1 Partes que intervienen.....	76

5.2	Diseño de capas	77
5.3	Propiedades de las capas	82
5.4	Capas del sistema	83
5.4.1	<i>Mapa de la traza de la ciudad.</i>	83
5.4.2	<i>Grafo de la ciudad.</i>	86
5.4.3	<i>Nombre de las calles de la ciudad.</i>	88
5.4.4	<i>Punto de localización.</i>	88
5.4.5	<i>Puntos de interés</i>	90
5.4.6	<i>Otras capas.</i>	90
5.5	Manejador de las capas.....	90
5.5.1	<i>Matriz de dibujado</i>	92
5.5.2	<i>Orden en el dibujado</i>	95
5.6	Otras funciones de las capas.....	96
5.7	Conclusiones	96
Capítulo 6.	Transformación de los archivos.....	98
6.1	Archivos de tipo gmp	98
6.2	Fuente de datos.....	99
6.3	Fuente de datos.....	102
6.4	Procesamiento de los archivos	106
6.4.1	<i>Archivos DXF y DWG.</i>	106
6.4.2	<i>DWG a DXF.</i>	107
6.4.3	<i>Extracción de vértices.</i>	107
6.4.4	<i>Extracción de cuadras de la ciudad.</i>	109

6.4.5 <i>División en cuadros (secciones del mapa)</i>	110
6.4.6 <i>Extracción de nodos</i>	116
6.4.7 <i>Cálculo del grafo</i>	116
6.4.8 <i>Extracción de nombres de las calles</i>	117
6.4.9 <i>Extracción de puntos de interés</i>	117
6.5 <i>¿Qué hacemos con los archivos?</i>	118
Conclusiones	120
Bibliografía	124

Introducción

Introducción

Hoy en día la tecnología avanza de manera vertiginosa en nuestra sociedad. Es notoria esta situación al comprar algún aparato electrónico nuevo con la ilusión de que es lo último de la tecnología y a la vuelta de poco tiempo nos damos cuenta de que ha salido algo mejor.

Podemos ver que día con día surgen nuevos dispositivos para cada una de las actividades de nuestra vida cotidiana. Por ejemplo, en el caso de las computadoras, cada breve período de tiempo sale al mercado procesadores más poderosos, o discos duros con mayor capacidad, memorias de mayor velocidad, o hardware que nos permite realizar tareas más completas y complicadas que antes. Otro ejemplo claro de esto son los automóviles, los cuales a cada momento son más potentes, ahorran más gasolina, duran más, entre otros muchos aspectos.

Esto ya es una tendencia normal, la tecnología que sale hoy, a la vuelta de medio año ya se nota como obsoleta.

Pero junto a esta tendencia que tiene la tecnología al avanzar de manera rápida también se presenta otra tendencia, donde los dispositivos que surgen en sustitución de los existentes son más pequeños en tamaño, no con esto haciéndolos menos capaces, sino todo lo contrario, tienen más capacidades aún a pesar de su menor tamaño.

Ejemplos claros de esta segunda tendencia son las computadoras. Desde su inicio, a mediados del siglo XX, cuando eran del tamaño de 3 pisos y no tan capaces, hasta nuestros tiempos, estos dispositivos electrónicos han variado mucho su tamaño hasta tal punto de tener computadoras portátiles de tamaño menor a la de un cuaderno y con capacidades muy superiores a las de las primeras computadoras. Luego de haber llegado a las computadoras portátiles, los desarrolladores de este tipo de tecnologías pasaron a la era de los dispositivos móviles, tales como los celulares, los biper y las Palm.

En este tipo de dispositivos móviles, también la tendencia es a tener un menor tamaño con cada modelo nuevo, y, por supuesto, con mayor capacidad que el modelo anterior. Esto lo podemos ver claramente con los celulares. En un principio los primeros celulares tenían un tamaño aproximado al de una caja para plumas, inclusive más grandes, ahora el más chico de ellos es aproximadamente del tamaño de la mano de un bebé.

Lo mismo pasa con las Palm, ya que estas, en un principio, tenían un tamaño considerablemente grande debido a que los circuitos que la formaban tenían un tamaño algo grande. Los nuevos modelos tienden a ser más delgados y con mayores capacidades.

Como podemos observar, la tecnología tiende a mejorar las capacidades de hardware de cada uno de los dispositivos que usamos en las actividades cotidianas para que

podamos realizar tareas más complejas, pero también tiende a ser más pequeña para que su utilización sea más cómoda.

Cuando hablamos de software para las computadoras y los dispositivos móviles la tendencia es a realizar programas de computadora cada vez más potentes y capaces, y esto, en gran parte, se debe a que los mismos dispositivos, ya sea computadoras de escritorio, portátiles o dispositivos móviles, permiten cada vez más este desarrollo.

Esta evolución podemos verla comparando los procesadores de texto que surgieron cuando el mundo del software empezaba a incursionar en la sociedad, los cuales eran solamente en texto, mientras que los actuales podemos manejar texto, imágenes, animaciones y muchas otras cuestiones.

Lo mismo pasa en el ámbito de los juegos, pues estos son cada vez más sorprendentes y complejos.

Pero por otro lado, existe una cuestión cultural, característica de cada país, que determina si las nuevas tecnologías y el nuevo software son usados en la sociedad en donde se pretende implementar y poner en marcha. Ejemplo claro de esto, en el caso de México, fue la difícil incursión de la tecnología de los celulares.

Lo anterior tuvo lugar en la época en la que los celulares empezaban a entrar a nuestra sociedad, pero ya fue superada esa etapa. Ahora, esa misma etapa está pasando con los dispositivos móviles como las Palm.

Estos dispositivos, a pesar de que muchos saben de su existencia, no son muy aceptados por la mayoría de los usuarios potenciales. Quizá sea por sus capacidades, quizá por las aplicaciones que se han hecho, el punto es que la mayoría de los usuarios en México no han querido hacer uso de ellos.

A pesar de estas dificultades, la tecnología que se desarrolla en este rubro es más capaz cada vez. Y en el mundo del software sucede lo mismo, las aplicaciones hechas son cada vez más poderosas y más útiles para la sociedad.

Entretenimiento, medicina, cartografía, negocios, bases de datos, y muchos más son los rubros sobre los cuales el software para las Palm ha depositado sus creaciones. Aplicaciones que van desde un simple anotador de textos hasta juegos de lo más complejo.

Así podemos observar, entre otras cosas, aplicaciones que ayudan al médico en su formación mediante la lectura de libros digitales, aplicaciones que permiten tener el control de los movimientos en un negocio, aplicaciones que permiten llevar el control de todas las actividades en el día de cualquier persona.

Dentro de todo este mundo de aplicaciones, una que ha sido muy útil en países en donde se ha implementado es el de sistemas de navegación en ciudades mediante dispositivos de tipo Palm. Me refiero a aquella aplicación en donde, mediante el dibujado de un mapa de la ciudad en cuestión, el usuario puede ubicarse dentro de la ciudad donde se

encuentra. Esta aplicación se ha desarrollado en otros países, y se ha implementado con gran éxito. Países como Estados Unidos, Japón y la mayoría de los países de Europa la han implementado en muchos ámbitos de su sociedad, y les ha ahorrado muchos recursos económicos y les ha traído grandes beneficios personales.

En el caso de México, tenemos la desventaja de importar dicha tecnología, tanto el software como el hardware. Atendiendo esta desventaja, en este trabajo de tesis se propuso el siguiente objetivo:

Sentar las bases para mejorar la aplicación del sistema de navegación, así como adaptarla a la ciudad de México¹. Es decir, lograr desarrollar una aplicación que contenga las características básicas que se mencionan más adelante como manejo de mapas, puntos de interés, localización geográfica y nombres de calles. Todo esto dentro de un dispositivo de tipo Palm.

Hablando un poco más a detalle de esta aplicación, y de sus alcances, podemos decir que consiste en varias partes, a saber:

- Dibujar un mapa dentro de la Palm. Se busca lograr dibujar el mapa de la ciudad dentro de la pantalla del dispositivo. Dicho mapa se presentará de manera segmentada, ya que, al igual que la Guía Roji², divide el mapa en cuadros de acuerdo a la zona que se quiera ver.
- Mostrar el nombre de cada una de las calles. Además del mapa que nos indique las cuadras que existen en la ciudad, esta aplicación nos permitirá ver el nombre de cada una de las calles que se estén desplegando en el mapa. Esto con la finalidad de una mejor ubicación para el usuario.
- Mostrar el punto de localización. Mediante un dispositivo llamado GPS³ el sistema recibe la posición exacta en la que se encuentra dentro del mundo, de tal manera que, mediante un cotejamiento del mapa con respecto a la posición geográfica en la que se encuentra este, puede pintar dentro de la pantalla el lugar donde se encuentra ubicado dicho dispositivo.
- Mostrar los puntos de interés dentro de la ciudad. El sistema también nos permitirá ver donde se encuentran los puntos de mayor interés en la ciudad, tales como museos, parques, iglesias, teatros, entre muchos otros.
- Desplegar la ruta mínima a seguir para llegar desde un punto dentro del mapa hasta otro punto indicado en el mismo.

¹ Menciono "sentar las bases" debido a que se realizará solamente con la delegación Cuauhtemoc, pero se dejará el software para poder llevar a cabo esto con las demás delegaciones de la ciudad más grande del mundo.

² Sistema de mapas que ayuda a la localización de los lugares de interés, y a la ubicación dentro de las ciudades más importantes del país.

³ Global Positioning System. Explicado en un capítulo posterior.

Como podemos observar, un sistema con las características que ya se mencionaron es muy útil para cualquier automovilista que ronde por la ciudad de México. Es como tener un mapa portátil y fácil de usar, que cabe en una mano, que uno lo pueda consultar cuando uno desee sin tener que estar lidiando con papeles enormes o buscando calle por calle hasta encontrar el lugar deseado.

Nos toparemos con diversidad de problemas como la manera de presentar los mapas, la forma de manejar los datos necesarios en la aplicación, la usabilidad del sistema, entre otros muchos dentro del desarrollo de esta aplicación, pero con ayuda de los conocimientos adquiridos en la Maestría en Ciencias de la Computación en la UNAM, así como los adquiridos de manera paralela al transcurso de la misma, confiamos en que sabremos resolver cada uno de dichos problemas.

Otra de los objetivos que se buscan con este trabajo es promover el desarrollo de este tipo de software orientado a los dispositivos móviles, ya que en nuestro país es muy pobre el desarrollo de software en general, y con mayor razón el software hecho para este tipo de dispositivos.

Partiendo de estos objetivos, planteamos la siguiente estructura en este trabajo de tesis.

- Sistema de Posicionamiento Global. Introduciremos al lector en este concepto, útil para la realización de este proyecto de tesis.
- Ruta Mínima entre dos puntos. Desplegaremos los conceptos teóricos acerca de los algoritmos utilizados para este rubro, importante para el desarrollo de uno de los principales objetivos de este trabajo de tesis.
- Mapas en la Palm. Mostraremos los diversos problemas que se pueden presentar a la hora de tratar de dibujar un mapa dentro de la Palm, y las posibles soluciones que se pueden tomar.
- Visión general del sistema. Plantearemos un panorama general del sistema GMAP⁴ que realizaremos, las partes que lo conforman y la manera, a grandes rasgos, de procesar los datos que estará necesitando.
- Diseño del sistema. Se dará detalle de cada una de las partes que conforman el sistema GMAP que se realizará en este trabajo de tesis. Esto nos permitirá ver la manera en que funcionará el sistema en cuestión.

⁴ Nombre con el que se bautizó a esta aplicación.

- Proceso de transformación de los archivos. Se detallará el proceso al que se deben someter los archivos originales para extraer los datos que el sistema GMAP usará para su funcionamiento.
- Finalmente se sacarán las conclusiones de este trabajo, partiendo de los objetivos planteados en este apartado.

Capítulo 1

Sistema de Posicionamiento Global

Capítulo 1. Sistema de Posicionamiento Global

A principios del siglo pasado, el ser humano, sobre todo en el ámbito de la navegación marítima, tenía un problema que se originaba a raíz del movimiento de la Tierra sobre su propio eje de rotación. Este problema consistía en que si uno se trasladaba 15° grados hacia el este tenía que adelantar una hora a su reloj; en cambio, si uno se traslada 15° hacia el oeste el reloj había que atrasarlo una hora. Esto resultaba ser un problema debido a que los barcos que navegaban de un continente a otro no sabían la hora exacta, y mucho menos la posición, en la que se encontraban; esto traía como consecuencias pérdidas en los trayectos, retrasos, y en algunos casos, naufragios.

A partir de este problema, surgieron varias necesidades, entre ellas la de tener una navegación más segura en cuanto a localización y trayectos no erróneos. A nivel militar y aviación también surgieron necesidades similares. Debido a esto se empezó a desarrollar la tecnología de los SPG (Sistema de Posicionamiento Global)⁵, y a partir de aquí otros sistemas que se basan en este último.

El SPG, desde su aparición, ha sido de gran utilidad para el ser humano. En muchos ámbitos de la sociedad ha incursionado con gran éxito, mediante aplicaciones que van desde los ámbitos topográficos, navegación marítima, e inclusive, y con gran aceptación, dentro de las actividades de recreación.

1.1 SPG

El Sistema de Posicionamiento Global es un método, que funciona a base de receptores electrónicos, de posicionamiento y navegación que se basa en una serie de señales que transmiten una serie de 24 satélites que se encuentran en la órbita terrestre (también llamada constelación de satélites NAVSTAR⁶). Dichas señales transmitidas por los satélites son captadas por receptores en tierra.

Los satélites se encuentran posicionados de manera estratégica en la órbita terrestre para darle este servicio a cualquier aparato receptor que se encuentre en cualquier parte del globo terráqueo. Más adelante se explicará bien este concepto.

⁵ Sus siglas en inglés son Global Positioning System (SPG), pero en este trabajo se le tratará como SPG, por sus siglas en español (Sistema de Posicionamiento Global).

⁶ Navigation Satellite Timing And Ranging por sus siglas en inglés.

Los SPG usan dicha constelación de satélites para calcular su posición en el mundo con una precisión de 2 a 5 metros. Inclusive, en algunos SPG avanzados, se puede calcular la posición con una precisión de unos cuantos centímetros.

Básicamente, la operación correcta de los SPG depende de una referencia muy precisa de tiempo. Dicha referencia de tiempo es brindada por el reloj atómico que se encuentra en el Observatorio Naval de los Estados Unidos. Cada satélite que se usa para estos fines tiene su propio reloj atómico.

Dichos satélites transmiten información de tiempo y posición constantemente, de tal manera que los aparatos receptores comparan todas las lecturas, y, a partir de las que son iguales en tiempo, calculan la posición en la que se encuentran; posición en tres dimensiones: latitud, longitud y altitud. La información que es transmitida por los satélites es gratuita para todos los usuarios que deseen usarla; y es transmitida durante las 24 horas del día y hacia cualquier parte del planeta.

Estas y otras características que se detallarán más adelante hacen que el SPG sea un sistema eficiente, confiable y muy útil.

Los SPG han evolucionado poco a poco, de tal manera que ahora son dispositivos muy chicos y baratos, accesibles a la mayoría de los usuarios de este tipo de sistemas.

Hoy en día este tipo de sistemas se encuentran en automóviles, barcos, aviones, equipo de construcción, computadoras portátiles, equipo de topografía, entre otras muchas áreas de la actividad de la sociedad.

Tanta ha sido su incursión en nuestra sociedad y su evolución que hoy en día los podemos encontrar en los celulares más recientes, los cuales ya incluyen estos sistemas para brindar otro agregado al servicio que proveen al cliente.

Pero este tipo de sistemas también tienen desventajas tales como que no transmiten ninguna señal a los satélites. Otra desventaja, aunque se le podría ver como una circunstancia del mismo medio, es que necesita forzosamente que el medio por el que viaja la señal, en este caso el aire de la atmósfera, esté libre, es decir, que no se encuentren obstáculos entre el aparato receptor y el satélite. Aunque, observando estas desventajas desde un punto de vista objetivo, se puede decir que estas no interrumpen el correcto funcionamiento de los aparatos receptores en cada una de las aplicaciones en las que se usa.

1.2 Breve historia del SPG

Los inicios del SPG se remontan a los años 60's del siglo pasado, después de que el ser humano se vio en la necesidad de contar con un sistema que le permitiera saber la posición exacta en la que se encuentra dentro del planeta. Esta necesidad, como ya se

mencionó, surgió a raíz de los problemas que experimentaba el sector navío en el mundo, y debido a que la milicia de Estados Unidos también necesitaba dicho sistema.

Es por esto que, al principio de los años 60's del siglo pasado, los departamentos de defensa (DoD)⁷, de transporte (DoT)⁸ y la agencia espacial norteamericana (NASA)⁹ adquirieron el interés de desarrollar dicho sistema, el cual determinara, por medio de satélites, la posición exacta de alguien dentro del planeta.

Dicho sistema, de acuerdo a los objetivos planteados en un principio, debía cumplir con la globalidad, es decir, abarcar toda la superficie del globo terráqueo; continuidad, es decir, que funcionara independientemente de las condiciones atmosféricas del lugar o de la época del año; y altamente dinámico para que fuera usado por aviones o por automóviles, y por supuesto que fuera preciso en sus mediciones para que fuera confiable.

Los primeros proyectos que se realizaron con respecto a este tema fueron el sistema Timation¹⁰ y el sistema 621B¹¹, los cuales se probaron en el desierto creando una simulación de diferentes comportamientos. Pero ninguno de estos proveía las características suficientes para resolver las necesidades planteadas. Debido a esto, empezaron a combinar tecnologías de dichos proyectos, hasta que, en Diciembre de 1973, se aprobó proceder con el desarrollo de lo que hoy se conoce como NAVSTAR¹², una constelación de satélites capaz de cubrir las necesidades planteadas. Constelación de satélites que consta de 24 elementos. (*Link 15 y 16*)

Por otro lado, Rusia desarrollo el sistema GLONASS¹³, el cual también consta de 24 satélites, y es la propia versión para resolver este problema. (*Link 14*)

Una vez desarrollados estos sistemas, lo que quedó por hacer fue desarrollar aplicaciones, las cuales han evolucionado también en gran medida.

Así se vino desarrollando, a grandes rasgos, el SPG. Hoy en día es un sistema que tiene muchas aplicaciones, tanto militares como civiles. Es un sistema muy usado y en constante crecimiento en cada uno de los ámbitos de la sociedad.

⁷ Por sus siglas en inglés, Department of Defense.

⁸ Por sus siglas en inglés, Department of Transport

⁹ Por sus siglas en inglés, National AeroSpacial Agency

¹⁰ Dirigido por Roer L. Easton

¹¹ Dirigido por la Fuerza Aérea de los Estados Unidos

¹² Nombre que se le dio a la primera constelación de satélites dirigida a resolver el Sistema de Posicionamiento Global.

¹³ Una vez que salió el NAVSTAR de los Estados Unidos, Rusia no se quiso quedar atrás y creó esta otra constelación de satélites, la cual también contiene 24 satélites.

1.3 Funcionamiento del SPG

En cuanto al funcionamiento de esta tecnología, primeramente, como ya se hizo una breve mención arriba, se basa en 24 satélites, posicionados en el espacio, que envían las señales al aparato receptor para poder, este último, calcular su posición.

Algunas características de estos satélites son las siguientes (*Link 2*):

- Se posicionaron cerca de la órbita terrestre, a una altura aproximada de 20,183 kilómetros.
- Dan dos vueltas a la tierra por día, es decir, su período orbital es de 12 horas.
- Inclinación de 55 grados
- Visibles, cada uno, por aproximadamente 5 horas.
- Posicionados en 6 órbitas planas.
- 4 categorías de satélites se han enviado

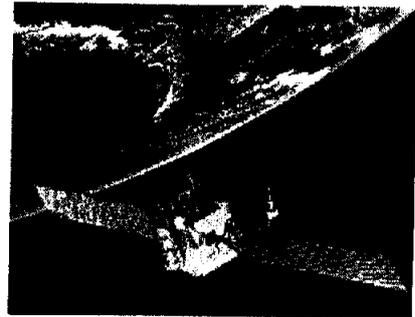


Figura 1. Tipo de satélite utilizado en la constelación de satélites del NAVSTAR (*Link 5*)

Cada satélite cuenta con un reloj atómico¹⁴, el cual les permite llevar la medida del tiempo con una precisión demasiado alta, viajan a una velocidad de 11,263 kilómetros cada hora aproximadamente, se alimentan de energía solar, tienen un tiempo de vida de aproximadamente 10 años y en caso de que lleguen a desviarse de la órbita en la que en un inicio fueron puestos tienen baterías extras y cohetes auxiliares para corregir la posición dentro de la órbita.

Cada satélite transmite señales de radio de bajo poder en diferentes frecuencias (designadas L1, L2, etc). En caso de los aparatos SPG para el uso de los civiles, estos reciben la señal en la frecuencia L1, la cual es de 1575.42 megahertz en la banda UHF.

Los satélites están constantemente enviando información a la Tierra. Información que toman los receptores para poder calcular la distancia que hay entre el receptor y el satélite que envió la señal.

Cada satélite transmite un único código, esto para que cada receptor pueda identificar la señal de dicho satélite, y, dado que otro de los datos que envía el satélite a tierra es el tiempo en el que envía la señal, este dato, junto con el código único, es tomado por el receptor para calcular cuáles cuatro satélites están más cerca a él para realizar la triangulación y obtener su posición exacta en el mundo.

¹⁴ Reloj de mucha precisión usado para cálculos de extrema importancia.

De esta manera, cada satélite funciona con un solo objetivo, proveer al usuario final, sea cual sea este, la capacidad de saber en que parte del mundo se encuentra en el momento en que recibe las señales. A esto se le llama posicionamiento.

Así, los dispositivos SPG reciben una señal desde los satélites que indica la posición exacta en la que se encuentran.

1.3.1 Señal emitida.

La señal que recibe el dispositivo SPG tiene tres bits diferentes de información: un código pseudo-aleatorio, datos de efemérides y datos de almanaque.

- a) El código pseudo-aleatorio es simplemente un identificador del satélite que envía la señal. Está formado por una serie impar de n bits de una duración de T segundos. Su espectro es similar al ruido. Cada uno de dichos códigos se asigna a cada uno de los satélites. Es muy importante que el receptor y el satélite estén sincronizados para que la correlación comience cuando llega la señal procedente del satélite. De esta forma calcularemos el retardo de dicha señal.
- b) Los datos de efemérides contienen información importante acerca del estado en que se encuentra el satélite, y la actual fecha y hora. Esta parte de la información es usada para determinar la posición del SPG.
- c) En cuanto a los datos de almanaque, estos le dicen al receptor SPG en que parte deben de estar cada satélite durante el día. Contiene información de la órbita donde se encuentra el satélite que envía dicha información.

El mensaje de navegación se transmite en un régimen binario de 50 bits por segundo (*Link 11*). La estructura de dicho régimen binario se muestra a continuación:

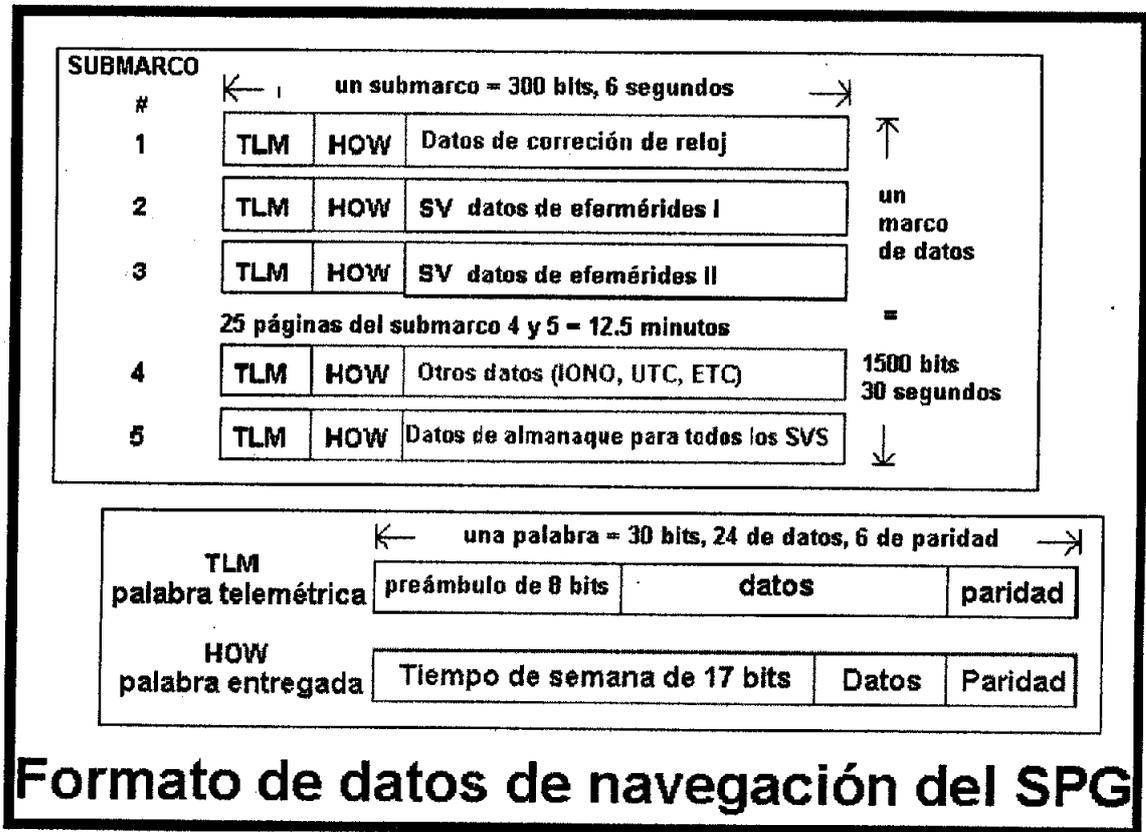


Figura 2 Estructura del régimen binario del mensaje de navegación (Link 11)

1.4 Errores en el SPG

El SPG es muy útil para muchos ámbitos de la sociedad, pero definitivamente no puede escaparse de la posibilidad de errores¹⁵. Este sistema, aunque no es muy susceptible, puede llegar a tener errores en el transporte de la señal. Dichos errores pueden deberse a las siguientes causas:

- Retardos en la ionósfera y en la tropósfera. Debido a que la señal, para ser recibida en tierra, tiene que pasar por la atmósfera, esto, en caso de que haya en la atmósfera mal clima o alguna clase de interferencia, o simplemente algún edificio, puede hacer que la señal se retrase o que no llegue al receptor de manera adecuada. El SPG cuenta con un modelo para calcular, de manera no exacta, el promedio de retraso de la señal, esto nos ayuda a reducir, mediante cálculos, el error del retraso de la señal.

¹⁵ Aquellos comportamientos del SPG que hacen que el mismo dispositivo no brinde las lecturas que debe brindar o simplemente no funcione.

- Rutas múltiples de la señal. Esto ocurre cuando la señal es desviada, por un edificio alto, por alguna roca o por algún otro objeto, antes de llegar al receptor SPG. Esto ocasiona que la señal llegue con un retraso, mismo que es ocasionado por dicha desviación. Esto se ilustra en la siguiente figura:

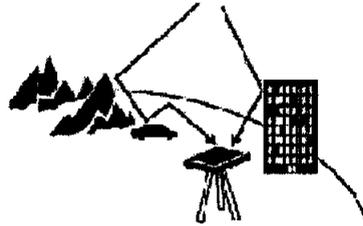


Figura 3. Señal bloqueada por los edificios (Link 1)

- Error del reloj del receptor. Dado que los receptores SPG no tienen relojes atómicos (debido a que se incrementaría su precio a tal grado que resultaría inaccesible para la mayoría de los usuarios), sino que tiene un reloj de baja precisión en comparación con un reloj atómico, esto puede ocasionar muchos errores de tiempo, con lo cual se pueden producir errores en el cálculo de la posición usando las señales llegadas de los satélites.
- Errores en la órbita. Este tipo de errores también son conocidos como 'errores de efemérides', los cuales se dan debido a que en ocasiones hay desajustes en el satélite con respecto a la órbita en la que se encuentran.
- Número de satélites visibles al receptor. El receptor tiene que, para tener un buen cálculo de la posición, recibir señal de por lo menos 4 satélites (generalmente se tiene lectura de 6 o más satélites), y cuando por circunstancias como muchos obstáculos entre el receptor y los satélites, o por cualquier otro tipo de circunstancias estas lecturas de datos provenientes de los satélites no son de cuatro o más, el receptor se verá incapacitado para poder realizar el cálculo de la posición.
- Degradación intencional de la señal del satélite. La milicia de los Estados Unidos tiene el derecho de degradar la señal del SPG para proporcionar menos precisión. Esto lo hacen si ellos piensan que es correcto, y lo pueden llevar a cabo por razones de seguridad para los Estados Unidos.
- Variación del campo gravitatorio, variaciones en la presión de la radiación solar, fricción del satélite con moléculas libres, etc.

(Link 15, 16 y 17)

1.5 Recepción.

En cuanto a la recepción de la señal enviada por los satélites, es llevada a cabo por los receptores SPG. Existen varios tipos de aparatos SPG, algunos con mayor precisión que otros, pero todos realizan la misma tarea. En cuanto al manejo de dicha recepción se mencionan algunos pasos que ejecuta el aparato receptor para empezar a calcular la posición en la que se encuentra.

1.5.1 Tipos de dispositivos.

Tres son los tipos de receptores: secuencial, continuo o multicanal y canales multiplexados.

En cuanto a los receptores de tipo secuencial, las características que lo definen son las siguientes:

- Solo cuentan con un canal
- Sigue de manera secuencial a los satélites visibles.
- El receptor permanece sincronizado con cada uno de los satélites al menos 1 segundo. Durante este tiempo adquiere la señal y calcula el retardo temporal.
- Extrae el retardo de 4 satélites y a partir de estos calcula la posición.
- Los satélites que elige son aquellos que tienen mejor SNR (Relación señal a ruido).
- Son los más baratos y los más lentos.
- Su precisión es menor que la de los otros tipos de receptores.
- Suele emplearse en aplicaciones de baja dinámica (barcos, navegación terrestre, etc.)

En cuanto a los receptores continuos o multicanal, las características que los definen son las siguientes:

- Disponen de al menos cuatro canales.
- A cada canal se le asigna el código de 1 satélite para que se sincronice con él y adquiera el retardo con ese satélite.
- Se miden los retardos simultáneamente.
- Son más rápidos que los receptores secuenciales a la hora de calcular la posición. Y por supuesto, son de mayor precisión.
- Están recomendados para aplicaciones de gran dinámica (aeronaves).

Los receptores de canales multiplexados tienen lo siguiente:

- 1 único canal físico (por medio de hardware)
- 4 o más bucles de seguimiento (por medio de software)

- De este modo se muestrean todos los satélites visibles en un tiempo inferior a 20 ms, pues así obtenemos la información recibida de todos los satélites visibles
- Complejidad de software mayor, por lo que se necesita procesador más potente.
- Tiene una ventaja sobre los receptores físicos, es menos sensible a las posibles variaciones de canal.

Vale la pena aclarar que cada canal sirve para localizar algún satélite y sincronizarse con él. Debido a esto los receptores de tipo secuencial son más tardados, ya que tienen que sincronizarse con cuatro satélites, igual que los otros dos, salvo que éste lo hace de uno por uno. Mientras que los multicanal lo hacen todos a la vez. En tanto que los de canal multiplexado, lo hace a través de un solo canal pero de manera paralela.

1.5.2 Cálculo de posición.

El proceso que sigue el receptor SPG es el siguiente:

- Una vez que el receptor se haya encendido tratará de establecer comunicación con 4 satélites.
- Cuando el receptor ya está en comunicación con un satélite se asigna el canal/es a los códigos de los satélites que están visibles y se inicia el proceso de establecimiento de comunicación con cada satélite.

Una vez sincronizados cada uno de los satélites, el receptor realiza la correlación y detecta los picos que superan un determinado umbral y a partir de ellos se obtiene el retardo temporal y con esto la ecuación de un esferoide donde está el usuario. Repitiendo este proceso para 4 satélites se obtiene la posición del usuario.

1.5.3 Determinación de la precisión

La precisión se refiere al error en metros que el receptor SPG experimenta debido a las causas ya mencionadas.

Para obtener una precisión exacta en la posición se necesitan 5 cosas:

- Triangulación¹⁶. Se hace con las distancias de los satélites al receptor.
- Distancias. Distancias desde el receptor a 4 satélites.
- Tiempo. El SPG necesita un control muy estricto de tiempo, ya que tiene que medir el tiempo en que viajan las señales.
- Posición. Se refiere a la posición de los satélites en el espacio.

¹⁶ Procedimiento que ejecutan los SPG para calcular, a partir de cuatro distancias a cuatro satélites, la posición exacta en el mundo en la que están.

- Corrección. El SPG debe corregir cualquier demora en el tiempo de viaje de la señal que esta pueda sufrir mientras atraviesa la atmósfera

Una vez que el receptor haya obtenido la localización del satélite y la distancia hacia el, se puede calcular la posición mediante el método de la triangulación.

Primero, con la distancia del primer satélite al receptor se obtienen una serie de puntos alrededor de dicho satélite, formando con esto una esfera de puntos. Esto nos indica que habría una cantidad infinita de puntos. Obviamente uno de dichos puntos es el que coincide con el receptor.

Para limitar el número de puntos se utiliza el segundo satélite, con el cual se obtiene una segunda distancia. Esto se ilustra a continuación.

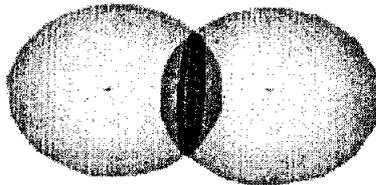


Figura 4. Área de cobertura con dos satélites (Link 1)

En este caso, la superficie sombreada es el conjunto de todos los puntos que coinciden con las dos distancias obtenidas.

Pero recordemos que debe de calcular únicamente un solo punto, la posición exacta del receptor.

Para esto se utiliza la distancia del tercer satélite, el cuál se expresa en la siguiente figura.

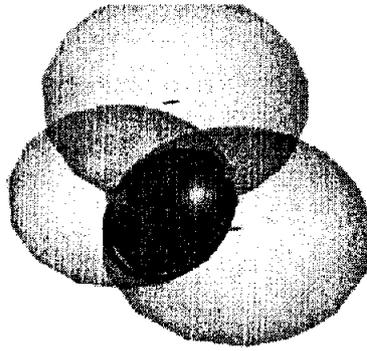


Figura 5. Área de cobertura con tres satélites (Link 1)

A estas alturas, ya son dos los puntos de coincidencia, pero debemos notar que uno de los puntos se encuentra fuera de la Tierra, y el otro es el receptor SPG. De aquí se obtiene el punto en el que se encuentra.

En cuanto a la cuarta medición, se utiliza debido a que el reloj que tiene el receptor no es tan preciso para determinar una buena posición. Nos permite sustituir la imprecisión en el reloj que se encuentra en el receptor.

Si todo fuera perfecto (es decir que los relojes de nuestros receptores SPG lo fueran), entonces todos los rangos (distancias) a los satélites se intersecarían en un único punto (que indica nuestra posición). Pero con relojes imperfectos, una cuarta medición, efectuada como control cruzado, NO intersecará con los tres primeros.

De esa manera la computadora de nuestro SPG detectará la discrepancia y atribuirá la diferencia a una sincronización imperfecta con la hora universal.

Dado que cualquier discrepancia con la hora universal afectará a las cuatro mediciones, el receptor buscará un factor de corrección único que siendo aplicado a sus mediciones de tiempo hará que los rangos coincidan en un solo punto.

Dicha corrección permitirá al reloj del receptor ajustarse nuevamente a la hora universal y de esa manera tenemos un reloj atómico en la palma de nuestra mano.

1.5.4 Posibilidades de incrementar la precisión.

El SPG nos brinda una precisión muy alta, pero existen otras técnicas, las cuales usan el mismo SPG, que nos ayudan a obtener una precisión mayor.

La principal de dichas técnicas en el llamado Diferencial de SPG (DGPS¹⁷ por sus siglas en inglés).

La estructura del DSPG consta de una estación monitora que conoce su posición con una precisión muy alta. Dicha estación tiene un receptor SPG y un microprocesador para calcular los errores del sistema SPG y para generar la estructura del mensaje que se envía a los receptores.

Existe un canal de datos unidireccional hacia los receptores, por lo que necesita un transmisor, la cual es la estación monitora, y los usuarios necesitarán un receptor para recibir dichos datos.

En los mensajes que se envían a los receptores próximos se pueden incluir dos tipos de correcciones:

- Una directamente aplicada a la posición, aunque esta tiene el inconveniente de que tanto el usuario como la estación monitora deberán emplear los mismos satélites, pues las correcciones se basan en esos mismos satélites.
- Una aplicada a las pseudo-distancias de cada uno de los satélites visibles, donde el usuario podrá hacer la corrección con los 4 satélites de mejor SNR. Esta corrección es más flexible.

Algunas de las ventajas que presenta esta técnica es que tiene una cobertura de 200 kilómetros en torno a la estación terrena. En cuanto al segmento de lo usuarios se eliminan los efectos de la ionosfera y troposfera y el parámetro que más afecta es el ruido del receptor.

En la actualidad se están desarrollando sistemas WADSPG¹⁸ (DSPG de área amplia) que no son otra cosa que un DSPG de gran cobertura. Está formado por varias estaciones monitoras DSPG cuyas áreas de cobertura están superpuestas.

1.6 Aplicaciones de los SPG

En cuanto a las aplicaciones de esta tecnología, se puede decir que en la actualidad hay un número muy grande de ellas, y dicho número sigue creciendo.

Si se pudiera clasificar por áreas de uso el tipo de aplicaciones, se haría de la siguiente manera:

¹⁷ Diferencial Global Positioning System

¹⁸ Wide Area Diferencial Global Positioning System

- Navegación marítima. Debido a que los barcos no necesitan SPG de gran calidad, estos últimos han penetrado de manera muy rápida en esta área.
- Navegación aérea. En este ámbito ha sido un poco más lento su intrusión.
- Navegación terrestre. Automóviles y receptores personales, transportes internacionales, redes de autobuses, policía, ambulancias, etc.
- Aplicaciones militares. Es un sistema desarrollado por el ejército, por lo que su intrusión ha sido de las más rápidas. Una de las aplicaciones más conocidas aquí es el guiado de misiles.
- Geografía y Geodesia.
- Otras aplicaciones. Referencia temporal muy exacta, localización y delimitación de zonas afectadas por grandes catástrofes y guiado de vehículos de auxilio.

En cuanto a sus limitaciones, una de sus principales es que depende de un solo país, Estados Unidos. Otra es que basta con que un satélite deje de transmitir la señal para que el SPG deje de funcionar.

Los dispositivos SPG, en cuanto a las aplicaciones dentro de la sociedad, los vemos en muchas de las actividades comunes. Como ejemplo podemos citar la localización de rutas en los taxis, o simplemente para ver el lugar donde están en la ciudad cuando están en horas de trabajo. Este tipo de tecnología les ayuda mucho a ubicarse dentro de la ciudad.

Otra aplicación que se les da a este tipo de tecnología, dentro de las actividades de la sociedad, es la de localización de los equipos de transporte de las diversas compañías de paquetería.

En fin, los usos que se les puede dar son muchos, pero si observamos bien, la mayoría de ellas son orientadas a las empresas, a negocios de transportes o a actividades fuera de la ciudad. Pero que pasa con los particulares que desean saber su ubicación dentro de la ciudad.

Las aplicaciones para los particulares son muy reducidas, pero hay tantas posibilidades de aplicaciones que uno puede pensar que ese terreno no está explotado.

1.7 ¿En qué nos ayudará estos dispositivos?

Una de las aplicaciones que resultaría ser muy práctica sería el tener una aplicación donde nos indicara el lugar exacto en el que uno se mueve dentro de la ciudad, para poder ubicarnos un poco más. Y para esto, el SPG, junto con una computadora portátil, nos serviría bastante ya que este dispositivo.

Y si en lugar de la computadora portátil usáramos un dispositivo más pequeño como una Palm, la aplicación sería mucho más conveniente debido a que es mucho más fácil de cargar y transportar. Otro de los puntos a favor es que estos dispositivos móviles son menos vistosos que una computadora portátil, lo cual los hace más seguros.

Y las dimensiones de la aplicación no pueden quedar hasta allí. Hasta aquí esa aplicación tendría mucha utilidad para lograr una mejor ubicación cuando uno quiera buscar un lugar.

Una aplicación debe ser agradable a la vista del usuario final para que pueda ser de utilidad, y simplemente mostrándole las coordenadas geográficas donde se encuentra no sería suficiente para que las personas usen dicha aplicación.

Para lograr lo anterior, el sistema puede presentarle en la misma Palm un mapa de la ciudad dibujado en su pantalla, para que, con la ayuda de su dispositivo SPG, dentro de dicho mapa pinte el punto donde se encuentra en dicho momento. Sería mucho más agradable y entendible para el usuario, algo que provocaría que el interés por usar el sistema creciera.

Así podemos agregarle otras cosas, como agregar un sistema de trazado de rutas, en donde se nos indique la manera en que nos tenemos que ir de un lugar a otro para ahorrar tiempos y distancias.

En toda esta aplicación el SPG sería de mucha utilidad para calcular el lugar en donde estamos y, a partir de allí, calcular la ruta más corta para llegar a algún otro lugar de la ciudad.

Definitivamente es una aplicación que, de manera obligada, tiene la necesidad de hacer uso de los tipos de dispositivos que hemos detallado en este capítulo, los SPG.

1.8 Conclusión

Por lo tanto, a manera de conclusión, la aplicación planteada en este trabajo de tesis tendrá un gran apoyo en el uso de estos dispositivos. La tarea de la que se encargarán estos dispositivos es obtener las coordenadas geográficas en donde se encuentra dicho dispositivo y poder indicarle al usuario, mediante la aplicación, donde está y por donde puede llegar de manera más fácil a otro lugar.

Capítulo 2

*Ruta mínima
entre dos puntos*

Capítulo 2. Ruta mínima entre dos puntos

Dentro de nuestra ciudad, sea chica o sea grande, en muchas ocasiones nos hemos topado con el problema de ir de un punto a otro de la misma y no sabemos que ruta tomar para acortar distancia.

Es muy común que cuando queremos trasladarnos de un punto a otro dentro de una ciudad buscamos el camino más corto entre el punto donde estamos y el lugar a donde tenemos que llegar. Y esto se puede deber a muchos diferentes motivos. Motivos que pueden ir desde simplemente llegar más rápido a nuestro destino, hasta acortar los gastos de gasolina o bien simplemente, en circunstancias especiales, evitar el tráfico que pueda haber en la ciudad.

En fin, esta necesidad se le presenta a casi todos los ciudadanos, independientemente del uso de un vehículo o no. Bien se podría tratar de una persona que se mueva en bicicleta o caminando y quiera llegar a un lugar de la manera más rápida acortando las distancias.

A este problema se le ha llamado de la ruta mínima. El cual consiste en ver cual es la ruta más corta entre dos puntos, pasando por puntos intermedios.

2.1 Grafos

Para ingresar a la siguiente sección primero tenemos que introducir algunos conceptos.

Un grafo¹⁹ es un conjunto de nodos²⁰ unidos por un conjunto de aristas²¹ que pueden representar alguna situación en particular.

En un grafo, tanto la cantidad de aristas como de nodos son un número finito. Las aristas se encargan de unir a los nodos.

Generalmente, los grafos constituyen una herramienta básica para modelar fenómenos discretos y son fundamentales para la comprensión de las estructuras de datos y el análisis de algoritmos.

¹⁹ Gráfica conformada de nodos y aristas que lleva la finalidad de representar una estructura de datos.

²⁰ Cada uno de los puntos que conforman a un grafo

²¹ Cada una de las líneas que unen a los nodos dentro de un grafo

Cuando el grafo tiene aristas que indican la dirección en la que se dirigen se les llama grafo dirigido.

Cuando tenemos un grafo dirigido y con pesos en sus aristas, se nos puede presentar un problema, el saber cual es el camino más corto de un nodo a otro a través de las aristas existentes.

En el mismo ejemplo de la red de computadoras, podría presentárenos el problema de saber por donde conviene que viaje un paquete de una computadora a otra para que pueda llegar en el menor tiempo posible, en el entendido de que entre más distancia recorra más tarda en llegar.

Este es un problema que puede ser aplicado en muchos ámbitos de la ciencia, por ejemplo en una ciudad en donde uno quiere ir de un lugar a otro, o simplemente cuando queremos encontrar la ruta mínima entre un nodo y otro dentro de un grafo.

2.2 Problema de la ruta mínima

En fin, el problema planteado anteriormente lo han estudiado muchas personas y han llegado a varias soluciones. Una de las más importantes, es la que planteó Dijkstra²², en la cual nos indica la manera de encontrar la ruta mínima que hay entre un par de nodos dentro de un grafo dirigido.

Este algoritmo, que se explicará más adelante en este capítulo, nos permite encontrar, de manera eficiente, el camino mínimo entre dos nodos dentro de un grafo dirigido y con pesos.

Este problema, el de la ruta mínima, es muy común de representar con el problema llamado "el problema del agente viajero". Este problema se detalla a continuación.

2.3 Problema del agente viajero

El problema del agente viajero consiste en encontrar cuál es la vuelta óptima (la más corta) que debe hacer un viajero que debe pasar por n ciudades, las cuales están interconectadas entre ellas por diferentes rutas. Matemáticamente, las ciudades y las rutas que las unen se representan por medio de un grafo, los nodos son las ciudades y las aristas son las rutas entre ellas.

²² Edsger Wybe Dijkstra nació en Rotterdam, Holanda en 1930. Creador del algoritmo de Dijkstra para calcular rutas mínimas entre dos nodos en un grafo dirigido y con pesos no negativos en 1956.

uno de los factores se toman por diferentes motivos, por ejemplo, se pueden tomar los siguientes factores:

- Tiempo de recorrido. Cuando se toma en cuenta el tiempo es porque lo que uno quiere es llegar cuanto antes a su destino, por lo que trata de que el recorrido sea por las calles que le lleve menos tiempo recorrerlas, y en el ejemplo citado arriba el recorrido que menos tiempo le lleva es por las calles donde haya menos tráfico, aunque esto implique que en cuanto a la distancia en metros recorrida no sea la óptima. Esto nos puede llevar a recorrer las calles aledañas a las avenidas principales para evitar el tráfico que existe dentro de éstas avenidas, aunque nos cueste más en cuestión a la longitud recorrida.
- Distancia en metros. En el caso de que tomemos como factor de medición la distancia recorrida podrá deberse a que nuestro vehículo no cuenta con mucho combustible, lo cual nos obligará a recorrer el tramo más pequeño en cuanto a longitud para que nuestro combustible pueda rendir. En este caso tendremos la necesidad de buscar la ruta mínima en cuanto a longitud, lo cual nos puede obligar a que vayamos por las avenidas grandes, las cuales pueden resultar, por ser casi siempre líneas rectas, la ruta más corta en cuanto a longitud.

En fin, como podemos ver, la distancia entre dos puntos la podemos medir de varias maneras de acuerdo a nuestras necesidades.

Transportando este ejemplo de la ciudad a un ámbito matemático, y representándolo mediante grafos podemos poner las siguientes premisas:

- Cada cruce de dos o más calles representará un nodo.
- Cada calle, entre dos nodos (ya definidos en la anterior premisa), representará una arista.
- En el caso de que los pesos de las aristas se midan en tiempos, dichos pesos representarán el tiempo en que se tarda el carro, o caminando o en algún otro transporte, en recorrer la longitud de la arista.
- En el caso de que los pesos de las aristas se midan en longitud, dichos pesos representarán la longitud de cada arista de acuerdo al tamaño real que representan en la ciudad.

Con las premisas planteadas podemos representar a nuestra ciudad a manera de grafo, para que se pueda plantear el problema de ruta mínima entre dos puntos. Con lo cual estaríamos planteando el problema de ir de un punto en la ciudad a otro, y lo podríamos resolver con alguno de los algoritmos que existen para tratar dicho problema.

Para el desarrollo de este trabajo, es necesario plantearlo de esta manera, ya que nos facilitará el trabajo en los cálculos planteados.

un sentido de la calle y otra en el sentido contrario. Y si observamos bien, ambas aristas con el mismo tamaño.

- Dado que el algoritmo de Dijkstra exige que se aplique en un grafo dirigido, cada arista tendrá una dirección, la cual estará definida por la dirección que lleve la calle en la realidad. Lo anterior aplica en caso de los vehículos, pero en caso de personas caminando, generalmente se puede circular en cualquiera de los sentidos sobre las calles, es por esto que se aplicará una doble arista.

Si desde el principio del recorrido hasta el final unimos los diferentes tramos que recorreremos en el camino deseado, vamos a obtener una serie de valores finales como distancia recorrida, o tiempo en que se recorrió dicha distancia, entre otros. Esto podemos hacerlo con diferentes recorridos, de tal manera que lleguemos a tener varios caminos recorridos, con sus respectivos datos, y podamos compararlos para ver cual resulta ser más óptimo en tiempo o en distancia. Si lo hacemos de manera manual puede resultar muy complicado, o en la mayoría de los casos puede resultar muy tardado, ya que implica recorrer en varias ocasiones el camino desde el punto de inicio hasta el punto final por las diferentes rutas que están marcadas como posibles dentro del mapa.

Viéndolo un poco más a detalle, y de manera gráfica, podemos ver que si dividimos nuestro recorrido en esos tramos mencionados podemos tener lo que se muestra en la figura 3:

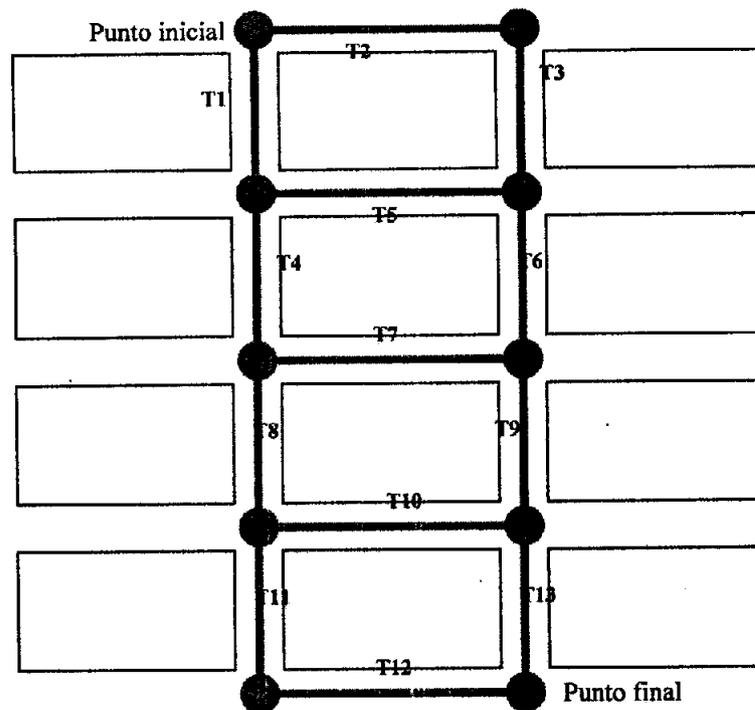


Figura 3. Recorrido de un auto dividido en tramos (aristas en el grafo)

Si vemos, en ésta gráfica tenemos 13 diferentes tramos que se pueden recorrer desde el punto inicial hasta el punto final, y con dichos tramos se pueden formar múltiples caminos, por ejemplo algunos se ilustran a continuación:

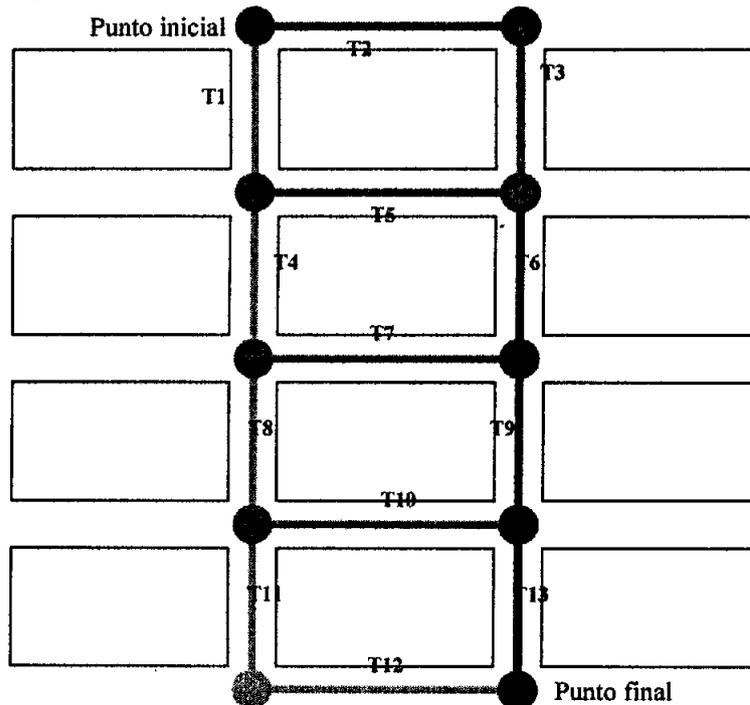


Figura 4. Diferentes caminos a recorrer dentro del grafo.

En la gráfica anterior se tienen dos posibles rutas, formadas por los tramos respectivos, pero se pueden formar otras muchas.

Y, si cada tramo tiene su valor en metros recorridos podemos tener lo siguiente:

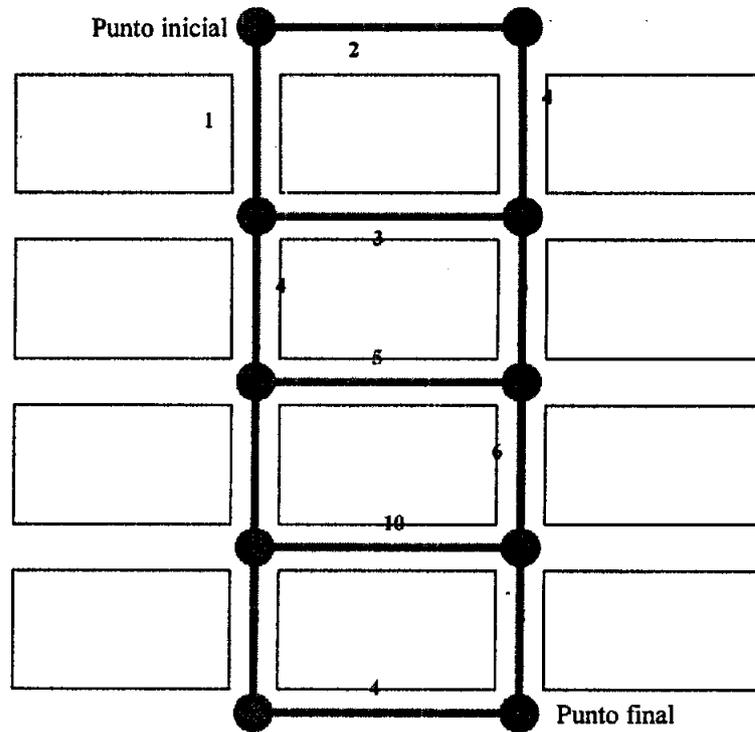


Figura 5. Tramos del recorrido con valores en distancias
(puede ser cualquier medida de distancia)

Suponiendo que los valores anotados en cada tramo son las distancias que se recorren, en metros, si uno recorre dicho tramo, podemos calcular las distancias recorridas en las rutas pintadas de azul y de verde en la ilustración anterior.

- En el caso de la ruta azul:

$$T_2 + T_3 + T_6 + T_9 + T_{13}$$

$$2 + 4 + 3 + 6 + 5 = 20$$

- En el caso de la ruta verde

$$T_1 + T_4 + T_8 + T_{11} + T_{12}$$

$$1 + 4 + 1 + 8 + 4 = 18$$

Como podemos observar, la ruta azul es más larga que la ruta verde.

Tomando en cuenta que las unidades son metros, querrá decir que la distancia recorrida es mayor en la ruta azul que en la ruta verde. Pero si las unidades marcadas en la ilustración son unidades de tiempo, querrá decir que el tiempo de recorrido en la ruta azul es mayor al tiempo de recorrido de la ruta verde.

Así, con todo lo planteado en las premisas anteriores, podemos formar un grafo dirigido y con pesos en sus aristas, tan grande como la ciudad lo exija, con nodos y aristas que contengan datos precisos y necesarios para que el algoritmo de Dijkstra pueda entrar a calcular rutas mínimas dentro de dicho grafo. Con esto podemos estar calculando las rutas más cortas entre dos puntos de la ciudad en donde estamos ejecutando el algoritmo.

De esta manera, una vez planteado la manera de tratar a nuestros mapas de la ciudad, podemos pasar a detallar la manera en que funciona el algoritmo de Dijkstra.

2.6 Algoritmo de Dijkstra

(Para el desarrollo de esta parte del capítulo se utilizó la siguiente bibliografía: [Cormen], hyperlinks 7, 8, 9)

Los dos principales algoritmos usados para atacar este tipo de problemas son el algoritmo de Dijkstra y el algoritmo de Bellman-Ford.

Tanto uno como otro son algoritmos eficientes y rápidos, y difieren en pocas características. Ambos recorren un grafo dirigido formado de aristas con peso (como el mostrado en la figura 6 y 7) y calculan la ruta mínima entre dos vértices dados.

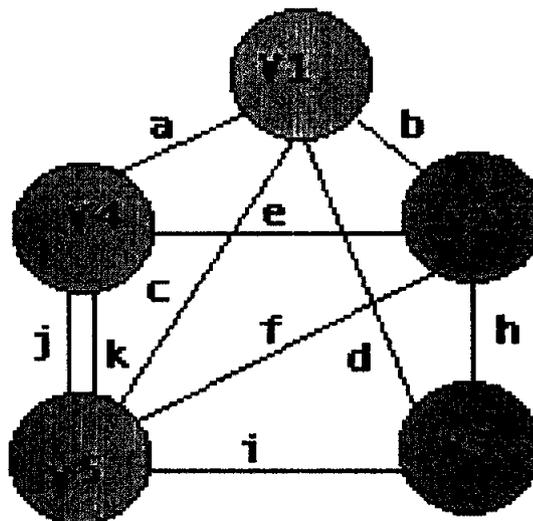


Figura 6. Grafo de aristas con peso

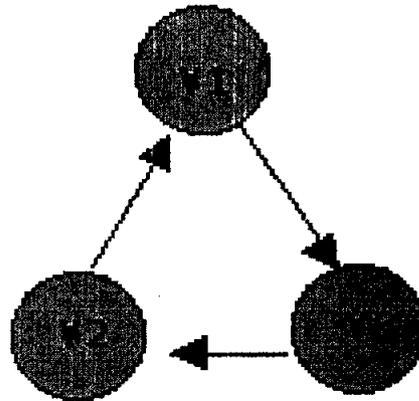


Figura 7. Grafo dirigido

Una de las cuestiones fundamentales en las que estos dos algoritmos difieren es que en el algoritmo de Dijkstra no se permiten pesos negativos en las aristas, ya que podría provocar errores en la obtención de las rutas mínimas, mientras que en el algoritmo de Bellman-Ford si se permite que las aristas tengan un peso negativo, sin provocar algún error.

Para el caso de la aplicación a desarrollar podemos tomar en cuenta que dichos mapas, si se abstraen en forma de grafos, tienen las siguientes características:

- Las aristas no pueden tener valores negativos, ya que se trata de distancias recorridas de un punto a otro.
- En el caso de las calles de doble sentido, el grafo desplegará dos aristas, una que va y otra que viene, del mismo peso, pero de diferentes sentidos.
- Las aristas estarán dirigidas de acuerdo al sentido que tenga la calle a la que represente.
- Los vértices estarán representados por las intersecciones de las calles que existan dentro del mapa.
- Habrá vértices que no serán intersecciones de calles, ya que se tratarán de calles cerradas. En estos casos el vértice estará representado por el final de dicha calle cerrada.

Dado que los mapas que se manejarán en este sistema tendrán las características que arriba se mencionan, bastará con utilizar el algoritmo de Dijkstra para calcular las rutas mínimas entre dos puntos dentro del mapa, ya que las aristas manejadas no tendrán pesos negativos, por lo cual el algoritmo citado no tendrá problemas en su ejecución.

Por lo tanto, de los dos algoritmos que se analizaron, se llega a la conclusión de que el algoritmo de Dijkstra es suficiente para las características que va a tener la funcionalidad de la búsqueda de las rutas mínimas entre dos puntos dentro del mapa.

Una vez planteado el algoritmo a usar, detallemos el mismo.

El algoritmo de Dijkstra nos encuentra la ruta más corta entre los nodos s (nodo inicial) y t (nodo final). Suponemos que la variable 'infinito' contiene el entero más grande posible. La variable 'distancia[i]' contiene el costo de la trayectoria más corta conocida y por tanto, alejada de s a i .

En un principio,
distancia[s]=0
distancia[i]=infinito, para todo i dentro del conjunto de nodos

Un conjunto 'perm' contiene todos los nodos de los que se conoce su distancia mínima de s , es decir, aquellos nodos cuyo valor de distancia es permanente y no cambiará. Si un nodo i pertenece a perm, distancia[i] es la distancia mínima de s a i .

En un principio, el único nodo perteneciente a perm es ' s '. Una vez que t se vuelve perteneciente a 'perm', se sabe que distancia[t] es la distancia más corta de s a t y termina el algoritmo.

El algoritmo conserva una variable, llamada 'actual', que es el nodo que se ha agregado a 'perm' más recientemente.

Inicialmente,
actual = s

Cada vez que se agrega un nodo 'actual' a 'perm', debe recalcularse la variable 'distancia' para todos los sucesores de 'actual'. Para todos los sucesores ' i ' de 'actual', si 'distancia[actual]' + peso(actual, i) es menor que distancia[i], la distancia de ' s ' a ' i ' a través de 'actual' es más pequeña que cualquier otra distancia de ' s ' a ' i ' encontrada hasta ese momento. Por tanto, distancia[i] debe restablecerse en este valor más pequeño.

Una vez que se ha recalculado 'distancia' para todos los sucesores de 'actual', distancia[j](para cualquier j) representa la trayectoria más corta de ' s ' a ' j ' que sólo incluye nodos de 'perm' (excepto tal vez j mismo). Esto significa que para el nodo k , que no está en 'perm' y para el cual distancia[k] es la más pequeña, no hay una trayectoria de ' s ' a ' k ' cuya longitud sea más corta que distancia[k]. distancia[k] ya es la distancia más corta a k que incluye sólo nodos en 'perm' y cualquier trayectoria a ' k ' que sólo incluya nodo 'nd' como su primer nodo que no está en 'perm' debe ser más larga, dado que distancia[nd] es mayor que distancia[k]. Por tanto, se agrega ' k ' a 'perm'. Después, se restablece 'actual' en ' k ' y se repite el proceso.

2.7 Aportación del algoritmo de Dijkstra en este trabajo de tesis

Definitivamente el algoritmo de Dijkstra será de las herramientas más importantes que se usen en la construcción del sistema al que queremos llegar en este trabajo de tesis.

Con él podremos calcular las rutas mínimas desde un punto a otro dentro del mapa que pintemos en la palm.

Será necesario tener toda la ciudad separada por segmentos, tal cual lo vimos en este capítulo, y aristas, con sus respectivos datos para poder enviarle al algoritmo de Dijkstra dichos datos y que este pueda calcular la ruta deseada. Con esto, el usuario final podrá tener la facilidad de moverse por el camino que más le convenga en su traslado.

Capítulo 3

*Mapas en la
Palm*

Capítulo 3. Mapas en la Palm

Hoy en día las Palm están viviendo una etapa de mucho auge dentro de la sociedad, ya que ésta última está aceptando la incursión de este tipo de tecnologías dentro de la misma.

Cada vez más las personas usan estos dispositivos dentro de sus actividades cotidianas. Lo vemos en los altos ejecutivos que los usan para mantener su agenda al día, y tener presente parte de sus actividades diarias. Pero no sólo en ellos lo vemos, ya que hoy en día las personas de cualquier índole están empezando a usarlos para ese mismo control en sus actividades, e inclusive para divertirse con juegos o con música.

Poco a poco se adentra en el gusto y en las necesidades de las personas de la sociedad.

3.1 Características de la Palm

Las Palm han ido mejorando en capacidades físicas, por ejemplo la memoria ram que tienen es mayor que cuando aparecieron, las pantallas que manejan tienen mejor resolución y manejan gran cantidad de colores, el procesador que tienen cada vez es más poderoso, entre otros aspectos de la cuestión física de estos dispositivos.

Este incremento en las capacidades físicas se ha dado de manera vertiginosa, ya que solamente unos años atrás, los primeros modelos de este tipo de dispositivos móviles aspiraban a capacidades físicas muy reducidas.

Dicho incremento en las capacidades ha venido a contribuir en la mejora de las aplicaciones que se pueden ejecutar dentro de estos dispositivos. En un principio sólo eran aplicaciones sencillas y sin mayor ciencia, ya que la memoria y el procesador que utilizaban no soportaba aplicaciones muy complejas.

Hoy en día existen aplicaciones altamente complejas para este tipo de tecnologías, y dado que sigue creciendo en sus capacidades físicas, la complejidad de las aplicaciones que salen al mercado también crece al mismo ritmo.

Vemos que el crecimiento de estos dispositivos en sus características de hardware pareciera hacerlo sin problemas, pero a pesar de esto se enfrentan a otro tipo de problemas. Problemas como su tamaño y forma, características que definen a este tipo de dispositivos y que no pueden cambiar. Con esto los dispositivos móviles de esta índole tienen la desventaja de que las aplicaciones que puedan ejecutarse en ellos están limitadas en el

sentido del tipo de entrada, la cual se hace no con el teclado sino con el lápiz y la pantalla sensible a los toques en la pantalla, la cual a pesar de la resolución que tiene no deja de ser de un tamaño muy pequeño, el procesador que maneja, el cual a pesar de los avances de la tecnología para este tipo de dispositivos no deja de ser un procesador limitado, y la capacidad de almacenamiento, la cual es muy reducida en comparación de las computadoras de escritorio, entre otras.

Lo anterior provoca que los desarrolladores de las aplicaciones para este tipo de dispositivos se las ingenien más para que, con esas características, desarrollen aplicaciones acordes a los mismos.

Detallando un poco más las características generales de estos dispositivos podemos decir lo siguiente:

- Pantalla: Alrededor de mide 5 x 5 centímetros. Generalmente son, en los últimos modelos, de mucha resolución y manejan gran cantidad de colores. Pero aún con estas disposiciones de los equipos nuevos el problema se presenta en el momento en que queremos acomodar los elementos que deseamos tener dentro de nuestra aplicación. Esta operación dentro de la creación de aplicaciones se vuelve difícil ya que generalmente no hay espacio para todos los elementos que llevaría una aplicación normal.
- Memoria. Generalmente, las aplicaciones necesitan una cantidad de memoria considerable para realizar todas las operaciones necesarias. En el caso de las Palm, a pesar de que están evolucionando rápidamente, tenemos una memoria reducida en comparación con la memoria que podríamos tener en una computadora de escritorio. Es por esto que en este tipo de aplicaciones tenemos que optimizar este recurso. En estos dispositivos no podemos darnos el lujo de estar desperdiciando los recursos con que cuenta, ya que son limitados.
- Procesador. Ciertamente que su evolución ha sido admirable, pero aún así sigue limitado. Su evolución no es suficiente para contar con aplicaciones grandes corriendo sin la preocupación de que se pueda ejecutar lenta. Por esta razón, las aplicaciones para Palm tienen que optimizar también el número de operaciones que se realizan dentro de cada una de las librerías que construimos. En este rubro y en el anterior tenemos que tener mucho cuidado, ya que este tipo de dispositivos suele quedarse en un estado en el que no responde porque se han acabado estos dos recursos.
- Dispositivos de entrada. En comparación con una computadora de escritorio, la Palm no tiene un ratón o un teclado que hagan más fácil la interacción entre el usuario y el dispositivo, esto contribuye a que el desarrollo de aplicaciones para este tipo de dispositivos sea más difícil.

Como vemos, las características de hardware que una Palm tiene pueden ser muy reducidas en comparación a las de una computadora portátil o una computadora de escritorio, lo cual puede resultar en un inconveniente a la hora de realizar software para dichos dispositivos.

3.2 La Palm para aplicaciones gráficas

Una de las grandes desventajas que tienen estos dispositivos es el tamaño de su pantalla, ya que esto nos limita en varios aspectos como lo veremos a continuación.

Generalmente en un dispositivo de este tipo es difícil armar una interfaz gráfica. En la mayoría de las ocasiones se debe al tamaño de su pantalla, ya que uno está acostumbrado, como desarrollado, a tratar con una pantalla grande.

Ahora bien, cuando se trata de aplicaciones donde se tiene que dibujar elementos de interfaz gráfica comunes, como botones, cajas de texto, listas de selección, entre otros, el acomodo en la pantalla de la Palm se vuelve menos difícil debido a que son elementos que uno sabe, con cierta facilidad, de que manera deben acomodarse. Esto, unido a un menú de diferentes opciones, puede lograr obtenerse un buen acomodo, teniendo en pantalla todo lo necesario para que la aplicación tenga la imagen que uno quiera.

Pero ¿qué pasa cuando se trata de una interfaz gráfica de usuario donde lo único que se dibuja es un mapa en toda la pantalla? En estos casos ¿donde pondríamos los menús de opciones, o los botones para ejecutar diversas funciones de la aplicación?

Si bien es cierto que en una aplicación como la que se quiere desarrollar en este trabajo debe tener dibujado el mapa en toda su pantalla, también es cierto que no solamente se limita a eso su funcionalidad. También tiene funciones como búsqueda de rutas mínimas, pintado del punto de localización, entre otras; y dichas funciones, de alguna manera, con mapa o sin mapa, tienen que ser accedidas. Bien es cierto también que, debido al tamaño de la pantalla del dispositivo, la distribución de dichos elementos tiene que ser bien planeada rigurosamente, ya que en una pantalla de dicho tamaño, o se coloca el mapa entero o se colocan los botones o el menú de opciones.

Es aquí donde las técnicas de presentación de los diferentes elementos de la aplicación, en nuestro caso el mapa y las diferentes maneras de acceder a la funcionalidad del sistema, deben ser muy eficientes y de mucha imaginación.

3.3 Forma en que se dibujarán los mapas

Sin duda, por lo que ya se comentó en el apartado anterior, cuando uno dibuja mapas dentro de estos dispositivos tiene que distribuir los elementos de manera muy conveniente por este tipo de limitantes. El menú de la aplicación, los botones de la aplicación y algunos detalles extra le quitan espacio muy valioso al área donde se puede dibujar el mapa.

Partiendo desde este punto de discusión, cuando uno realiza este tipo de aplicaciones puede diseñar varias maneras de presentar los datos.

1. Dedicar una parte de la pantalla al dibujado de mapas. Esta técnica consiste en dedicarle solamente una parte de la misma pantalla al dibujo del mapa, mientras que la otra parte contendrá los botones y los elementos de la interfaz gráfica necesarios para que el usuario le indique al sistema las funciones que aquel quiere que este lleve a cabo.

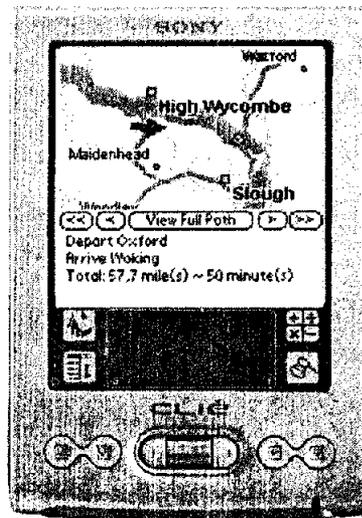


Figura 4. Mapa dibujado solamente en una sección de la pantalla de la Palm (Link 12)

Podemos observar que solamente una parte de la pantalla está dedicada al pintado del mapa. La parte superior de la pantalla se dedica a mostrar datos que la aplicación te provee, datos que, de acuerdo al diseño del sistema que se hizo, son presentados en el mismo momento en que se presenta el mapa que se usa.

Esta manera de presentar los datos puede ser muy útil si esos datos extra son obligatorios para que el usuario los vea, ya que puede tener presente ambas cosas dentro de la misma pantalla. Pero tiene la desventaja de que le quita espacio al mapa, por lo que se dibujará un mapa más pequeño, lo que puede implicar que sea menos agradable a los ojos del usuario final.

2. Mapa en toda la pantalla. Otra manera de presentar estos datos, es pintando el mapa en toda la pantalla, sin dejar espacio para los datos extra. En esta opción, dado que es obligatorio por la manera en que se trabaja dentro de la pantalla, se dejaría el espacio para un menú contextual, dentro del cual se presentarían todas las opciones necesarias para la interacción del usuario con el sistema.

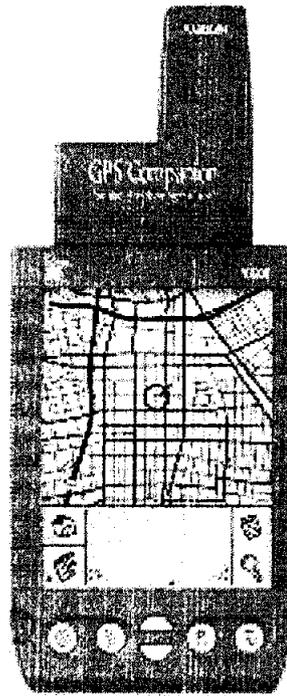


Figura 5. Mapa dibujado en toda la pantalla (Link 14)

Aquí el mapa se presenta más grande, con más detalle y, quizá, podría resultar más atractivo al usuario final.

Mientras que el acceso a la demás funcionalidad del sistema puede accederse a través del menú contextual, el cual se tiene acceso dando un clic en la parte superior de la pantalla. En dicho menú se pueden acomodar todos los botones de acceso a todas las demás funcionalidades.

El problema de presentar de esta manera la aplicación es que cuando el usuario acceda a las diferentes opciones del menú, se tendría que desaparecer el mapa, con lo cual el usuario pierde la vista del mismo. Y, en un momento dado, si el usuario necesita estar viendo el mapa, podría causar alguna incomodidad.

3. Una manera muy vistosa en que se pueden presentar los datos es pintando en todo el espacio de la pantalla el mapa, y dejando el acceso al menú en alguna

parte pequeña de la pantalla, visible mediante un pequeño dibujo, pero sin desplegarse. En el momento en que se de clic en dicho dibujo, representante del menú, aparezca este ocupando una parte más vistosa de la misma pantalla pero sin dejar de dibujar el mapa en la parte trasera de la misma. En esta forma, el mapa nunca se borraría y el usuario tendría la posibilidad de tener acceso, de una manera más dinámica y vistosa a las diferentes opciones de la funcionalidad del sistema desarrollado.

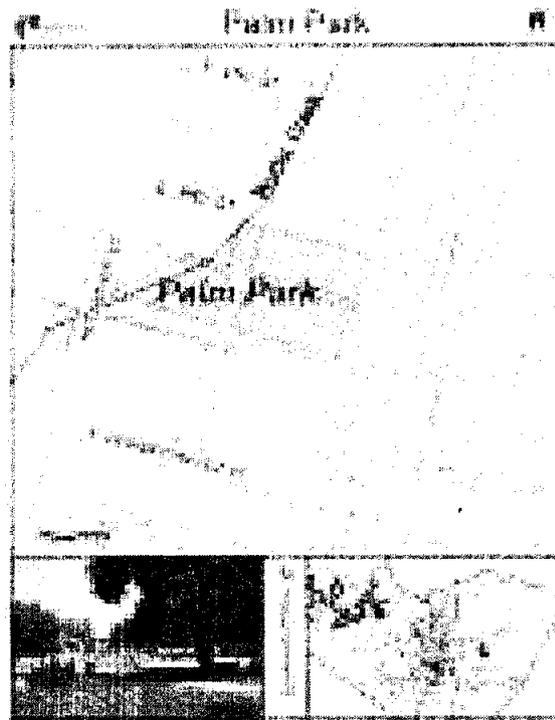


Figura 6. Mapa en toda la pantalla con acceso a menú en a parte inferior.

De esta forma, podemos observar que cuando el menú se despliega, ocupa una parte de la pantalla, pero no provoca que el mapa desaparezca o deje de ser visible para el usuario.

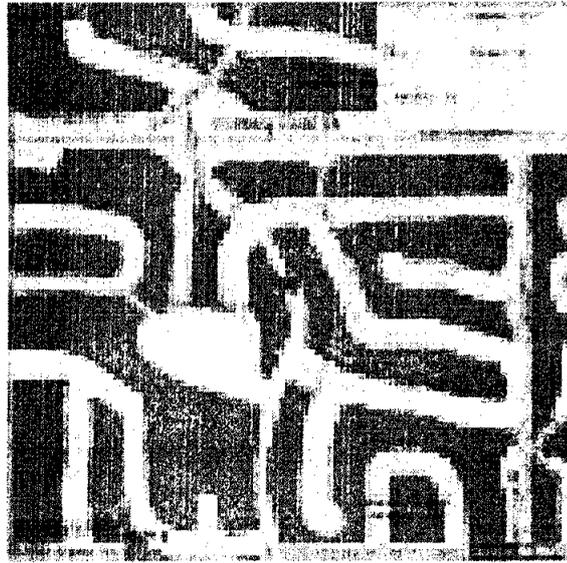


Figura 7. Menú desplegado dentro del mapa.

Inclusive, en esta opción, si se quiere presentar algo más allá del menú de opciones de la aplicación, se le puede asignar otra parte de la pantalla que no sea la que ocupa el mismo menú, de la misma manera, sin modificar el área donde se pintó el mapa. Como se puede observar, podemos agregar más detalles arriba del mapa sin dejar de presentar a este en toda la pantalla.

Las formas en que este tipo de aplicaciones se pueden presentar son muy diversas. Recordemos que los mapas no dejan de ser un conjunto de píxeles²³ dibujados, por lo que podemos hacer con ellos un uso a nuestra conveniencia para pintarlos de diferentes formas. Algunas de estas formas son más complicadas, en cuestión de uso de procesador y memoria, que otras, pero se debe idear la mejor manera de presentárselo al usuario, ya que a este sólo le interesa que su aplicación funcione, y que se sienta a gusto utilizándola.

Cuando un mapa lo recibe y lo lee la Palm, se puede tratar no solamente de un conjunto de píxeles, sino se puede tratar de otros tipos de datos. La interpretación, y utilización, de dichos datos nos dará la pauta para decidir la manera en que podemos presentar el mapa contenido en dichos datos.

²³ Unidad de medida de la resolución de las pantallas en las computadoras, o en las Palm's. Cuando uno dibuja algo en la computadora, se dice que está dibujando píxeles, los cuales son cada punto que está contenido en dicho dibujo. Estos píxeles, a la hora de dibujarse, se dibujan con un color en específico, y, junto con los demás píxeles dentro de la pantalla dan forma a lo dibujado.

3.4 Formato de los datos de un mapa

Un mapa puede ser representado de dos maneras:

- a) **Raster.** Es un conjunto de puntos que se pintan en la pantalla sin ninguna modificación alguna. Esta manera de representar imágenes nos limita su manejo, ya que no tenemos en nuestras manos las características internas de la imagen, sino simplemente tenemos una representación de los píxeles que se quieren dibujar, con lo cual no podremos realizar operaciones tales como zoom de manera precisa y conveniente.

Mencionamos que estas imágenes son un conjunto de puntos que se pintan en la pantalla, pues dicho conjunto de puntos tienen un color definido, el cual será usado cuando se tenga que desplegar en algún programa. Podemos observar que estos datos no nos dicen mucho acerca del mapa, de hecho no nos dicen que se trata de un mapa, pero nos dicen lo suficiente para, por lo menos, dibujarlo en la pantalla, aún sin saber que se trata de un mapa.

Uno de los más conocidos dentro de este tipo de archivos es el de extensión BMP, el cual guarda todos los píxeles que contiene una imagen, de tal manera que cuando se trata de volverla a pintar en la aplicación que lo lee, solamente se tiene que leer cada uno de los píxeles para pintarlos en dicha aplicación. Este tipo de archivos resulta ser muy grande en tamaño, algo que puede resultar en un difícil manejo del archivo.

Otros tipos de archivo son los archivos GIF y JPEG. Se crearon para manejar archivos más ligeros en cuanto tamaño, para que resulten más manejables, pero de alta calidad. Al no ser de gran tamaño, o por lo menos tener un tamaño menor al de los archivos de tipo BMP, son muy utilizados en el mundo del internet. Cada uno de ellos tiene su correspondiente algoritmo de construcción y de lectura bien definido, lo cual los hace altamente accesibles a la hora de leerlos.

Así como estos dos tipos de archivos, también existen muchos otros tipos que se usan dentro del ámbito de las imágenes en la computación.

Lo importante de este tipo de archivos es la manera en que manejan las imágenes, es decir, a través de la representación, mediante algoritmos de compresión o en bruto, de cada uno de los píxeles que la misma imagen contiene en su interior.

En el caso del dibujado de mapas, este tipo de imágenes es muy útil porque de esta manera se puede pintar el mapa de una manera en que el usuario lo entienda

fácilmente, ya que lo que va a observar es lo que existe, a manera de traza, en la realidad.

Lo anterior por el lado del usuario es muy útil, pero para fines de la aplicación, esto puede resultar muy ineficiente, ya que la aplicación no necesita píxeles para realizar la búsqueda de las rutas mínimas, sino que necesita los datos que representan dichos píxeles dentro del mapa y dentro de la ciudad en la que se está trabajando, y que el mapa representa.

- b) Vectorial. Este es un tipo de imagen que se maneja a través de un conjunto de vectores.

La imagen entera está formada por un conjunto de vectores que, a la hora de pintarlos en la pantalla, se lee cada uno de dichos vectores y se pinta de acuerdo al gusto del programador.

Esta manera de manejar las imágenes nos permite una gran flexibilidad en la manipulación de la misma, ya que podemos realizar operaciones sobre la imagen más detalladas y complejas como un zoom o una rotación, entre muchas otras. Dichas operaciones se podrán realizar, con esta forma de representar a los mapas, de manera más precisa.

De estas dos opciones presentadas, la más viable, por sus características, y adaptable a nuestra aplicación es la segunda, las imágenes vectoriales.

Dentro del mundo del formato vectorial, existen muchas maneras de representar los vectores que se encuentran dentro de una imagen y dentro de los mapas. Entre los más conocidos se encuentra el formato en XML que lleva por nombre de SVG²⁴.

Pero existen otras maneras de representar los vectores dentro de un archivo. Por ejemplo, el formato que se usa para el manejo de vectores en las imágenes difiere mucho de aplicación en aplicación, así lo podemos ver cuando desarrollamos dibujos de alta calidad en AutoCad y CorelDraw, ya que estos dos software de dibujo manejan archivos de imágenes vectoriales, pero el formato que genera cada uno en su aplicación, así como el que lee cada uno, es diferente. Esto se debe a que cada uno necesita diferentes datos en la representación y el manejo de estos vectores.

Volviendo un poco al formato SVG, una de las maneras en que se representan los vectores de la imagen que se está usando es a través del formato xml²⁵. Este formato tiene la ventaja de ser muy fácil a la hora de leerlo y resulta, así mismo, fácil también de entenderlo, y como el formato xml es un estándar que todo mundo puede usar, pues esto facilita que cualquier programa pueda leerlo, interpretarlo y procesarlo, pues bastaría que el

²⁴ Scalable Vector Graphics, por sus siglas en inglés.

²⁵ Lenguaje de etiquetas que nos ayuda a transportar datos de manera fácil. Existe un estándar definido acerca de la manera en que se debe escribir el xml.

programa tenga el dato de cómo tiene que interpretar cada una de las etiquetas que maneja el xml.

El formato SVG define, a través de etiquetas xml, los múltiples vectores y características vectoriales de las que está formada la imagen que representa el archivo, para que este, al ser leído e interpretado pueda tener lo necesario para construir la imagen que se quiere construir.

La ventaja de este tipo de imágenes, con respecto a las imágenes raster, es que ocupan menos espacio en disco duro, ya que pueden definir vectores muy largos con simples líneas de caracteres, mientras que en los raster tienen que definir píxel por píxel.

Otra de las ventajas que tiene este tipo de archivo es que puede ser leído por los ojos de un ser humano y, a la vez, entendido. Con lo cual no se necesita de ningún programa para saber que se tiene en dicha imagen.

Si a dicho formato le agregamos, porque el xml nos lo permite²⁶, otras etiquetas creadas por nosotros mismos, podemos obtener un formato xml ideal para manejar toda la funcionalidad de nuestro sistema.

También existen los formatos DWG, DXF²⁷ y CGF²⁸. Los cuales se analizarán debido a que son los tipos de formato en que vendrán los mapas en su formato original.

3.5 SVG más pequeño que RASTER

Para dejar más claro este concepto del porque el formato SVG ocupa menos espacio en los archivos que el formato Raster, haremos un ejemplo de dibujo donde se pintarán 4 cuadros del mismo tamaño en un espacio de 1024 puntos por 768 puntos.

De esta manera si tenemos en la pantalla el siguiente dibujo:

²⁶ El xml, de acuerdo a su definición, nos indica que podemos crear nuestro propio conjunto de etiquetas para representar datos para poder combinarlas con otro tipo de etiquetas, que representan otros datos, para tener una mayor funcionalidad en el formato xml final. De esta forma tendremos un formato híbrido.

²⁷ DWG y DXF son formatos que fueron creados, y son manejados por sus aplicaciones, por Autocad. La descripción del formato de este tipo de archivos puede obtenerse en la página "www.myfileformats.com"

²⁸ Formato que, a base de texto, representa los vectores contenidos en una imagen. La descripción de este formato se puede obtener en la página "www.airporttools.com/apecs/animator/docs.html"

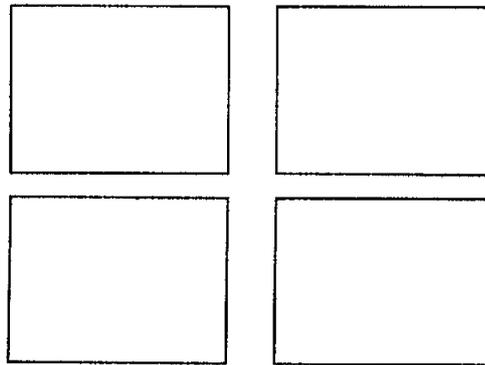


Figura 5. Cuatro cuadros del mismo tamaño.

que sea de tamaño 1024 x 768. Con la metodología raster, el archivo puede tener varias longitudes, y dicha longitud dependerá del número de colores que maneje. Por ejemplo si para el manejo de cada color basta con 1 bits, es decir blanco y negro, el tamaño del archivo será de

$$(1024 \times 768 \times 1) + \text{Tamaño de cabecera en bits}$$

y esto nos da como resultado a más de 98 Kb, es decir casi 100 Kb.

Pero si el número de bits que se usa para el manejo de los colores es de 24 bits, entonces el tamaño se calcula de la siguiente manera

$$(1024 \times 768 \times 24) + \text{Tamaño de cabecera en bits}$$

por lo que el tamaño del archivo de dicha imagen será de aproximadamente 2,359,296 bytes, es decir 2 megas. Y esto solamente para dibujar un cuadro de 1024 puntos por 768 puntos, con una calidad de color de 24 bits.

Respecto a este mismo ejemplo, pero escrito de manera vectorial, lo único que se tendría que hacer es definir el cuadro que se encuentra en la parte superior izquierda. Una vez definido dicho cuadro, a este se le hace 3 copias y las coloca en las posiciones correctas. Con lo cual se logra tener un tamaño de archivo mucho menor, y esto depende de la manera en que se declare cada uno de los cuadros.

La forma de declarar cada uno de los cuadros mediante vectores²⁹ puede ser de la siguiente manera:

Si se tiene el siguiente cuadro



Vemos que está compuesto de dos vectores:

```
EO,5,0,0,#FFFFFF
EO,5,0,2.5,#FFFFFF
NS,2.5,0,0,#FFFFFF
NS,2.5,5,0,#FFFFFF
```

Donde el primero, utilizando esta nomenclatura personalizada, significa que es un vector que va de Este a Oeste y mide 5 unidades y empieza desde la coordenada (0 , 0) y tiene el color FFFFFFFF representado en notación hexadecimal, así como el segundo. Y el tercero va de Norte a Sur y mide 2.5 unidades y empieza desde la coordenada (0 , 0) y tiene el color FFFFFFFF representado en notación hexadecimal, así como el cuarto vector.

En cuanto a las unidades, estas pueden ser cualquier tipo de unidades, y estas se pueden definir en un sólo renglón del archivo, para no estar repitiendo la palabra cada que haya que indicar unidades. Y se puede indicar mediante la primera línea con un par de letras, por ejemplo me para metros.

De esta manera podemos definir nuestro archivo con el siguiente texto

```
Me
EO,5,0,0,#FFFFFF
EO,5,0,2.5,#FFFFFF
NS,2.5,0,0,#FFFFFF
NS,2.5,5,0,#FFFFFF
```

En donde podemos observar que solamente ocupa 80 bytes. Y si esto lo repetimos cuatro ocasiones para pintar los cuatro cuadros podemos obtener un archivo de

²⁹ El formato que aquí se define para representar a los vectores dentro de un dibujo se construyó para fines de este ejemplo, y no es usado por ninguna aplicación en el mundo. Es simplemente, por comodidad del manejo interno, y para dejar claro este ejemplo, para fines educativos.

aproximadamente 350 bytes (el sobrante de bytes es por aquello de los saltos de línea y de la opciones de cabecera extra).

Si observamos, el archivo que se generó con el formato inventado para este ejemplo resultó ser de mucho menor tamaño que el archivo de la imagen raster. Esta diferencia se nota porque este archivo sólo define lo necesario, lo demás lo omite, en cambio la imagen de tipo raster guarda todos y cada uno de los píxeles que existen, aunque estos se repitan. Vale la pena aclarar que el formato que se creó para este ejemplo es inventado con la finalidad de hacer ver la diferencia entre los archivos vectoriales y los archivos de tipo raster.

Ahora bien, dentro de las imágenes raster existen métodos para comprimir³⁰ el tamaño de la imagen sin degradar mucho su calidad. Estos métodos realizan, comúnmente una búsqueda de los píxeles que se repiten y los guardan una sola vez, con esto ayudando a comprimir la imagen creada. El procesamiento que lleva la lectura y escritura de estas imágenes comprimidas es mayor que las que no están comprimidas.

Ahora bien, el protocolo que se usa para la creación de vectores varía de acuerdo a las necesidades del usuario, esto nos puede llevar a tener archivos de mayor tamaño que el calculado aquí, pero la diferencia en tamaños seguirá siendo abismal.

Y, en el caso en que esta diferencia se acortara mucho, inclusive que el formato vectorial resultara ser más grande que el formato raster, seguirá siendo conveniente este formato para esta aplicación por la cantidad de datos que están contenidos en este, más allá de la manera en que se va a dibujar la imagen.

3.6 En qué nos ayudarán los formatos de los mapas

A la hora de presentar los mapas en la Palm, tenemos que leerlos de algún lado para pintarlos de alguna manera.

Pero la función del sistema a desarrollar no se puede quedar allí, ya que se necesitarán datos como los siguientes:

- c) Coordenadas de cada punto representado en la imagen. Esto con la finalidad de pintar el punto donde nos encontramos de manera correcta dentro del mapa. Cuando el SPG nos da las coordenadas exactas en la que estamos en el mundo, no nos servirán de nada si no sabemos en que parte del mapa se encuentran.

Otra ventaja de esto es que sabremos de cada nodo del grafo su posición exacta, con lo cual se facilitará el proceso de búsqueda de la ruta mínima.

³⁰ Técnicas usadas para que la imagen dentro del archivo, sin perder sus características de dibujado, ocupe menos espacio. Generalmente se realiza buscando los píxeles repetidos y guardándolos una sola vez.

- d) Definición de cada uno de los bloques de la ciudad. Esto nos ayuda a definir cada una de las cuadras de la ciudad, y, por ende, cada una de las calles. Con esto podemos identificar, dentro del sistema, cuales lugares son aptos para transitar y cuales lugares no lo son.

- e) Sentido y dirección en que van cada una de las calles. Observamos de manera clara que si una imagen de tipo raster no nos puede decir cuales son las calles, con mayor razón no nos podrá decir el sentido y dirección de las mismas. En cambio, el formato vectorial nos puede facilitar el trabajo delimitándonos las calles que existan en el mapa, indicándonos con esto donde se encuentran cada uno. Pero de no mucho servirían si no nos dice el sentido y la dirección que recorren, ya que, por sí mismo, el sistema no es capaz de identificar estos datos.

Otra ventaja que nos presenta este tipo de datos es que son datos que se usarán para construir el grafo con el cual se calculará la ruta mínima que hay entre dos nodos, aparte que nos ayudará a saber por donde se tiene que pintar la ruta mínima obtenida.

Con estos y otros datos se realizará una aplicación más completa, lo cual nos permitirá darle mayor funcionalidad a la misma.

En cuanto a la manera de dibujarlo, creemos que el segundo de los métodos planteados es el más conveniente para el usuario. Y, aunque es más complicado el procesamiento para la Palm, sabemos que es más agradable a los ojos del usuario final, y, recordemos, si al usuario le agrada como se ve es casi seguro que compra³¹ el producto desarrollado.

3.7 Conclusión

En conclusión, la manera de manejar los datos de un mapa conviene que sea de manera vectorial, porque de otra manera se vuelve muy grande el archivo, y sabemos que en las Palm existe el problema de almacenamiento, por lo que se verá afectado ese rubro en el dispositivo.

Otra de las ventajas que se vieron en el formato vectorial, es que los datos pueden estar contenidos en dicho archivo, ya que el mismo archivo puede tener datos que ayuden mucho a la realización correcta de las operaciones dentro del sistema.

Pero, aun tomando en cuenta el manejo de datos, tenemos que ver también que a la hora de pintar hay que leer los datos generados en el archivo de texto y pintar el mapa de acuerdo a lo que se lee e interprete, y esto puede resultar en mayor procesamiento para el

³¹ Expresión usada para indicar que el usuario querrá usar el producto

procesador del dispositivo. Este pintado resulta ser punto por punto porque no se puede hacer vector por vector debido a que la pantalla de nuestro dispositivo recibe solamente píxeles.

Todo lo anterior nos habla de que se usan dos tipos de metodologías en el manejo de imágenes para el simple pintado de un mapa en la pantalla, y por lo tanto nuestra aplicación: la metodología de imágenes vectoriales para representar los mapas en archivos³², ya sea de texto o binarios, de tamaño reducido, y la metodología raster a la hora de pintar dichos mapas en la pantalla de la Palm, después de un proceso de conversión de los datos en el formato vectorial a los píxeles en el formato raster.

De esta manera se puede atacar el problema del pintado de mapas. En tanto que para transportar los datos se utilizan archivos binarios o de texto en formato vectorial, que representen cada una de las imágenes de mapas, mientras que el pintado en el dispositivo se hace píxel por píxel.

³² Vale la pena aclarar que estos archivos no sólo contendrán los vectores que definen el mapa, sino que también contendrán datos que ayuden al sistema a hacer cálculos dentro de la funcionalidad del mismo, o a realizar el pintado correcto del mismo mapa.

Capítulo 4

Visión general del sistema

Capítulo 4. Visión General del Sistema

Para resolver cada una de las necesidades que se presentan en este sistema, tenemos que realizar un planteamiento en donde se cuente con la opción de poder expandir la funcionalidad del mismo sistema, aparte de la básica planteada.

Lo que se busca con este planteamiento del sistema son dos cuestiones: cubrir las necesidades básicas del sistema y dejar abierta la posibilidad de expansión de funcionalidad dentro del mismo.

Con todo esto, podemos empezar a introducirnos en el diseño de dicho sistema, para, luego de esto, detallar cada una de sus partes.

4.1 Diseño del sistema

El sistema a desarrollar, visto desde un diagrama gráfico, tendrá la faceta que se muestra en la figura 1.

Vemos que el sistema se divide en 4 partes:

- **Imágenes:** se tienen los diferentes tipos de archivos de imagen para el manejo de mapas.
- **Filtrado de imágenes:** Se filtra los diferentes tipos de archivos de imágenes.
- **Obtención de archivos utilizados por la aplicación GMAP:** Se tienen los diferentes archivos que utilizará nuestra aplicación, los cuales se obtuvieron del filtrado hecho en la segunda etapa del sistema.
- **Utilización de archivos en aplicaciones para Palm y computadora de escritorio:** se utilizarán dichos archivos en las diferentes aplicaciones, ya sea en la Palm o en la computadora de escritorio.

En la figura 1 se presenta el diagrama que se encarga de describir, de manera gráfica, el bosquejo de la aplicación, y un poco el flujo de los datos³³ que intervienen en ella a lo largo de su funcionalidad.

³³ Al decir datos nos referimos a los mapas que se usarán, la ubicación de los nodos y las aristas, así como los puntos de interés dentro del mapa. Los cuales, generalmente, se manejarán en archivos.

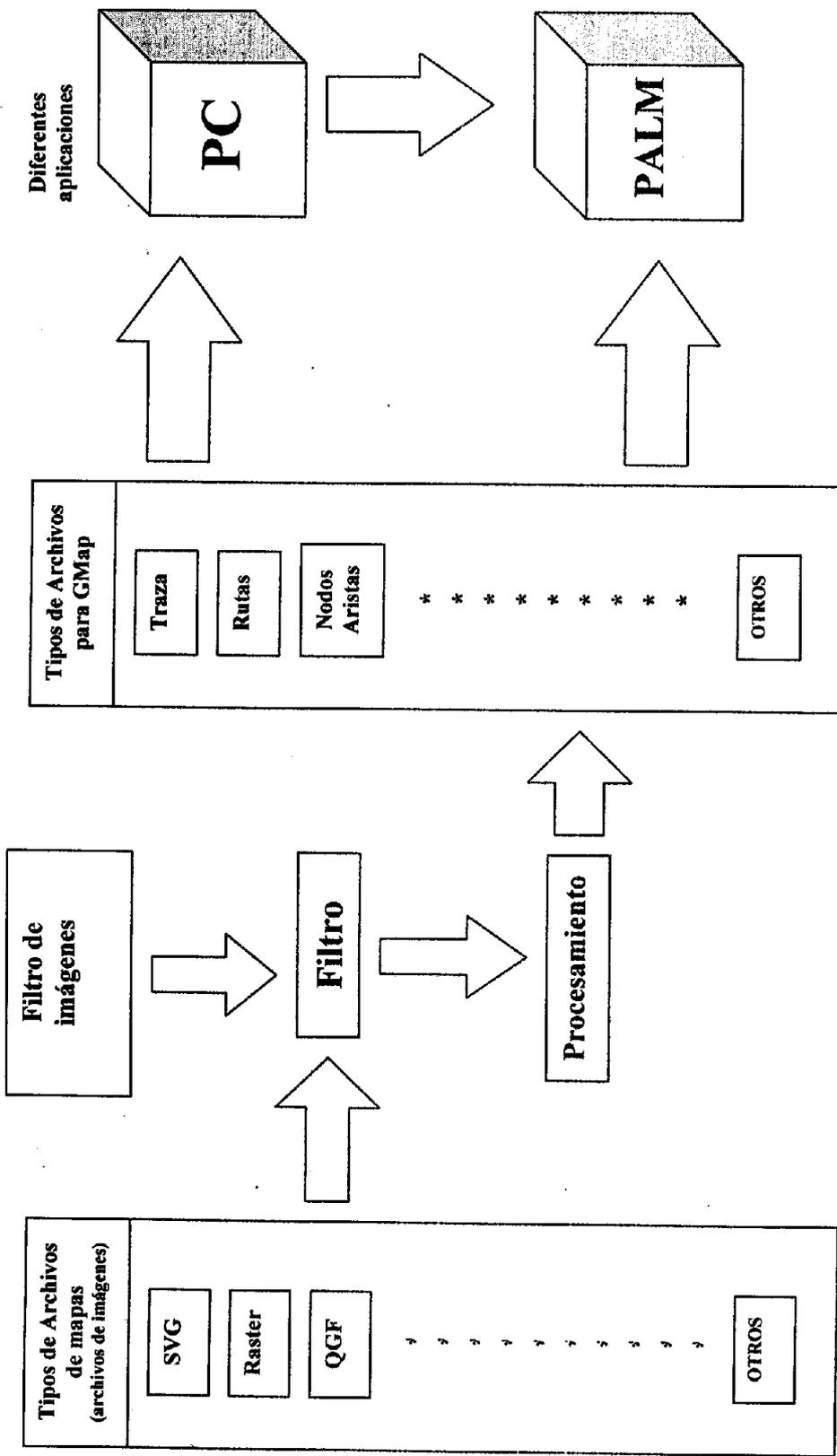


Figura 1. Diseño general del sistema GMAP

4.2 Detalle de cada elemento

Dentro del proceso que conlleva la transformación de los diferentes tipos de archivos contenedores de mapas, podemos observar, que consta de varias etapas.

En la primera etapa se tienen los diferentes archivos que representan a los mapas³⁴.

En la segunda etapa, los archivos de imágenes pasan a un filtro, el cual desglosará cada mapa en diferentes tipos de archivos, cada uno con una finalidad particular.

Aplicando el filtro, se obtienen los archivos de la tercera etapa. Dichos archivos fueron el resultado del desglose del mapa introducido al filtro.

En la cuarta etapa, los archivos de la tercera etapa son transportados a las diferentes aplicaciones a desarrollar para que estas puedan hacer uso de aquellos.

Detallando un poco cada una de las etapas podemos decir lo siguiente:

4.2.1 Tipos de Archivo de Mapas.

En el mundo de la computación existen muchos tipos de archivos para el manejo de imágenes, algunos de los ejemplos más socorridos son el formato GIF, el formato JPEG, el formato BMP y el formato JPEG, los cuales son archivos de tipo raster.

Cada uno de estos tipos de archivos para el manejo de imágenes tiene sus características y algoritmos de creación y lectura; algunos de estos algoritmos se desarrollaron para guardar imágenes en un formato más comprimido; otras tienen la funcionalidad de guardar las imágenes en formato vectorial. En fin, cada uno se desarrolló con un propósito en particular.

Toda la gama de tipos de archivos de imágenes que existen se pueden clasificar en 2 grandes categorías, las cuales se mencionaron en el capítulo anterior.

Son dos las categorías en las que las imágenes pueden dividirse. Cada una tiene ventajas y desventajas, pero ninguna de las dos reúnen, en su totalidad, todas las características que necesitaría una aplicación como la que se quiere construir en este trabajo.

³⁴ Debemos tomar en cuenta que los tipos de archivos que representan a los mapas pueden variar. En el caso de este trabajo de tesis se utilizaron archivos de tipo DXF y DWG.

4.2.2 Ventajas y desventajas de RASTER y Vectoriales.

En cuanto a las ventajas de las imágenes de tipo RASTER para nuestra aplicación podemos mencionar las siguientes:

- Resulta muy práctico, y conveniente, a la hora de dibujarlas, ya que únicamente se tienen que dibujar los píxeles de la imagen. Es muy conveniente porque para el usuario es una manera mucho más fácil de ver que los vectores.

En cuanto a las desventajas de las imágenes de tipo RASTER para nuestra aplicación podemos mencionar las siguientes:

- Resulta muy difícil obtener los datos necesarios para que la aplicación realice las operaciones necesarias. Es decir, lo único con lo que contamos, al manejar este tipo de imágenes, es con el conjunto de píxeles que contiene. No podemos obtener, en caso de que sea una imagen de un mapa de una ciudad, el tamaño de cada una de las calles, la localización geográfica de estas. Esto nos dificultaría, y en ocasiones sería imposible, realizar operaciones como el cálculo de rutas mínimas dentro de los mapas o el cálculo de distancias de los mismos mapas, entre otras.
- Resulta difícil realizar operaciones como zoom, traslación, rotación, etc.
- El tamaño que ocupan generalmente puede ser alto. Esto depende de la resolución y de la calidad de la misma imagen.

En relación con las desventajas que presentan las imágenes de tipo vectorial para nuestra aplicación, podemos mencionar las siguientes:

- No se pueden dibujar directamente ya que representan vectores, y generalmente el detalle de los mismos está escrito en texto.
- De acuerdo al detalle de este tipo de archivos, pueden omitir datos importantes como la ubicación exacta de cada uno de los vectores. Es decir, pueden omitirnos las coordenadas en las que se encuentra cada vector dentro del globo terráqueo. Esto imposibilitaría la mayoría de las operaciones dentro de nuestra aplicación.

En relación con las ventajas que presentan las imágenes de tipo vectorial para nuestra aplicación, podemos mencionar las siguientes:

- Tamaño pequeño en comparación con las imágenes de tipo RASTER.
- Más facilidad de lectura, referente a los datos que representan.
- Contiene datos muy útiles para la realización de diversas operaciones dentro de la aplicación deseada.

De esta forma, podemos ver, que ninguno de los dos grandes tipos de representación de imágenes cumple, en su totalidad, con lo deseado para que la aplicación trabaje por completo.

Por esta razón es necesario realizar una combinación de ambas para poder lograr la funcionalidad total de nuestra aplicación.

De acuerdo a todo lo mencionado en este apartado, podemos ver que en la práctica nos vamos a topar con mapas que estén contenidos en diferentes tipos de imágenes, donde en ningún caso puede la imagen darle a la aplicación lo que esta necesita en su totalidad. Esto nos obliga a pasar la imagen a través de un filtro, que la convierta y le deje lista para trabajar con ella.

A continuación se detalla dicho filtro.

4.2.3 Filtro convertidor.

Previo a la utilización de las imágenes de tipo raster o vectoriales, donde están contenidos los mapas, por parte de la aplicación se tienen que someter a un tratamiento para obtener los archivos que nos podrán ayudar a resolver cada una de las funcionalidades de nuestra aplicación. Esto dependerá del tipo de imagen, que constará de varios pasos, y dichos pasos variaran de acuerdo al formato de la imagen que se tenga.

El inicio de dicho proceso se encuentra en el módulo que hemos llamado filtro. Después a este vendrá una etapa donde se obtendrán los archivos que la aplicación necesitará explícitamente para su funcionamiento.

El camino para explicar de mejor manera las tareas del filtro es primero saber los tipos de archivos que la aplicación necesita, para saber de que manera tiene que procesar los archivos que recibe. De esta forma, la manera en que el filtro trabaja se detallará al final de este capítulo, junto con la manera global en que el diseño de este sistema se plantea.

4.2.4 Convertidor.

Una vez que el filtro haya hecho su función, el resultado, que será una serie de archivos que contengan la imagen en un formato vectorial, tendrá un formato más apegado a los que la aplicación necesita. Pero este formato no será suficiente para que nuestra aplicación trabaje.

Debido a esto, la imagen tiene que procesarse aún más, para que nuestra aplicación pueda realizar lo planeado.

La imagen, hasta este momento se encuentra en formato svg, es decir, en un formato vectorial representado en xml. Lo que la aplicación necesita es los diferentes archivos que nos detallarán cada una de las partes de un mapa, los cuales se detallarán en el siguiente apartado de este capítulo.

Este módulo se encargará de tomar la imagen, analizarla y desglosarla de tal manera que pueda generar los archivos que nuestra aplicación necesita.

De forma básica, y a manera de ejemplo, a reserva que más adelante se detalle mejor, una de los archivos que se necesita es el que nos detalle la posición de cada uno de los nodos, una información que el archivo en svg que tenemos, por sí mismo, no nos dice. Es por esto que este módulo interviene en dicho archivo para obtener los nodos que se necesitarán para el cálculo de las rutas mínimas, el cálculo de las distancias, entre otras muchas tareas.

Mediante este módulo, el archivo en formato vectorial de tipo SVG podrá analizarse, agregarle los datos necesarios y convertirse en los archivos necesarios para la aplicación.

Otro detalle que se hará en este módulo es el posicionamiento del mapa. Es decir, un mapa, cuando se encuentra en un archivo de este tipo, o de tipo RASTER, tiene lo necesario para ser dibujado en alguna parte. Pero hay algo que el mapa no contiene, las coordenadas exactas en el mundo en el que se encuentra. Por ejemplo, un mapa de la ciudad de México lo identificamos cuando lo vemos, pero eso no nos dice algún dato de la posición geográfica³⁵ en las que se encuentran. En el caso del mapa de una colonia, si la vemos podremos, quizá, reconocer cual colonia es pero no sabremos las coordenadas geográficas en el mundo en la que se encuentra, y, con mucho menos razón, sabremos las coordenadas geográficas relativas que tiene dentro de la ciudad de México, el cual es el mapa que la contiene. Y, dado que el sistema va a interactuar son el SPG para darle al usuario final la ubicación exacta en el mundo en la que se encuentra, debe el sistema saber las coordenadas exactas de cada una de los elementos del mapa.

Por la razón expuesta en el párrafo anterior, se debe tener alguna de las dos siguientes opciones:

- La coordenada geográfica de un par de puntos dentro del mapa. Esto con la finalidad de poder obtener las dimensiones reales de dicho mapa. Esto significa que el mapa tiene que representar las medidas exactas que tiene en la realidad el lugar geográfico que representa. Esto se puede obtener a partir de la coordenada de dos puntos en el mapa. Con dichas coordenadas se puede obtener la distancia exacta que hay entre ellas y, a partir de eso, calcular las distancias en todo el mapa.

³⁵ Coordenadas de latitud, longitud y altitud en el mundo. Generalmente se miden en grados y nos ayudan a ubicar un punto dentro del globo terráqueo.

- La coordenada de un punto dentro del mapa y las distancias exactas. Otra manera de tener representados las dimensiones exactas dentro del mapa, es teniendo la coordenada geográfica de un punto dentro de este mapa y que el mismo mapa nos indique las dimensiones exactas y reales que tiene cada uno de los elementos del mapa. Con esto se puede calcular, mediante un algoritmo sencillo las coordenadas geográficas de todos los componentes del mapa.

De esta manera podemos contar, dentro del mapa, con las coordenadas exactas de cada uno de los componentes del mapa, algo que nos ayudará a calcular otros aspectos como las coordenadas geográficas de los nodos existentes dentro del mapa, las aristas y distancias de las mismas, esto en caso de que a un mapa se le vea como a un grafo.

4.2.5 Tipos de archivos de GMAP³⁶.

Dentro del funcionamiento total de la aplicación GMap, podemos observar, en el diagrama que se presentó al principio de este apartado, que necesita varios archivos. Esto con la finalidad de que maneje cada una de los datos necesarios para su funcionamiento de manera independiente. De esta forma resultará más fácil el cálculo de las operaciones realizadas al llevar a cabo cada una de las operaciones dentro de la funcionalidad de la aplicación.

Colocando todo esto en la perspectiva de la aplicación, podemos tener las siguientes necesidades:

- Dibujado de mapa. Dentro de la aplicación se tendrá la necesidad imperante de dibujar el mapa del lugar por donde vamos pasando. Esto puede ser de dos maneras diferentes.

La primera consiste en representar, dentro del dibujo, a cada calle con una línea delgada, en donde dentro del mapa sólo podamos ver líneas que nos representen las calles por las cuales podemos transitar. Las aplicaciones que hay de este estilo nos presentan el mapa de esta forma, con lo cual no nos detalla lo que existe dentro de cada cuadra en los mapas. Esto se debe que los usuarios que hacen uso de estas aplicaciones se conforman con ver esto, pero para un usuario que prefiera ver un detalle más profundo de cada mapa puede resultar poco agradable el ver solamente las calles delineadas, sin ver la manera en que se forman cada una de las cuadras. Una imagen de un ejemplo que nos ayuda a ver un poco más claro esta manera de pintar los mapas es la siguiente:

³⁶ GMAP es el nombre con el que se ha bautizado a esta aplicación.

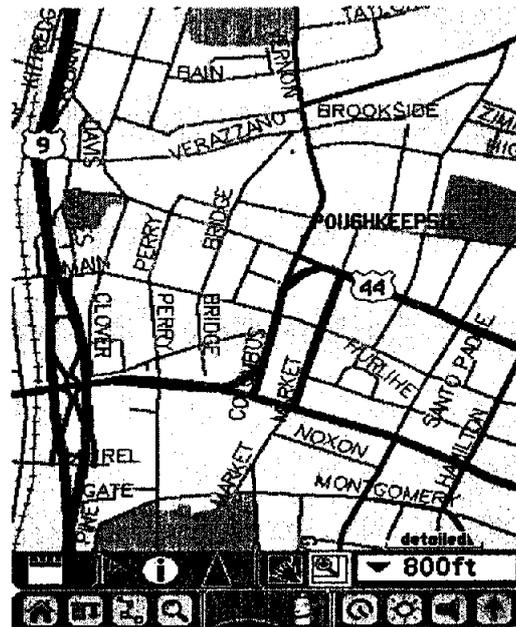


Figura 2. Mapa cuyas calles son representadas por una línea.

En esta manera de presentar el mapa, como se puede observar, resulta un poco limitada ya que no podemos ver el detalle de cada una de las cuadras, pero es suficiente para una aplicación donde simplemente trace rutas dentro del mapa, ya que no se necesita más que el delineado de cada una de las calles por donde puede pasar la ruta trazada.

Por lo anterior explicado, si se desea representar los mapas de esta manera, se necesitarán archivos con el suficiente detalle para identificar los nodos³⁷ y las aristas³⁸ del grafo³⁹ que representaría la ciudad en cuestión.

La segunda manera consiste en presentar cada una de las calles en su justa dimensión, es decir, con su anchura y largo real, para que el usuario pueda realizar la lectura⁴⁰ de las mismas de manera más intuitiva y agradable. Con esta manera de realizar el pintado del mapa, el sistema se volvería más fácil para el usuario, con esto el mismo usuario podría localizar de manera más rápida los lugares cercanos a donde se encuentra localizado debido a que el mismo mapa se los va presentado uno a uno.

³⁷ En este caso cada nodo representaría a una esquina existente en la ciudad ya que es el punto en donde se unen una o más aristas. Por lo tanto, habría tantos nodos como esquinas o cruceros en la ciudad.

³⁸ En nuestro caso cada arista representaría un tramo de cada calle dentro de la ciudad, ya que son las que unen a dos nodos. Por lo tanto habrá tantas aristas como tramos de calles que unen a dos nodos.

³⁹ Toda la ciudad se puede representar en un grafo, ya que se puede ver como un conjunto de nodos que se entrelazan por medio de las aristas.

⁴⁰ El que el usuario pueda ver el mapa e interpretarlo de acuerdo a lo que el mismo usuario entiende en lo que está viendo.

Otra de las ventajas de esta segunda forma de presentar los mapas es que en el se puede dibujar figuras más completas⁴¹ para indicar la manera en que se tiene que dirigir al lugar deseado después del cálculo de dicha ruta.

Por todo lo anterior, si se desea presentar el mapa en el segundo formato se necesitaría un mapa donde se representara cada una de las cuadras, calles, avenidas y demás elementos que una ciudad puede tener, representados en el mapa que se va a generar.

- Cálculo de las rutas mínimas. Para el cálculo de las rutas mínimas, como ya se explicó en el capítulo correspondiente, se puede utilizar el algoritmo de Dijkstra, el cual lo debemos aplicar en un grafo. Para esto es necesario identificar los nodos y las aristas, así como las distancias⁴² que tienen dichas aristas. Y estos datos no pueden ser proporcionados directamente por un mapa con las características del archivo en formato SVG, en caso de que se use este tipo de archivos.

En este caso, para poder tener los datos de los nodos que existen dentro del mapa, así como las aristas, y su tamaño, que unen a estos nodos, tenemos que tomar el mapa y calcular la posición de dichos nodos y aristas. De otra manera es muy tardado, en cuanto a procesamiento, calcular la ruta mínima, ya que el cálculo de los nodos se tendría que realizar cada que uno quiera calcular la ruta mínima. Y recordemos que cuando hablamos del procesamiento dentro de una computadora de escritorio, que generalmente tiene capacidades suficientes para realizar esta tarea de manera considerablemente rápida, podemos hablar de un tiempo de respuesta relativamente rápido; pero los dispositivos a los que se van a enfocar esta aplicación es a los móviles, los cuales no tienen la capacidad que tiene una computadora de escritorio, sino una capacidad mucho menor. Esto nos conduce a optimizar los procesos al máximo.

Por todo esto se necesitaría algún archivo que nos indicara la cantidad de nodos⁴³, identificándonos el lugar exacto en el que se encuentra, así como las aristas⁴⁴ del mismo grafo⁴⁵ para poder llevar a cabo las tareas de la aplicación de manera más optimizada.

- Localización de puntos de interés. En el caso de que se quiera localizar puntos de interés dentro de la ciudad, por ejemplo museos, lugares para comer, parques, entre otros muchos, se tendrá que tener una base de datos que nos indique el lugar en el mapa donde se encuentra cada una de estos lugares de interés, ya que, por sí

⁴¹ Entiéndase figuras como flechas con mayor grosor, o los nombres de las calles, entre otras dentro del mapa que se está dibujando.

⁴² En el lenguaje de grafos, representarían el peso de cada una de las aristas.

⁴³ En este caso, cada nodo representaría cada esquina o crucero en la ciudad.

⁴⁴ En este caso, cada arista representaría a cada una de las calles de la ciudad.

⁴⁵ El grafo general sería una representación abstracta del mapa entero del que estamos hablando.

mismo, el archivo que represente el mapa no podría tener dicha información porque solamente representaría el mapa.

De esta manera, al igual que citamos las tareas básicas de este sistema, podemos citar cada una de las tareas que puede tener, en un futuro con las siguientes versiones, esta aplicación. Con esto nos podemos dar cuenta de que para cada funcionalidad que se le de a este sistema, se necesitarán datos específicos para llevar a cabo la solución de dicha funcionalidad. Por lo que en todo el sistema se necesitarán datos de diversos tipos, y no solamente el dibujo del mapa.

A partir de esto, podemos ver que la problemática de resolver cada una de las funciones que el sistema requiere se puede plantear, con todo lo anterior dicho, de dos maneras, a saber:

- Un archivo por área geográfica⁴⁶. Tener un archivo que represente a un área geográfica, el cual nos diga la manera en que se encuentra el mapa, para pintarlo, el lugar donde se encuentran los nodos y las aristas, para poder calcular las rutas mínimas y las operaciones que tengan relación con estos dos elementos, los puntos de interés y demás datos que surjan a partir de las necesidades que se vayan presentando de cada una de las funciones que se agreguen al sistema.

Como se puede observar, esta manera de resolver la situación nos implicaría el manejo de un archivo, en el formato que nosotros elijamos, muy extenso en tamaño, con lo cual puede resultar incómodo, inclusive hasta inconveniente, en su manejo, ya que estamos hablando de dispositivos móviles. Y cada archivo de este tipo se repetiría con cada una de las diferentes zonas geográficas con que se quiera trabajar.

- Varios archivos con los datos necesarios por cada zona geográfica. Se puede tener varios archivos por cada zona geográfica con la que se quiera trabajar. Por ejemplo, un archivo que contenga la manera de dibujar el mapa, otro que contenga la ubicación de cada uno de los nodos, otros, se puede pensar, con la descripción de cada una de las aristas, y otro que contenga la base de datos de los puntos de interés, etc. Y otros archivos para representar datos específicos para cada una de las funciones que se agreguen a este sistema.

Ahora bien, este conjunto de archivos se repetiría, por lo menos en su estructura, en cada una de las zonas geográficas que queramos manejar, ya que la funcionalidad total del sistema se tiene que aplicar a todas las zonas geográficas que se manejen. Esto nos puede traer implicaciones de manejo de muchos archivos dentro del sistema, pero esto tiene la ventaja de que resultan archivos más pequeños en tamaño y más manejables desde la perspectiva de la aplicación.

⁴⁶ Más adelante se explicará, cuando se explique la manera en que trabajará el sistema, que el mapa estará dividido en zonas geográficas, algo que facilita el trabajo de todo el mapa en su conjunto. Así mismo se explicará el porque se tomó la decisión de trabajarlo de esta manera. La división del mapa en dichas áreas geográficas se realiza en la etapa del filtro y procesamiento.

Con estas dos maneras planteadas se puede atacar el problema de los archivos que necesitará el sistema.

4.3 Ventajas y desventajas en el trato de los datos

Transportando todo lo explicado en este apartado al ámbito de la aplicación podemos decir que lo más conveniente y recomendable es utilizar la segunda manera de manejar los datos. Las razones son varias, y provienen, en gran medida, de las ventajas y desventajas que presentan cada una de las opciones que se plantearon.

En cuanto a las ventajas que presenta la primera opción se encuentran las siguientes:

- En un solo archivo se manejaría todo. El que en un solo archivo se maneje todo nos traer como ventaja que no se manejaría una gran cantidad de archivos, solamente tantos archivos como zonas geográficas se manejen, por lo que a la aplicación, al manejar una sola zona geográfica a la vez, a la hora de cargar los datos podría hacerlo de manera más rápida ya que solamente leería un archivo, el de la zona geográfica que esté manejando.

En cuanto a las desventajas que presenta la primera opción se encuentran las siguientes:

- Archivos muy grandes en tamaño. Dado que en un solo archivo se colocarían todos los datos necesarios, estos archivos, aún cuando estemos hablando de que se trabaja por zonas geográficas, serían extensos en tamaño. Esto acarrearía como consecuencias el que sea difícil de transportarlos, ya sea en tarjetas de memoria o a través de internet. Esto implicaría un problema algo serio, ya que dichos datos son indispensables para el funcionamiento total de la aplicación. Otra consecuencia de esto es la tardada lectura de parte de la aplicación.
- Mucha memoria sin darle utilidad. Cuando se lea dicho archivo, lo más conveniente es que se tenga que guardar en memoria todos los datos necesarios para llevar a cabo la funcionalidad completa contenidos en el archivo en la primera ocasión en que se lea, para evitar leer en varias ocasiones. Los dispositivos que van a estar utilizándolos son dispositivos limitados en memoria, por lo cual se puede consumir de manera muy rápida. En el caso de que la aplicación esté usando todas sus funciones, esto sería lo ideal ya que los datos que necesitaría estarían en memoria, pero si solamente se está usando una parte de la funcionalidad, dado que todos los datos estarían en memoria, se estaría desperdiciando la memoria dedicada a almacenar los datos de las funciones que no está utilizando la aplicación. En otras palabras, todos los datos de un archivo estarían levantados a memoria aunque no se estuviera usando dichos datos, algo

que sería contraproducente de acuerdo a las capacidades de hardware con que cuentan los dispositivos móviles.

- Escalabilidad complicada. Dado que en un solo archivo se describe toda la zona geográfica, cuando uno como desarrollador quiere agregarle una nueva función a la aplicación, tiene que introducir los datos de dicha función en el archivo que existía, por lo que el tamaño se incrementaría. Otra dificultad que presenta en este rubro es que los usuarios, al querer actualizar el sistema con esa nueva funcionalidad tendrá que obtener de nuevo los datos que ya tenía más los nuevos datos, ya que dichos datos nuevos están contenidos en el archivo de la nueva versión y no se pueden separar de los datos de las anteriores versiones. Esto implica que se repita el proceso de obtención de los datos de las primeras versiones tanto como nuevas versiones vayan surgiendo.

Como podemos observar, la única ventaja que nos da la primera forma de plantear la solución es relativa, ya que un par de las desventajas que se presentan (el tamaño del archivo que se formaría y la escalabilidad complicada) viene a contradecir dicha ventaja. Todas estas desventajas que se presentan con la primera forma de solución, se enmiendan con la segunda manera de tratar a los datos que la aplicación va a necesitar. Esto se podrá observar a continuación que se explican las ventajas de esta segunda solución planteada.

En cuanto a las ventajas que presenta la segunda opción se encuentran las siguientes:

- Archivos más organizados. Dado que cada una de las zonas geográficas con que cuenta el sistema tendrá sus propios archivos descriptivos, cuando la aplicación tenga que cambiar de zona geográfica de trabajo, lo único que tendrá que tener en cuenta es cargar los datos contenidos para dicha zona geográfica, y dado que estarían todos los archivos ordenados, el sistema no batallaría en encontrar los que vaya a necesitar.
- Sólo carga datos que vaya a utilizar. Dado que los datos de cada funcionalidad estarán colocados en un solo archivo, dentro de los archivos pertenecientes a cada zona geográfica, será necesario cargar aquellos archivos de dicha zona geográfica que vaya a utilizar de acuerdo a la funcionalidad que el usuario del sistema le indique. Por ejemplo, si el usuario solamente pide pintar el mapa, será necesario cargar solamente el archivo que defina la manera en que se dibuja el mapa de esa zona geográfica, mientras que los demás archivos descriptores de la zona geográfica se mantendrán sin ser leídos. Hasta que sea necesario leer los otros archivos se leerán, mientras tanto el sistema no los lee. De esta manera se pueden manejar muchos datos pero solamente se cargarán los que estrictamente se estén utilizando.
- Escalabilidad sencilla. Cuando el desarrollador quiera agregarle una funcionalidad extra, dado que dicha funcionalidad necesitará datos específicos,

bastará, por estar cada zona geográfica organizada en varios archivos, agregar un archivo que contenga los datos necesarios para realizar las operaciones que impliquen dicha funcionalidad por cada zona geográfica. Al igual que el usuario final, sólo bastará que adquiera el archivo extra para la nueva funcionalidad, ya que el módulo correspondiente se encargará de usarlo cuando sea necesario, mas no tendrá que adquirir de nuevo todo los datos de las versiones, o funcionalidades, con que ya contaba.

- Archivos más pequeños en tamaño. Ya que se tendrán cierto número de archivos para cada zona geográfica, el tamaño de estos será más pequeño. Al final de la construcción, la suma del tamaño de cada uno de los archivos generados por cada zona geográfica quizá sea muy similar al tamaño de un archivo con todos los datos (como se plantea en la primera solución) de la misma zona geográfica, inclusive cabe la posibilidad de que aquella sea mayor que esta última, pero la ventaja de hacerlo de esta manera se observará a la hora de que la aplicación ejecute cada funcionalidad.
- Modularidad. Con esta manera de manejar los datos de la aplicación, nos permitirá manejarla a través de módulos, en donde cada módulo estará representado por una funcionalidad, el cual consistirá la parte del programa que llevará a cabo dicha funcionalidad y los datos que se necesitarán para dar cause a la misma. Con esto resultar más fácil hacer escalable el sistema.

A partir de las ventajas y desventajas de cada una de las maneras de manejar los datos, podemos concluir que es mejor, de acuerdo a la funcionalidad que se le quiere dar al sistema, la cual se detallará más adelante, que la segunda solución propuesta para el tratamiento de datos es la más adecuada que podemos utilizar.

Es por esto, que el módulo del filtro y procesamiento que aparecen en el diagrama planteado en un principio en este capítulo son para convertir los diferentes tipos de archivos que contengan un mapa a este tipo de archivos. De allí radica la importancia de este par de módulos.

Mediante estos dos módulos, ya explicados anteriormente, podemos llegar a la obtención de todos los archivos necesarios para cada una de las funcionalidades del sistema. Una vez obtenidos dichos archivos, lo único que faltaría es usarlos en la aplicación para los diferentes dispositivos móviles o computadoras de escritorio para las cuales se va a realizar su respectiva versión de este sistema.

4.4 Proceso de los mapas

En resumen, el procesamiento de los mapas que serán utilizados por cada funcionalidad en este sistema se divide en varias etapas:

- Mapas en bruto. En esta etapa los mapas están en su formato inicial, que consiste en los diferentes tipos de archivos para manejar imágenes, donde los mapas solamente tienen definidos los trazos con que están formados, es por esto que no cuentan con ningún otro dato que a la aplicación pueda servir más que para pintarlo, ya sea en los dispositivos móviles o en la computadora de escritorio.
- Procesamiento del mapa. En esta etapa el mapa es convertido del archivo que simplemente contiene los trazos necesarios para dibujarlo a un archivo en formato xml para poder almacenar los datos necesarios para cada funcionalidad que contenga el sistema. Esta etapa le permitirá a la aplicación obtener los archivos necesarios que contengan los datos suficientes para que la aplicación pueda trabajar de manera adecuada. El mapa será dividido en zonas geográficas, las cuales se definirán en otro documento, que nos permitan tener modularizada la funcionalidad de esta aplicación. Y para cada zona geográfica identificada en dicho mapa, se generarán los archivos necesarios de acuerdo a la funcionalidad que se le quiera dar.
- Uso de los archivos de datos. Una vez que se pasa por la etapa anterior, el sistema ya cuenta con los archivos que contienen los datos para que cada aplicación, ya sea computadora de escritorio o dispositivos móviles, puedan utilizarlos. Lo único que restaría es enviarlas a dicha computadora de escritorio o al dispositivo móvil que las vaya a utilizar.

4.5 Conclusiones

De lo tratado en este capítulo podemos ver un bosquejo, que en los capítulos siguientes iremos desglosando poco a poco, del funcionamiento total del sistema.

Vimos que son diversas las maneras de tratar los datos en nuestra aplicación, ya sea en formatos SVG (XML) o en formato RASTER.

Como pudimos observar, el formato SVG es muy conveniente para este tipo de aplicaciones debido a que es capaz de transportar los datos necesarios para que la aplicación trabaje adecuadamente.

Pero debido a que dicho formato está basado en XML, la nomenclatura utilizada por las etiquetas características de este formato puede hacer que el archivo resulte ser lo suficientemente grande en tamaño como para provocar un procesamiento más tardado dentro de la Palm. Por esta razón se ha optado por manejar los datos en formato vectorial pero en una manera más reducida. Lo más óptimo en el caso de estos dispositivos es el manejo de los datos en forma binaria, buscando reducir al máximo el tamaño de cada uno de los archivos que se manejen.

Lo que dichos archivos contendrán son las coordenadas necesarias para el dibujo del mapa dentro del sistema GMAP.

Capítulo 5

Diseño del sistema

Capítulo 5. Diseño del Sistema

Dentro del dispositivo móvil Palm, al desplegar todas las características de los mapas dentro de la aplicación GMap se tiene que hacer de tal manera que se aproveche al máximo las capacidades del dispositivo, ya que este último, como ya vimos, es limitado en comparación de una computadora de escritorio o una portátil.

Por todo esto tenemos que plantear una manera de trabajar cada una de las partes que intervienen en el sistema.

5.1 Partes que intervienen

Definitivamente este sistema, al buscar que sea altamente escalable, como se mencionó en capítulos anteriores, puede llegar a tener múltiples partes. Pero para este proyecto solamente necesitaremos las siguientes:

- Mapa de la traza de la ciudad. Esta parte comprende únicamente los datos que nos indican cada una de las cuadras⁴⁷ de la ciudad en cuestión. No contiene otro dato más que dicha traza. Claro está que cada una de las cuadras debe venir acompañada de las partes que la conforman, de los cuales algunas de estas son las siguientes: el conjunto de vértices con sus respectivas coordenadas de localización; identificador de la cuadra, entre otras.
- Grafo de la ciudad. Grafo que se forma a partir de cada una de las calles, representadas por medio de las aristas del grafo, y las intersecciones entre estas, identificadas por medio de los nodos pertenecientes a dicho grafo.
- Nombre de las calles de la ciudad. Conjunto de nombres de cada una de las calles de la ciudad. Con esto el usuario del sistema podrá identificar la calle por la cual está transitando.
- Punto de localización. Es el punto de localización del usuario dentro del mapa. Esto se representará dentro de la Palm mediante un punto redondo indicando el lugar donde se encuentra el usuario dentro del mapa dibujado en nuestra Palm.
- Puntos de interés. Es el conjunto de lugares más conocidos como museos, parques, escuelas, iglesias, mercados, etcétera, que al usuario le podrían interesar. Por supuesto que si el usuario desea agregar un nuevo punto de interés el sistema deberá tener la capacidad de hacerlo y actualizarse.

⁴⁷ Cada una de las manzanas de la ciudad.

Como podemos observar solamente 5 partes son las necesarias para que nuestro sistema haga todo lo que queremos para el usuario.

Tomando en cuenta estas partes, ahora pasemos a ver la manera en que se acomodarán dentro de la aplicación.

5.2 Diseño de capas

Para que nuestra aplicación sea altamente escalable tiene que tener una manera de colocar todas las partes que intervienen en el sistema en una forma tal que se puedan estar manipulando con cierta libertad. Esto implica que cada una de las partes mencionadas arriba tiene que funcionar independientemente de las otras. Así tiene que pasar con las partes que en un futuro se agreguen a este sistema.

Esta independencia entre las partes también se debe a que cada una de las partes tendrá sus propias operaciones. Por ejemplo, en el caso de los puntos de interés el usuario podrá agregar o quitar los puntos de interés que están marcados dentro del sistema, pero este cambio no debe afectar a las otras partes porque estaríamos cayendo en una dependencia entre las partes.

Es cierto que debemos abogar por que la dependencia de las partes que intervienen en el sistema se de, pero en algunos casos las operaciones tendrán que implicar que se modifique dos o más de las partes que intervienen en el sistema, por ejemplo cuando el usuario nota que la geografía de una parte de la ciudad ha cambiado por la remodelación de alguna calle o por alguna otra razón. En este caso se tendrá que cambiar la traza de la ciudad, el grafo formado por la misma y quizá el nombre de la calle. Vemos que se modifican, con la operación mencionada, varias de las partes del sistema, pero aún así, no tenemos porque apartarnos de la independencia de las partes, ya que esta operación se divide en tres sub-operaciones realizadas sobre cada una de las partes en cuestión y de esta forma se conserva la independencia de cada una de las partes.

De esta manera, con la independencia de cada una de las partes podemos obtener muchos beneficios, algunos de los cuales se mencionan a continuación:

- Operaciones atómicas. Cada una de las partes mencionadas tiene sus operaciones propias, y en caso de una operación que afecte a más de una parte, la modificación se hace a cada una como una operación atómica dentro de otra operación general. De esta manera se podrá hacer modificaciones sobre cada parte de manera independiente. Si no fueran independientes se tendría que modificar a un todo aunque fuera una operación sencilla.

- Visibilidad independiente. Al ser independientes cada una de las partes con respecto a las otras que intervienen, podemos verlas a cada una de estas sin tener que ver a las demás. Esto puede que no se vea como útil, pero imaginemos por un momento que el usuario está caminando por la ciudad y tiene todas las partes que intervienen en el sistema desplegado en su Palm, es decir, la traza de la ciudad, los nombres de las calles, el grafo que se forma, el punto de localización y los puntos de interés, y pensemos que el usuario ya sabe donde está el punto a donde quiere llegar, y no quiere que la pantalla se le sature de información. Con esta características el usuario puede indicarle al sistema que desaparezca el punto de localización, los nombres de las calles y los puntos de interés, de tal manera que solo queda en la pantalla la información de la traza de la ciudad y el camino que tiene que recorrer para llegar al lugar al que desea. Si no fueran independientes no habría forma de ocultar alguna de las partes que intervienen en el sistema.

Estas son sólo algunas de las bondades que nos da esta manera de tratar a cada una de las partes del sistema.

Habiendo visto algunas de las bondades que la independencia de cada una de las partes que intervienen con respecto a las otras que la acompañan nos da podemos pasar a ver la manera en que proponemos para el funcionamiento del sistema en cuestión del manejo de dichas partes dentro del mismo.

El diseño que se plantea aquí es un diseño a base de capas, en donde cada capa representa a cada una de las partes que se mencionan arriba que intervienen en la aplicación. De tal manera que cada una de las operaciones que haya que hacer se realice a cada una de las capas que existen.

Esto, visto de manera gráfica, lo tenemos de la siguiente forma. Imaginémonos a cada una de las capas de la aplicación⁴⁸, de manera independiente la una de la otra.

Cada capa podemos verla en las figuras 1, 2, 3, 4 y 5.

⁴⁸ Entendiendo por capas de aplicación a cada una de las partes que intervienen en la aplicación que va a correr en la Palm

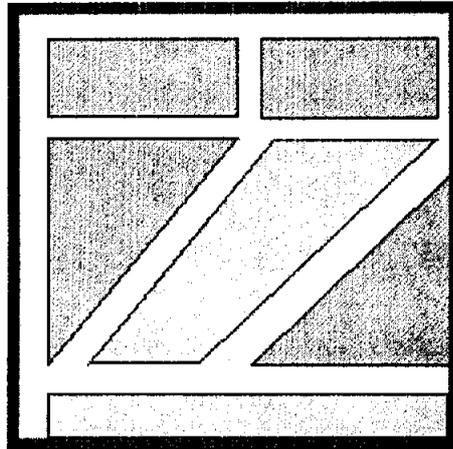


Figura 1. Capa de las cuadras de la ciudad

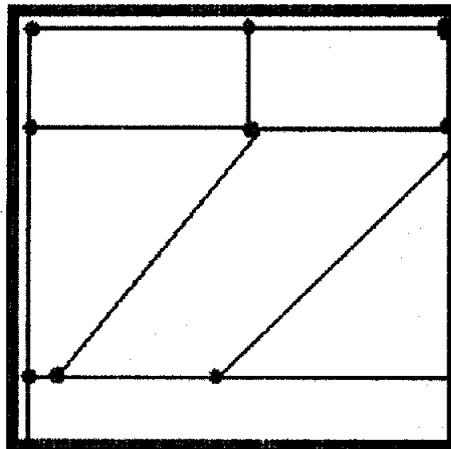


Figura 2. Capa del grafo de la ciudad.

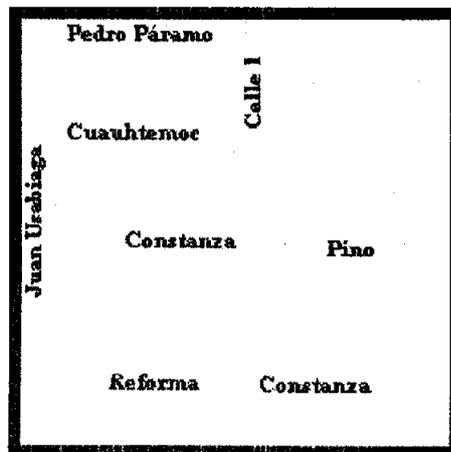


Figura 3. Capa de los nombres de las calles de la ciudad

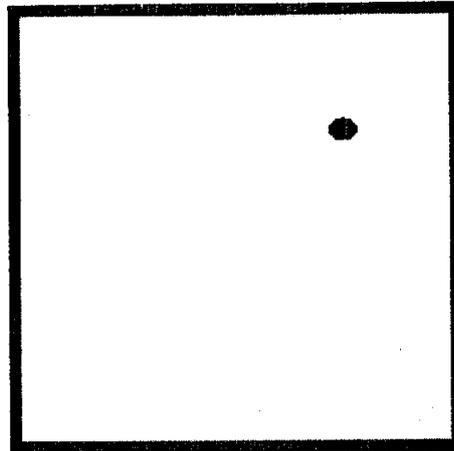


Figura 4. Capa de punto de localización

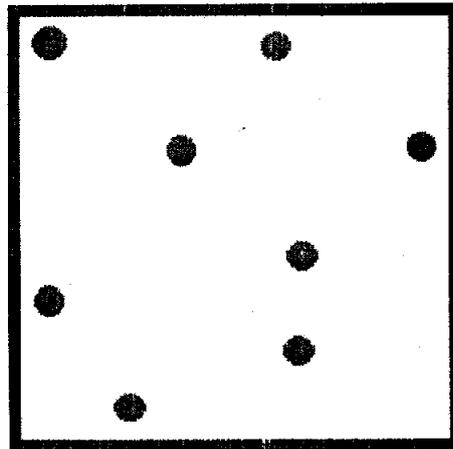


Figura 5. Capa de puntos de interés.

De esta manera se verían las 5 capas de la aplicación, de manera independiente unas de otras.

Nuestra aplicación estará manejando cada una de las capas mediante un arreglo de capas, de tal manera que, a la hora de tener que manipularlas podrá ser mucho más fácil. Esto, visto de manera gráfica, podemos observarlo en la figura 6.

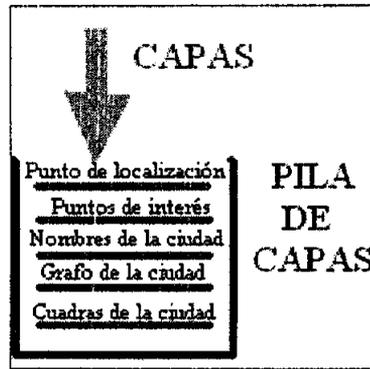


Figura 6. Pila de capas

Cuando se vaya a realizar alguna operación como el dibujado de las capas que hayan sido elegidas por el usuario se tiene que dibujar capa tras capa empezando por la inferior, es decir, por las cuadras de la ciudad, así como se observa en la figura 7.

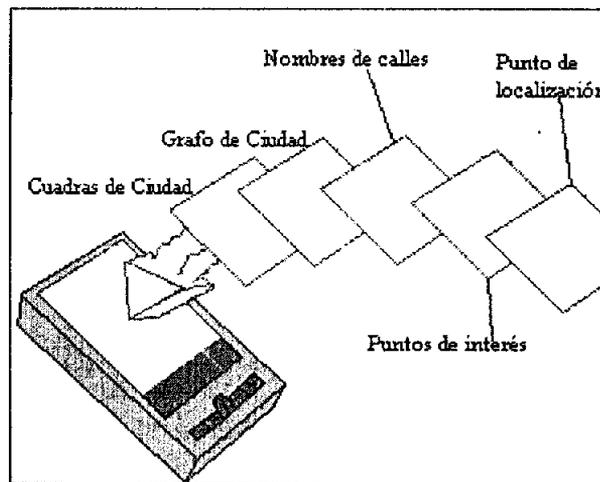


Figura 7. Dibujando todas las capas

Entonces, de acuerdo a la figura 7, la primera capa, la que se trata de las cuadras de la ciudad, en caso de que esté activada será dibujada en primer lugar. Luego de esta, encima del dibujo ya hecho por la primera capa, se dibujará, en caso de que esté activada, la segunda capa, es decir, la que se trata del grafo de la ciudad. Esta segunda capa no borrará lo que se había dibujado de la primera porque solamente se dibujará las partes necesarias, es decir, donde implique dibujar una arista o un nodo dentro de la pantalla, y lo demás se quedará como estaba. Así se aplica con la tercera capa, así como con las demás capas.

Cada una de las capas tendrá una bandera de activación, la cual le indicará al sistema si la tendrá que dibujar o no. Esto es, en caso de que, por ejemplo, la primera capa esté desactivada, y las demás estén activadas, dentro de la pantalla no se dibujarán las

cuadras de la ciudad. Es cierto que, en este ejemplo, la capa que se encontrará desactivada es la más importante, ya que se trata de la capa que dibujará las cuadras de la ciudad; bien es cierto que esta capa debería siempre estar dibujada, y no tendría que tener bandera de activación, sino que de manera predeterminada se dibujara, pero se le coloca la bandera de activación debido a que es más fácil para el sistema trabajarla así. Por otro lado, el que tenga esta bandera de activación le permitirá al usuario, en un momento dado, poder hacer que no se dibuje esta capa, por ejemplo, cuando el usuario solamente quiera visualizar el grafo correspondiente a la ciudad. Por esta razón, todas las capas que pertenezcan a esta aplicación tendrán la bandera de activación.

Dentro de dicho arreglo, la primera capa en entrar es la primera que se dibujará, después la segunda que se haya introducido, luego la tercera y así sucesivamente. En cuanto a la bandera de activación, en caso de que esté desactivada no querrá decir que la capa salga del arreglo que maneja las capas, sino solamente significará que, a pesar de que la capa exista dentro del manejador de capas, esta no será dibujada dentro de la pantalla. Esto se detallará más adelante.

Así mismo, con esta manera de trabajar dentro de la aplicación, el usuario puede, de manera manual, activar o desactivar cada una de las capas para que estas se desplieguen de acuerdo a su gusto.

5.3 Propiedades de las capas

Ahora bien, cada una de las capas, para que pueda la aplicación trabajar de manera completa con cada una de ellas tendrá que tener varias características, entre las cuales se pueden enumerar las siguientes.

- **Bandera de activación.** Esta propiedad nos ayuda a indicarle a la aplicación si la capa tiene que ser dibujada o simplemente queda invisible. En el caso de que sufra un cambio, se generará un evento dentro del sistema que las otras capas, o alguna otra parte del sistema podrá manejar.
- **Capacidad de realizar operaciones sobre ella.** Esto es debido a que el usuario podrá modificar las capas del sistema.
- **Independencia entre ellas.** Esto para que se puedan manipular cada una por separada.
- **Contenedor para dibujarlas.** Estarán dentro de un contenedor que las manejará en el sentido del dibujado de las mismas sobre la pantalla.
- **Índice dentro del contenedor.** Cada una tendrá su índice dentro del contenedor, el cual lo usarán para desplegarlas en el orden en el que estén, es por eso que la

capa de las cuadras de la ciudad debe estar siempre como primer lugar, es decir debajo de todas las demás.

Ninguna de estas características puede estar ausente de la capa que se quiera manejar en el sistema.

Adicionalmente, cada capa tendrá una manera particular de pintarse, la cual será definida únicamente por el interior de la capa, ya que si esta sufre modificaciones, es esta misma la que tiene que modificar su contenido, por lo que las modificaciones se tendrán que hacer dentro de la capa.

5.4 Capas del sistema

Ya mencionamos en el primer apartado de este capítulo que las partes que conforman el sistema básico son 5, a saber:

- Mapa de la traza de la ciudad.
- Grafo de la ciudad.
- Nombre de las calles de la ciudad.
- Punto de localización.
- Puntos de interés.

Dado que cada una de estas partes, dentro del sistema estará representadas por una capa, en este apartado detallaremos de manera más completa la función de cada una de ellas, así como las operaciones que el usuario podrá hacer sobre ellas en el sistema que se desarrollará.

Vale la pena dejar en claro que el sistema no se limita a lo que se detalla aquí, dado que puede agregársele nuevas funcionalidades, pero, para los objetivos de este trabajo, el sistema solamente tendrá la funcionalidad detallada aquí.

5.4.1 Mapa de la traza de la ciudad.

Esta capa es una de las más importantes para nuestro sistema, ya que es la que muestra cada una de las cuadras que existen en la ciudad.

Vale la pena aclarar que esta capa no es necesariamente exacta a la realidad de la parte de la ciudad a la que representa debido a que depende mucho de la fuente donde se

haya obtenido el mapa. Para nuestro trabajo, la fuente de datos fue el INEGI⁴⁹, y debemos tomar en cuenta que esta institución realiza actualizaciones a sus mapas cada 5, o en ocasiones cada 10, años, por lo que la exactitud del mapa que se muestre en la pantalla de la Palm puede no ser alta.

Ya pudimos observar más atrás en este capítulo una imagen que nos muestra la forma que tendrá la capa. Para mayor comodidad me permitiré citarla de nuevo en la figura 8.

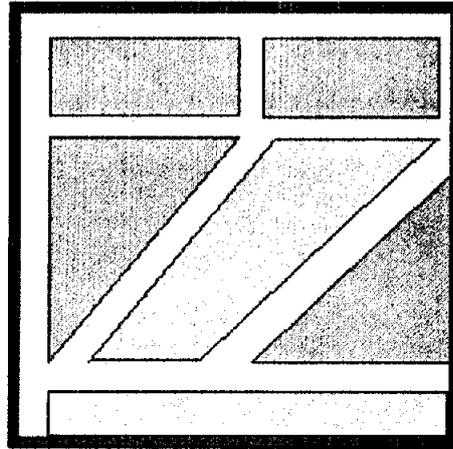


Figura 8. Capa de las cuadras de la ciudad

Como podemos observar, despliega cada una de las cuadras que conforman a la ciudad, de acuerdo, como ya se explicó, a como se encuentran en la fuente de datos de la cual se obtuvo.

Dado que con esto el usuario podrá darse cuenta de la manera en que se conforma la colonia en la que se encuentra, esta capa pasa a ser una de las capas más importantes del sistema. Imaginemos si esta capa no estuviese dentro de la pantalla del sistema, el mismo usuario no sabría ubicarse con exactitud y su orientación se vería reducida al mismo grado que si no tuviera el sistema. Es por esto que esta capa tiene que existir por fuerza.

En cuanto a su bandera de activación, se pensó que la debía tener fija, pero por funcionalidad para el usuario se le dejó dinámica para que puede, el mismo usuario, aparecerla y desaparecerla en el momento en que este lo desee.

Por la cuestión mencionada en el párrafo anterior consideramos que la bandera de activación para esta capa debe ser editable.

⁴⁹ Instituto Nacional de Estadística, Geografía e Informática. Instituto de donde se obtuvo el mapa de la delegación Cuauhtemoc, con el cual trabajaremos en este proyecto.

Volviendo un poco a retomar la importancia de esta capa dentro de nuestro sistema, podemos concluir que la manera de dibujar el mapa dentro de la pantalla del dispositivo es sumamente importante para que el usuario se sienta contento con el uso de la aplicación.

Partiendo de este punto debemos considerar los colores con los cuales se van a dibujar cada una de las cuadras de la ciudad. Y para esto debemos tomar en cuenta que dentro de un mapa solamente encontramos calles delimitadas por asentamientos de diversos tipos a los cuales, en este trabajo de tesis, les hemos llamado cuadras. Pero desglosando un poco más este término, podemos llegar a que hay varios tipos de cuadras, entre los cuales los más básicos y comunes se encuentran los siguientes:

- Manzana con casas o negocios. Este tipo de manzanas es el que predomina en cualquier ciudad, y es la que comúnmente está formada por casas o por negocios.
- Manzana de áreas verdes. Este tipo de manzanas es la que está formada por algún parque o algún área que contiene áreas verdes, y no casas.
- Manzana mixta. Es toda aquella manzana que consta tanto de áreas verdes relativamente grandes como de casas o negocios.
- Pasos peatonales. Habrá asentamientos pequeños que funcionen únicamente como pasos peatonales, los cuales no tendrán ni áreas verdes ni negocios ni casas, inclusive puedan pasar por en medio de estas.

Así como los mencionados aquí, habrá otros tipos de manzanas, o lugares que delimiten a las calles.

Hemos decidido usar los siguientes colores para el dibujado de cada una de las partes que conforman el mapa que se va a presentar en la pantalla.

- Azul claro. Este color se usará para pintar todas las orillas de cada una de las figuras que estén dibujadas en el mapa. Es como cuando uno dibuja algo con lápiz, las líneas están del color de la punta del lápiz (generalmente negro), y el relleno de las figuras que se hacen con el lápiz varía dependiendo de lo que represente.
Así, nuestra aplicación dibujará las orillas de todas las cuadras en color azul claro.
- Amarillo claro. El interior de las manzanas que sean de negocios o casas se pintarán con este color. Esto para hacer un contraste agradable al ojo humano con las líneas de color azul claro con que se pintarán las orillas de cada manzana.
- Amarillo fuerte. Este color se usará para representar las calles de la ciudad. Esto debido a que se pueda ver contraste entre las manzanas y las calles de la ciudad.

- Verde claro. Utilizado para las manzanas en donde predominen las áreas verdes. Esto porque el usuario, dentro de un mapa, identifica este color con las áreas verdes.

Por lo pronto estos son los colores con los que se cubrirán todas las necesidades principales de nuestros mapas a la hora de dibujarlos sobre la pantalla del dispositivo.

En el caso de algún dispositivo en donde su pantalla solamente acepte dos colores, el blanco y el negro, el dibujado tendrá que cambiar, ya que solamente podremos dibujar dos colores. Dado que para este trabajo de tesis se utiliza un dispositivo que acepta colores en su pantalla, por ahora no nos ocuparemos de este problema, y lo dejaremos para un trabajo posterior.

Por último, el usuario no podrá llevar a cabo operaciones en la versión que se hará para este trabajo de tesis. Esto quiere decir que esta capa estará estática en cuanto a su contenido, y la única manera de modificarla es a la hora de que la fuente de datos sea modificada.

5.4.2 Grafo de la ciudad.

Esta capa, sin ser más importante que la anterior, es muy importante ya que el usuario, a través de esta capa, podrá ver la ruta mínima que se hará desde el lugar donde se encuentra hasta algún otro lugar deseado dentro de la ciudad.

La bandera de activación también la tendrá esta capa, con la finalidad de que el usuario disponga su aparición dentro de la pantalla del dispositivo.

Esta capa tendrá la finalidad de presentarle al usuario la ruta más corta entre la posición en la que se encuentra y algún lugar al que desee llegar.

Tendrá la capacidad de segmentarse de acuerdo a la parte del mapa que el usuario está viendo. Esto quiere decir que, por ejemplo, si el usuario está viendo solamente una parte del mapa (como comúnmente sucede) y pide al sistema que encuentre la ruta más corta hasta un lugar que se encuentre fuera de la parte del mapa que está viendo, el sistema le mostrará al usuario la parte de la ruta que tendrá que seguir dentro del mapa que se encuentra viendo, y cuando pase a la siguiente parte del mapa, el sistema le mostrará la parte de la ruta que pertenece al nuevo segmento de mapa. De manera gráfica se puede observar dentro de la figura 9 y 10.

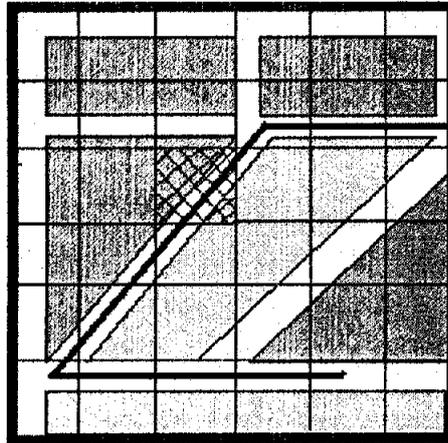


Figura 9. Mapa segmentado en cuadros de un tamaño definido.

Supongamos que la ruta mínima que el sistema nos traza es la que se ve en azul dentro de la figura anterior. Y las líneas delgadas rojas son los límites de cada uno de los segmentos del mapa. En términos prácticos, el usuario va a estar viendo dentro de la Palm el cuadro que se encuentra sombreado por líneas rojas de un lado a otro. Por lo que solo verá lo que se despliega en la figura 10.

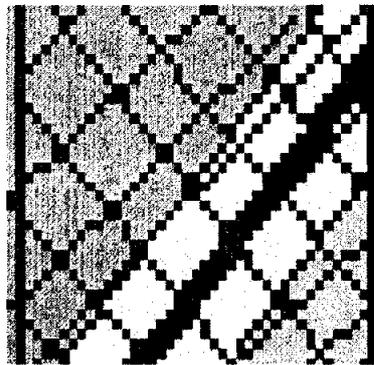


Figura 10. Segmento de mapa ampliado.

Adicionalmente a esto, también podrá ver el punto de localización. Pero si podemos observar, el principio de la ruta mínima no empieza en dicho segmento, sino que empieza dos segmentos a la derecha y tres segmentos hacia abajo del cuadro que el usuario está observando. Con esto podemos deducir dos cuestiones

- El usuario ya ha recorrido todo la primera parte del camino marcado por la ruta mínima hasta el segmento que se ve en el diagrama.

- En el segmento del mapa que se muestra solo se pinta el segmento de la ruta mínima que pertenece a la parte del mapa que se muestra, y no se pinta lo demás.

La primera puede resultar un poco obvia, debido a que es lo que se puede ver en el dibujo. Y, en cuanto a la segunda, se da de esa manera porque para el usuario puede no resultarle muy trascendente el estar viendo la parte que ya recorrió, por lo que dicha parte ya no la necesita. En cuanto a la parte que falta por recorrer probablemente sea trascendente que si lo pueda ver, pero es más importante que vea la parte de la ruta mínima que está recorriendo, así como la parte inmediata a esta, que es la que le toca seguir.

5.4.3 Nombre de las calles de la ciudad.

Esta capa del sistema tiene la finalidad de mostrarle al usuario cada uno de los nombres de las calles que se despliegan en la primera capa explicada.

Esta capa, al igual que las otras, podrá el usuario aparecerla y desaparecerla, sin modificar nada del contenido de cada una de las otras capas que estén en el sistema, las ocasiones que el mismo desee. En el caso de que el usuario decida desactivar esta capa, al igual que las otras también existirá dentro del sistema, lo único es que no se podrá desplegar dentro de la pantalla hasta que vuelva a activarla.

Esta capa no podrá ser modificada por el usuario, pues solamente podrá ser modificada con la aplicación dentro del servidor. La razón es porque se trata de una capa que contiene únicamente letras (los nombres de las calles), y poseen cierta inclinación, la cual va con la inclinación que tienen las calles con respecto a la horizontal de la pantalla.

Esta capa tendrá dos capacidades fundamentales, a saber:

- Segmentación. La segmentación para esta capa funcionará igual que en las demás capas del sistema.
- Los nombres de las calles tendrán la misma inclinación que las mismas calles. Esto para que el usuario identifique con mayor facilidad la calle a la que pertenece el nombre.

El color que se utilizará para pintar las calles será negro, para que puedan resaltar con respecto de las otras cosas dibujadas dentro de la pantalla.

5.4.4 Punto de localización.

Esta capa tiene una importancia grande dentro del sistema GMAP, ya que es la capa que le va a permitir al usuario ver el punto exacto en el que se encuentra dentro del mapa que se despliega en el interior de la pantalla.

Al igual que las demás capas, por ser característica común a todas las capas, también esta capa contiene una bandera de activación en su interior.

La finalidad de que esta capa se active y desactive es porque en muchas ocasiones al usuario le basta con ver las calles de la ciudad, ya que se podrá ubicar dentro de ellas sin la necesidad del punto de localización que le provee el sistema. Debido a esto, dicho usuario puede querer omitir esta parte del sistema, por lo menos verla dentro de la pantalla.

Ahora bien, algo importantísimo que debemos tomar en cuenta es que esta capa del sistema depende, y va íntimamente relacionado, con el aparato GPS que se le conecte a la Palm, el cual da la posición exacta del lugar donde va caminando el usuario, con lo cual puede desplegarlo dentro del mapa que se dibuja en la pantalla del dispositivo. En caso de que dicho aparato, el GPS, no se encuentre conectado a nuestro dispositivo, esta capa no se podrá desplegar dentro del sistema, ya que no tendría fuente de datos para saber el lugar donde debe aparecer el punto de localización dentro de la pantalla. En este caso el sistema no carga la capa a la pila que va a hacer el manejo de las capas del sistema para no gastar recursos extra de dicho dispositivo.

En el caso de que el GPS esté conectado al dispositivo, mediante el menú del sistema, el usuario podrá activar esta capa. En el momento en que el usuario haga esta operación, el sistema tratará de localizar el GPS conectado, y en caso de que lo encuentre cargará la capa al sistema, y en caso de que no lo encuentre la capa seguirá sin cargarse.

El que el dispositivo GPS conectado a la Palm sea encontrado por el sistema depende en gran medida de la implementación de esa parte del sistema donde vaya a ser usado, ya que se trata de manejo de dispositivos externos a la Palm y su funcionamiento, o su forma de lectura, puede variar de dispositivo a dispositivo.

Finalmente, esta capa le proveerá al usuario una de las funcionalidades principales de este sistema, ya que es la que le indicará el lugar donde se encuentra dentro del mapa, a partir del cual puede empezar a trazar rutas mínimas a otros puntos de la ciudad. Si la capa no está cargada, el sistema no podrá saber de manera automática cual es el punto inicial de la ruta mínima que se quiere calcular, por lo que, en este caso, el mismo usuario, de manera manual, tendrá que introducir el punto inicial al sistema. De esta manera, aunque el dispositivo GPS no este conectado al dispositivo, la funcionalidad de la búsqueda de rutas mínimas la tendrá el sistema.

En cuanto al color con que se pintará el punto de localización será el rojo, debido a que este color resalta a la vista en medio de los otros colores claros que ya se han propuesto para las otras capas de este sistema.

5.4.5 Puntos de interés

Por medio de esta capa, el usuario podrá visualizar dentro de la pantalla los puntos de interés de la ciudad que se encuentran en la parte del mapa que se está dibujando en el momento en que se solicita este servicio al sistema.

Esta capa, al igual que las otras, también contará con la bandera de activación.

Esta capa, a diferencia de las demás dentro de este sistema, podrá ser modificada por el usuario, con lo cual podrá agregar o quitar puntos de interés dentro del mapa de acuerdo a su gusto y preferencia. Esta modificación podrá hacerse dentro de la parte del mapa que se encuentre dibujada y se podrá llevar a cabo en el momento en el que el sistema esté en funcionamiento. Para que se pueda hacer dicha modificación tiene que estar la capa activada, en caso contrario no podrá el usuario acceder a dicha funcionalidad dentro del sistema.

Vale la pena aclarar que en el caso de estas operaciones, cuando se terminan de realizar, el mismo sistema se actualiza y al desplegar la capa por siguiente ocasión, esta se despliega con las modificaciones respectivas hechas por el usuario. Esto permitirá al usuario trabajar con las nuevas modificaciones que el sistema tenga en tiempo real.

Finalmente, en cuanto al color que se le destinará al nombre de cada uno de los puntos de interés, se usará el azul fuerte para que resalte a los ojos del usuario y pueda distinguir cada uno de los puntos de interés, dentro del mapa, de cada una de las otras partes que conforman a las demás capas dentro del sistema.

5.4.6 Otras capas

Por el momento, este sistema no tendrá más que las 5 capas que en este capítulo se mencionan, pero se dejará abierta la posibilidad, a nivel código, para que en trabajos posteriores se agreguen cuantas capas puedan ser manejadas por el dispositivo. Y esto dependerá en gran medida de las capacidades de hardware del mismo dispositivo. Esto se debe a que el manejo de capas, al usar muchos recursos del sistema por todas las operaciones que hará sobre cada una de las capas, incrementa el número de recursos usado con cada capa extra que maneja.

5.5 Manejador de las capas

Una vez explicadas cada una de las capas que este sistema tendrá dentro del mismo, ahora nos resta realizar una breve explicación de algunos aspectos del funcionamiento del

manejador de las capas, ya que esta parte del sistema es fundamental para el funcionamiento del mismo.

Aunque, dentro de lo poco que hemos introducido acerca del funcionamiento de esta parte del sistema, trabaja a través de un sistema de capas, estas capas solamente están ligadas a este sistema a través de funciones similares en todas las capas, funciones que el sistema necesita para el manejo interno de las mismas capas.

Esta parte del sistema, el manejador de capas, básicamente, realizará las siguientes funciones:

- Recibir capas nuevas. Va a poder recibir capas que el mismo sistema va a manejar a través de una interfaz creada para la manipulación de cada una de las capas.
- Quitar capas. El sistema no podrá contener un número infinito de capas debido a las capacidades de memoria y procesador de los dispositivos. Esto nos fuerza a que tengamos, en un momento dado, que eliminar alguna capa de las que el sistema está manejando, de tal manera que dicha capa salga de la memoria y podamos usar los recursos en otra cuestión del mismo sistema. Es por esto que esta parte del sistema tiene esta capacidad de alterar el número de capas que se encuentran dentro.
- Pintar capas. El proceso de pintado es obligatorio para esta parte del sistema. Y, dado que es obligatorio, este módulo dentro del sistema tiene que tener la interacción plena con la pantalla del dispositivo en el cual se va a manejar, ya que es el que va a pintar las capas sobre la pantalla de dicho dispositivo. La manera en que serán pintadas las capas es en el mismo orden en el que estén dentro de la pila que las va a contener. De tal manera que la primera capa que se introduzca al sistema (es muy recomendable que se introduzca como primera capa la que se refiere a las cuadras de la ciudad) es la primera que se va a pintar sobre el dispositivo, después la segunda capa, posteriormente la tercera, y así sucesivamente. Lo anterior se dará de esa manera en caso de que la bandera de activación con que cuentan las capas está activada. En el caso de que una capa sea borrada, independientemente si tenía la bandera de activación activada o desactivada, dicha capa sale del sistema, por lo cual ya no se podrá pintar, y mucho menos trabajar con ella referente a las operaciones relacionadas con dicho objeto.
- Acomodar las capas. El sistema, a través de este módulo, podrá darle otro acomodamiento a las capas que el mismo sistema esté manejando. Esto con la finalidad de que al pintarlas el usuario tenga otro panorama con respecto a las mismas. A pesar de que esta característica permitirá pintar las capas en otro orden, se recomienda que la capa de las cuadras de la ciudad esté siempre en primer lugar porque esta ocupa toda la pantalla del dispositivo, y si se coloca en

un lugar que no sea el primero, a la hora de que el manejador de las capas la pinte obstruirá la visibilidad de las otras capas que se pintan antes que esa.

Si observamos, son funcionalidades que, aunque están ligadas a cada una de las capas, no se usan al interior de esta, ya que es el mismo sistema, a través de su manejador de las capas, el que va a utilizar dichas funcionalidades, por lo cual son funciones internas a las capas usadas por el sistema.

5.5.1 Matriz de dibujado

Respecto a lo último mencionado en el apartado anterior, referente a las funcionalidades que tiene cada una de las capas, que son de uso exclusivo del módulo manejador de capas del sistema, tenemos aquella que se trata de dibujar lo que la capa contiene.

Para detallar la manera de dibujar a cada capa, por parte del manejador de capas, dentro de la pantalla tenemos que definir las siguientes premisas, inherentes del mismo dispositivo.

La pantalla del dispositivo es una matriz de pixeles, sobre la cual el sistema pintará las diferentes capas que existan dentro del sistema. Esto significa que en esa única matriz de pixeles de dos dimensiones, a la hora de ser pintadas dos o más capas va a presentarse el problema de que varios pixeles se pinten uno sobre otro. Esto sucederá porque el dibujado de cada una de las capas es secuencial, es decir primero se dibuja la primera, luego de esta última se dibuja la segunda y así hasta terminar de dibujar todas las capas correspondientes. Por esta razón se debe tener una manera de manejar todos los pixeles que se pinten sobre otro para poder saber cual se va a ver y cual píxel no se verá porque se encuentra debajo de otro.

Con respecto a este punto, el manejador de capas podrá manejar de manera práctica a todas las capas, ya que podrá resolver el problema planteado en el párrafo anterior. Esto se debe a que dibujará a las capas como se muestra en la figura 11.

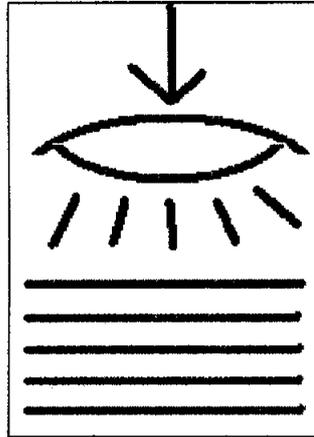


Figura 11. Manera de ver las capas

Como se observa en la figura anterior, el manejador de capas tiene que dibujar las capas de tal manera que como si el ojo estuviera viendo desde arriba a las capas. Aunado a esto, podemos decir que algunas capas tendrán píxeles sin pintar, es decir, invisibles, por lo que, al no pintarse dicho píxel, se visualizará el píxel correspondiente al mismo lugar pero de la capa inferior.

Lo anterior sucede con frecuencia en la capa del punto de localización. En esta capa vale la pena pintar solamente el punto de localización, y lo demás de la matriz de píxeles no tiene que llevar color, es decir, tiene que quedar sin pintar. Y, recordemos lo que mencionamos más atrás en este trabajo de tesis, esta capa es la última capa que se introduce al sistema manejador de capas, es decir, es la última que se dibuja dentro de las capas que se van a dibujar, pues antes de ella se dibujarán las otras capas, ya que se dibujarán en el orden en el que se hayan introducido.

Por otro lado, veamos el ejemplo donde las capas 2, 3 y 4 estén desactivadas, es decir, no se van a pintar dentro de la pantalla del dispositivo. Por esta razón, solamente se van a dibujar dos capas, la capa de las cuadradas de la ciudad y la capa del punto de localización.

Esto nos lleva a que tengamos que pintar una sobre otra las siguientes dos capas, tal como se muestra en la figura 12.

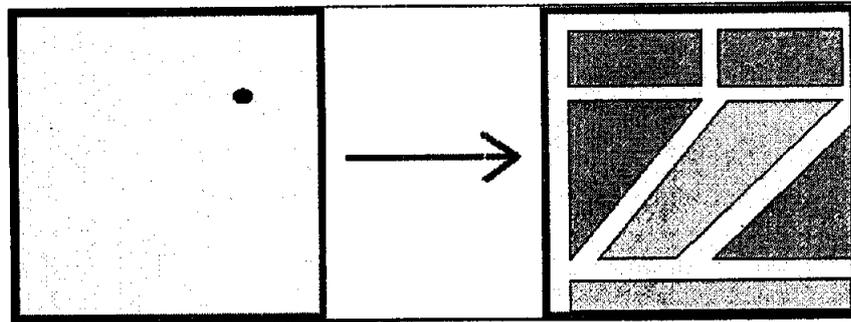


Figura 12. Capas a juntar

En la figura 12 podemos observar que solo son dos capas, la de cuerdas de la ciudad (colocada a la derecha) y la del punto de localización (colocada a la izquierda), la cual esta última se colocará sobre aquella debido a que aquella se introduce primero al sistema y es la primera que se dibuja.

Esta última capa, la del punto de localización, por cuestiones de visibilidad tiene de fondo un color amarillo claro, pero esto no significa que ese color sea el ideal para la capa. El color ideal para la capa es un color transparente en donde se encuentra el color amarillo claro dentro de la ilustración.

Al ser un color transparente, tenemos que toda la capa es transparente, en su color, a excepción del lugar donde se pinta el punto de localización. Es por esto que a la hora de juntar las dos imágenes nos encontremos con alguna imagen como la mostrada en la figura 13.

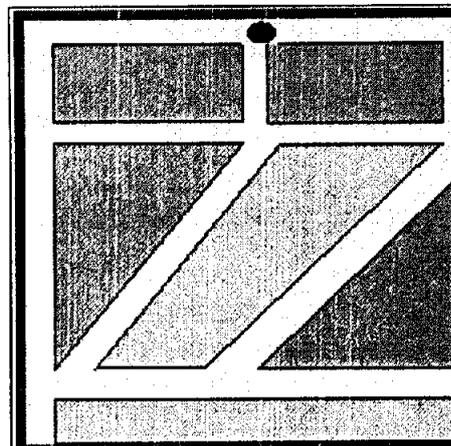


Figura 8. Ambas capas juntas

Aquí ya podemos observar a ambas capas juntas, y si observamos, la capa del punto de localización, la cual se coloca encima de la capa de calles de la ciudad, no impide que se vea la capa inferior debido a que solamente tiene color, dentro de dicha capa, el lugar donde se encuentra el punto de localización.

Esto hará que lo que se observe sobre la pantalla del dispositivo es el último píxel pintado a causa del dibujado de cada una de las capas, en cuanto a píxeles de cada una de las mismas, lo cual provocará que los píxeles que se dibujen primero ya no serán observados por el usuario cuando haya un píxel, de capas subsiguientes, en la misma posición del píxel en cuestión.

Las capas del sistema se dibujarán cada que haya un evento de cambio de capa, y sucederá cuando lo haya con cada una de las capas que se estén dibujando.

La condición de que un evento de este tipo se lance es que la capa donde sea lanzado dicho evento esté activada, ya que no tiene caso que sea lanzado dicho evento si la capa no está dibujada.

Con esta manera de funcionar del manejador de capas, podemos observar que este módulo del sistema solamente se dedicará a dibujar y redibujar las capas cuantas veces sea necesario a cada capa dependiendo de las circunstancias de las mismas. Esto simplifica la funcionalidad de esta parte del sistema.

En cuanto a las otras funciones de cada una de las capas depende de la funcionalidad del sistema y se ejecutarán de manera interna en la capa.

5.5.2 Orden en el dibujado

Es importante saber en que orden serán dibujadas de nuevo las capas cuando se modifica alguna.

En un principio, el dibujado de las capas, en el caso de que hubiera 5 y todas activadas, se hace en el siguiente orden: 1, 2, 3, 4, 5.

Si alguna de las capas sufre alguna modificación, se vuelve a dibujar de nuevo todas en ese orden. Esto se hace de esta manera debido a que el cambio en dicha capa pueda implicar que alguna de las parte de la capa tengan que ser borradas e impresas de manera diferente, por lo cual resulte complicado el que solamente se pinte de nuevo dicha capa y se dejen intactas a las otras. Es por esta razón que el pintado cada que haya un cambio en alguna de las capas se hace de esta manera.

Este método, en el momento de implementarlo, nos pueda provocar problemas en el desempeño de nuestra aplicación, esto debido a que se dibuja toda las capas cada que haya un cambio, por mínimo que sea en alguna de las capas, lo cual puede implicar que se usen mucho los recursos del mismo dispositivo. Si es así, habrá que crear métodos más optimizados para el manejo de esta parte de nuestro sistema.

5.6 Otras funciones de las capas

En cuanto a las demás funciones de cada una de las capas podemos mencionar que el sistema las estará llamando a partir de la ejecución de comandos dentro del menú contextual.

Cuando se mande llamar alguna función mediante la manera citada, el sistema ejecutará los siguientes pasos.

- Se indicará al sistema que se quiere llamar una de la funcionalidad de la capa específica.
- Se toma la capa a la que pertenece la función que se manda llamar mediante el botón de comando.
- Se ejecuta, de manera interna, por medio de un método de la capa la función necesaria.
- Al modificarse la capa, si es necesario se lanza un evento de modificación de capa para que vuelvan a enviarse al dibujado todas las capas que están activadas dentro del manejador de capas.
- El manejador de las capas dibuja a todas las capas de nueva manera de la forma indicada en el apartado anterior.
- Regresa el control al sistema para que el usuario siga utilizándolo.

Como podemos observar, cuando el usuario quiere ejecutar una función sobre la capa en cuestión, a este solamente le basta ejecutarla por medio de algún comando del menú contextual de la aplicación.

5.7 Conclusiones

Como podemos observar, la realización de esta parte del sistema no es muy complicada, ya que se trata del manejo de las capas que puedan estar dentro del sistema.

En el capítulo siguiente, explicaremos la manera en que se van a manejar los datos de acuerdo a cada una de las capas.

Así mismo, se explicará la manera de procesarlos de acuerdo a la fuente de datos que tenemos, que se trata de los mapas que edita el INEGI, hasta que tengamos los archivos que le harán falta a la aplicación para poder funcionar de manera correcta.

Capítulo 6

Transformación de los archivos

Capítulo 6. Transformación de los archivos

Definitivamente una de las necesidades fundamentales de nuestra aplicación es tener los archivos de los mapas necesarios para que nuestro dispositivo móvil pueda desplegarlos en la pantalla, ya sea esta la de un celular o, como es el caso específico de nuestros objetivos en este proyecto, en una Palm.

Ya vimos que este proyecto es desarrollado para desplegar los respectivos mapas en un dispositivo móvil, y, tomando en cuenta las características de este tipo de dispositivos, como se vio en el capítulo 3, podemos deducir que los archivos no podrán ser igual que los que se usen en una aplicación parecida dentro de una computadora de escritorio o portátil. Debido a esto podemos decir lo siguiente de los archivos a crear para nuestra aplicación.

6.1 Archivos de tipo gmp

Primeramente tendremos que mencionara que los archivos especiales que la Palm utilice estarán relacionados con las capas que el sistema maneja.

Ya vimos que cada una de las capas que el sistema va a tener está dividida en cuadros dentro del mapa. Cuadros que ayudan a que el usuario vea solamente una sección del mapa, y no todo el mapa, con lo cual podrá ver las cuadras aledañas a el de manera más ampliada que si viera todo el mapa.

Debido a esto, los archivos también estarán divididos por sección en cada capa. Esto quiere decir que aparte de tener diferentes capas que trabajen en el sistema, cada capa estará dividida en secciones (al estilo de la guía roji en la ciudad de México, mediante la división del mapa utilizando cuadros del mismo tamaño) que delimiten un área para que al usuario se le haga más fácil su visualización. Aunado a esto, podemos decir que cada sección en que esté dividida la capa tendrá sus propios archivos que definan dicha capa.

Archivos individuales, que se utilicen para determinada función, necesarios para definir la totalidad de la funcionalidad de cada una de las secciones de la capa, son los que se tendrán presente dentro del sistema.

Dichos archivos, aquellos que se encarguen de definir todos los datos necesarios para que la funcionalidad de cada una de las secciones de las capas, tendrán la extensión gmp.

Dado que cada capa utilizará uno o varios archivos para definir los datos de cada una de las secciones en que se divida la capa, entonces cada capa se encargará de definir el nombre que llevará cada uno de dichos archivos mencionados. Pero junto con el nombre que se le de a dicho archivo, para identificación de la sección a la que pertenece el archivo,

llevará el identificador de la sección a la cual represente antes del nombre. El identificador de cada sección de las capas de definirá más adelante.

De esta manera, el nombre completo para los archivos que definen los datos de cada una de las secciones de las capas del sistema, tienen la siguiente forma

IDCapa_nombrearchivo.gmp

En un futuro puede este nombre cambiar en su formato por alguna necesidad extra del sistema.

Ahora bien, cada uno de estos archivos va a tener su propio formato, el cual se definirá cuando se esté desarrollando la manera en que se va a obtener dichos archivos.

Para obtener los archivos gmp es necesario llevar a cabo un proceso de transformación, el cual se define en los siguientes rubros de este capítulo.

6.2 Fuente de datos

Hay dos maneras de obtener los mapas necesarios para el buen funcionamiento del sistema GMAP. Una de estas maneras es creando un sistema que nos permita ir dibujando el mapa, de alguna manera, dentro del dispositivo y posterior a eso lo guardemos para que se pueda utilizar en posteriores ocasiones. Y la segunda manera se trata de buscar una fuente de datos confiable y, a partir de dicha fuente de datos, convertir los archivos que proporcione a archivos de tipo gmp necesarios para el funcionamiento del sistema.

Definitivamente la segunda es la mejor, ya que la primera implica que tengamos que recorrer cada uno de los centímetros del mapa que queramos construir, puesto que la construcción del mismo sería un proceso meramente manual y realizado por el usuario.

Otra de las desventajas de esta segunda manera de realizar esta operación es que el usuario, al ir trazando el mapa dentro de su dispositivo, puede incurrir en equivocaciones o en trazos que no concuerden con la realidad de las cuadras de la ciudad, por lo que con este método, el mapa trazado es muy propenso a errores de traza.

En cambio, la primera manera que se plantea, la de tomar los mapas de una fuente de datos confiable, es el camino más seguro, dentro de esta aplicación, para llegar al fin deseado.

Ahora bien, después de haber decidido por cual opción nos íbamos para obtener los mapas, lo que nos queda es decidir por una fuente de datos confiable.

Las fuentes de datos confiables que tenemos dentro de la ciudad de México, y que son las que tomamos en cuenta, son dos: Los mapas realizados por el INEGI, y los mapas

realizados por la “guia roji”. Veamos cuales son las ventajas y las desventajas de cada una de estas dos opciones.

En cuanto a las ventajas de la guia roji tenemos las siguientes.

- Son mapas que ya están divididos en cuadros (secciones) de cierto tamaño.
- Son mapas que ya tienen indicado todos los puntos de interés con cierta figura para cada uno.
- Son mapas que dividen muy bien, por colores, cada una de las partes de dicho mapa, como calles, colonias, parques y jardines, entre muchas otras cosas dentro del mapa.
- Son mapas que el usuario, por lo menos el que vive en la capital de México, está acostumbrado a usar.
- Se actualizan muy seguido.
- El precio es bajo
- Son pequeños en tamaño.

Y de las ventajas de los mapas que produce el INEGI podemos decir lo siguiente:

- Son mapas que podemos ver dentro del programa Autocad
- Son mapas de la traza de la ciudad
- Están a una escala exacta de 1:5,000
- Son mapas hasta cierto punto vectorizados
- Contiene los nombres de las calles de la ciudad
- Contiene algunos de los puntos de interés más importantes.

Como vemos, ambos tienen ventajas muy útiles para nuestro sistema. Por ejemplo la “guia roji” viene con los colores deseados para cada una de las partes de la ciudad, así como con las anotaciones correspondientes. En cambio los archivos de autocad, son mapas vectorizados y también tienen, aunque solamente los más importantes, los puntos de interés en la ciudad y los nombres de las calles.

En fin, desafortunadamente cada una de estas dos opciones también tienen desventajas, entre las cuales podemos mencionar las siguientes para el caso de la “guía roji”.

- No son dibujos vectoriales.
- Son imágenes tipo raster
- La extracción de los datos necesarios para los mapas de nuestro sistema es más difícil.
- No hay escala exacta

En cuanto a las desventajas de los archivos que nos provee el INEGI podemos citar lo siguiente:

- Resultan ser muy caros
- No vienen todos los puntos de interés de la ciudad
- Son muy extensos en cuanto a tamaño.
- Su actualización es cada 5 o 10 años.

Vemos que también las desventajas de cada uno de estos dos archivos nos afectaría en la realización del sistema.

Pero, a pesar de las ventajas y desventajas, podemos decir que ninguno de los dos, de manera completa, cumple con los requisitos para que nuestro sistema trabaje de manera satisfactoria para el usuario del mismo sistema. Por este motivo tenemos que hacer una selección de alguno de ambos, tomando en cuenta que cualquiera que elijamos tendremos que realizar una serie de cambios y adaptaciones para poder llegar a los archivos necesarios para que nuestro sistema los pueda leer.

Para este sistema, en cuanto a la elección de una de las dos fuentes de datos que tenemos, elegimos trabajar con los datos que nos proporciona el INEGI, ya que estos tienen las siguientes ventajas, las cuales fueron decisivas dentro de nuestra elección.

- Datos vectoriales. Esto nos dará la facilidad de manipularlos y convertirlos a nuestro gusto en el formato necesario para que el sistema los pueda leer.
- Puntos de interés. Aunque son limitados los puntos de interés que tienen estos mapas del INEGI, es una parte del sistema que resulta ser dinámico para el

usuario, es decir, el mismo usuario va a poder agregar o quitar dichos puntos de interés.

- **Traza de la ciudad.** La traza de la ciudad es lo más importante de este sistema, ya que dicha traza es la que nos va a dar el mapa que se le va a mostrar al usuario dentro del sistema.
- **Escala.** La escala de la traza nos facilita mucho el trabajo en cuanto a que podemos definir el tamaño exacto en que se va a dibujar el mapa dentro del dispositivo. Esto nos facilitará en ubicar el mapa respecto a las coordenadas, de cada una de las cuadras, exactas en las que se encuentran.
- **Nombre de las calles.** Contiene los nombres de las calles de la ciudad, lo que nos permitirá colocarlas dentro del mapa que vamos a pintar en el dispositivo.

Dos de los puntos que nos detuvo en la decisión por un momento es el precio de los mapas del INEGI y el tiempo de actualización de los mismos.

Por un lado, el precio era demasiado alto con respecto a los mapas de la “guía roja”, lo cual eleva demasiado el costo de la aplicación.

Por otro lado, el tiempo de actualización de los mapas es de 5 a 10 años, y tomando en cuenta que las ciudades están creciendo a un paso que supera a ese tiempo, podemos llegar a la conclusión que un período de actualización de 5 a 10 años es mucho tiempo para una aplicación de esta índole.

Pero, a pesar de estas dos desventajas grandes, se decidió tomar como fuente de datos al INEGI.

Y, por obvias razones, estos archivos no nos sirven en el formato en que vienen del INEGI, por lo que tenemos que someterlos a un proceso de conversión de formato para que nuestro sistema los pueda interpretar.

Dicho proceso, lo detallamos en este capítulo a partir del siguiente apartado.

6.3 Fuente de datos

La fuente de datos, como se mencionó en el anterior apartado en este capítulo, que se han elegido son los mapas del INEGI.

Estos mapas tienen varias características

- Podemos adquirir los mapas por ciudades, por delegaciones, o por otro tamaño de extensión de territorio.
- Son tres capas las que te proporcionan, los nombres de las calles de la ciudad, los puntos de interés, y la traza de la ciudad (cuadras de la extensión territorial que uno adquiere).
- Te dan toda la extensión territorial que pidas en una sola pieza, respetando las tres capas mencionadas, lo que nos impide, de manera manual, separarla por cuadros de secciones de mapa.

Vemos que las características que mencionamos son muy útiles en la aplicación Autocad, pero recordemos que dichos archivos están hechos para ser leídos por dicha aplicación.

Para nuestro sistema se adquirió el mapa de la delegación Cuauhtemoc de la ciudad de México. Esto se hizo de esa manera por varios motivos, los cuales, los más importantes, se mencionan a continuación, así como una pequeño detalle de cada motivo.

- En la delegación Cuauhtemoc se encuentra el Zócalo capitalino, y es el lugar que se le denominó, por nosotros, como zona cero, la cual representa el punto central del país, a partir del cual todas las coordenadas de las partes del mapa van a tener su referencia.
- Para empezar con la primera etapa del sistema es suficiente con esta parte del mapa. En posteriores versiones del mismo sistema se adquirirán las otras partes de la ciudad de México.
- Basta con una delegación para hacer las pruebas necesarias que el sistema requiere.

En Autocad, estos archivos se ven como se muestran en la figura 1.



Figura 1. Mapa de las cuadras de la delegación Cuauhtemoc

Podemos ver que es un mapa que, aunque no se nota del todo por la resolución, define cada una de las cuadras de la delegación Cuauhtemoc.

Si este mapa, tomando en cuenta la resolución que tiene, se pinta dentro de un dispositivo de tipo Palm tal cual lo vemos en la ilustración anterior tendríamos dos dificultades. La primera es que el usuario no podría distinguir las cuadras que conforman a la delegación, ya que se verían casi nulas por el tamaño de la pantalla del dispositivo. Y en segundo lugar, el usuario sentiría incomodidad por ver todo ese mapa, cuando el mismo está en una parte mínima del mismo.

El otro archivo que nos proporciona el INEGI es el que contiene los nombres de las calles y los puntos de interés, el cual, cuando se despliega en Autocad lo podemos observar como se muestra en la figura 2.

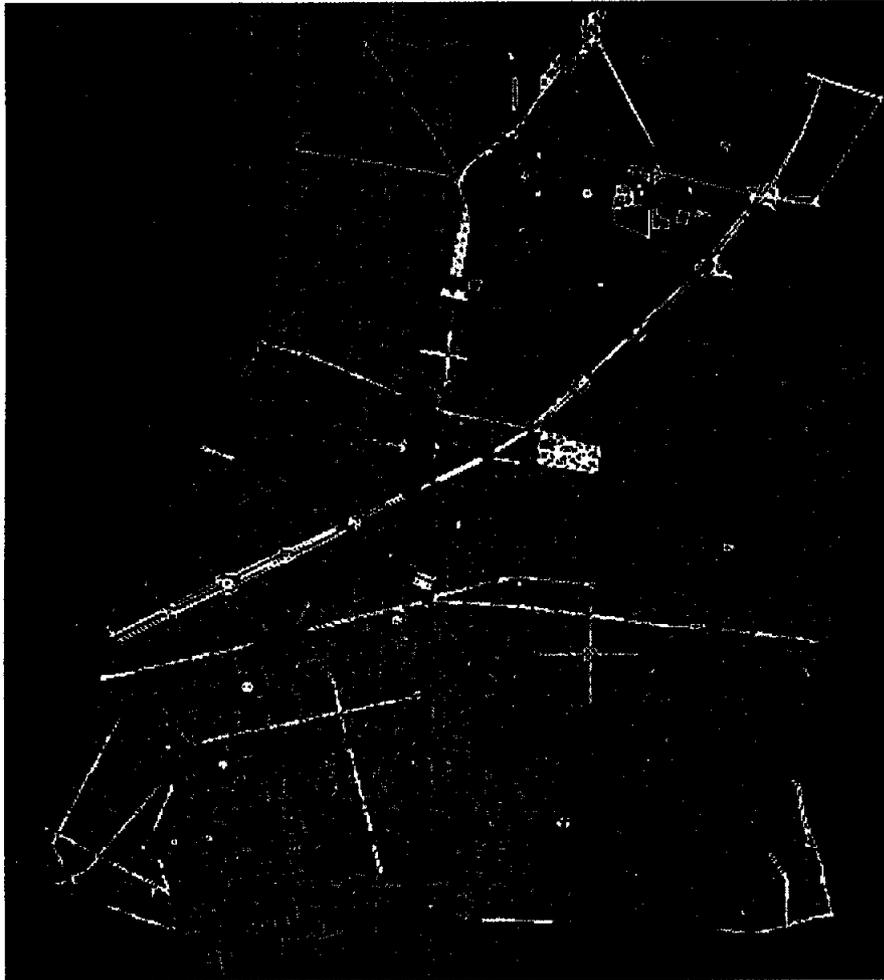


Figura 2. Nombres de las calles y puntos de interés.

Este archivo contiene los nombres de las calles y los puntos de interés, por lo que de este otro archivo tendremos que obtener dichos datos.

Igual que en el anterior archivo mostrado como ilustración, a este archivo también tendremos que dividirla en cuadros seccionales.

Dejando atrás la cuestión de que son dos archivos individuales, lo cual nos hace más difícil la extracción de datos, estamos conscientes que cada uno de estos dos archivos se tiene que trabajar para poder adaptarlos al dispositivo, lo cual nos lleva una serie de pasos.

Los dos archivos, si se ven uno encima de otro, todas los nombres de las calles de uno de los archivos coinciden, en posición, con las calles del otro archivo. Esto nos ayuda a cotejar los datos de un archivo a otro.

6.4 Procesamiento de los archivos

Para poder explicar las etapas por las que pasan los archivos que nos proporciona el INEGI tenemos que describir un poco de las características de dichos archivos.

Dentro de las características que tenemos que saber para comprender el proceso de transformación están las siguientes.

- Las cuadras de la ciudad, dentro del mapa, están definidas como figuras polígonas, donde cada figura es definida por una serie de puntos que representan cada uno de sus vértices.
- Los puntos que representan sus vértices vienen en coordenadas bidimensionales, es decir, una coordenada en 'x' y otra coordenada en 'y'.
- El punto de referencia de las coordenadas que se manejan son respecto a la esquina superior derecha. Pero esta esquina no tiene la coordenada (0,0), sino que tiene una coordenada específica. Esto significa que la coordenada (0,0) no se encuentra en la delegación Cuauhtemoc, lo que nos hace suponer que la coordenada origen se encuentra en otro lugar de la ciudad de México, e inclusive puede estar en alguna otra ciudad de la republica.
- Los nombres de las calles y los puntos de interés también manejan las mismas coordenadas, y el punto de referencia, que manejan las cuadras de la ciudad.

Después de haber mencionado las características básicas de estos archivos, podemos empezar a detallar el proceso de tratamiento de los datos para lograr obtener los archivos necesarios para que el sistema trabaje como debe.

6.4.1 Archivos DXF y DWG.

El formato inicial de los archivos son dos, dxf y dwg.

Ambos formatos son muy similares, la diferencia entre ambos es su contenido y el tamaño. El formato dwg es más difícil de leer por estar en formato binario, pero básicamente es lo mismo que el formato dxf en cuestión de contenido.

Las cuadras de la ciudad están en el formato dwg, el cual lo hace más pequeño en espacio en disco duro.

Los nombres de las calles y los puntos de interés están en el mismo archivo, y este archivo está en el formato dxf. Este tipo de formato es más fácil leerlo porque es en formato texto, inclusive de manera normal, sin ningún tipo de procesamiento podemos leerlo debido a que es texto legible al ojo humano.

6.4.2 DWG a DXF.

Dado que las cuadras de la ciudad están en el formato dxf, y ese formato, como ya lo mencionamos, es más difícil de procesar, el primer paso que tenemos que realizar es el que tenemos que convertir el archivo de tipo dxf a un archivo de tipo dwg.

Este paso se lleva a cabo con una herramienta gratuita llamada DWG2DXF, el cual se obtuvo de la página de Internet "http://www.autodwg.com/DWG_DXF_Converter/".

Lo que hace esta herramienta es realizar ese primer paso que nos hace falta dentro de nuestro proceso, convertir el archivo dwg al formato dxf.

La herramienta utilizada es gratuita y basta con obtenerla en la página de Internet para poder usarla.

La desventaja de esta herramienta, aunque es desventaja del formato dxf, es que el archivo, en cuestión de tamaño en disco duro, ocupa cerca del doble que en el formato dwg ya después de estar convertido. Pero esto tiene solución en uno de los siguientes pasos dentro del tratamiento de conversión.

Una vez hecho esto, ahora tenemos dos archivos de tipo dxf, el de las cuadras de la ciudad y el de los nombres de las calles y puntos de interés de la misma.

6.4.3 Extracción de vértices.

De los archivos que ahora tenemos, los cuales resultaron de la previa transformación, los leeremos para extraer todos los vértices que en su interior existan.

Llamamos vértice a cada uno de los puntos que existen dentro del archivo de formato DXF. Puntos que son el principio o el final de una línea dentro del dibujo.

Explicado de manera más clara, las cuadras de la ciudad son un conjunto de líneas unidas una tras otras hasta que la última de éstas llega al punto inicial. Cada una de las esquinas de dichas cuadras son los vértices, los cuales están unidos por líneas que van de vértice a vértice, con lo cual se conforman las cuadras.

Dentro del mapa que aparece en Autocad, el cual se muestra más atrás en este trabajo, podemos verlo más claramente si aplicamos la operación de zoom sobre el mismo. Esto lo observamos en la figura 3.

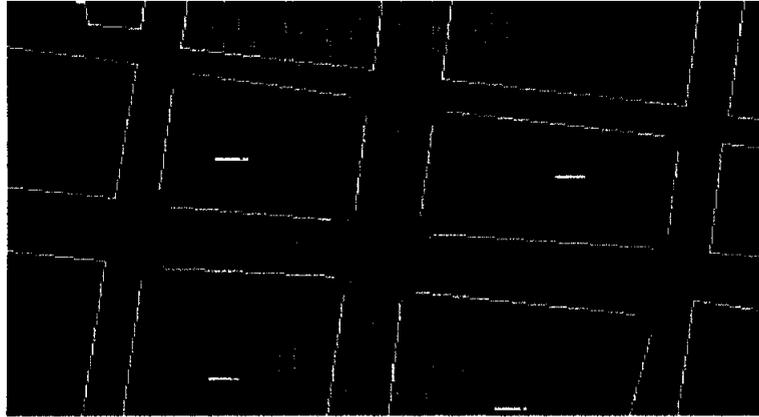


Figura 3. Ampliación del mapa de la delegación Cuauhtemoc

En la figura 3 podemos ver una sección del mapa de la delegación Cuauhtemoc pero ampliada. Los puntos rojos que vemos son puntos editados, para indicar cuales son los vértices dentro del mapa.

Como podemos observar, los vértices son los puntos que conforman cada una de las cuadras de la ciudad. Uniendo los vértices que pertenecen a cada cuadra mediante líneas podemos lograr ver la cuadra entera de la cual se trata.

Los datos de la posición de los vértices son dados por el archivo de formato DXF, inclusive la cuadra a la que pertenece cada vértice son indicados en el mismo archivo. Esto nos ayuda a la hora de leer dichos vértices.

En este paso se extraen todos los vértices indicados en el archivo de formato DXF y se almacenan en un archivo con el siguiente formato.

`#:CoordX:CoordY`

Vemos que lo que se almacena son tres cuestiones.

- Consecutivo para identificar el vértice dentro del mapa. Este número consecutivo se inicializa en 1 y termina en el número de vértices del archivo. La finalidad de este número es solamente llevar un control de los vértices que tiene el archivo, y para optimizar lo más que se pueda, en tamaño del archivo, el archivo donde se indican las cuadras.
- Coordenada X del vértice. Este dato se guarda con la finalidad de saber en que parte del mapa se encuentra el vértice, y así poder localizar su ubicación para realizar los posteriores cálculos.

- Coordenada Y del vértice. Este dato se guarda con la finalidad de saber en que parte del mapa se encuentra el vértice, y así poder localizar su ubicación para realizar los posteriores cálculos.

Las coordenadas X y Y no nos sirven si no tenemos el punto de referencia exacto con el cual la tenemos que comparar. Es por esto que necesitamos saber a partir de que se obtiene dicho dato, ya que, sino lo tenemos, será imposible ubicarlo dentro de un mapa.

Respecto al punto de referencia para ubicar a todos los vértices dentro del mapa, podemos decir que el mismo archivo de formato DXF nos indica dicho punto de referencia, por lo que también lo guardamos, pero en otro archivo. En dicho archivo guardaremos datos que nos van a apoyar en la realización de los cálculos posteriores.

6.4.4 Extracción de cuadras de la ciudad.

Una vez obtenidos todos los vértices, lo que sigue es obtener todas las cuadras que existan en el mapa que estamos leyendo a través del archivo DXF.

Sabemos que las cuadras son la unión de algunos vértices por medio de líneas, por lo que basta registrar los vértices que conforman a la cuadra y el orden en el que van a ser unidos, esto para poder unir dichos vértices y encontrar la forma que tiene la cuadra a la cual pertenecen dichos vértices dentro de la aplicación donde se usen los datos.

Basta con esos datos porque el dibujo total de la cuadra se hará directamente en el sistema que esté instalado en el dispositivo, con lo cual no hay necesidad mas que de almacenar los vértices que conforman a cada una las cuadras de la ciudad, junto con sus coordenadas.

Las cuadras de la ciudad, tanto los vértices como las coordenadas de estos se guardarán en otro archivo con el siguiente formato.

`#:#vértice:#vértice:#vértice:#vértice:#vértice:#vértice:#vértice:#vértice...`

Donde primero vemos un número antes del primer signo de dos puntos, el cual nos indica el número de la cuadra que estamos registrando. Este nos servirá para identificar las cuadras de acuerdo a su número.

Posterior a esto, viene una serie de vértices. Son números que representan, cada uno, un vértice de la cuadra. Esto es, cada número es el identificador de uno de los vértices que conforman a la cuadra. Dicho identificador se relaciona con el consecutivo que se le asignó a cada vértice cuando se extrajeron del archivo DXF, con lo cual podemos realizar dicha comparación para ver las coordenadas donde se encuentra la cuadra.

El orden en que aparecen es el orden en que se deben unir para poder formar la cuadra completa. Ya habíamos mencionado que la unión se hará a través de líneas.

El número de vértices que aparecerán es variable, pues una cuadra puede no tener forma de cuadrado, en cuyo caso serían solamente cuatro vértices, sino tener la forma de un polígono, e irregular, por lo que el número de vértices incrementa. Por tal motivo, el número de vértices es muy variable, podemos tener cuadras desde tan solo tres vértices hasta un número muy grande de vértices.

Con esto logramos tener el registro de las cuadras de la delegación Cuauhtemoc, o de la ciudad entera si es el caso, separadas en otro archivo.

6.4.5 División en cuadros (secciones del mapa).

Ya vimos en unos de los apartados anteriores que dentro de la pantalla de la Palm no podemos dibujar todo el mapa que tenemos, en este caso el mapa de la delegación Cuauhtemoc. Con mayor razón no podemos colocar dentro de la misma pantalla el mapa de toda la ciudad de México. Esto porque el usuario no distinguiría ninguna de las calles por la resolución que tendría el mapa al ser dibujado dentro de la pantalla.

Para dar solución a esto, habíamos mencionado que habría que dividir el mapa en secciones pequeñas que, al ser dibujadas dentro de la pantalla del dispositivo, puedan ser fácilmente distinguidas cada una de las partes del mapa.

Dichas secciones serán cuadros relativamente pequeños en los que se dividirán todos los mapas. Estos cuadros son la parte del mapa que se estará dibujando dentro de la pantalla de la Palm. Esto le permitirá al usuario poder ver una sola parte del mapa, con la finalidad de que todas las calles, cuadras y demás partes del mismo, puedan ser distinguidos de mejor manera.

Para empezar a definir los cuadros de los que estamos hablando tenemos que definir el punto de referencia a partir del cual vamos a empezar a tomarlos en cuenta. Es decir, no podemos definir cuadros sin un orden específico.

Por otra parte tenemos que definir las dimensiones que tendrán cada cuadro, de tal manera que dichas dimensiones son las que va a poder el usuario ver en un momento dado dentro de la pantalla de la Palm.

En cuanto al primer punto a tomar en cuenta, el punto de referencia que definimos es el centro, o por lo menos un punto muy cercano al mismo, del zócalo capitalino.

Y para encontrar dicho punto dentro del mapa y de los datos que el INEGI nos da, lo que tenemos que hacer es identificar primero cual de todas las cuadras del mapa es la que representa al zócalo capitalino. Para esto podemos mencionar que, dentro de los datos que proporciona el INEGI, además de las cuadras dibujadas, también viene el identificador que

se utiliza para cada una de dichas cuadras. Vemos que el identificador correspondiente al zócalo capitalino es el número 090150001077-11, el cual se encuentra dibujado dentro de la cuadra que representa al zócalo. Las coordenadas donde se dibujo este identificador, dentro del mapa, son las siguientes:

Coordenada en X: 486018.0288479999
Coordenada en Y: 2148520.605235

Esto nos habla de que dicho punto se encuentra dentro del zócalo. Esto quiere decir que las coordenadas obtenidas de los datos del INEGI son de un punto, dentro del mapa, que se encuentra en el interior de la cuadra que representa al zócalo.

Ahora bien, el centro del zócalo no es ese punto definido, pero eso tampoco nos preocupa ya que para que el sistema se facilite en los cálculos definimos un punto cercano al ya definido pero con las coordenadas definidas como un número entero.

Respecto a lo anterior, definiendo un punto cercano al ya definido cuyas coordenadas estén definidas en un entero podemos definir el siguiente

Coordenada en X: 486000
Coordenada en Y: 2148500

Por lo tanto ya tenemos el punto a partir del cual van a definirse todos los cuadros en los que se seccione nuestro mapa. A esta coordenada le vamos a llamar coordenada cero.

Ahora lo que nos resta es definir las dimensiones de los cuadros.

Para esto, los datos que maneja el INEGI en el mapa que nos proporcionó vienen en unidades no definidas. Independientemente de lo que signifiquen dichas unidades, definimos primeramente un número de 1200 unidades para cada lado de los cuadros en los que se dividirán los cuadros⁵⁰.

El número de 1200 unidades por un lado lo elegimos porque dicho número tiene más divisores que el 1000, que es el que generalmente se usa por terminar con un mayor número de ceros. El que tenga más divisores nos ayudará en un momento dado realizar un recálculo de cuadros menores.

Una vez definido esta medida, ahora obtenemos las coordenadas que van a tener la esquina superior izquierda y la esquina inferior derecha del cuadro central. Para esto tenemos que tomar en cuenta dos cosas; una es que la coordenada cero será el centro del primer cuadro en que se dividirá el mapa; y otra cuestión a tomar en cuenta es que dentro

⁵⁰ El definir un número al azar para las dimensiones de los lados del cuadro se hizo porque de esa manera podemos llegar a una conclusión en la que nos podamos basar, de manera visual, si conviene para la aplicación. Esto significa que a partir de este número definido podremos visualizar en el Autocad todas las cuadras que el usuario podrá ver dentro de la pantalla, y, a partir de dicha visualización, podemos ajustar dicho número definido de acuerdo a lo que mejor convenga al usuario.

del mapa del INEGI las coordenadas en el eje x crecen hacia la derecha, mientras que las coordenadas en el eje y crecen hacia arriba, esto se define en la figura 4.

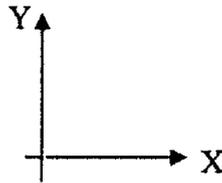


Figura 4. Plano cartesiano utilizado

Una vez visto estos dos puntos importantes ahora presentamos las coordenadas de la esquina superior izquierda y la inferior derecha del primer cuadro en que se dividirá el mapa.

Esquina superior izquierda. Coordenada X: 485400
Esquina superior izquierda. Coordenada Y: 2149100
Esquina inferior derecha. Coordenada X: 486600
Esquina inferior derecha. Coordenada Y: 2147900

Este cuadro, observándolo dentro del mapa que se dibuja en Autocad podemos ver las cuadras alrededor del zócalo capitalino tal como se muestra en la figura 5. En dicha figura podemos ver la cuadra que representa al zócalo con un punto rojo en medio.

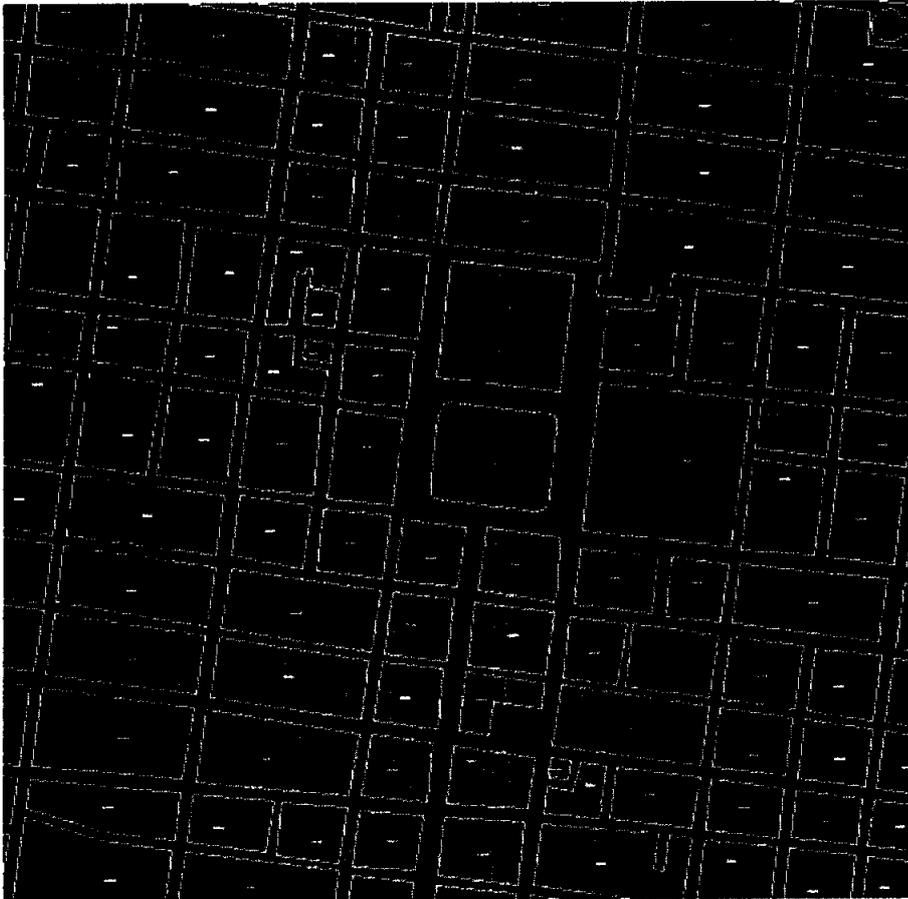


Figura 5. Contenido del cuadro de 1200 unidades cuadradas.

Si tomamos en cuenta cuadros con las dimensiones ya definidas, 1200 unidades por cada lado, tendríamos cuadros con el contenido que arriba se muestra en la ilustración. Ahora bien, si lo que vemos en la ilustración anterior nos lo imaginamos dentro de la pantalla de la Palm podemos llegar a la conclusión de que es mucha información dentro de la misma, con lo cual puede implicar que el usuario no pueda observar con la suficiente precisión cada una de las partes del mapa que se le despliega en la pantalla.

Tomando en cuenta lo anterior mencionado, llegamos a la conclusión que las dimensiones para los cuadros definidas como primera opción son muy grandes. Por esta razón decidimos probar con dimensiones de la mitad de lo definido.

Las siguientes dimensiones a probar son las de 600 unidades por cada lado del cuadro definido.

Con estas nuevas dimensiones, podemos concluir en que las coordenadas de las esquinas superior izquierda y la inferior derecha del nuevo cuadro son las siguientes

Esquina superior izquierda. Coordenada X: 485700

Esquina superior izquierda. Coordenada Y: 2148800
Esquina inferior derecha. Coordenada X: 486300
Esquina inferior derecha. Coordenada Y: 2148200

Con estas nuevas dimensiones, ahora, el cuadro que veremos dentro de la pantalla de la Palm, tomando en cuenta toda su información interna (nos referimos a las cuadras y nombres de las calles), toman la forma que se muestra en la figura 6.

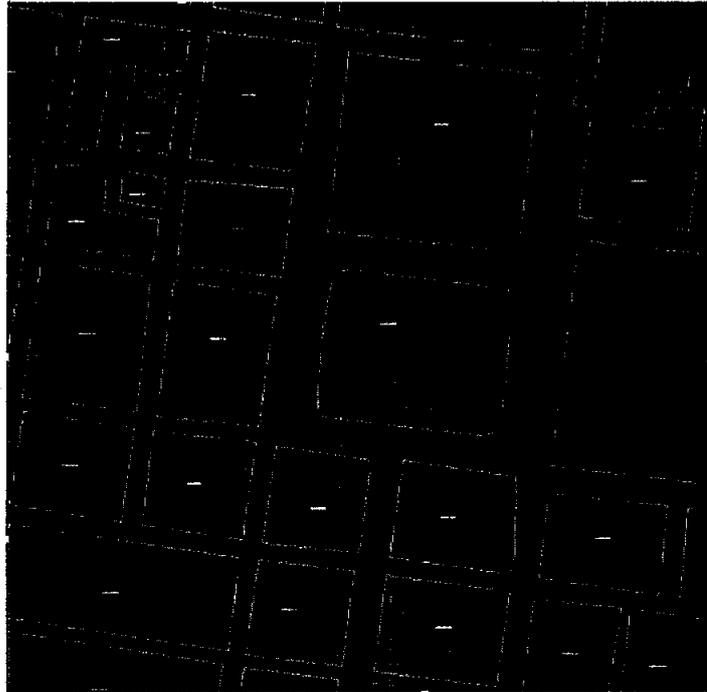


Figura 6. Contenido del cuadro de 600 unidades cuadradas.

En la figura 6 podemos ver que la carga de información sobre la pantalla ya no es mucha, por lo que esto resultará como beneficio para el usuario ya que podrá ver todo esto con mayor claridad sobre la pantalla.

En estas proporciones, podemos realizar el zoom necesario si dividimos el cuadro resultante como se muestra en la figura 7.

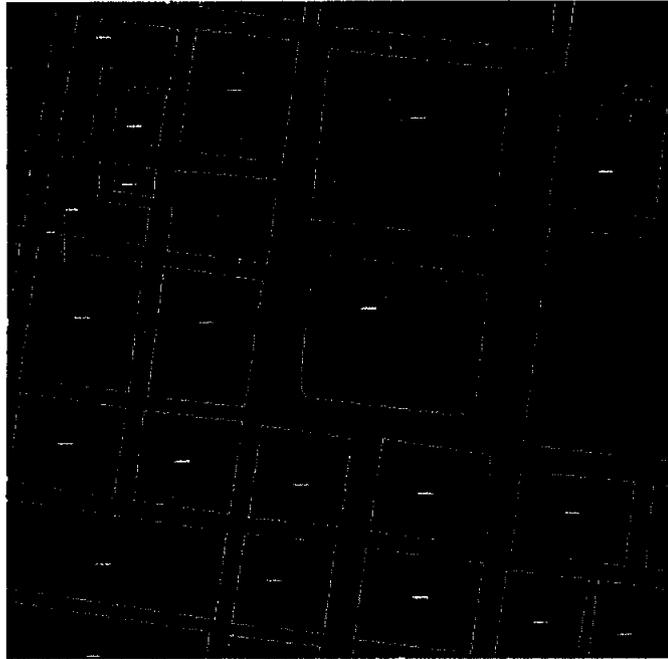


Figura 7. División del mapa para la operación del zoom

De esta manera el usuario podrá tener la opción de observar el mapa con una mayor resolución. Pero esta modalidad se realizará en la segunda versión de este software, es decir, en este trabajo de tesis no se desarrollará.

Con todo lo expuesto, podemos llegar a la conclusión de que los cuadros tendrán una dimensión de 600 unidades por cada lado y el cuadro central tendrá como centro la coordenada cero, la cual están definidas sus coordenadas más arriba.

La definición de las dimensiones de cada cuadro nos abre el camino para formar los diferentes archivos que representarán cada cuadro dentro del mapa.

Dichos archivos llevarán el nombre de las coordenadas que tengan en la forma siguiente

`coordX.coordY`

seguido por un punto y luego por el identificador de la capa a la que representa.

En el caso de las coordenadas, estas se miden en cuadros avanzados de manera vertical, para las coordenadas en x, o cuadros avanzados de manera horizontal, para las coordenadas en y, a partir del cuadro central. El cuadro central, aquél que tiene como centro las coordenadas cero, tendrá la coordenada x como 0 y la coordenada y como 0. El cuadro que se encuentre de bajo del cuadro central tendrá las coordenadas "-1,0", el que se encuentre a la izquierda de este último tendrá las coordenadas "-1,-1", y así sucesivamente.

De esta manera, la primera parte del nombre del cuadro central será "0.0".

Para el caso de las capas que habíamos definido, los identificadores que se le asignaron a cada una son las siguientes

- Traza de la ciudad. A esta capa se le asignó el número 0 como identificador.
- Grafo de la ciudad. A esta capa se le asignó el número 1 como identificador.
- Nombres de las calles de la ciudad. A esta capa se le asignó el número 2 como identificador.
- Puntos de interés. A esta capa se le asignó el número 3 como identificador.
- Punto de localización. A esta capa se le asignó el número 4 como identificador.

6.4.6 Extracción de nodos.

Una vez teniendo los dos anteriores en archivos, el que contiene los vértices de la ciudad y el de las cuadras de la misma, ahora procedemos a obtener los nodos que pertenecen al grafo que se formaría de la ciudad.

Recordemos que tenemos que ver la ciudad como un grafo, en donde cada esquina de la misma es un nodo y cada calle que una a dos esquinas se tratará de una arista.

Para esto, dentro del Autocad, marcamos los nodos que queremos que nuestro sistema contenga.

Una vez marcados, podemos extraerlos del archivo resultante. Pero antes de extraerlos, podemos realizar, dentro del mismo Autocad, el trazado del grafo de la ciudad.

6.4.7 Cálculo del grafo.

El grafo se puede generar dentro del Autocad, dibujando las líneas (aristas) que queremos que contenga el grafo.

Para dibujar las aristas del grafo, basta con basarse en los nodos que ya se habían trazado.

Una vez que se tiene tanto los nodos como las aristas, ya se pueden extraer del archivo de tipo DXF resultante.

Cuando se extraen tanto los nodos como las aristas del grafo, estos datos se guardan en un archivo con el nombre de

grafo.palm

Como podemos observar, este archivo no se guarda seccionado. Esto es porque se va a usar para la funcionalidad de la búsqueda de las rutas mínimas, y como los puntos de origen y destino de dichas rutas mínimas pueden estar en varios cuadros, entonces implicaría mayor problema a la hora de pintarlo.

Atendiendo a esta diferencia en este archivo con respecto a los demás, la capa desplegará la sección de la ruta trazada que pasa por la sección de mapa dibujado en la pantalla del dispositivo.

6.4.8 Extracción de nombres de las calles.

La extracción de los nombres de las calles de la ciudad se realiza a través de uno de los archivos que provee el INEGI. En este archivo vienen todos los nombres de la porción del mapa que se adquirió, en nuestro caso solamente la delegación Cuauhtemoc.

Los datos que nos proporciona el archivo mencionado son la ubicación del nombre dentro del mapa con las coordenadas manejadas por el INEGI, la inclinación que va a tener el nombre dentro del mapa y el mismo nombre al que hace referencia.

Los datos obtenidos de los archivos proporcionados por el INEGI son almacenados en diferentes archivos, respetando la división, planteada anteriormente, del mapa.

Los nombres de cada uno de los archivos para esta capa serán igual que en el caso de los archivos utilizados para la capa de las cuadras de la ciudad, es decir

coordX.coordY.nombrescalles

De esta manera, el sistema podrá identificar cuales nombres de calles tendrá que ser pintadas dentro de la pantalla del dispositivo, de acuerdo a la sección del mapa que se está desplegando.

6.4.9 Extracción de puntos de interés.

Los puntos de interés son proporcionados, sólo los que maneja el INEGI, en el mismo archivo en donde se proporcionan los nombres de cada una de las calles.

La manera en que el INEGI presenta estos datos es la coordinada en la que está ubicado el punto de interés, y lo que representa dicha coordenada, es decir, el tipo de punto de interés que es.

Los puntos de interés que el INEGI maneja dentro de sus datos son los siguientes:

- Cementerios
- Escuelas
- Iglesias
- Mercado
- Palacio o ayudantía
- Jardines
- Servicio Médico

Debido a esto, los únicos puntos de interés, por el momento, que se tendrán son los que se acaban de mencionar.

6.5 ¿Qué hacemos con los archivos?

Una vez que se obtuvieron todos los archivos, estos son transmitidos a la Palm para que este los guarde dentro del mismo dispositivo y pueda trabajar con ellos.

Con estos archivos obtenidos, el sistema GMAP puede trabajar de acuerdo a lo planeado en el principio de este trabajo.

Debemos tomar en cuenta que en este trabajo solamente se adquirió el mapa de la delegación Cuauhtemoc de la fuente de datos INEGI. Y solamente se trabajo con una parte del mismo.

Conclusiones

Conclusiones

En definitiva, bastaría con tener los recursos para poder alcanzar las metas que uno desea.

A lo largo de este trabajo de tesis nos hemos topado, en el ámbito del desarrollo del sistema GMAP, con diferentes detalles, como los impedimentos que nos provee los dispositivos Palm respecto a sus características físicas, el dibujado de mapas, el tamaño de los mismos, los colores que debemos manejar, las formas y contenidos de los datos del mapa. En fin, muchos han sido dichos detalles, que se han ido solucionando conforme se presentaron.

Y, al darle solución a cada uno de estos problemas con que nos topamos hemos ido dando paso a la realización de la aplicación GMAP, que conformaba el principal objetivo de este trabajo de tesis.

Tomando en cuenta que faltaron algunos detalles por resolver, podemos concluir en que los objetivos planteados en un principio en este trabajo de tesis se han cumplido.

Se han sentado las bases para mejorar las aplicaciones que existen en otros países referentes a mapas para este tipo de dispositivos. Con el sistema GMAP se ha logrado realizar una aplicación cuya funcionalidad es extensible, tanto como las características físicas del mismo dispositivo nos lo permitan. Esto favorece a la escalabilidad del sistema, con la finalidad de que en un futuro se pueda complementar este sistema con otras funcionalidades dentro del mismo.

Lo anterior se logró a través de la proposición del sistema de capas que contiene GMAP en su interior para el control de todos y cada uno de los datos que el mismo sistema debe tener.

En cuanto a la solución de los diversos problemas que se nos presentaron en el camino podemos listar los siguientes:

- Mapas en formato DXF y DWG. Los mapas adquiridos se nos proporcionaron en el formato DXF y DWG, formatos utilizados por Autocad. Para poder leerlos y obtener los datos necesarios se construyó una librería en java con la capacidad de lectura y extracción de los datos de dichos archivos. De esta manera se pudo obtener cada una de las cuadras marcadas en el mapa para los fines de la aplicación.
- Pintado de los mapas dentro de la pantalla del dispositivo Palm. Los mapas, al ser muy grandes para ser pintados en su totalidad dentro de la pantalla del dispositivo, tuvieron que ser divididos en secciones cuadradas con la finalidad de poder desplegar dichas secciones, por separado, dentro de la pantalla del

dispositivo. Con esto, el mapa es legible ante los ojos del ser humano cuando se despliega dentro de la pantalla.

- **Compresión de datos.** Otro de los problemas que se nos presentó fue que el mapa de la delegación Cuauhtemoc, mapa que se utilizó en esta tesis, medía cerca de 3.5 mb. Este tamaño impedía que pudiera entrar con facilidad a nuestra Palm, ya que si se usaba este archivo se iban a tener dos problemas: no se dejaría espacio para los siguientes mapas y el procesamiento de dichos archivos sería muy lento e ineficiente. Este problema se solucionó extrayendo únicamente los datos que eran necesarios para el sistema GMAP que el archivo del mapa nos proporcionaba. De esta manera, el espacio se logró reducir considerablemente.
- **Características de hardware y procesamiento de los datos.** Dado que las características de hardware de nuestro dispositivo son limitadas en comparación de la computadora donde se desarrolló el sistema, esto nos orilló a que el procesamiento dentro del dispositivo de tipo Palm fuera el mínimo, de tal manera que fuera lo menos tardado posible. Esto implicó que el cálculo de la división del mapa en pequeñas secciones lo realiza la computadora de escritorio, y los datos obtenidos, que son de tamaño pequeño, los lee el dispositivo para ser dibujados dentro del sistema. De esta manera, al dejar el procesamiento fuerte del lado de la computadora de escritorio, a la Palm solamente le queda el leer los datos procesados previamente para desplegarlos en la pantalla.
- **Envío de datos al dispositivo.** Dado que la Palm con la que se probó el sistema cuenta con red inalámbrica, fue por este medio por el cual se le transmitieron los datos necesarios para el buen funcionamiento del sistema. A saber, los mapas seccionados, los nombres de las calles, los puntos de interés, y otros. Con esto se pudo resolver la transmisión de datos al dispositivo de tipo Palm. Vale la pena aclarar que la Palm con la que se probó tenía esta característica integrada (la de red inalámbrica), por lo que no se tuvo la necesidad de invertir más en dicho dispositivo. Habrá dispositivos que no cuenten con dicha característica, y para estos dispositivos se tendrá que buscar la manera de transmitir los datos a la Palm.
- **Localización geográfica.** Para resolver el problema de la localización geográfica dentro del sistema GMAP, se tuvo que adquirir un dispositivo SPG. El dispositivo tiene las siguientes características:

Marca: HOLUX
Modelo: GM-210

Este dispositivo nos permitió obtener la posición exacta en la que se encuentra la Palm dentro del globo terráqueo, con lo cual se pudo cotejar con el mapa obtenido en el INEGI

- Puntos de interés. En cuanto a los puntos de interés, dentro de los mapas obtenidos en el INEGI, vienen marcados algunos de los más importantes de estos. La librería construida para la lectura de los archivos DXF y DWG nos permitió extraer dichos datos para que fueran presentados dentro del sistema.

Como podemos observar fueron muchos los problemas que se presentaron a lo largo de la realización del sistema en cuestión. A todos se les dio cause y solución de una u otra manera.

Con esto, podemos concluir que el principal objetivo de esta tesis se cumplió, a saber:

Sentar las bases para mejorar dicha aplicación, así como adaptarla a la ciudad de México. Es decir, lograr desarrollar una aplicación que contenga las características básicas que se mencionan arriba como manejo de mapas, puntos de interés, localización geográfica y nombres de calles. Todo esto dentro de un dispositivo de tipo Palm.

Y, en cuanto a los objetivos secundarios, aquellos que implicaban dar solución a los detalles y problemas que se irían presentando a lo largo de la realización de este sistema también se cumplieron, ya que se dio solución a todos los problemas que se presentaron.

Pensar que este es un sistema no expandible sería una locura, por lo que podemos mencionar que muchas partes que podría tener este sistema quedaron fuera del alcance de este trabajo, entre las cuales podemos mencionar las siguientes:

- Tráfico en las aristas. El peso de las aristas se puede manejar de manera proporcional al tráfico que haya en la calle que representa a la arista. Esto implica tener la infraestructura para ubicar el tráfico que hay en cada calle de la ciudad para después procesar dicho tráfico y obtener el cálculo del peso de la arista.
- Semáforos. El manejo de semáforos dentro de las calles de la ciudad puede ser un aditamento muy útil a este sistema, ya que nos permitiría ver si el estado en que se encuentra algún semáforo delante de nosotros, y así prevenir la velocidad de nuestro avance.
- Otras señalizaciones. Señalizaciones como topes, estado de las calles, tiempo en que se tarda el carro en recorrer cada calle de la ciudad, puentes peatonales, entre otras muchas.

Vemos que son muchas los aditamentos que se le pueden agregar a este sistema, pero por cuestiones de tiempo no las hemos agregado al mismo, dejando así abierta la posibilidad de continuar con este trabajo en un futuro con otro trabajo de tesis.

Bibliografía

Bibliografía

Autodesk, Inc (2001). DXF Reference Guide.

Daryl Wilding-McBride (2003). Java Development on PDAs, Building Applications for PocketPC and Palm Devices. Addison-Wesley

John W. Muchow (2002). Core J2ME Technology & MIDP. Sun Microsystems

Jonathan Knudsen (2002). Wireless Java Developing with Java 2 Micro Edition. Apress.

Kim Topley (2002). J2ME in a Nutshell, a desktop quick reference. O'reilly

Qusay H. Mahmoud (2002). Learning Wireless Java. O'reilly

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. Introduction to Algorithms.
Pag: 463-630.

Yu Feng, Dr. Jun Zhu (2001). Wireless Java Programming with J2ME. SAMS

1. <http://www.elgps.com/documentos/comofuncionagps/comofuncionagps.html>
2. <http://www.elgps.com/documentos.html>
3. <http://www.dlacuadra.com/gps.htm>
4. <http://www.alsitel.com/tecnico/gps/historia.htm>
5. <http://www.clubdelamar.org/sistemagps.htm>
6. <http://personales.upv.es/~arodrigu/grafos/Dijkstra.htm>
7. <http://www.alumnos.unican.es/uc900/Algoritmo.htm>
8. <http://www.it.uc3m.es/~pablo/asignaturas/rysc1/alumnos/04-EncEjerciciosSolucion.pdf>
9. <http://www.w3.org/Graphics/SVG/>
10. <http://biblioweb.dgsca.unam.mx/bd/dig/imagenes.html>
11. <http://www.tel.uva.es/personales/jpozdom/telecomunicaciones/tutorial/apartado33.html>

12. http://www.esri.com/software/busmap/busmap_travel/about/maps.html
13. http://www.livtech.co.uk/articles/GPS/gps_article_3.htm
14. <http://telecom.iespana.es/telecom/gps/>
15. <http://homepages.mty.itesm.mx/al584299/mypaper.htm>
16. <http://www.acarioja.com/digital2000/digital200004.html>
17. <http://www.nautigalia.com/otrostemas/articulos.php4?id=2&pag=3>