



UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ARAGON"

**PROCESOS DE DIBUJO
PARA EL TRAZO DE CARRETERAS
CON AUTOCAD**

TESIS PROFESIONAL
Que para obtener el título de:
INGENIERO CIVIL

Presenta:
ANTONIO AGUILAR MALAGON

San Juan de Aragón, 1996.

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



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.



UNIVERSIDAD NACIONAL
AUTÓNOMA
DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN
DIRECCIÓN

ANTONIO AGUILAR MALAGON
PRESENTE.

En contestación a su solicitud de fecha 10 de noviembre del año en curso, relativa a la autorización que se le debe conceder para que el señor profesor, M en I DANIEL VELAZQUEZ VAZQUEZ, pueda dirigirse el trabajo de Tesis denominado "PROCESOS DE DIBUJO PARA EL TRAZO DE CARRETERAS CON AUTOCAD", con fundamento en el punto 6 y siguientes, del Reglamento para Exámenes Profesionales en esta Escuela, y toda vez que la documentación presentada por usted reúne los requisitos que establece el precitado Reglamento; me permito comunicarle que ha sido aprobada su solicitud.

Aprovecho la ocasión para reiterarle mi distinguida consideración.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPÍRITU"
San Juan de Aragón, México., 15 de noviembre de 1995
EL DIRECTOR

M en I CLAUDIO C. MERRIFIELD CASTRO

c c p Jefe de la Unidad Académica.
c c p Jefatura de Carrera de Ingeniería Civil.
c c p Asesor de Tesis.

CCMC/AIR/unac.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
CAMPUS ARAGÓN

UNIDAD ACADÉMICA

Me enl **DANIEL VELÁZQUEZ VÁZQUEZ**
Jefe de la Carrera de Ingeniería Civil,
Presente.

En atención a la solicitud de fecha 23 de febrero del año en curso, por la que se comunica que el alumno **ANTONIO AGUILAR MALAGÓN**, de la carrera de Ingeniería Civil, ha concluido su trabajo de investigación titulado **"PROCESOS DE DIBUJO PARA EL TRAZO DE CARRETERAS CON AUTOCAD"**, y como el mismo ha sido revisado y aprobado por usted, se autoriza su impresión, así como la iniciación de los trámites correspondientes para la celebración del Examen Profesional.

Sin otro particular, le reitero las seguridades de mi distinguida consideración.

ATENTAMENTE
"POR MI RAZA HABLARA EL ESPÍRITU"
San Juan de Aragón, Méxco., 28 de febrero de 1998.
EL JEFE DE LA UNIDAD


LIC. ALBERTO BARRERA ROBAS

c.c.p. Asesor de Tesis.
c.c.p. Interesado.


AIR/ta.

El Ser Excelente

El ser excelente es hacer las cosas, no buscar razones para demostrar que no se pueden hacer.

Ser excelente es comprender que la vida no es algo que se nos da hecha, sino que tenemos que producir las oportunidades para alcanzar el éxito.

Ser excelente es comprender que en base a una férrea disciplina, es factible forjar un carácter de triunfadores.

Ser excelente es trazarse un plan y lograr los objetivos deseados a pesar de todas las circunstancias.

Ser excelente es saber decir: Me equivoqué y proponerse no cometer el mismo error.

Ser excelente es levantarse cada vez que se fracasa, con un espíritu de aprendizaje y superación.

Ser excelente es reclamarse a sí mismo el desarrollo pleno de nuestras potencialidades buscando incansablemente la realización.

Ser excelente es entender que a través del privilegio diario de nuestro trabajo podemos alcanzar la realización.

Ser excelente es ser creador de algo: un sistema, un puesto, una empresa, un hogar o una vida.

Ser excelente es ejercer nuestra libertad y ser responsables de cada una de nuestras acciones.

Ser excelente es sentirse ofendido y lanzarse a la acción en contra de la pobreza, la calumnia y la injusticia.

Ser excelente es levantar los ojos de la tierra, elevar el espíritu y soñar con lograr lo imposible.

Ser excelente es trascender a nuestro tiempo legando a las futuras generaciones un mundo mejor.

Ser líder de excelencia de esta talla son los que necesita el mundo y los reclama Dios.

Miguel Ángel Cornejo

Contenido

INTRODUCCION.....	1
CAPITULO I. AUTOCAD COMO HERRAMIENTA DE DISEÑO.....	5
1.1 Características generales de AutoCAD.....	5
1.2 Unidades, escalas y tamaños de papel.....	8
1.3 Láminas de dibujo.....	9
1.4 Integridad del dibujo desplegado.....	11
1.5 Dibujos prototipos.....	12
1.6 Conceptos de gráficas tridimensionales.....	13
CAPITULO II. TIPOS DE SISTEMAS DE COORDENADAS.....	16
2.1 Tipos de Sistemas de Coordenadas.....	16
2.2 Transformación de Coordenadas.....	17
2.3 Sistema de Coordenadas en AutoCAD.....	20
2.4 Sistema de Coordenadas de Usuario.....	22
2.5 Icono del Sistema de Coordenadas.....	23
CAPITULO III. DIGITALIZACION.....	24
3.1 Conceptos Básicos.....	24
3.2 Area Activa.....	25
3.3 Proximidad.....	25
3.4 Resolución.....	25
CAPITULO IV. COMANDOS RELACIONADOS DE AUTOCAD.....	28
4.1 COPY.....	28
4.2 CHPROP.....	30
4.3 DOUGHNUT.....	30
4.4 DXFIN.....	32
4.5 DXFOUT.....	33
4.6 ERASE.....	33
4.7 FILLET.....	34
4.8 LAYER.....	35
4.9 MOVE.....	35
4.10 OFFSET.....	37
4.11 PAN.....	40
4.12 PEDIT.....	40
4.13 PLINE.....	42
4.14 REDRAW.....	46
4.15 REGEN.....	46
4.16 TRIM.....	47
4.17 VIEW.....	49
4.18 ZOOM.....	50
CAPITULO V. CURVAS HORIZONTALES SIMPLES.....	52
5.1 Definición de elementos.....	52
5.2 Trazo de las curvas en el terreno.....	56

5.3 Trazo de las curvas en AutoCAD.....	60
CAPITULO VI. DISEÑO DEL PROGRAMA VIAS.C.....	64
6.1 Descripción.....	64
6.2 Diagrama de flujo.....	68
6.3 Listado 1.....	69
6.4 Listado 2.....	74
CAPITULO VII. IMPRESION EN FOTOCOMPOSICION.....	77
7.1 Fotocomposición.....	77
7.2 El lenguaje PostScript.....	77
7.3 Características del lenguaje PostScript.....	78
7.4 Evolución del lenguaje PostScript.....	81
7.5 Un vistazo al PostScript Nivel 2.....	83
CAPITULO VIII. APLICACION.....	87
8.1 Procedimiento.....	87
8.2 Paso 1. Definición de Láminas.....	88
8.3 Paso 2. Digitalización de Caminos.....	89
8.4 Paso 3. Dar Calidad a los Trazos.....	91
8.5 Paso 4. Colocación de Símbolos.....	92
8.6 Paso 5. Edición.....	93
8.7 Paso 6. Rotulación.....	94
8.8 Paso 7. Impresión.....	95
CONCLUSIONES.....	96
BIBLIOGRAFIA.....	97

Reconocimientos

Agradezco a DIOS por haberme concedido la gracia de concluir satisfactoriamente este trabajo, por brindarme los dones que me han permitido seguir adelante ante todas las adversidades y que sabré perfeccionar para hacer el bien.

Especialmente agradezco a mi pequeña pero amorosamente unida familia, formada por mi esposa Silvia Vizuet Vizuet y nuestro hijo Irving Aguilar Vizuet los cuales son el motivo de mi constante búsqueda de la superación, gracias por su confianza, por su apoyo, por su amor, por creer en mí.

Agradezco a mis padres Francisco Aguilar Rico y Amelia Malagón Lara, por que han sabido darnos una base firme de valores héticos, morales y religiosos, por consagrar su vida a la formación y bienestar de sus hijos.

Gracias a mis hermanos Francisco, Armando, Yolanda, Maricela y Gerardo por ser los amigos de toda la vida y por que siempre me apoyaron en mi formación profesional.

A todos mis amigos les estoy muy agradecido por la sinceridad de su amistad, por los que se quedan en los momentos difíciles.

Agradezco a mi asesor de tesis Daniel Velázquez Vázquez por la amistad que nos une y por su apoyo y buena disposición en la realización de este trabajo.

Gracias a nuestra amiga Ma. del Carmen Oliva Miranda por su valiosa y especial contribución.

Reconocimiento de Marcas Registradas

- *MicroSoft, es marca registrada de MicroSoft Corporation.*
- *Borland C++, es marca registrada de Borland International, Inc.*
- *PostScript, es una marca registrada de Adobe Systems Incorporated.*
- *AutoCAD, es una marca registrada de Autodesk, Inc.*

INTRODUCCION

Desde los años sesenta cuando los primeros pasos de dibujo asistido por computadora fueron dados, muchos libros han sido escritos alrededor del tema. La mayoría de esos libros tienden a describir los componentes y el potencial del dibujo asistido por computadora, siempre hablan de los periféricos, de las impresoras, graficadores, monitores y de algunas de las aplicaciones a las que se deriva el dibujo computacional.

El CAD, las siglas en Inglés de Computer Aided Design = Diseño Asistido por Computadora, es una técnica que tiene el potencial de mejorar la calidad, el flujo y el uso del producto realizado por las Ingenierías, y las materias relacionadas al dibujo y al diseño; y sobre todo el resultado, se ve reflejado en la productividad, reducción en los costos y en los ciclos del proceso de diseño.

Los sistemas CAD se utilizan en más de 60 países, las bases instaladas de estaciones de CAD alrededor del mundo son aproximadamente de quinientas mil, el número de usuarios varía de 2 a 1 y/o 3 a 1 por estación, por lo tanto, hay más de un millón de usuarios alrededor del CAD.

Los componentes básicos de un equipo de CAD son: la computadora, el monitor, el digitalizador, la impresora, el graficador y el programa. El sistema operativo y las posibles aplicaciones que acompañan al CAD como lo pueden ser un paquete Integrado de Ingeniería, o de perspectiva de modelaje, etc., existirán también algunos otros periféricos, y medios contenedores (discos duros, ópticos, discos compactos y otros medios) del resultado generado en estos paquetes.

Muchas compañías han adquirido sistemas CAD, pues ellos piensan que pueden optimizar sus tiempos como una consecuencia de esa adquisición, en el orden de 500% a un 5000% (dependerá del área y

departamento donde sea usado, así como el equipo adquirido). Algunos otros comentan que no vieron ninguna alza en la productividad hasta después de tres años de adquirido el equipo.

Las empresas han comprendido que al adquirir equipo incrementan su productividad, sin embargo, sin una estrategia clara, esta productividad en el consenso general de la empresa es muy pequeña, por lo tanto, el éxito del CAD no radica en la compra del sistema, sino en el manejo organizacional y el diseño de las estrategias de adquisición de equipos y herramientas adecuadas a la empresa.

La importancia de elegir el equipo de cómputo adecuado, va a depender en gran medida de las necesidades del software que se elija para la producción, en este caso, del software de CAD que se piense utilizar. La elección del paquete de CAD, debe hacerse con un respaldo de un análisis de las actividades principales de la empresa, tener claro los objetivos de la misma y conocer a grandes rasgos las posibilidades del paquete que se proponga.

Para evaluar el programa se pueden emplear las siguientes preguntas:

- ¿Existen Personas capacitadas que puedan enseñar y dar cursos?
- ¿Es fácil de aprender?
- ¿Es fácil de usar?
- ¿El sistema es tecnología de punta?
- ¿Tiene soporte técnico?
- ¿Existe distribuidor en México?
- ¿Tiene la suficiente precisión?
- ¿Corre en red?
- ¿Se comunica con otros paquetes y/o aplicaciones?
- ¿Es un programa estático o crece continuamente (lo mejoran)?
- ¿Existen Consultores y Asesores para el programa evaluado?
- ¿Existen revistas informativas de dicho programa?
- ¿Existen libros fáciles de conseguir de ese programa?
- ¿Utiliza sistema métrico decimal?

Se puede cuestionar la flexibilidad de la siguiente forma:

- ¿Es fácil de personalizar?

- ¿Le permite modificarlo a sus necesidades?
- ¿Existen aplicaciones para ese paquete?

Recordando que el programa es el que exige el equipo y no al revés como mucha gente piensa, los programas tienen diferentes requerimientos, por ejemplo un programa de contabilidad tiene diferentes necesidades de equipo a las de un programa de dibujo, el segundo requerirá más velocidad y capacidad. Para evaluar las plataformas existentes para un sistema CAD, se pueden sugerir las siguientes preguntas:

- ¿Bajo qué sistemas operativos corre el programa?
- ¿Cuáles plataformas ofrece el mercado mexicano y que tengan el soporte técnico que se necesita?
- ¿Corre en red?
- ¿Tiene conectividad?
- ¿Tiene portabilidad?
- ¿Se comunica entre diferentes plataformas?
- ¿Es compatible con otros sistemas CAD, con cuáles?

Junto con los anteriores cuestionamientos se pueden incluir los de la empresa que produce el software que se piensa adquirir, con las siguientes:

- ¿Es una empresa sólida?
- ¿Realiza investigación y desarrollo?
- ¿Es vanguardista?
- ¿Cuántas copias de ese programa ha vendido?

Si tomamos como ejemplo una empresa que tiene como objetivo principal generar planos cartográficos, y que cuyas reflexiones de las consideraciones anteriores la llevaron a adquirir lo siguiente:

- Como plataforma de desarrollo
AutoCAD

- Como equipo de cómputo
Computadora 486 DX2 66Mhz
8 MB en RAM

540 MB en disco duro
Sistema de video Super VGA color

- Como sistema de impresión
Fotocomponedora

Podríamos pensar que se tiene el material suficiente, como equipo y programas para el rediseño de los procesos que involucran la creación de dibujos. Con esta infraestructura, es posible optimizar tiempos de procesamiento, mejorar la calidad del trabajo, perfeccionar la presentación de los productos, además del cambio importante de la visión general de lo que se puede hacer con este equipo, tales aspectos que antes no se podrían estudiar por las limitantes que se presentan al producir de una forma casi artesanal.

Todas estas herramientas no pueden lograr las metas fijadas por sí solas, se necesita de un proceso bien planeado, estudiado y documentado para su aplicación.

El objetivo de esta tesis es, precisamente, presentar una propuesta de un proceso de dibujo para el trazo de carreteras, con las herramientas adquiridas en el ejemplo anterior.

Está estructurada de tal forma que se pueda dar el seguimiento adecuado, para que se contemplen todas las consideraciones y se tengan todas las herramientas, desde la concepción del diseño por computadora, hasta el proceso de dibujo mismo.

CAPITULO I.

AutoCAD COMO HERRAMIENTA DE DISEÑO

1.1 Características generales de AutoCAD

AutoCAD es un programa de aplicación de dibujo asistido por computadora de propósito general, diseñado para ser usado por un usuario simple, en una computadora personal de escritorio y como una estación de trabajo gráfica. Este fue desarrollado cerca del año 1980 por Autodesk Inc., en Sausalito, California, y actualmente es distribuido en Australia por Autodesk Australia, localizado en Sidney.

AutoCAD es un sistema de dibujo interactivo diseñado para permitir al usuario construir o editar un dibujo en una pantalla de dispositivo gráfico. Esto es análogo a un programa de procesamiento de palabras, a excepción de que en este caso las cosas que se van a procesar son dibujos. Cada dibujo es almacenado en un archivo del disco, y AutoCAD solamente está disponible para editar un dibujo a la vez. La similitud de un procesador de palabras es reflejado en el hecho de que la función principal del componente de AutoCAD es conocido como editor de dibujos.

Hasta la versión 10, AutoCAD fue esencialmente un sistema de dibujo de dos dimensiones. Las siguientes versiones soportan dimensiones tridimensionales. Esto ha tenido el efecto en la característica de AutoCAD que al soportar dibujos de dos dimensiones tengan el mismo contenido y puedan ser usados como un sistema de dos dimensiones sin tomar en cuenta las características tridimensionales.

Un dibujo de AutoCAD está hecho de entidades. Estas pueden ser ya sea primitivas gráficas simples (tales como líneas, arcos, círculos, texto, etc.) o bloques (grupos de entidades).

Las primitivas gráficas son definidas geoméricamente en términos del sistema normal de coordenadas cartesianas (en el eje de las X los valores positivos son a la derecha, en el eje de las Y los valores positivos son hacia arriba, en el eje de las Z los valores positivos son hacia el usuario). Por lo tanto, las líneas son definidas por su coordenada del punto final (x,y,z), mientras los círculos son definidos por las coordenadas del centro y el radio. Cada entidad también tiene ciertos atributos asociados con ésta, tales como estilos de líneas, tipos de letras y color.

Un bloque es un grupo de entidades que pueden ser manipuladas como una unidad simple. Una vez creado, un bloque puede ser movido, escalado, rotado, copiado o borrado. Un bloque puede ser creado por la unión de otros grupos de entidades del mismo dibujo (del dibujo que está siendo editado) y asignando un nombre simple a tal grupo. Alternativamente, un dibujo ya creado (almacenado en el disco) puede ser insertado dentro del dibujo actual como si fuera un bloque. Igualmente, un bloque del dibujo actual puede ser escrito o guardado a un archivo como un dibujo independiente.

Es importante entender que un bloque, en AutoCAD, es único para un dibujo específico. Esto es, cuando un dibujo ya existente, es insertado dentro del dibujo actual como un bloque, AutoCAD simplemente copia las gráficas de tal dibujo y lo incluye como un objeto simple. El dibujo almacenado no se afecta por la acción. En efecto, si el dibujo ya existente (del cual los bloques fueron creados) fuera subsiguientemente editado entonces estos cambios no podrían afectar los bloques insertados. Es posible, sin embargo, actualizar una definición de bloque en un dibujo, AutoCAD redefine el bloque usando la versión actual del dibujo original.

Como una alternativa para insertar un dibujo en otro dibujo como un bloque, AutoCAD ahora proporciona la facilidad para asignar un dibujo al

dibujo actual como una referencia externa. Un dibujo externo es tratado como un bloque, exceptuando que su actualización es automática cada vez que el dibujo en el cual está asignado es cargado con naturalidad, si el dibujo externo es alterado en un tiempo significativo, entonces la referencia externa podría ser actualizado para reflejar estos cambios.

Otra característica importante de un bloque (incluyendo referencias externas de un bloque) es que éste puede ser duplicado muchas veces dentro del mismo dibujo, en cada copia se apunta a una descripción gráfica, pero teniendo su posición, escala y factor de rotación único. Esto es conocido como instancia. Hay dos ventajas que se pueden aprovechar:

- **Salvar espacio** - La descripción gráfica es almacenada una sola vez.
- **Facilidad de redefinición** - La descripción gráfica de un bloque puede ser alterado y redefinido en orden para actualizar cada instancia en el dibujo.

Es posible asociar atributos nombrados como un bloque. Por ejemplo, un bloque representando una puerta puede tener atributos tales como tipo, manufactura, acabados, tipo de chapa, etc. Cada instancia del bloque podría tener un valor asociado por cada definición de atributo. AutoCAD proporciona facilidades para planear los datos de atributos asociados con bloques insertados en un dibujo.

El concepto final para explicar un bloque es la noción de desagruparse. Cualquier instancia de un bloque puede ser desagrupado en orden para reducir su regreso a sus entidades gráficas separadas. Similarmente, una referencia externa de bloque puede ser agrupado para que sea un bloque local y entonces, en cambio, pueda ser desagrupado.

1.2 Unidades, escalas y tamaños de papel

Una comprensión clara de la forma en que AutoCAD maneja unidades de medida, escala y tamaños de papel es necesario antes de que un dibujo pueda ser creado o editado. AutoCAD proporciona un control completo sobre estas cosas por distinción entre espacios modelo y espacios de papel. Todos los dibujos, ya sea de dos dimensiones o tridimensionales deben ser diseñados como un modelo de una entidad de un mundo real (sea un dibujo o una descripción tridimensional de un objeto de un mundo real). En tal caso, la descripción es mantenida en un espacio modelo. El espacio de papel puede ser concebido como un modelo de una pieza de tamaño estándar de papel en el cual se pueden establecer vistas (regiones rectangulares) en donde AutoCAD podría desplegar vistas a escala del dibujo que se haya creado en el espacio modelo. Se pueden adicionar líneas de trabajo en el espacio de papel para formar bordes y anotaciones.

En orden para comprender esto, es mejor dibujar un espacio modelo como una pieza muy larga de papel localizada en algún lado en un espacio de dos dimensiones. (AutoCAD actualmente puede dibujar en cualquier lado en espacios tridimensionales). Desde el tamaño y la locación de tal pieza de papel que está al criterio del dibujante se puede reflexionar para escoger una cosa que haga el dibujo tan simple como posible. Hoy en día, los Ingenieros civiles Australlanos usan milímetros como sus unidades de medida y podrían dibujar objetos en su tamaño real. Esto quiere decir que cuando dibujan un plano de una construcción que es de 15 metros cuadrados, el tamaño del papel (en un espacio modelo) se puede fijar a algo mayor que 15,000 x 15,000 unidades (donde las unidades son igual a milímetros). Estrictamente hablando, no es necesario fijar un espacio modelo del tamaño del dibujo, pero hacerlo ayuda en orden para establecer un contexto para el dibujo.

Cuando se trabaja en el dibujo en un espacio modelo, se pueden imaginar las pantallas gráficas como una ventana a través de la cual se puede ver

todo o parte del mismo dibujo. Así, cuando se está viendo un dibujo, una escala larga de despliegue podría permitir ver solamente parte del dibujo en la pantalla. Esto está referenciado como un acercamiento en el dibujo, tanto como se incrementa la escala, se podrá ver menos dibujo, pero lo que se pueda ver podría ser a gran detalle. Si se aleja el dibujo se podría ver más dibujo, pero en menor detalle. AutoCAD proporciona unas herramientas muy sofisticadas para hacer acercamientos y alejamientos del dibujo y para mover en diferentes direcciones el área que se está viendo.

A pesar de que es posible imprimir desde un espacio modelo, es normal definir una hoja en el espacio papel y crear una o más ventanas mostrando diferentes vistas del dibujo en el espacio modelo. Cada vista es desplegada a una escala fija, y el dibujo en el espacio papel es trazado a una escala 1:1. En este caso, los textos y otras anotaciones en el dibujo es mejor adicionarlas en el espacio papel, y trazados a su tamaño actual. Cualquiera texto que es adicionado al modelo espacio tiene que ser dibujado a un tamaño artificial largo, tanto, que cuando sea escalado para trazarlo, éste finalice en el tamaño requerido. Por ejemplo, textos en un dibujo a escala 1:100 puede fijarse a 250 milímetros de altura en un espacio modelo para crear textos trazados que son de 2.5 mm de altura en el trazo.

1.3 Láminas de dibujo

AutoCAD también soporta la notación estándar de laminación. Esto quiere decir que cualquier entidad de dibujo puede ser asignada a cualquier lámina de dibujo. Esto permite la separación de porciones de dibujo. Por ejemplo, todas las columnas pueden ser asignadas a una lámina, todo el trabajo eléctrico puede ser asignado a otra, y así sucesivamente. De esta forma, cuando se traza, solamente se seleccionan las láminas necesarias para incluir en el dibujo físico que es producto del dibujo realizado en AutoCAD.

Las láminas también pueden utilizarse para tomar ventaja cuando se está editando un dibujo. Por ejemplo, suponiendo que un dibujo de un plano es substancialmente completo con solamente el trabajo de las paredes para hacer. Si todo el trabajo de las paredes está en una lámina, todas las demás láminas pueden ser apagadas para que desaparezcan momentáneamente y únicamente el trabajo de las paredes pueda ser visible y seleccionable. Esto hace posible que el trabajo que se realice no afecte los demás trazos o alguna otra cosa.

El laminado en AutoCAD es manipulado a través de propiedades que pueden ser asignadas a otra lámina. Una vez que la interacción entre las propiedades de la lámina es comprendida, entonces éstas pueden ser manipuladas para tomar una gran ventaja. A continuación se describen algunas propiedades de las láminas.

Nombre de la lámina.

Un número ilimitado de láminas puede ser creado. Cada una es identificada por un nombre único asignado por el usuario. El nombre puede ser de hasta 31 caracteres de longitud, formado por letras mayúsculas, dígitos y tres caracteres especiales (\$ - _).

Lámina en uso.

AutoCAD mantiene la noción de una lámina en uso, siendo la lámina a través de la cual todas las entidades nuevas creadas son colocadas. Cualquiera de las láminas existentes puede ser seleccionada como en uso en cualquier momento.

Visibilidad.

Cada lámina puede ser turnada ya sea visible (ON) o invisible (OFF). Solamente las láminas visibles pueden ser desplegadas o trazadas. Las láminas pueden prenderse o apagarse en cualquier momento, en cualquier combinación. Las entidades pueden seguir adicionándose a una lámina invisible (inclusive es posible tener una lámina en uso que es invisible), pero las entidades que no son visibles no pueden ser seleccionadas para edición.

Color y tipo de línea.

Cada lámina está asociada con un color y tipo de línea. Así como las entidades se van dibujando, a ellas le son asignadas, por omisión, el color y el estilo de línea de la lámina en uso. (Esta asignación se puede cambiar asignando a cada entidad su propio color y su estilo de línea). Por omisión, las láminas nuevas son asignadas al color 7 (blanco) y estilo de línea continuo.

Estado Congelado/Descongelado.

En cualquier momento el usuario puede congelar una o más láminas del dibujo. Una lámina congelada se hace invisible. Cualquier entidad en una lámina congelada es completamente ignorada u olvidada por AutoCAD durante el proceso de edición. Naturalmente, una lámina congelada puede ser descongelada en cualquier momento.

La discusión anterior se refiere a la manipulación de láminas en general, ya sea en un espacio modelo o en un espacio en papel. AutoCAD también permite manipular las láminas (prenderlas, apagarlas, congelarlas) independientemente en cada ventana en espacio papel. Esto quiere decir que una lámina puede ser visible en una ventana, pero no en otra. AutoCAD también te permite manipular las láminas que están contenidas dentro de referencias externas a un dibujo.

1.4 Integridad del dibujo desplegado.

Claramente, con cualquier sistema de dibujo, es importante que las imágenes desplegadas en la pantalla por el usuario sean una representación real del dibujo que está guardado por computadora. Al mismo tiempo, es igualmente importante que el sistema responda rápidamente al usuario. En orden para conseguir un balance fino entre la velocidad de operación y el mantenimiento de la integridad del dibujo. AutoCAD configura una base de datos de despliegue, separada de la

base de datos completa del dibujo. La base de datos del despliegue permite una rápida interacción gráfica con la pantalla, pero necesita ser regenerado de vez en cuando en orden para mantener la integridad. Por lo tanto, en AutoCAD se distingue entre redibujar la pantalla y regenerar el dibujo.

Redibujar la pantalla es más rápido (pero menos perfecto) que regenerarla y AutoCAD podría ofrecer hacerlo automáticamente después, solo cuando sea realmente necesario. Se podría encontrar que se desea redibujar la pantalla con frecuencia mientras se está usando AutoCAD, mientras, por otro lado, se podría rara vez necesitar forzar a AutoCAD a regenerar. En estas ocasiones cuando se ofrece, generalmente se puede regenerar automáticamente sin interferencia. Es posible deshabilitar completamente la regeneración automática, pero esto debe ser hecho con precaución, como las restricciones severas de la imagen desplegada.

En la búsqueda de la velocidad contra la integridad, hay otras tres facilidades en AutoCAD que pueden ser usadas para conservar un balance entre velocidad de interacción y la calidad de la imagen desplegada. Estas son cosas que normalmente no se podrían necesitar a menos que el dibujo sea particularmente complejo. Los textos pueden ser representados en la pantalla como cajas rectangulares en lugar de la forma completa de las letras (esto es referenciado como un texto rápido). Círculos y arcos pueden ser desplegados como una serie de segmentos de líneas en lugar de curvas bien definidas. Áreas de rellenos sólidos y líneas gruesas pueden ser mostradas como delineadas solamente.

1.5 Dibujos prototipos.

Como es muy discutido, hay muchas composiciones en AutoCAD, las cuales forman una parte integral de cualquier dibujo para el cual ellas se

aplican. Composiciones tales como nombre de láminas y atributos asociados, escalas, tamaños de dibujo y parámetros de despliegue pueden ser almacenados a un archivo como un dibujo, inclusive si no existen gráficas en tal dibujo.

Esto permite tener dibujos prototipos los cuales pueden ser usados como plantillas en la producción de otros dibujos. Este es un concepto que permite establecer estándares en una oficina los cuales deben ser accesibles a todos los usuarios del sistema.

1.6 Conceptos de gráficas tridimensionales.

Todas las entidades gráficas en AutoCAD están completamente definidas dentro de un espacio tridimensional. Sin embargo, en casos normales, donde el usuario no hace intentos por definir los objetos en tercera dimensión, éstos son definidos con una base plana de $Z=0$ del sistema de coordenadas mundiales (WCS) definidas por AutoCAD. En tal caso, el valor de z de todos los puntos es cero y el usuario no tiene necesidad de enterarse de ello.

Para soportar usuarios de tres dimensiones, AutoCAD proporciona cinco formas básicas para introducir información tridimensional:

Primero. La mayoría de las entidades gráficas básicas (líneas, arcos y curvas) pueden tener un espesor asociado con ellos, los cuales tienen una altura efectiva en la dirección z , esto siempre ha sido una característica de AutoCAD y es algunas veces referenciada como dimensión (2.5).

Segundo. Cuando se especifica la posición de cualquier punto durante el proceso de dibujo, los tres valores de la coordenada pueden ser introducidos por el usuario, si solamente dos coordenadas son especificadas, el punto es asignado en una base plana -

haciendo notar que algunas entidades deben ser perfectamente planas, por lo tanto, si se intenta introducir un punto tridimensional que podría ser tal como un objeto no planar, entonces AutoCAD podría desplegar un mensaje de error.

Tercero. AutoCAD soporta un número específico de entidades gráficas en tercera dimensión, incluyendo polinomios tridimensionales, caras y mallas, otra vez de un lote de comandos que pueden ser usados para crear objetos tridimensionales con ellos.

Cuarto. Y más significativo, AutoCAD permite al dibujante hacer una configuración temporal de sistemas de coordenadas de usuario (UCS) el cual puede ser posicionado y orientado en cualquier lugar en el espacio (relativo al sistema de coordenadas universales). Así, se pueden pintar los ejes estándares x,y y z del sistema de coordenadas universales y después imaginar una configuración de ejes equivalentes (el UCS) siendo movido de cualquier forma del origen y entonces transformado y rotado en cualquier orientación que se desee, una vez que un UCS se establece, todos los dibujos en dos dimensiones son hechos relativos a estas coordenadas hasta que otro UCS sea definido o el usuario regrese al sistema de coordenadas universales.

Finalmente. De la versión 11 en adelante, AutoCAD ha provisionado un paquete de extensiones llamado AME - Advanced Modelling Extension (Extensión Avanzada de Modelación) el cual proporciona unas capacidades de modelación de sólidos en tercera dimensión, esto permite formar objetos sólidos complejos, por adición, sustracción e intersección de un lote de primitivas de sólidos simples, tales como, cajas, conos, cilindros, etc., por ejemplo, se puede formar una caja con un hoyo circular sustrayendo un cilindro de la caja, y así sucesivamente.

Se hace notar que las características tridimensionales de AutoCAD emergen de las tradicionales facilidades de dos dimensiones, por lo

Capítulo I. AutoCAD como herramienta de Diseño. 11

tanto, es posible pero no común usar AutoCAD como si éste no tuviera todas las facilidades tridimensionales.

CAPITULO II. TIPOS DE SISTEMAS DE COORDENADAS

2.1 Tipos de Sistemas de Coordenadas

Los sistemas de coordenadas se pueden distinguir, según la terminología GKS (Graphical Kernel System), en:

- *coordenadas universales*
- *coordenadas normalizadas*
- *coordenadas de aparato*

Las coordenadas universales sirven para definir el espacio que interesa para una aplicación en particular. Por ejemplo, un mapa de la región de México puede usar el espacio $(-50,-50)$ $(50,50)$ en kilómetros con origen en el Zócalo.

Las coordenadas normalizadas sirven para esconder los diferentes espacios de las mismas, es decir, que un punto se presentará en la misma posición independientemente de las coordenadas del aparato. Define un periférico virtual.

Las coordenadas de aparato describen un periférico particular. Por ejemplo, un graficador $(0,0)$ $(32767,32767)$ en milímetros o una pantalla de video $(0,0)$ $(1024,768)$ en pixeles.

Las coordenadas de aparato no corresponden necesariamente al espacio donde son descritas las entidades que tienen que desplegarse. Cada espacio de coordenadas de aparato describe un solo tipo de periférico; habría que modificar la aplicación si se utilizan diferentes tipos de periféricos. Por este motivo, se utilizan las coordenadas universales y las coordenadas normalizadas.

Cuando se dispone de los tres tipos de coordenadas, generalmente utilizará las coordenadas universales para especificar las primitivas de salida. Cuando se desea indicar coordenadas en la pantalla (con el ratón, por ejemplo), el dispositivo de entrada generará las coordenadas del dispositivo de salida.

Para estos cambios se requieren de ciertos procedimientos u operadores llamados transformaciones de coordenadas.

Para desplegar algo, el sistema debe transformar las coordenadas universales en coordenadas normalizadas por medio de la transformación de normalización, luego, debe transformar las coordenadas normalizadas en coordenadas de dispositivo por la transformación del dispositivo.

Para la indicación de coordenadas, se transforman las coordenadas del dispositivo, en coordenadas normalizadas, por medio de la transformación inversa del dispositivo, se transforman las coordenadas normalizadas, en coordenadas universales, por medio de la transformación de normalización inversa.

2.2 Transformación de Coordenadas

Supongamos que tenemos dos pantallas diferentes, una de 1024 por 1024 pixeles, y otra de 512 por 512 pixeles; un espacio de coordenadas

Capítulo II. Tipos de Sistemas de Coordenadas. ■

normalizadas de (0,0) a (1,1); un espacio de coordenadas universales de (0,0) a (50,50); se requerirá de una transformación de normalización y dos transformaciones de dispositivo.

Usando el modelo de transformaciones de GKS:

- Transformación de normalización:

Para un punto (X_{cu}, Y_{cu}) perteneciendo al espacio de las coordenadas universales, y el punto transformado (X_{cn}, Y_{cn}) , se verifican las siguientes ecuaciones:

$$\frac{X_{cu} - X_{cu \min}}{X_{cu \max} - X_{cu \min}} = \frac{X_{cn} - X_{cn \min}}{X_{cn \max} - X_{cn \min}}$$

$$\frac{Y_{cu} - Y_{cu \min}}{Y_{cu \max} - Y_{cu \min}} = \frac{Y_{cn} - Y_{cn \min}}{Y_{cn \max} - Y_{cn \min}}$$

En este caso:

$$\begin{aligned} X_{cu \min} &= 0 & X_{cu \max} &= 50 \\ Y_{cu \min} &= 0 & Y_{cu \max} &= 50 \end{aligned}$$

$$\begin{aligned} X_{cn \min} &= 0 & X_{cn \max} &= 1 \\ Y_{cn \min} &= 0 & Y_{cn \max} &= 1 \end{aligned}$$

El espacio $(X_{cu \min}, Y_{cu \min}), (X_{cu \max}, Y_{cu \max})$ se llama ventana (es parte del espacio de coordenadas universales).

El espacio $(X_{cn} \text{ min}, Y_{cn} \text{ min})$, $(X_{cn} \text{ min}, Y_{cn} \text{ max})$ se llama puerta de visión (es parte del espacio de coordenadas normalizadas).

La transformación de normalización y su inversa son definidas por la asociación de una ventana y de una puerta de visión.

- Transformación de dispositivo:

Para un punto (X_{cn}, Y_{cn}) perteneciendo al espacio de las coordenadas normalizadas, y el punto transformado (X_{cd}, Y_{cd}) , se verifican las siguientes ecuaciones:

$$\frac{X_{cn} - X_{cn \text{ min}}}{X_{cn \text{ max}} - X_{cn \text{ min}}} = \frac{X_{cd} - X_{cd \text{ min}}}{X_{cd \text{ max}} - X_{cd \text{ min}}}$$

$$\frac{Y_{cn} - Y_{cn \text{ min}}}{Y_{cn \text{ max}} - Y_{cn \text{ min}}} = \frac{Y_{cd} - Y_{cd \text{ min}}}{Y_{cd \text{ max}} - Y_{cd \text{ min}}}$$

En este caso:

$$\begin{aligned} X_{cn \text{ min}} &= 0 & X_{cn \text{ max}} &= 1 \\ Y_{cn \text{ min}} &= 0 & Y_{cn \text{ max}} &= 50 \end{aligned}$$

$$\begin{aligned} X_{cd \text{ min}} &= 0 & X_{cd \text{ max}} &= 512 \\ Y_{cd \text{ min}} &= 0 & Y_{cd \text{ max}} &= 512 \end{aligned}$$

$$\begin{aligned} X_{cd \text{ min}} &= 0 & X_{cd \text{ max}} &= 1024 \\ Y_{cd \text{ min}} &= 0 & Y_{cd \text{ max}} &= 1024 \end{aligned}$$

El espacio $(X_{cn} \text{ min}, Y_{cn} \text{ min})$, $(X_{cn} \text{ max}, Y_{cd} \text{ max})$ se llama ventana de dispositivo (es parte del espacio de coordenadas normalizadas).

El espacio $(X_{cd\ min}, Y_{cd\ min}), (X_{cd\ max}, Y_{cd\ max})$ se llama puerta de visión del dispositivo (es parte del espacio de coordenadas de dispositivo).

Cada transformación de dispositivo y su inversa son definidas por la asociación de una ventana de dispositivo y de una puerta de visión de dispositivo.

2.3 Sistema de Coordenadas en AutoCAD

AutoCAD utiliza un sistema fijo de coordenadas cartesianas, como se muestra en la figura 2.1, para localizar puntos en la pantalla. El eje de las X indica la distancia horizontal y el eje de las Y indica la distancia vertical. El origen es donde el valor de ambos ejes X y Y vale cero. A este sistema de coordenadas se le puede llamar sistema de coordenadas universales (WCS).

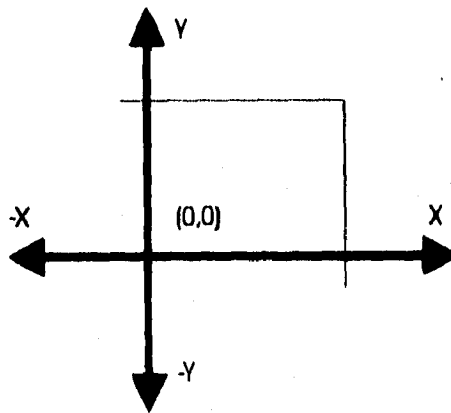


figura 2.1

Un punto es expresado como un par de coordenadas (x,y) (esto es, un valor de X y un valor de Y separados por una coma. Para localizar un punto en particular en el sistema de coordenadas universales, se puede imaginar una línea perpendicular a través del eje de las X en el punto del valor de X, y otra línea perpendicular en el eje de las Y en el punto del valor de Y. El punto se localiza donde las dos líneas se intersectan.

Otra forma de visualizar esto es imaginarse una malla rectangular emanada desde el origen del sistema de coordenadas. Para localizar un punto en la malla con las coordenadas $(5,4)$, se pueden contar cinco líneas a la derecha en la malla a partir del origen (dirección positiva de X), y cuatro líneas en la malla hacia arriba a partir del origen del sistema de coordenadas (dirección positiva de Y).

Para dibujos de dos dimensiones, todo el trabajo puede ser hecho usando coordenadas (x,y) . Sin embargo, si el trabajo involucra dibujos o modelos tridimensionales, el eje de las Z's puede ser adicionado para localizar puntos tridimensionales usando una coordenada triple (x,y,z) . El eje de las Z's es perpendicular al plano definido por los ejes X y Y (también llamado plano XY). En la figura 2.2 se muestra la posición del punto $(5,4,6)$.

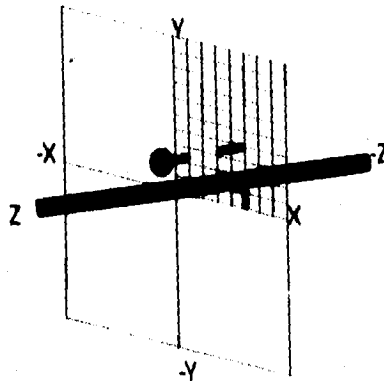


figura 2.2

Cuando AutoCAD espera a que se introduzca un punto tridimensional, con frecuencia se puede introducir sólo como un par de coordenadas

(x,y), omitiendo el valor de Z, AutoCAD asigna el valor omitido como el valor de la elevación. Desde que los apuntadores trabajan solamente en dos dimensiones, AutoCAD siempre usa la elevación activa como el valor de Z para puntos que se introduzcan con un apuntador (mouse, tableta). El plano XY en la elevación activa es llamado plano construcción.

2.4 Sistema de Coordenadas de Usuario.

El sistema de coordenadas universales es fijo y no puede ser alterado en AutoCAD, sin embargo, se puede definir un sistema de coordenadas arbitrario, llamado Sistema de Coordenadas de Usuario (UCS), dentro del WCS. El origen del nuevamente definido sistema de coordenadas puede ser relocalizado en cualquier lugar en el sistema de coordenadas universales, estos ejes pueden ser girados e inclinados de cualquier forma que se elija.

El Sistema de Coordenadas de Usuario permite cambiar el plano construcción y simplificar las locaciones de los puntos tridimensionales. Por ejemplo, el dibujar el tejado de una casa o el lado de una parte mecánica es mucho más fácil si se define un sistema de coordenadas de usuario posicionado y orientado en relación a la superficie que se desea crear.

No existe límite en el número de coordenadas de usuario que se deseen usar en un dibujo. Con frecuencia, relocalizar el origen da a un complejo problema de tres dimensiones una simple solución de dos dimensiones. La casa en la figura 2.3, por ejemplo, tiene varias elevaciones y orientaciones. Aún adicionando un tragaluz en cada tejado, es una tarea simple: solamente se define un sistema de coordenadas de usuario por cada tejado, con el origen en la esquina inferior izquierda. Cada tragaluz es fácilmente localizado y colocado desde el origen de su propio sistema de coordenadas de usuario.

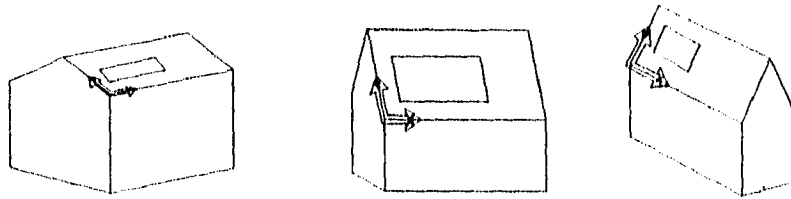


figura 2.3

El sistema de coordenadas de usuario puede ser manejado en dos dimensiones tan bien como en aplicaciones tridimensionales. Por instancia, cambiando el origen y rotando el plano XY alrededor del eje Z, se podría establecer un sistema local de coordenadas rectangulares por cada cuarto en una planta arquitectónica, inclusive si algunos de los cuartos que tienen un giro o irregulares en su forma.

2.5 Icono del Sistema de Coordenadas.

Se puede referir al icono del sistema de coordenadas como ayuda a ubicar la orientación de los diferentes sistemas de coordenadas en un dibujo. El icono muestra la orientación del sistema de coordenadas de usuario activo indicando las direcciones positivas del eje de las X's y del eje de las Y's. Esta información es mostrada como en la figura 2.4.

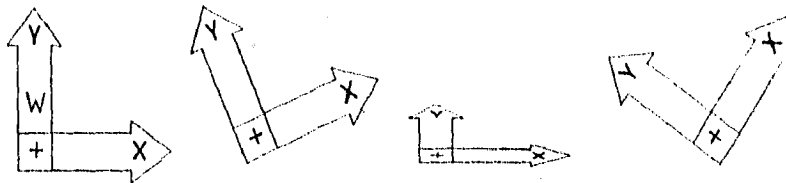


figura 2.4

CAPITULO III. DIGITALIZACION

3.1 Conceptos Básicos

Un digitalizador es un dispositivo de entrada que permite trasladar información gráfica, tales como dibujos, fotografías, planos, etc. dentro de un formato legible por la computadora.

De esta forma trabajan los digitalizadores tales como el ratón, la pluma óptica, la tableta, etc. La tableta es el aparato más utilizado para la digitalización de planos.

Una tableta tiene dos partes principales, una tabla y un apuntador, el apuntador también puede ser el cursor/punta. Una malla dentro de la tableta emite una señal de baja intensidad. El cursor/punta lee la señal cuando éste se mueve a través de la tableta.

De esta forma, para que el cursor/punta pueda sentir las señales, debe de estar sobre la malla o área activa, y suficientemente cerca de la malla, o en aproximidad. Las tabletas digitalizadoras expresan la posición del cursor/punta como un par de coordenadas X,Y. el cual es llamado un reporte. Los reportes válidos solamente pueden ser recolectados cuando el cursor/punta está en el área activa de la tableta y en aproximidad. Los reportes están en coordenadas absolutas, acotadas a partir del origen de la tableta, y son expresadas en pulgadas, milímetros, o cuentas de resolución.

3.2 Área Activa

El área activa es la región en la superficie de la tableta en la cual la posición del cursor/punta puede ser identificada. Esto está marcado en la tableta como brackets en las esquinas.

3.3 Proximidad

Proximidad es la distancia máxima sobre el área activa en la cual el cursor/punta pueda ser reconocido, y reportar una posición. El límite de proximidad para la mayoría de las tabletas es de 0.5 pulgadas. Aunado a esto, una área activa y proximidad establecen un volumen tridimensional dentro del cual el cursor/punta puede sacar reportes. Los reportes derivados desde afuera de este volumen están fuera de proximidad, y de esta forma, no representan la posición actual del cursor/punta.

3.4 Resolución

Resolución es el más fino detalle que la tableta puede distinguir. La resolución está expresada en líneas por pulgada (lpi) o líneas por milímetro (lpmm).

Los reportes son acotados en cuentas de resolución. Una cuenta es la distancia entre dos líneas de resolución. Como se muestra en la figura 3.1, cada cuadro es una cuenta de resolución. Las tabletas reportan las mismas coordenadas para cualquier punto dentro del cuadro.

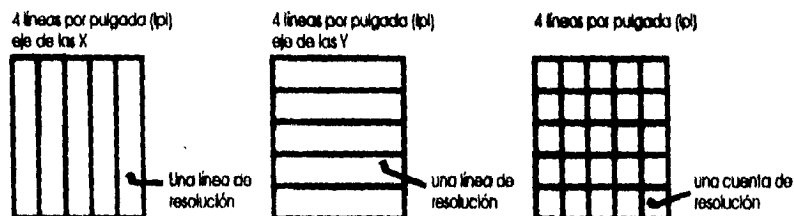


figura 3.1

Con diferentes resoluciones, se pueden obtener diferentes reportes para la misma localización de la tableta. En la figura 3.2 el punto A y B se localiza en la misma posición en la tableta, pero sus coordenadas son diferentes, esto por la diferente configuración de la resolución.

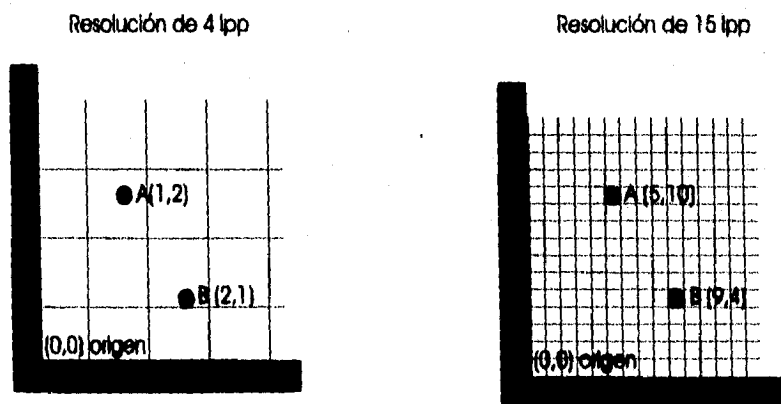


Figura 3.2

La mayoría de las tabletas soportan dos tipos de resoluciones, la resolución fija y la variable.

La punta es un instrumento en forma de pluma usado para localizar puntos en la tableta. Ya que es similar en tamaño y forma a un lápiz o

pluma, es natural a la mano y es recomendada para dibujo a mano alzada. Esta se conecta a la tableta y puede ser de dos o tres botones.

El cursor es un dispositivo que se arrastra con la mano, y es usado para localizar puntos en la tableta. El cursor tiene dos hilos cruzados para dar precisión en la localización de puntos y es recomendado para trazar o seleccionar opciones de un menú, puede ser de dieciséis o cuatro botones.

El cursor y la punta tienen la misma funcionalidad. Ambos pueden ser usados para trazar, dibujar, etc. La punta es más natural a la mano, por lo tanto es más recomendada para dibujo a mano alzada o selección de puntos. El cursor tiene hilos cruzados para obtener más precisión en la selección de puntos, por lo tanto es más recomendado para trazar o seleccionar opciones de menú.

Los botones del cursor y la punta son programados por el software de aplicación que se esté usando.

El tamaño de las tabletas puede variar desde 12" hasta 48" dependiendo de la marca y del avance de la tecnología. Su tamaño físico es diferente del tamaño del área activa, esto depende del modelo de la tableta.

Las tabletas utilizan una tecnología electromagnética y su resolución puede ser de hasta 2000 líneas por pulgada dependiendo del avance de la tecnología.

CAPITULO IV. COMANDOS RELACIONADOS DE AutoCAD

Esta es una descripción breve de algunos de los comandos básicos y necesarios para el trazo de carreteras con el paquete AutoCAD.

4.1 COPY

Para copiar objetos existentes, se teclea el comando COPY, COPY es similar al comando MOVE, pero éste deja copias del objeto seleccionado en el desplazamiento especificado, dejando el original intacto. Las copias son orientadas y escaladas en la misma forma que el original. Se puede inclusive preguntar al comando COPY si se desea hacer otro desplazamiento, permitiendo con esto hacer fácilmente múltiples copias del objeto seleccionado. Cada resultado de una copia es completamente independiente del original y puede ser editado y manipulado como cualquier otra entidad simple.

Command: COPY

Select objects: (muestra los objetos a copiar)

<Base point or displacement> /Multiple: (primer punto o distancia x,y,z)

Second point of displacement: (segundo punto o solamente ENTER)

Como el comando MOVE, se puede usar "DRAG" para especificar el desplazamiento.

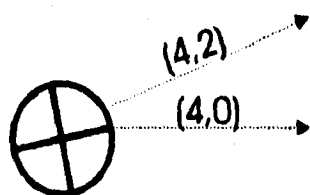
Copias Múltiples

Para hacer copias múltiples usando el comando COPY, basta responder en la pregunta "Base point" con "Multiple" (o solamente "M"). La pregunta "Base point" reaparece seguida de repetidas preguntas "Second point", y la copia del objeto seleccionado es hecha por cada desplazamiento que se introduzca. Cada desplazamiento es relativo al punto base original. Cuando se han realizado todas las copias que se necesitan, se contesta con una respuesta nula en la pregunta "Second point".

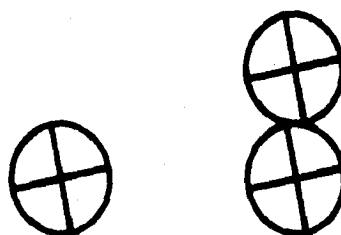
NOTA: Desde una señal de respuesta nula que se necesita para dejar de hacer más copias, el desplazamiento no puede ser introducido como un duplicado x,y,z seguido por ENTER. Se puede causar el mismo efecto, sin embargo, introduciendo 0,0,0 como el punto base, y el deseado x,y,z duplicado como el segundo punto del desplazamiento.

En el ejemplo siguiente, se hacen dos copias de un objeto existente.

Command: COPY
Select objects: se hace
<Base point or displacement>/Multiple: M
Base point: 0,0
Second point of displacement: 4,0
Second point of displacement: 4,2
Second point of displacement: ENTER



DESPLAZAMIENTOS



MÚLTIPLES COPIAS

Si se necesitan múltiples copias arreglados en un patrón circular o rectangular, se utiliza el comando ARRAY.

4.2 CHPROP

EL comando CHPROP (Change Properties) perfecciona una parte del comando CHANGE. Las propiedades de las entidades que se pueden cambiar vía CHPROP son color, tipo de línea, lámina, y espesor. CHPROP trabaja con todas las entidades, independientemente de su dirección. El formato del comando es:

Command: **CHPROP**
Select objects: *(se hace)*
Change what property (Color/LAyer/LType/Thickness) ?

Las opciones son las mismas que las de la opción "properties" del comando CHANGE, excepto que la opción "Elevation" no se proporciona.

4.3 DOUGHNUT

Se puede usar el comando DOUGHNUT (o DONUT) para dibujar anillos o círculos rellenos. El diálogo del comando comienza así:

Command: **DOUGHNUT** o **DONUT**
Inside diameter <current>:
Outside diameter <current>:

Los diámetros de la dona más reciente son mostrados como current. Para especificar un diámetro nuevo, se introduce un valor numérico o se designan dos puntos para mostrar el diámetro a AutoCAD. Un círculo relleno puede ser dibujado especificando un diámetro interno con valor cero. El valor del diámetro externo debe de ser diferente de cero. Cuando se están especificando los diámetros, AutoCAD pregunta repetidamente

por las localidades en las cuales la dona va a ser dibujada de esta forma:

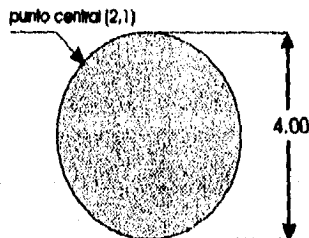
Center of doughnut:

Se puede desplazar el punto del centro si se desea. A medida que se introduce un punto, se dibuja una dona con los diámetros especificados. Para finalizar el comando DOUGHNUT, se responde con una respuesta nula a esta pregunta.

El comando DOUGHNUT estructura una polilínea cerrada (compuesta de segmentos de arco cerrados) representando el objeto específico. Consecuentemente, una dona se puede editar con el comando PEDIT o cualquier otro comando de edición que opere con polilíneas. El relleno sólido de la dona está sujeta al modo de relleno.

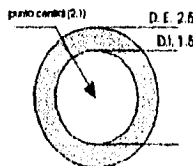
Por ejemplo, suponiendo que se desea dibujar un círculo relleno con el centro en (2,1) con un diámetro de 4 unidades de dibujo. Se podría usar la siguiente secuencia de comandos:

Command: **DOUGHNUT**
Inside diameter <0.25>: **0**
Outside diameter <0.35>: **4**
Center of doughnut: **2.1**
Center of doughnut: **ENTER**



Similarmente, si se desea una dona con centro en (2,1), con un diámetro interno de 1.5 unidades de dibujo y un diámetro externo de 2.5 unidades de dibujo, ésta sería la secuencia de comandos:

Command: **DOUGHNUT**
Inside diameter <0.00>: **1.5**
Outside diameter <4.00>: **2.5**
Center of doughnut: **2.1**
Center of doughnut: **ENTER**



4.4 DXFIN

El comando DXFIN es usado para convertir archivos con formato DXF (archivo de intercambio de dibujo) a dibujos de AutoCAD, el formato es:

Command: DXFIN File name: (nombre)

Se introduce el nombre del archivo de intercambio de dibujo a ser cargado.

Para cargar un archivo DXF completo, se debe de usar el DXFIN en un dibujo vacío, antes de que cualquier entidad haya sido dibujada, y antes de cualquier definición de bloque, layer, tipos de línea, tipos de letras, nombres de vistas, nombres de sistemas de coordenadas, o nombres de configuraciones de vistas hayan sido creadas.

Si se detecta un error durante la entrada, el dibujo nuevo es descartado. De otra forma, un ZOOM automático se ejecuta para ver las extensiones del dibujo.

Si el actual dibujo de trabajo no está vacío, DXFIN carga solamente la sección de ENTITIES del archivo DXF, adicionando las entidades que se encuentren allí al dibujo actual de trabajo. En este caso, DXFIN despliega el siguiente mensaje:

Not a new drawing -- only ENTITIES section will be input.

Si se detectan errores durante la entrada parcial de un archivo DXF, el dibujo regresa al estado en el que se encontraba antes de ejecutar la instrucción. De otra forma se dibujan las nuevas entidades.

4.5 DXFOUT

Se puede generar un archivo de intercambio de dibujo desde un dibujo existente, es decir, desde el comando DXFOUT del editor de dibujo. El formato del comando es el siguiente:

Command: **DXFOUT** File name <default>: *(nombre o ENTER)*

El nombre default para el archivo de salida es el mismo que se esté empleando para el actual dibujo de trabajo, pero con el tipo de formato DXF, si se especifica un nombre de archivo diferente, no se debe de incluir la extensión ya que siempre será ".dxf". Si un archivo con el mismo nombre ya existe, éste es borrado. A continuación, DXFOUT, pregunta por la precisión que se desea para los valores numéricos de punto flotante y permite la salida parcial de un archivo DXF conteniendo solamente los objetos seleccionados.

Enter decimal places of accuracy (0 to 16)/Entities/Binary <6>:

La opción de "Binary" genera un archivo con información compacta, con datos binarios, a diferencia del tipo normal que es un archivo ASCII editable. Cuando se seleccionan objetos, AutoCAD vuelve a preguntar por el número de precisión.

Enter decimal places of accuracy (0 to 16)/Binary <6>:

4.6 ERASE

El comando ERASE te permite especificar entidades que se desean remover permanentemente del dibujo. El formato del comando es:

Command: **ERASE**
Select objects: *(objetos deseados)*

Una característica especial es el "ERASE Last" variación del comando ERASE. Se puede utilizar para borrar la entidad más recientemente dibujada. Se puede usar repetidamente, para seleccionar hacia atrás paso por paso a través del dibujo, borrando la entidad más reciente en cada paso. Por ejemplo, si se ha creado solamente una línea seguida de dos círculos, la secuencia de comandos mostrada abajo borrará los dos círculos:

Command: **ERASE**
Select objects: **L**
Select objects: **ENTER** (fin de la selección)
Command: **ENTER** (ocasiona una repetición del comando)
Select objects: **L**
Select objects: **ENTER** (fin de la selección)

4.7 FILLET

El comando FILLET conecta dos líneas, arcos o círculos con un segmento de arco de un radio específico. Este ajusta la longitud de la línea o arco original de tal manera que finalice exactamente en el arco definido.

Command: **FILLET**
Polyline/Radius/<Select two objects>: *(Point to two objects)*

FILLET basa algunos de sus cálculos en la distancia entre el punto seleccionado y el punto final del objeto mostrado en la pantalla. Sin embargo, éste trabaja mejor en una vista plana. Si la dirección de la vista es oblicua al plano XY del actual UCS, FILLET despliega un mensaje en el que advierte que el resultado podría no ser el que se espera.

4.8 LAYER

El comando LAYER sirve para crear láminas, para definir láminas nuevas, se selecciona la capa activa, se asigna un número de color, un tipo de línea y éste se aplica a todos los elementos que pertenezcan a esta lámina, sirve para activar o desactivar una lámina, y listar las láminas definidas. Estas funciones se seleccionan desde una lista de opciones, como se muestra a continuación:

Command: **LAYER**
? Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw: (Se selecciona una)

La operación que sigue depende de la función seleccionada. Cuando se completa una función, aparece nuevamente la pregunta de cuál opción se desea, permitiendo definir varias láminas antes de salir del comando. Cuando se desea salir del comando, se da una respuesta nula a la pregunta, si se selecciona alguna función que pueda cambiar el despliegue, una operación de regeneración de redibujo es aplicado automáticamente cuando se sale del comando.

4.9 MOVE

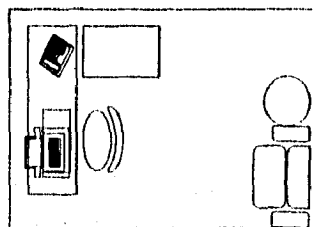
El comando MOVE permite mover una o más entidades desde su localidad presente, a una nueva, sin cambiar su orientación o tamaño. Después de que se selecciona el lote de objetos a ser movidos, AutoCAD pregunta por un vector de desplazamiento (x,y,z) para indicar qué tan lejos va a ser movido el objeto, y en qué dirección. Si se da una respuesta nula en la pregunta del segundo punto, el primer punto es interpretado como una distancia x,y,z. El vector de dirección que se indica es aplicado a todas las entidades seleccionadas. Cabe hacer notar que se está supliendo un desplazamiento relativo; si se usa el método de dos puntos, el primer punto no tiene que ser uno de los objetos seleccionados, sin

embargo puede ayudarte a visualizar el desplazamiento. El formato del comando es:

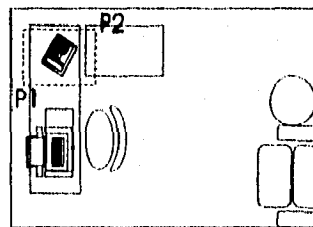
Command: **MOVE**
Select objects: *(muestra cual mover)*
Base point or displacement: *(primer punto o distancia x,y,z)*
Second point of displacement: *(segundo punto o solamente ENTER)*

Se puede introducir el comando "DRAG" cuando se pregunte por el desplazamiento. Esto permite arrastrar los objetos seleccionados dentro de una posición visual.

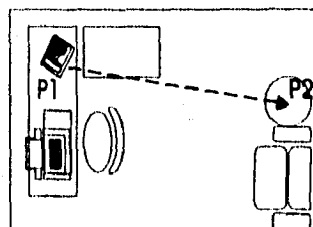
La siguiente secuencia de figuras muestra el uso del comando MOVE. En este ejemplo, se utiliza la técnica de selección de ventana para designar los objetos a mover, pero se puede usar cualquier método de selección que sea más conveniente.



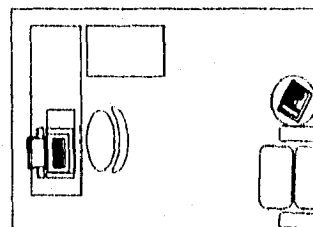
1. ORIGINAL



2. SELECCION DE VENTANA



3. DESPLAZAMIENTO



4. RESULTADO

4.10 OFFSET

El comando **OFFSET** construye una entidad paralela a otra entidad a una distancia específica o hacia un punto específico. El diálogo del comando inicia así:

Command: OFFSET
Offset distance or Through <last>:

Donde "last" es la última especificación dada al comando **OFFSET**, ya sea una distancia o la palabra "Through". Se puede introducir una distancia explícitamente, mostrando la distancia a AutoCAD con dos puntos, o introduciendo "Through" (o simplemente "T") para indicar que el doble podría ser hacia un punto específico indicado posteriormente. Seguido por la pregunta:

Select object to offset:

Solicita apuntar al objeto a ser duplicado para seleccionarlo. No se puede usar la selección "Window", "Crossing" o "Last" para escoger el objeto.

La dirección de la extrusión de la entidad que se selecciona debe ser paralela al eje de las Z's del activo Sistema de Coordenadas de Usuario. Si ésto no es así, AutoCAD despliega el siguiente mensaje:

Entity not parallel with UCS.

Y repite la pregunta "Select object to offset:".

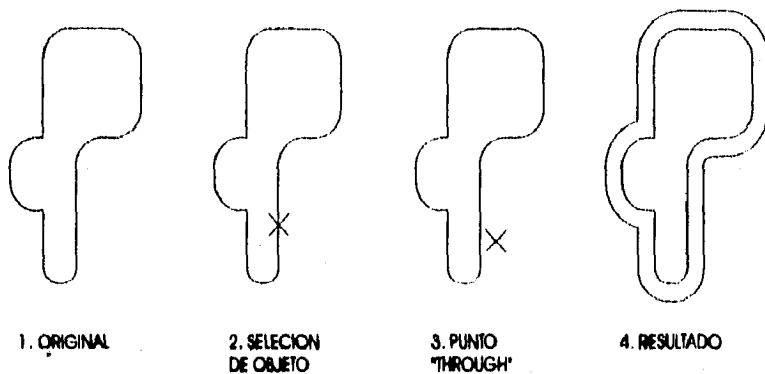
Dependiendo del tipo de definición de la distancia, ya sea por selección o por posición, en la pregunta original, uno de los siguientes mensajes podría aparecer a continuación.

Side to offset:
Through point:

Capítulo IV. Comandos Relacionados de AutoCAD.

Se responde con un punto en el lado del objeto que se desea duplicar. Si se especifica una distancia, el punto solamente indica el lado lógico del objeto en el cual el duplicado va a ser construido. Si se especifica "Through", el duplicado es construido hacia el punto que se especifica.

Un ejemplo de duplicar una polilínea usando el método "through point" se muestra a continuación.



Si se sigue las operaciones del duplicado, la pregunta "Select object to offset" es rehusada. Entonces se puede seleccionar otro objeto (asumiendo la misma distancia de duplicado o el mismo modo "through point"), o solamente ENTER para salir del comando al modo principal de AutoCAD.

Si se usa el método "distance", la distancia que se especifica debe ser mayor que cero. Para duplicar desde una polilínea ancha, se acota la distancia del duplicado desde el centro de la línea de la polilínea.

Si se duplica una entidad de línea y las especificaciones en la nueva línea dibujada sobre la original, el mensaje:

Invalid though point.

es usado y se ignora el comando. Si se duplica un Círculo o Arco y se ha especificado una distancia y lado de duplicado en el cual el objeto no

está definido (por ejemplo, podría tener un radio negativo), el siguiente mensaje podría aparecer:

No parallel at that offset.

Si se intenta duplicar una entidad diferente a una Línea, Arco, Círculo, o Polilínea bidimensional, aparecerá el siguiente mensaje:

Cannot offset that entity.

Aunque se puede optar fácilmente por un duplicado de Polilínea, cabe hacer notar que una línea paralela a una curva arbitraria no está bien definida. Hay con frecuencia un número infinito de curvas paralelas para una curva dada, pasando todas a través del mismo punto. El comando OFFSET trata de aplicar reglas razonables para generar la mejor forma paralela, lo más cercano a lo que se desea. Si falla OFFSET al dar la curva deseada, se puede usar PEDIT, TRIM, y EXTEND para transformar la curva resultante a una que se tenga en mente. Puede ser útil usar el comando EXPLODE en la Polilínea original y duplicar las piezas individualmente. Duplicar Polilíneas con anchos variables puede producir un resultado inesperado. Se puede usar PEDIT para corregir cualquier consecuencia indeseada.

En general, OFFSET trabaja propiamente tan largo como se le indique hacer cosas razonablemente; para generar buenas definiciones, curvas paralelas únicas. Si se piden hacer cosas ridículas, es posible que pase, ejercita su propio juicio en caso de ambigüedad, y podría estar en desacuerdo con lo que se desea. Recordando que siempre se puede fijar el resultado más tarde.

El duplicado se aplica relativamente al plano X,Y del activo UCS. Si la vista es oblicua al plano activo X,Y AutoCAD sale con un mensaje cuando se inicia el comando OFFSET, advirtiendo que el resultado de un OFFSET podría no ser lo que se espera. Se puede evitar esto usando el comando PLAN o VPOINT para cambiar la dirección de la vista al plano (0,0,1) relativo al activo UCS, y posteriormente perfeccionar el comando OFFSET.

4.11 PAN

El comando PAN permite ver una porción diferente del dibujo en la ventana activa, sin cambiar la magnitud. Esto permite ver los detalles que quedaron fuera de la pantalla antes del comando PAN. Para visualizar los efectos de este comando, imagina que se está viendo un dibujo a través de una ventana de la pantalla y que se puede deslizar el dibujo a la derecha, izquierda, arriba y abajo sin mover la ventana.

Por supuesto, se debe indicar al comando PAN en qué dirección mover el dibujo, y qué tan lejos moverlo. Esta información es llamada el desplazamiento. Se puede introducir un par de coordenadas simples indicando el desplazamiento relativo del dibujo con respecto a la pantalla, o se pueden designar dos puntos, en este caso AutoCAD calcula el desplazamiento desde el primer punto al segundo.

4.12 PEDIT

Pollíneas de dos dimensiones, y mallas poligonales tridimensionales todas son variaciones de una entidad de Pollínea. El comando PEDIT permite editar estas entidades. El grupo de operaciones de edición depende del tipo de Pollínea, como se muestra en la siguiente tabla.

Tipo de entidad	Operaciones de edición
Pollínea bidimensional	a) Cambiar la Pollínea entera para tener un ancho uniforme. b) Cambiar el ancho de segmentos individuales de la Pollínea. c) Cerrar una Pollínea abierta, o abrir una cerrada.

	<p>d) Borrar todos los espirales y curvas entre dos vértices.</p> <p>e) Separar una Polilínea en dos Polilíneas.</p> <p>f) Juntar cualquier número de Líneas contiguas, Arcos, y Polilíneas bidimensionales dentro de una simple Polilínea bidimensional.</p> <p>g) Mover vértices seleccionados de una Polilínea, o adicionar nuevos vértices.</p> <p>h) Ajustar una curva de arcos o una curva B-spline a todos los vértices en la Polilínea, con especificaciones opcionales de la tangente a cada vértice.</p>
Polilínea tridimensional	<p>a) Cerrar una Polilínea abierta, o abrir una cerrada.</p> <p>b) Borrar todos los espirales entre dos vértices.</p> <p>c) Separar una Polilínea en dos Polilíneas.</p> <p>d) Mover vértices seleccionados de la Polilínea, o adicionar nuevos vértices.</p> <p>e) Ajustar una curva tridimensional spline a todos los vértices de la Polilínea.</p>
Malla poligonal	<p>a) Cerrar una malla abierta, o abrir una cerrada, en la dirección M o N.</p> <p>b) Mover vértices seleccionados de la malla.</p> <p>c) Ajustar una igualación de superficie a los vértices de la malla.</p>

Se puede editar una Polilínea bidimensional solamente si la dirección extrusiva es paralela al eje de las Z's del activo UCS. Para editar cualquier tipo de Polilínea, se introduce el comando PEDIT.

Command: **PEDIT**
 Select polyline:

Es válido usar cualquier método de selección de objetos para seleccionar una entidad simple. Si la entidad seleccionada es una Línea o un Arco, AutoCAD pregunta:

Entity selected is not a polyline.
Do you want it to turn into one?

Si se responde "Y", el objeto es convertido en un segmento de Polilínea bidimensional simple que puede, por lo tanto, ser editada. Se puede usar esta operación para juntar Líneas y Arcos dentro de una Polilínea.

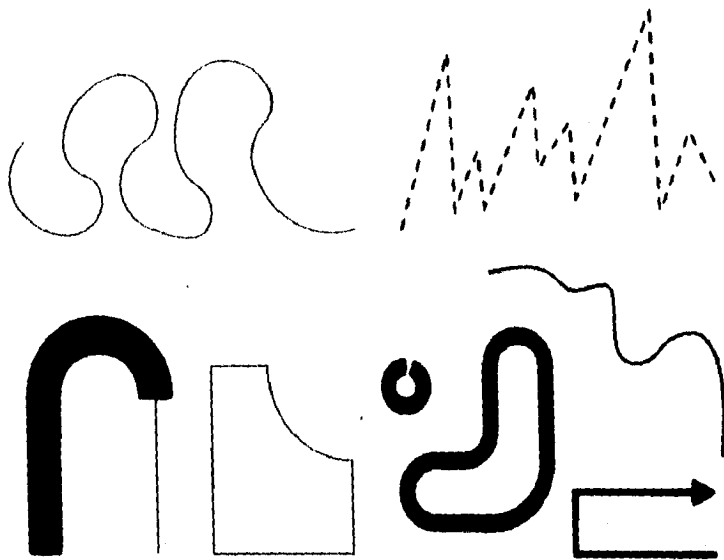
Las preguntas subsiguientes y opciones de edición dependen del tipo de Polilínea (2D, 3D, o malla) que se haya seleccionado.

4.13 PLINE

Con este comando se pueden dibujar Polilíneas bidimensionales o tridimensionales. Una Polilínea bidimensional es una secuencia conectada de segmentos de línea y arcos, y es tratado por AutoCAD como una entidad simple. Las Polilíneas bidimensionales tienen las siguientes propiedades adicionales:

- a) Ellas pueden ser dibujadas con tipos de línea de puntos y rallas.
- b) Ellas pueden ser anchas o angostas.
- c) Una polilínea ancha puede ser usada para formar círculos rellenos o donas.
- d) La secuencia de líneas y arcos pueden formar un polígono cerrado o una elipse.
- e) Las polilíneas bidimensionales pueden ser editadas para insertar, mover, o borrar vértices o para unir Líneas, Arcos, y polilíneas dentro de una polilínea.
- f) Se pueden adicionar cortes o redondeos de esquina, donde se deseen.
- g) Se pueden ajustar curvados o líneas "S" en una polilínea bidimensional.
- h) El área y perímetro de una polilínea bidimensional puede ser calculado.

Las figuras siguientes ilustran diversos tipos de polilíneas.



Una polilínea tridimensional esta formada de segmentos de línea con un ancho de cero. Se pueden suplir coordenadas tridimensionales (x,y,z) para cada vértice. Las polilíneas tridimensionales pueden ser ya sea abiertas o cerradas.

Para dibujar una polilínea bidimensional, se introduce el comando PLINE.

Command: **PLINE**
from point:

Cuando se responde con el punto inicial de la Polilínea, el actual ancho de línea es desplegado:

Current line-width is *nnn*

Este ancho podría ser usado para todos los segmentos de la Polilínea hasta que se seleccione un ancho diferente. Ahora se puede empezar a introducir puntos y otras especificaciones.

Inicialmente, el comando PLINE espera que se introduzcan segmentos de línea, y sale con la pregunta:

Arc/Close/Halfwidth/Length/Undo/Width/<Endpoint of line> :

Como la pregunta indica, varias opciones están disponibles. La opción por omisión es desplegada con signos de menor que y mayor que "<>", e indica que podría pasar si se introduce simplemente un punto. En este caso, un punto es interpretado como el punto final de un segmento de línea, y un segmento de línea es dibujado desde el punto anterior hasta el nuevo punto, igual que en el comando LINE. AutoCAD pregunta a continuación por otro segmento de línea.

Las otras respuestas de la pregunta de PLINE son opciones que modifican la acción del comando. Para seleccionar una opción, se introduce solamente la letra mayúscula indicada en la pregunta. Por instancia, se introduce "W" para seleccionar la opción de tamaño "Width". Cada opción se describe a continuación.

- Arc** Cambia el comando PLINE al modo de arco y resulta una pregunta diferente, con sus correspondientes opciones.
- Close** Obliga a AutoCAD a dibujar una línea desde la posición actual a la posición inicial de la Polilínea, creando un polígono cerrado. El comando PLINE concluye con esta opción. El ancho de la línea que cierra podría ser el que estaba definido antes de seleccionar esta opción.
- Length** Este te permite dibujar un segmento de línea al mismo ángulo que el segmento anterior, especificando solamente la longitud del segmento nuevo. Si el segmento previo fue un arco, éste produce un segmento de línea tangente a ese arco.
- Undo** Borra la línea más reciente o segmento de arco adicionado a la Polilínea activa. Se puede aplicar este método "n" número de veces, hasta que quede solamente un punto en la Polilínea. Si se termina el comando PLINE mientras éste contiene solamente un punto, no se crea ninguna Polilínea. La siguiente pregunta y la dirección que asume para

subsecuentes segmentos de arco podría basarse en el último segmento que quedó después de la eliminación. Esto es, el último segmento que queda es un arco, el comando PLINE podría cambiar al modo arco.

Width

Este te permite especificar el ancho del siguiente segmento de la Polilínea. Un ancho de cero produce una línea simple que aparece con el mínimo ancho desplegable dependiendo de la tecnología del desplegador. Un ancho mayor que cero produce líneas amplias similares a trazos, rellenos sólidos si el modo relleno está prendido. AutoCAD pregunta por ambos, el ancho inicial y, el ancho final del segmento, permitiendo angostar la línea de esta forma:

Starting width <0,0000>: 0.2
Ending width <0,2000>: 0.4

El ancho inicial que se introduce se convierte en el valor por omisión para el ancho final. El ancho final, es almacenado, y se convierte en el ancho uniforme para todos los segmentos subsecuentes hasta que se cambie el ancho nuevamente. Los puntos iniciales y finales de los segmentos de la línea amplia están en el centro de la línea, como se muestra a continuación:



Halfwidth

Esta opción permite especificar el ancho desde el centro de un segmento de Polilínea ancha a uno de sus extremos o en medio del ancho total. La pregunta es:

Starting half-width <current>:
Ending half-width <current>:

Se encuentra que esta opción es muy útil cuando se le indica a AutoCAD el ancho por puntos, desde un final de la banda de la cinta el cursor es ensanchado al punto central del final del segmento de línea ampliada. Todos los demás

comentarios descritos para la opción "Width", también se aplican para "Halfwidth".

4.14 REDRAW

Aunque diversos comandos redibujan la imagen en la pantalla automáticamente (generalmente, cuando se cambia la visibilidad de una lámina o cuando se prende o apaga la malla), éste es usado algunas veces para forzar a redibujar la pantalla. El comando REDRAW limpia la ventana activa y remueve las marcas que pueda tener y redibuja los objetos que pudieran borrarse parcialmente con la eliminación de las marcas por la edición o por otros objetos.

Command: REDRAW

El comando REDRAW también puede ser usado mientras otro comando está en progreso. Para hacer esto, se introduce REDRAW en cualquier pregunta diferente de texto. Se puede abortar el proceso de REDRAW presionando Ctrl-C. Con esto se puede ahorrar tiempo si se está saliendo de otro comando que también redibuja la pantalla. REDRAW afecta solamente la ventana activa. Si están activas múltiples ventanas, se puede usar el comando REDRAWALL para redibujar todos los objetos que se encuentren en todas las ventanas.

4.15 REGEN

El comando REGEN fuerza a AutoCAD a regenerar las entidades de dibujo y redibujar la ventana activa. Este es un proceso más tardado que REDRAW y no es necesario con frecuencia.

Command: REGEN

Los comandos ZOOM, PAN, y "VIEW Restore" regeneran automáticamente el dibujo sobre las mismas condiciones. Existen otros comandos que también regeneran las entidades sobre ciertas circunstancias. La regeneración se puede abortar presionando Ctrl-C. Para ahorrar tiempo el se está saliendo de un comando que también utiliza una regeneración. REGEN afecta solamente la ventana activa. Si están activas múltiples ventanas, se puede usar el comando REGENALL para regenerar todos los objetos que se encuentren en todas las ventanas.

4.16 TRIM

Para recortar un objeto y que finalice en un punto de intersección preciso, definido por uno o más objetos, se utiliza el comando TRIM. Primero se define el punto de intersección a la cual la entidad va a ser recortada.

Command: TRIM
Select cutting edge(s) ...
Select objects:

Se puede usar cualquiera de los métodos de selección para definir la entidad de intersección. La dirección extrusiva de la entidad que se selecciona como punto de intersección debe ser paralela al eje de las Z del Sistema de Coordenadas de Usuario que se encuentre activo. Si no es así, AutoCAD despliega el mensaje:

Select objects: *n* selected, 0 found, (*n* not parallel with UCS).

y se repite la pregunta "Select objects". Líneas, Arcos, Círculos, y Polilíneas bidimensionales pueden servir como puntos de intersección; otros objetos diferentes serán ignorados. Cuando una Polilínea

Capítulo IV. Comandos Relacionados de AutoCAD. ■

bidimensional sirve como punto de Intersección, su ancho es ignorado de tal manera que el corte se realiza en el centro de la línea.

Se presiona ENTER cuando todos los elementos de Intersección han sido seleccionados. TRIM pregunta posteriormente:

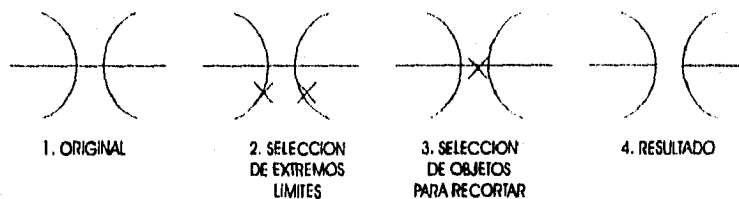
Select objects to trim:

Se seleccionan los objetos para ser recortados a la entidad previamente seleccionada como Intersección, del lado deseado para ser recortado, es decir, se eliminará la porción del elemento seleccionado desde el punto de Intersección, hasta el extremo o Intersección más cercana. Los objetos a ser recortados no pueden ser seleccionados por ventana, Atravezamiento, o Ultima selección. Al igual que las entidades que Interseccionan, la dirección extrusiva de las entidades que se desean recortar deben ser paralelas al eje de las Z del actual UCS. Si no es así, AutoCAD despliega el mensaje:

Entity not parallel with UCS.

Se repite la pregunta "Select object to trim" infinidad de veces, que permite recortar un número ilimitado de entidades, hasta que se responda con una respuesta nula.

La figura que se presenta a continuación, muestra cómo se usa el comando TRIM para eliminar una porción de Línea que se encuentra entre dos Arcos.



4.17 VIEW

El comando VIEW guarda una porción de dibujo para que se pueda manipular posteriormente. El formato es:

Command: **VIEW**
?/Delete/Restore/Save/Window: (se selecciona una)
View name: (name)

Las opciones válidas para esta pregunta quieren decir:

- | | |
|-------------|---|
| ? | Lista las vistas nombradas y reconocidas en el dibujo activo. |
| D (Delete) | Borra una vista de la lista de vistas almacenadas. |
| R (Restore) | La vista que se especifica reemplaza la vista que se está presentando en la pantalla por la porción de dibujo que previamente se almacenó. AutoCAD reconoce el punto central y la magnitud de cada vista almacenada y ejecuta un "ZOOM Center" con esta información cuando se selecciona esta opción. |
| S (Save) | La porción de video que se presenta en la pantalla es almacenada en el mismo archivo que contiene la información del dibujo activo en una vista con un nombre específico (reemplaza alguna vista existente con el mismo nombre). |
| W (Window) | Permite almacenar una porción rectangular del dibujo activo sin la necesidad de ubicarse en esa porción de dibujo para ser presentada en la pantalla. AutoCAD solicita dos puntos para describir los límites de la ventana. |

La ventana que se designa puede ser de diferente forma que la pantalla. Cuando se restaura esta vista, la pantalla despliega la vista junto con el

dibujo que pudiera estar fuera de la ventana es que las medidas de la ventana y la pantalla son diferentes. Sin embargo, cuando se traza esta vista, objetos fuera de esta ventana no aparecen.

Los nombres de las vistas pueden ser de hasta 31 caracteres de longitud y pueden contener letras, números, y algunos otros caracteres (\$-_) estos son almacenados en letras mayúsculas.

Una aplicación del comando VIEW podría ser el asignar una vista de ventana a cada cuarto de una planta y asignarle su nombre correspondiente (COCINA, GARAGE, BAÑO). De esta forma, el se desea desplegar en la pantalla o editar la porción de dibujo que contiene la cocina, simplemente se introduce:

Command: **VIEW**
?/Delete/Restore/Save/Window: **R**
View name: **COCINA**

Cuando se edita un dibujo existente contenido en una vista nombrada, se puede especificar una de estas vistas para ser desplegada cuando el editor de dibujo primero carga el dibujo. Así mismo se puede trazar una porción de dibujo por nombre de vista.

4.18 ZOOM

El comando ZOOM actúa como un lente zoom de una cámara; éste te permite incrementar o aumentar el tamaño aparente del elemento que se está viendo en la pantalla o ventana activa, aunque su tamaño permanece constante. Como se incrementa el tamaño aparente del objeto, se ve una área más pequeña del dibujo en mayor detalle; aumentar el tamaño aparente permite ver una área más larga.

Capítulo IV. Comandos Relacionados de AutoCAD. 81

El comando ZOOM ofrece diversos caminos para designar la magnitud y la porción del dibujo a ser desplegado. Después que se introduce el comando ZOOM, la siguiente pregunta aparece:

All/Center/Dynamic/Extents/Left/Previous/Window/<Scale(X)>:

Se responde con un factor de escala o una de las opciones indicadas del comando ZOOM, cada una de las cuales puede ser abreviada con una letra.

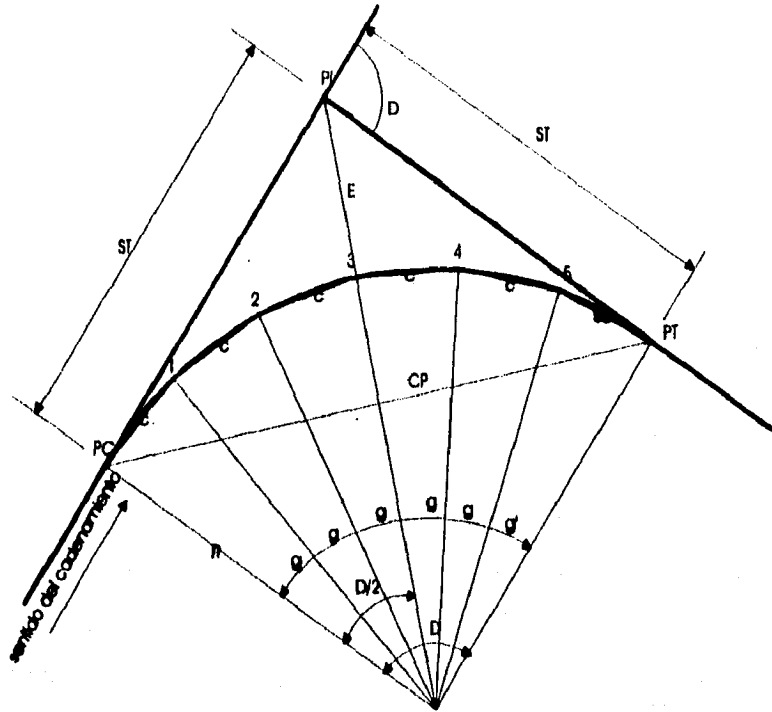
CAPITULO V. CURVAS HORIZONTALES SIMPLES

5.1 Definición de elementos

Las curvas horizontales desde un punto de vista topográfico, son las que se emplean en las vías de comunicación para cambiar de una dirección a otra, uniendo los tramos rectos, "tangentes". Estas curvas son arcos de circunferencia.

Se clasifican las curvas horizontales en simples y compuestas, las primeras están formadas por una sola circunferencia y las segundas por más de una curva simple. En este trabajo describiremos las curvas simples que son las que se aplican para desarrollar el programa auxiliar "VIAS.c", descrito en el siguiente capítulo.

Los elementos que deben de considerarse para trazar las curvas horizontales simples son:



En donde:

- D - Deflexión
- PI - Punto de Intersección
- PC - Punto de Comienzo
- PT - Punto de Término
- R - Radio
- ST - Subtangentes
- C - Cuerda
- g - Grado de curvatura: Es el ángulo bajo el cual se ve la cuerda unitaria desde el centro de la curva (la cuerda unitaria que normalmente se emplea es de 20 m.)
- SC - Subcuerda
- g' - Subgrado
- CP - Cuerda principal
- LC - Longitud de curva
- M - Ordenada media
- E - Externa

Como el cadenamiento debe seguir continuo, deben marcarse igual que las tangentes, cada 20 m, lo cual se hace mediante cuerdas. En algunos casos se usan cuerdas menores.

Los datos de que se parte para calcular los demás elementos de la curva son:

Deflexión	D
Cuerda	C
Radio	R

- (D) Es el ángulo externo que se forma al cambiar de dirección un segmento del trazo, se puede calcular restando a 180 el ángulo interno entre los dos segmentos.
- (C) Es la cuerda que se emplea, según la curva a trazar. Lo más común es que se tome $C = 20\text{m}$ si el grado (g) no pasa de 10° , ya que para ese valor el radio se excede de 100m y el arco es sensiblemente igual a la cuerda. Para curvas con (g) entre 10° y 20° se usan cuerdas de 10m, y para (g) entre 20° y 40° se usan de 5m.
- (R) Queda al criterio del proyectista, quien deberá tratar de que el radio sea lo mayor posible para no tener curvas forzadas, pero adaptándose lo mejor que se pueda a la configuración del terreno para no producir terracerías costosas.

Escogido el radio que convenga, se calcula el (g) correspondiente, y después se adopta como definitivo el (g) cerrado más cercano, y que sea un número par, para facilitar el trazo. El radio varlará con esto, y debe procurarse que sea en aumento, pero no tiene importancia que quede como una cantidad fraccionaria, pues sólo sirve para los cálculos.

Teniendo estos datos, los elementos restantes de la curva se recalcularán así:

$$\frac{C}{2} = R \cdot \text{sen}\left(\frac{g}{2}\right) \therefore R = \frac{C}{2 \text{sen}\left(\frac{g}{2}\right)}$$

$$R = \frac{C / 2}{\text{sen}\left(\frac{g}{2}\right)} \text{ y para } C = 20\text{m}: R = \frac{10}{\text{sen}\left(\frac{g}{2}\right)}$$

En el triángulo Rectángulo (O-PC-PI):

$$\frac{ST}{R} = \tan\left(\frac{D}{2}\right) \therefore ST = R \cdot \tan\left(\frac{D}{2}\right)$$

Número de cuerdas enteras = D/g ; al hacer esta división queda un residuo que no debe dividirse ya, pues será el valor del subgrado (g'):

Número de cuerdas = $D/2 + \text{residuo}$; **residuo = g'**

$$SC = 2R \cdot \text{sen}\left(\frac{g'}{2}\right)$$

$$LC = \frac{D}{g} \times 20\text{mts} \text{ (para } R > 100 \text{ mts).}$$

o también

$$LC = \text{número de cuerdas enteras} + SC$$

Cuando $R < 100\text{m}$, puede multiplicarse por el arco de la cuerda de 20m para tener un valor más aproximado de la longitud; $LC = D/g \times \text{arco}$.

En la figura de la curva se tiene:

$$\frac{R+E}{R} = \sec\left(\frac{D}{2}\right)$$

$$R + E = R \cdot \sec\left(\frac{D}{2}\right)$$

$$E = R \cdot \sec\left(\frac{D}{2}\right) - R \therefore E = R \left[\sec\left(\frac{D}{2}\right) - 1 \right]$$

$$CP = 2 \left(R \cdot \operatorname{sen}\left(\frac{D}{2}\right) \right)$$

$$M = R - R \cdot \cos\left(\frac{D}{2}\right) = R \left(1 - \cos\left(\frac{D}{2}\right) \right)$$

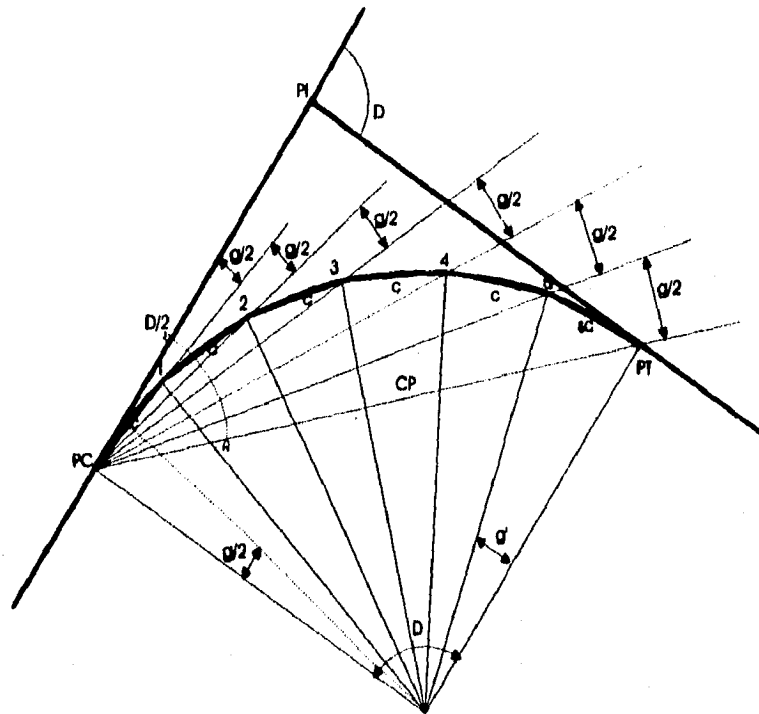
$$M = R \cdot \operatorname{sen}\left(\frac{D}{2}\right)$$

Ya conocida la longitud de la curva, se calculan los cadenamientos para continuarlos por la curva y luego por la siguiente tangente. El cadenamiento del (PI) se conoce gráficamente midiendo en el proyecto, o en el terreno cuando se tiene trazada la línea definitiva.

- CADENAMIENTO (PC) = Cadenamiento (PI) - ST
- CADENAMIENTO (PT) = Cadenamiento (PC) + LC

5.2 Trazo de las curvas en el terreno

El trazo de las curvas con tránsito y cinta se hace por DEFLEXIONES con estación en el (PC) ó (PT).



El origen de las deflexiones será la tangente, es decir, la visual al PI. Como estos ángulos de deflexión son la mitad de los ángulos centrales, para ir marcando cada cuerda que es abarcada por (g) desde el centro, las deflexiones irán variando (g/2). Entonces, poniendo en cero el tránsito y viendo PI, las deflexiones que habrá que ir marcando son g/2, g, 3g/2, 2g, ... hasta llegar a ver el PT, (previamente marcado con la medida de ST a partir del PI).

Para cada deflexión se mide la cuerda desde el punto anterior, y en la intersección estará el nuevo punto de la curva.

El trabajo se puede comprobar:

Angularmente: viendo PT, la graduación del tránsito debe marcar (D/2) con una tolerancia de 01°.

Linealmente: La distancia entre el último punto trazado, y PT, será la (SC) previamente calculada, con una tolerancia de 0.10m.

Para mayor exactitud se recomienda trazar la mitad de la curva desde PC y la otra mitad desde PT para encontrarse al centro, con el objeto de disminuir errores acumulativos que pudieran arrastrarse al hacer el trazo continuo total.

El trazo antes explicado supone que en el PC se inicia la primera cuerda; pero el caso general que se presenta, tratándose de vías de comunicación, es que el PC no calga en un punto de cadenamiento cerrado, y como éste debe de continuarse por el eje de la vía, pasando por la curva, el primer tramo o la primera cuerda que debe marcarse, será lo que le falte al cadenamiento que le toque al PC para llegar a la siguiente estación cerrada. Esto requiere el poder trazar puntos de la curva a una distancia cualquiera del punto de tangencia Inicial PC. La deflexión que corresponda se calcula conociendo la deflexión por metro de curva, la cual se obtiene dividiendo la deflexión para trazar la longitud de una cuerda, entre la cuerda, o la deflexión para trazar toda la curva entre la longitud de la curva.

$$d = \frac{D / 2}{LC} \text{ DEFLEXION POR METRO}$$

Hay tablas ya calculadas con los diferentes grados, radios y deflexiones usuales.

Al hacer el trazo se lleva un registro con los datos necesarios.

Ejemplo:

Datos: $D = 60^{\circ} 30' 1$
 $g = 6^{\circ} 1$
 $PI = 2 + 226.00$

$$ST = R \tan(D/2) = 191.07 + \tan(30^{\circ} 15') = 117.04 \text{ m.}$$

Capítulo V. Curvas Horizontales Simples. ■

Longitud de la curva:

$$LC = \frac{20D}{g} = \frac{20 \times 60.5}{6} = 201.67 \text{ m.}$$

$$PC = PI - ST = 2 + 226.00 - 117.04 = 2 + 108.96$$

$$PT = PC + LC = 2 + 101.96 + 201.67 = 2 + 310.63$$

En el registro de tránsito aparecerán los datos para trazar la curva como sigue:

Estación	P.V.	Deflex.	Datos curva	RMO	RAC	Notas	
	PT 2+310.63	30°15'	R = 191.97 m ST = 117.04 m LC = 201.67 m PI = 2+226.00 D = 60°30'1 g = 6°1	N 12°00'E	N 21°42'E		
	300	28°39'					
	280	25°39'					
	260	22°39'					
	240	19°39'					
	220	16°39'					
	200	13°39'					
	180	10°39'					
	160	7°39'					
	140	4°39'					
	2+ 120	1°39'					
PC 2+108.96				N 72°30'E	N 82°12'E		

En el campo se fijará primero el PI y se cadenará la ST = 117.04 m para fijar el PC, con la mayor precisión tanto en alineamiento como en cadenamiento. Se pasará el aparato al PC. Con los ceros del limbo y la alidada coincidiendo, se visa el PI, se fija el movimiento general y se dará la primera deflexión igual a 1°39', para obtener la primera estación 2 + 120.00 m.

Esta deflexión se calcula: $d = \frac{30°15'}{201.67 \text{ m}} = 09'$

o también: $d = \frac{3^\circ}{20m} = 09'$

entonces $11.04m \times 09' = 1^\circ 39'$

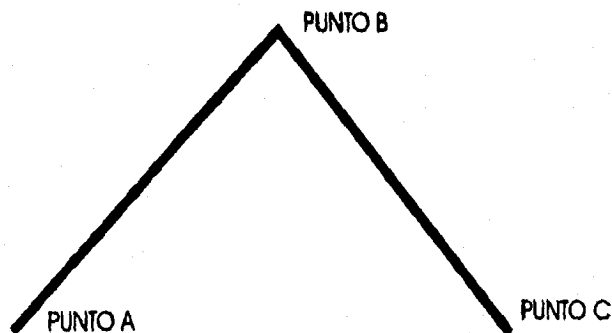
Con la primera deflexión de $1^\circ 39'$, se miden 11.04m; para el segundo punto de la curva se mide 20m. y la deflexión será $4^\circ 39'$, y así sucesivamente hasta llegar a la visual correspondiente a la estación $2 + 300$. Ahora, igual que al principio, se requiere calcular la deflexión necesaria para 10.63 m para visar el PT.

$$10.63 \times 9 = 1^\circ 36'$$

Como comprobación, la última deflexión viendo el PT será igual a la mitad del (D). El PT no se fija mediante el trazo de la curva sino desde el PI, con la ST.

5.3 Trazo de las curvas en AutoCAD

Si tomamos como dato el siguiente trazo:



Capítulo V. Curvas Horizontales Simples. 61

Para darle curvatura a este vértice, eligiendo los criterios anteriores, pero aplicados al programa de AutoCAD, se toma como base el archivo DXF generado con la información del trazo anterior:

0	0	0
SECTION	SECTION	VERTEX
2	2	8
HEADER	ENTITIES	0
.	0	10
.	POLYLINE	4.717505
.	8	20
.	0	7.089801
0	66	30
ENDSEC	1	0.0
0	10	0
SECTION	0.0	VERTEX
2	20	8
TABLES	0.0	0
.	30	10
.	0.0	6.352734
.	0	20
.	VERTEX	5.156809
0	8	30
ENDSEC	0	0.0
0	10	0
SECTION	2.809738	SEQEND
2	20	8
BLOCKS	5.169782	0
0	30	0
ENDSEC	0.0	ENDSEC
		0
		EOF

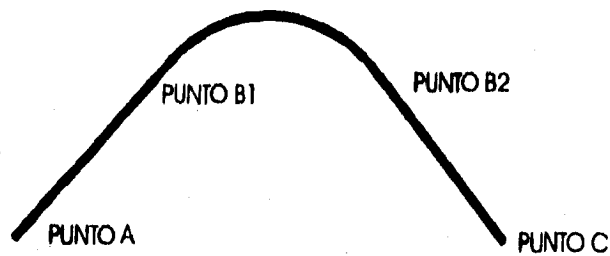
Este archivo está dividido en 4 secciones que son:

- Encabezado
- Definición de Tablas
- Definición de bloques
- Entidades

En la sección de entidades, una Polilínea se forma por grupos de vértices. En un archivo DXF un vértice guarda las coordenadas (x,y,z) identificadas por las cabeceras (10,20,30). para dar curvatura en el punto B, AutoCAD descompone el vértice en dos puntos, los cuales se ubicarán en la misma dirección y sentido del vector que los contiene,

Capítulo V. Curvas Horizontales Simples.

pero en localidad diferente para dar cabida al arco que los une, quedando como la figura siguiente:



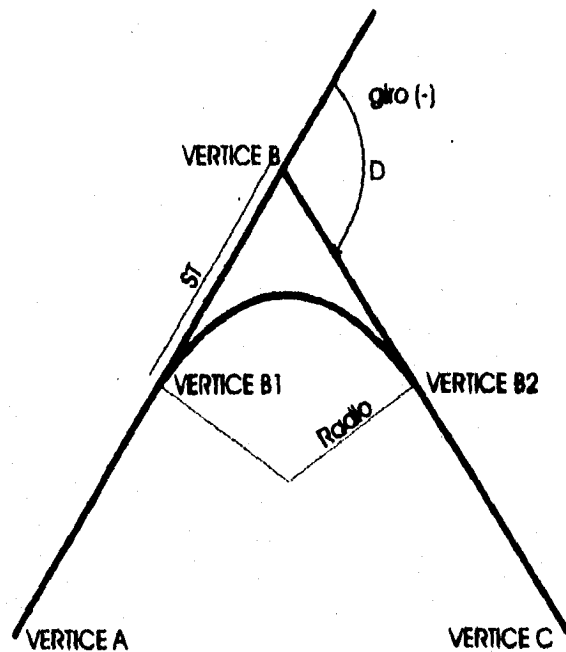
El archivo DXF queda formado con la siguiente información:

0	POLYLINE	0.0
SECTION	8	42
2	0	-0.439769
HEADER	66	0
.	1	VERTEX
.	10	8
.	0.0	0
.	20	10
0	0.0	5.421759
ENDSEC	30	20
0	0.0	6.257308
SECTION	0	30
2	VERTEX	0.0
TABLES	8	0
.	0	VERTEX
.	10	8
.	2.809738	0
.	20	10
0	5.169782	6.352734
ENDSEC	30	20
0	0.0	5.156809
SECTION	0	30
2	VERTEX	0.0
BLOCKS	8	0
0	0	SEQEND
ENDSEC	10	8
SECTION	3.948932	0
2	20	0
ENTITIES	6.316293	ENDSEC
0	30	0
		EOF

Capítulo V. Curvas Horizontales Simples. 43

La cabecera 42 es la que contiene la información del arco en radianes y el signo indica la dirección del giro.

Los elementos que intervienen en el trazo de curvas en AutoCAD relacionados con el diseño de curvas horizontales, son los siguientes:



Con estos elementos, se puede automatizar la operación con ayuda de un programa, mismo que se describe en el siguiente capítulo.

CAPITULO VI. DISEÑO DEL PROGRAMA VIAS.C

6.1 Descripción

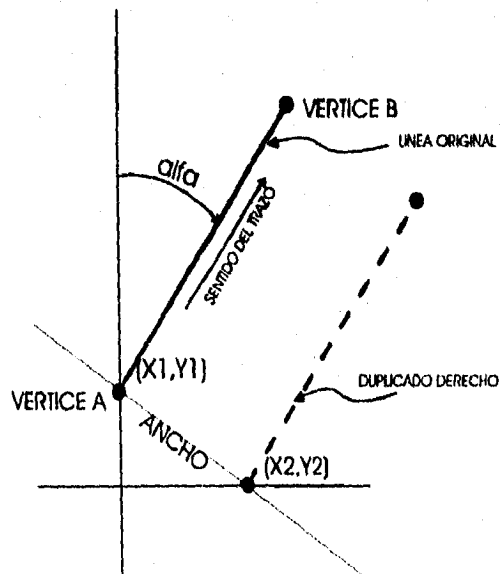
Este programa fue diseñado para agilizar el proceso de edición de dibujo, específicamente en el proceso de suavizar esquinas, duplicar y darle calidad a las líneas, bajo las siguientes consideraciones.

Para generar la copia paralela de los vértices, se aplica la siguiente fórmula:

$$X_2 = X_1 + \text{Ancho} \cdot \text{Sen}(\text{alfa})$$

$$Y_2 = Y_1 - \text{Ancho} \cdot \text{Cos}(\text{alfa})$$

Elementos que se muestran en la siguiente gráfica:



Para curvar un ángulo se aplican las siguientes fórmulas:

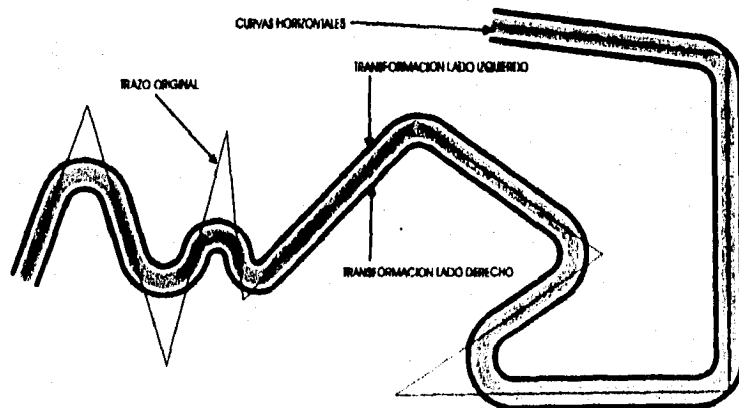
$$\text{Radio} = \frac{ST}{\tan\left(\frac{D}{2}\right)}$$

$$ST = \text{Radio} \cdot \tan\left(\frac{D}{2}\right)$$

$$b = \text{signo} \cdot \tan\left(\frac{D}{4}\right)$$

En donde (b) es el valor del arco en radianes y (signo) indica la dirección del giro (negativo a la derecha y positivo a la izquierda), los demás elementos son descritos en el capítulo anterior.

En la siguiente figura, se presenta el trazo original y los efectos del programa Vias.c, junto con la variante del programa llamado CurvHor.c, estas transformaciones ahorran considerablemente el tiempo de edición.



Opera bajo el sistema operativo MS-DOS y necesita un archivo DXF generado con AutoCAD, éste a su vez, genera otro archivo DXF con los cambios adecuados.

Capítulo VI. Diseño del Programa Vias.c. ■

El programa se ejecuta desde el Sistema Operativo MS-DOS y el formato es el siguiente:

VIAS <Fuente> <Destino> <Layer> <Ancho> [Radio] [Pluma]

Donde:

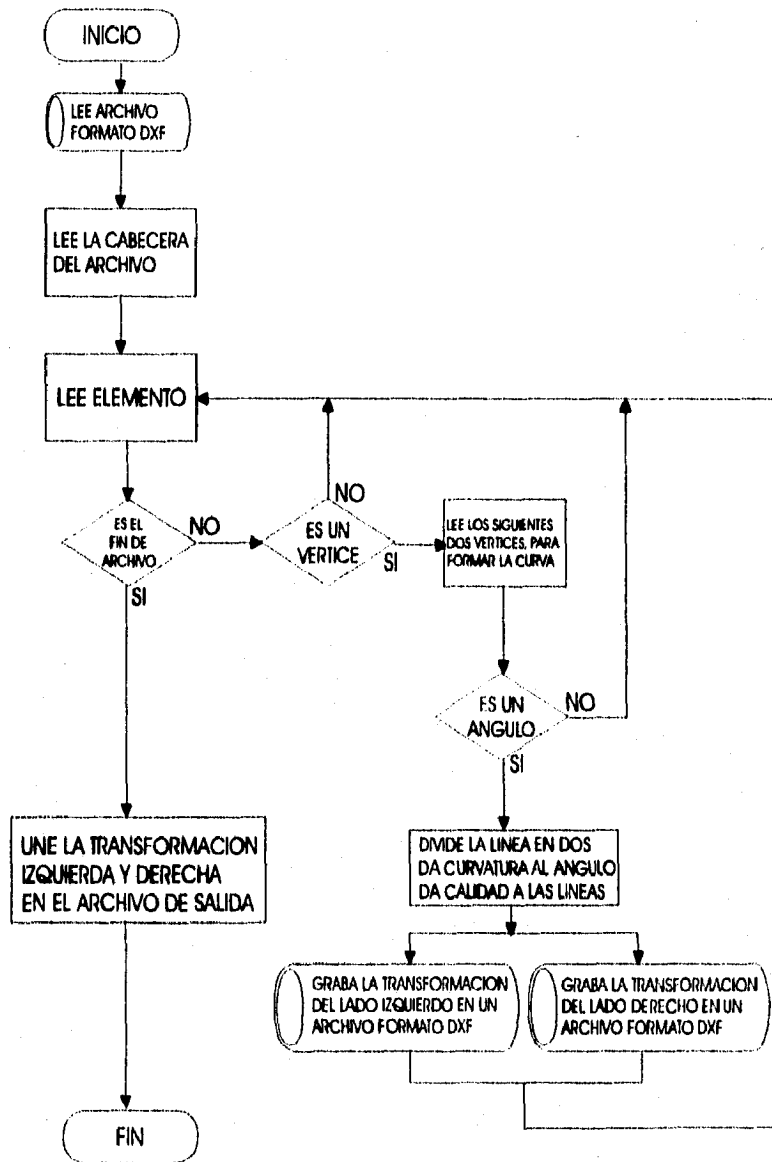
- Fuente** Es el nombre del archivo que contiene la información de los trazos, en formato DXF.
- Destino** Es el nombre del archivo que se genera al ejecutar el comando y donde se guarda la información de las transformaciones, en formato DXF.
- Layer** Es el nombre de la lámina que contiene la información de los trazos a transformar, permitiendo con esto realizar diferentes transformaciones con las diferentes láminas que contenga el archivo original DXF.
- Ancho** Es la distancia en la cual se coloca el duplicado de la línea original, desde el centro de la línea y en ambos lados, izquierda y derecha.
- Radio** Es la longitud del radio máxima para suavizar las curvas, el programa ajusta los arcos al radio óptimo de tal manera que sea posible dar curvatura a segmentos con longitud menor al radio máximo. El valor por omisión es de 0.2 unidades de dibujo.
- Pluma** Es la calidad de línea final en la transformación. El valor por omisión es de 0.01 unidades de dibujo.

Este programa está desarrollado en lenguaje C utilizando el compilador de Borland C++ Ver. 3.0. Los requerimientos mínimos de equipo y software para que se pueda ejecutar son:

- Computadora 8088
- 640 Kbytes de memoria RAM.
- Espacio libre: cinco veces más que el tamaño del archivo original formato DXF.
- Sistema Operativo MS-DOS Ver. 3.0

En el Listado 1 se encuentra el programa completo Vias.c y en el Listado 2 se presenta una variación del programa Vias.c llamado CurvHor.c en el cual no se generan duplicados y se mantiene la línea original y es esta misma la que se curva en sus vértices, con los mismos parámetros y consideraciones del programa anterior.

6.2 Diagrama de flujo




```

remove("fder.dxf");
remove("fizq.dxf");
}

void Cabecera()
{
    char *sentinel;

    if ( !(sentinel == new char[22]) )
    {
        printf("\nERROR : Insuficiente espacio en memoria");
        exit(1);
    }
    read(fs,(void *)sentinel,22);
    write(fs,(void *)sentinel,22);
    delete sentinel;
}

void Genera()
{
    int i;
    long nVertice=0L, fsize;
    DATOS *Dato;
    COORD Punto[3], Pder[2], Pizq[2];

    if ( !(Dato = new DATOS) )
    {
        printf("\nERROR : Insuficiente espacio en memoria");
        exit(1);
    }
    do{
        GetData(Dato);
        WriteData(fs,Dato);
    }while ( strcmp(Dato->llave,"ENTITIES") != 0 &&
    strcmp(Dato->llave,"EOF") != 0);

    while ( strcmp(Dato->llave,"EOF") != 0 )
    {
        while ( strcmp(Dato->llave,"POLYLINE") != 0 &&
        strcmp(Dato->llave,"ENDSEC") != 0 )
            GetData(Dato);
        if ( strcmp(Dato->llave,"POLYLINE") == 0 )
        {
            GetData(Dato);
            if ( strcmp(Dato->llave,LAYER) == 0 )
            {
                Dato->codigo=0x00;
                strcpy(Dato->llave,"POLYLINE");
                WriteData(fder,Dato);
                WriteData(fizq,Dato);

                Dato->codigo=0x08;
                strcpy(Dato->llave,LAYER);

                do{
                    WriteData(fder,Dato);
                    if ( strcmp(Dato->llave,"BRECHA") != 0 )
                        strcpy(Dato->llave,"BRECHA2");
                    WriteData(fizq,Dato);
                    GetData(Dato);
                } while ( strcmp(Dato->llave,"VERTEX") !=
                0);

                nVertice = 0L;
                Dato->codigo=0x28;
                Dato->punto = LineWIDTH;
                WriteData(fder,Dato);
                WriteData(fizq,Dato);
                Dato->codigo=0x29;
                WriteData(fder,Dato);
                WriteData(fizq,Dato);
            }
        }
    }
}

```

```

do{
    GetData(Dato);
    nVertice++;
    Punto[0] = Punto[1];
    Punto[1] = Punto[2];
    Punto[2] = GetPoint();

    if ( nVertice == 2 )
    {
        extremo(&Punto[2],&Punto[1],&Pizq[1],&Pder[1]);
        WriteVertex(fder,&Pder[1]);
        if (
        strcmp(LAYER,"BRECHA") != 0 )
            Brecha = 1;
        WriteVertex(fizq,&Pizq[1]);
    }

    if ( nVertice > 2 )
        Convierte(Punto,Pder,Pizq);
    GetData(Dato);
    } while ( strcmp(Dato->llave,"SEQEND") !=
    0);

    extremo(&Punto[1],&Punto[2],&Pder[0],&Pizq[0]);
    WriteVertex(fder,&Pder[0]);
    if ( strcmp(LAYER,"BRECHA") != 0 )
        Brecha = 1;
    WriteVertex(fizq,&Pizq[0]);

    Dato->codigo=0x00;
    strcpy(Dato->llave,"SEQEND");
    WriteData(fder,Dato);
    WriteData(fizq,Dato);

    GetData(Dato);
    WriteData(fder,Dato);
    if ( strcmp(Dato->llave,"BRECHA") != 0 )
        strcpy(Dato->llave,"BRECHA2");
    WriteData(fizq,Dato);
}

else
{
    if ( nVertice == 0 )
    {
        printf("ERROR : No se encuentra el LAYER :
        %s",LAYER);
        exit(1);
    }
    MergeData();
    if ( (fs = open(OutFile,O_RDWR|_BINARY))
    == -1 )
    {
        printf("\nERROR : No puedo usar el archivo
        %s",OutFile);
        exit(1);
    }
    fsize = filelength(fs)-1L;
    lseek(fs,fsize,SEEK_SET);
    WriteData(fs,Dato);
    GetData(Dato);
    WriteData(fs,Dato);
    printf("\nTotal de v.rtices : %d\n",nVertice);
}

delete Dato;
}

void GetData(DATOS *Dato)
{
}

```

```

char *codigo;
byte grupo;
double punto;
int flag;
char llave[127];

if ( !(codigo = new char[1]) )
{
    printf("nERROR: Insuficiente espacio en memoria");
    exit(1);
}
read(fe,(void *)codigo,1);
Data->codigo = (byte) *codigo;
if ( Data->codigo <= 9 )
    grupo = 1;
else if ( Data->codigo >= 60 && Data->codigo <= 79 )
    grupo = 3;
else
    grupo = 2;

switch( grupo ){
case CADENA :
    GetString(llave,127);
    strcpy(Data->llave,llave);
    break;
case FLOTANTE :
    read(fe,(void *)punto,8);
    Data->punto=punto;
    break;
case ENTERO :
    read(fe,(void *)flag,2);
    Data->flag=flag;
    break;
}
delete codigo;

void GetString(char *String, int n)
{
    char *c;
    int ret;
    int ni=0;

    if ( !(c = new char[1]) )
    {
        printf("nERROR: Insuficiente espacio en memoria");
        exit(1);
    }
    ret = read(fe,(void *)c,1);
    while ( ret != 0 && *c != NULL && ni < n )
    {
        *String=*c;
        String++;
        ni++;
        ret = read(fe,(void *)c,1);
    }
    *String=*c;
    delete c;
}

void WriteData(int f, DATOS *Data)
{
    byte grupo;

    write(f,(void *)&Data->codigo,1);
    if ( Data->codigo <= 9 )
        grupo = 1;
    else if ( Data->codigo >= 60 && Data->codigo <= 79 )
        grupo = 3;
    else
        grupo = 2;
}

```

```

switch( grupo ){
case CADENA :
    write(f,(void *)Data->llave,strlen(Data->llave)+1);
    break;
case FLOTANTE :
    write(f,(void *)&Data->punto,8);
    break;
case ENTERO :
    write(f,(void *)&Data->flag,2);
    break;
}

void MergeData()
{
    int f, nbytes, i, j;
    long fsize;
    char *FileTemp[2] = {"der.dxf","izq.dxf"};
    char buf[127];

    write(f,(void *)CtrlZ,1);
    close(f);
    if ( (f = open(OutFile,O_RDWR|O_BINARY)) == -1 )
    {
        printf("nERROR : No puedo usar el archivo
        %s",OutFile);
        exit(1);
    }

    fsize = filenlength(f)-1L;
    lseek(f,fsize,SEEK_SET);
    write(f,ctrlz,(void *)CtrlZ,1);
    close(f);
    write(f,flag,(void *)CtrlZ,1);
    close(f);

    for ( i=0; i < 2; i++)
    {
        if ( (f = open(FileTemp[i],O_RDWR|O_BINARY)) == -1 )
        {
            printf("nERROR : No puedo usar el archivo
            %s",FileTemp[i]);
            exit(1);
        }

        fsize = filenlength(f)-1L;
        nbytes = 0;
        while ( nbytes < fsize )
        {
            read(f,(void *)buf,sizeof(buf));
            j=0;
            while ( nbytes < fsize && j < sizeof(buf) )
            {
                write(f,(void *)&buf[j++],1);
                nbytes++;
            }
        }
        close(f);
    }
    write(f,(void *)CtrlZ,1);
    close(f);
}

void extremo(COORD *P1, COORD *P2, COORD *Pder,
COORD *Pizq)
{
    int sx,sy=1;
    double x,y,Alfa;
}

```



```

y = P1->y - P2->y;
x = P1->x - P2->x;

Alfa = x != 0 ? atan(y/x) : P1/2;
sx = x != 0 ? x/fabs(x) : 1;
sy = y != 0 ? y/fabs(y) : 1;

*Pder = GetOffset(P2, Alfa, sx, sy, 1);
*Pizq = GetOffset(P2, Alfa, sx, sy, -1);
}

COORD GetOffset(COORD *P, double Alfa, int sx, int sy,
int Side)
{
COORD Retorno;

sx = sx == 0 ? -1 : sx;
sy = sy == 0 ? -1 : sy;
if (Alfa < 0)
{
sx = (-1)*sx;
sy = (-1)*sy;
Alfa = fabs(Alfa);
}

Retorno.x = P->x + Side*sx*ANCHO*sin(Alfa);
Retorno.y = P->y - Side*sy*ANCHO*cos(Alfa);
Retorno.b = P->b;

return Retorno;
}

void WriteVertex(int f, COORD *Punto)
{
DATOS *Dato;

if ( !(Dato = new DATOS) )
{
printf("nERROR : Insuficiente espacio en memoria");
exit(1);
}
Dato->codigo = 0x00;
strcpy(Dato->llave, "VERTEX");
WriteData(f, Dato);

Dato->codigo = 0x08;
if ( Brecha )
{
strcpy(Dato->llave, "BRECHA2");
Brecha = 0;
}
else
strcpy(Dato->llave, "LAYER");
WriteData(f, Dato);

Dato->codigo = 0x0A;
Dato->punto = Punto->x;
WriteData(f, Dato);
Dato->codigo = 0x14;
Dato->punto = Punto->y;
WriteData(f, Dato);
Dato->codigo = 0x1E;
Dato->punto = 0.0;
WriteData(f, Dato);
if ( Punto->b != 0 )
{
Dato->codigo = 0x2A;
Dato->punto = Punto->b;
WriteData(f, Dato);
}
delete Dato;
}

```

```

}

void Convierte(COORD *Punto, COORD Pder[2], COORD
Pizq[2])
{
double Delta;
Alfa1, Alfa2, Alfa;
Grado1, Grado2;
St;
r1x, r1y, r2x, r2y;
L1, L2, RadioOp;
int
sx, sy, signo, i;
char flag=0;
COORD Paux, PuntAntD, PuntAntI, PuntDespD,
PuntDespI;

r1y = Punto[0].y - Punto[1].y;
r1x = Punto[0].x - Punto[1].x;
r2y = Punto[1].y - Punto[2].y;
r2x = Punto[1].x - Punto[2].x;
L1 = sqrt(pow(r1x,2)+pow(r1y,2));
L2 = sqrt(pow(r2x,2)+pow(r2y,2));

//-----Obtención de ngulos Alfa1, Alfa2-----
if (r1x == 0.0)
Alfa1 = P1/2;
else
Alfa1 = r1y == 0.0 ? 0 : atan(r1y/r1x);
if (r2x == 0.0)
Alfa2 = P1/2;
else
Alfa2 = r2y == 0.0 ? 0 : atan(r2y/r2x);

//-----Obtención de grados absolutos Grado1,
Grado2 -----
if (r1y >= 0)
Grado1 = r1x >= 0 ? Alfa1 : P1 + Alfa1;
else
Grado1 = r1x > 0 ? 2*P1 + Alfa1 : P1 + fabs(Alfa1);
if (r2y > 0)
Grado2 = r2x >= 0 ? P1 + Alfa2 : 2*P1 + Alfa2;
else
Grado2 = r2x > 0 ? P1 + Alfa2 : Alfa2;

//-----Obtención del Angulo Delta y del sentido del
giro-----
Delta = Grado1 - Grado2;
signo = fabs(Delta) > P1 ? (-1) * Delta/fabs(Delta) :
Delta/fabs(Delta);
Delta = fabs(Delta) > P1 ? fabs(Delta) - P1 : P1 -
fabs(Delta);

//-----Obtención del Radio Optimo -----
RadioOp = RADIO;
St = RadioOp*tan(Delta/2);
flag = 0;
if ( St > L1 || St > L2*3/4 )
{
flag = 1;
St = L1 < L2*3/4 ? L1 : L2*3/4;
RadioOp = St/tan(Delta/2);
}
if ( flag )
printf("nRadio Optimo = %d", RadioOp);
Paux.b = signo*tan(Delta/4);

Alfa = Alfa1;
Alfa1 = Alfa1 == P1 ? 0 : fabs(Alfa1);
sx = r1x != 0 ? r1x/fabs(r1x) : 0;
Paux.x = Punto[1].x + sx * St * cos(Alfa1);
}

```

```

sy = r1y - 0.7 * r1y / fabs(r1y); 0;
Paux.x = Punto[1].y + sy * S1 * sin(Alfa1);

PuntAntD = Pder[1];
PuntAntI = Pizq[1];
Pder[0] = GetOffset(&Paux, Alfa, sx, sy, 1);
Pizq[0] = GetOffset(&Paux, Alfa, sx, sy, -1);

Alfa = Alfa2;
Alfa2 = Alfa2 == PI ? 0 : fabs(Alfa2);
sx = r2x != 0.0 ? r2x / fabs(r2x) : 0;
Punto[1].x = Punto[1].x + sx * S1 * cos(Alfa2);
sy = r2y != 0.0 ? r2y / fabs(r2y) : 0;
Punto[1].y = Punto[1].y + sy * S1 * sin(Alfa2);

Pder[1] = GetOffset(&Punto[1], Alfa, sx, sy, 1);
Pizq[1] = GetOffset(&Punto[1], Alfa, sx, sy, -1);
PuntDempD = GetOffset(&Punto[2], Alfa, sx, sy, 1);
PuntDempI = GetOffset(&Punto[2], Alfa, sx, sy, -1);

ChecaCruce(&PuntAntD, &Pder[0], &Pder[1], &PuntDempD)
;
ChecaCruce(&PuntAntI, &Pizq[0], &Pizq[1], &PuntDempI);
WriteVertex(fder, &Pder[0]);
if (strcmp(LAYER, "BRECHA") == 0)
    Erucha = 1;
WriteVertex(fizq, &Pizq[0]);
WriteVertex(fder, &Pder[1]);
if (strcmp(LAYER, "BRECHA") == 0)
    Erucha = 1;
WriteVertex(fizq, &Pizq[1]);
}

void ChecaCruce(COORD *Seg1a, COORD *Seg2a,
COORD *Seg1b, COORD *Seg2b)
{
    double Ca, Cb;
    char flag = 0;
    COORD *Cruce;

    if (!(Cruce = new COORD))
    {
        printf("ERROR : Insuficiente espacio en memoria.");
        exit(1);
    }
    Ca = Seg1a->y - ((Seg2a->y - Seg1a->y) / (Seg2a->x -
Seg1a->x)) * Seg1a->x;
    Cb = Seg1b->y - ((Seg2b->y - Seg1b->y) / (Seg2b->x -
Seg1b->x)) * Seg1b->x;
    Cruce->x = (Seg2a->x - Seg1a->x) * (Seg2b->x - Seg1b-
>x) / (Cb - Ca) /
        ((Seg2b->x - Seg1b->x) * (Seg2a->y - Seg1a-
>y) / (Seg2a->x - Seg1a->x)
        + (Seg2b->y - Seg1b->y));
    Cruce->y = (Seg2a->y - Seg1a->y) * Cruce->x / (Seg2a->x -
Seg1a->x) + Ca;
}

```

```

if (Seg2a->x >= Seg1a->x && Seg2a->y >= Seg1a->y)
if (Cruce->x >= Seg1a->x && Cruce->x <= Seg2a->x
&&
    Cruce->y >= Seg1a->y && Cruce->y <=
Seg2a->y)
    flag = 1;
if (Seg2a->x < Seg1a->x && Seg2a->y > Seg1a->y)
if (Cruce->x >= Seg2a->x && Cruce->x <= Seg1a->x
&&
    Cruce->y >= Seg1a->y && Cruce->y <=
Seg2a->y)
    flag = 1;
if (Seg2a->x > Seg1a->x && Seg2a->y < Seg1a->y)
if (Cruce->x >= Seg1a->x && Cruce->x <= Seg2a->x
&&
    Cruce->y >= Seg2a->y && Cruce->y <=
Seg1a->y)
    flag = 1;
if (Seg2a->x < Seg1a->x && Seg2a->y < Seg1a->y)
if (Cruce->x >= Seg2a->x && Cruce->x <= Seg1a->x
&&
    Cruce->y >= Seg2a->y && Cruce->y <=
Seg1a->y)
    flag = 1;
}

if (flag)
{
    Seg2a->x = Cruce->x;
    Seg2a->y = Cruce->y;
    Seg1b->x = Cruce->x;
    Seg1b->y = Cruce->y;
}

delete Cruce;
}

COORD OutPoint()
{
    COORD Punto;
    DATOS *Dato;

    if (!(Dato = new DATOS))
    {
        printf("ERROR : Insuficiente espacio en memoria.");
        exit(1);
    }
    GetData(Dato);
    Punto.x = Dato->punto;
    GetData(Dato);
    Punto.y = Dato->punto;
    GetData(Dato);
    Punto.b = Dato->punto;

    delete Dato;
    return Punto;
}

```



```

Dato->codigo = 10;
Dato->punto = Punto->x;
WriteData(Dato);
Dato->codigo = 20;
Dato->punto = Punto->y;
WriteData(Dato);
Dato->codigo = 30;
Dato->punto = 0x0;
WriteData(Dato);
if ( Punto->b != 0 )
{
    Dato->codigo = 42;
    Dato->punto = Punto->b;
    WriteData(Dato);
}
delete Dato;
}

void Convierta(COORD *Punto)
{
    double Delta,
           Alfa1, Alfa2, Grado1, Grado2,
           St,
           r1x, r1y, r2x, r2y,
           signo,
           L1, L2, RadioOp;
    char flag=0;
    COORD Paux;

    r1y = Punto[0].y - Punto[1].y;
    r1x = Punto[0].x - Punto[1].x;
    r2y = Punto[1].y - Punto[2].y;
    r2x = Punto[1].x - Punto[2].x;
    L1 = sqrt(pow(r1x,2)+pow(r1y,2));
    L2 = sqrt(pow(r2x,2)+pow(r2y,2));

    //-----Obtención de ángulos Alfa1, Alfa2-----
    if ( r1x == 0.0 )
        Alfa1 = PI/2;
    else
        Alfa1 = r1y == 0.0 ? 0 : atan(r1y/r1x);
    if ( r2x == 0.0 )
        Alfa2 = PI/2;
    else
        Alfa2 = r2y == 0.0 ? 0 : atan(r2y/r2x);

    //-----Obtención de grados absolutos Grado1,
    Grado2 -----
    if ( r1y >= 0 )
        Grado1 = r1x >= 0 ? Alfa1 : PI + Alfa1;
    else
        Grado1 = r1x > 0 ? 2*PI + Alfa1 : PI + Alfa1;
    if ( r2y > 0 )
        Grado2 = r2x >= 0 ? PI + Alfa2 : 2*PI + Alfa2;
    else
        Grado2 = r2x > 0 ? PI + Alfa2 : Alfa2;

    //-----Obtención del Angulo Delta y del sentido del
    giro-----
    Delta = Grado1 - Grado2;
    signo = fabs(Delta) > PI ? (-1) * Delta/fabs(Delta) :
    Delta/fabs(Delta);
    Delta = fabs(Delta) > PI ? fabs(Delta) - PI : PI -
    fabs(Delta);

    //-----Obtención del Radio Optimo -----
    RadioOp = RADIO;
    St = RadioOp*tan(Delta/2);
    flag = 0;

```

```

if ( St > L1 || St > L2*3/4 )
{
    flag = 1;
    St = L1 < L2*3/4 ? L1 : L2*3/4;
    RadioOp = St/tan(Delta/2);
}
if ( flag )
    printf("\nRadio Optimo = %f",RadioOp);
Paux.b = signo*tan(Delta/4);

Alfa1 = Alfa1 == PI ? 0 : fabs(Alfa1);
signo = r1x != 0.0 ? r1x/fabs(r1x) : 0;
Paux.x = Punto[1].x + signo * St * cos(Alfa1);
signo = r1y != 0.0 ? r1y/fabs(r1y) : 0;
Paux.y = Punto[1].y + signo * St * sin(Alfa1);
WriteVertex(&Paux);

Alfa2 = Alfa2 == PI ? 0 : fabs(Alfa2);
signo = r2x != 0.0 ? r2x/fabs(r2x) : 0;
Punto[1].x = Punto[1].x - signo * St * cos(Alfa2);
signo = r2y != 0.0 ? r2y/fabs(r2y) : 0;
Punto[1].y = Punto[1].y - signo * St * sin(Alfa2);
}

void GetString(char *String, int n)
{
    char *c;
    int ret;
    int ni=0;

    if ( ! (c = new char[1]) )
    {
        printf("ERROR : Insuficiente espacio en memoria\n");
        exit(1);
    }

    ret = read(fc,c,1);
    while ( ret != 0 && *c != NULL && ni < n )
    {
        *String=*c;
        String++;
        ni++;
        ret = read(fc,c,1);
    }
    *String=*c;
    delete c;
}

void GetData(DATOS *Dato)
{
    char *codigo;
    byte grupo;
    double punto;
    int flag;
    char llave[127];

    if ( ! (codigo = new char[1]) )
    {
        printf("ERROR : Insuficiente espacio en memoria\n");
        exit(1);
    }

    read(fc,codigo,1);
    Dato->codigo = *codigo;
    if ( Dato->codigo != 9 )
        grupo = 1;
    else if ( Dato->codigo >= 60 && Dato->codigo <= 79 )
        grupo = 3;
    else
        grupo = 2;

```

```

switch( grupo ){
case CADENA :
    GetString(llave,127);
    strcpy(Dato->llave,llave);
    break;
case FLOTANTE :
    read(fe,(char *)&punto,8);
    Dato->punto=punto;
    break;
case ENTERO :
    read(fe,(char *)&flag,2);
    Dato->flag=flag;
    break;
}
delete codigo;
}

COORD GetPoint()
{
COORD Punto;
DATOS Dato;

do {
    GetData(&Dato);
} while ( Dato.codigo != 0x0a);
Punto.x = Dato.punto;
GetData(&Dato);
Punto.y = Dato.punto;
GetData(&Dato);
Punto.h = Dato.punto;

return Punto;
}

void Genera()
{
long nVertice=01;
DATOS *Dato;
COORD Punto[3];

if ( !(Dato = new DATOS) )
{
printf("ERROR : Insuficiente espacio en memoria\n");
exit(1);
}

do{
    GetData(Dato);
    WriteData(Dato);
} while ( strcmp(Dato->llave,"ENTITIES") != 0 &&
strcmp(Dato->llave,"EOF") != 0);

while ( strcmp(Dato->llave,"EOF") != 0 )
{
while ( strcmp(Dato->llave,"POLYLINE") != 0 &&
strcmp(Dato->llave,"ENDSEC") != 0 )
    GetData(Dato);
if ( strcmp(Dato->llave,"POLYLINE") == 0 )
{
    GetData(Dato);
if ( strcmp(Dato->llave,LAYER) == 0 )
{
        Dato->codigo=0x00;
        strcpy(Dato->llave,"POLYLINE");
        WriteData(Dato);
    }
}
}
}

```

```

Dato->codigo=0x08;
strcpy(Dato->llave,LAYER);
do {
    WriteData(Dato);
    GetData(Dato);
} while ( strcmp(Dato->llave,"VERTEX")
!= 0);

nVertice = 01;
Dato->codigo = 0x28;
Dato->punto = LineWIDTH;
WriteData(Dato);
Dato->codigo = 0x29;
WriteData(Dato);
do {
    GetData(Dato);
    nVertice++;
    TotVertice++;
    Punto[0] = Punto[1];
    Punto[1] = Punto[2];
    Punto[2] = GetPoint();

    if ( nVertice == 1 )
        WriteVertex(&Punto[2]);
    if ( nVertice > 2 )
    {
        Convierte(Punto);
        WriteVertex(&Punto[1]);
    }
    GetData(Dato);
} while ( strcmp(Dato->llave,"SEQEND")
!= 0 );

WriteVertex(&Punto[2]);
Dato->codigo=0x0;
strcpy(Dato->llave,"SEQEND");
WriteData(Dato);
GetData(Dato);
WriteData(Dato);
}

else
{
WriteData(Dato);
GetData(Dato);
WriteData(Dato);
printf("\n Total de v.rtices : %ld\n",TotVertice);
}

delete Dato;
}

void Cabecera()
{
char *sentinel;

if ( !(sentinel = new char[22]) )
{
printf("ERROR : Insuficiente espacio en memoria\n");
exit(1);
}

read(fe,sentinel,22);
write(fs,sentinel,22);
delete sentinel;
}

```

CAPITULO VII. IMPRESION EN FOTOCOMPOSICION

7.1 Fotocomposición

Para el proceso de imprenta de grandes volúmenes de ejemplares, se necesitan métodos sofisticados con maquinaria especial, así, existen métodos tales como serigrafía, flexografía, offset, etc. Para la mayoría de estos métodos, entre otras cosas, se necesita de una impresión en un negativo de las entidades que se pretenden imprimir, en este aspecto, hoy en día existen equipos de impresión que permiten imprimir cualquier contenido de un trabajo realizado en una computadora, en un dispositivo. Este equipo hace bastante ágil el trabajo de imprenta, ya que se ahorran muchos pasos intermedios que anteriormente se realizaban.

Estas impresoras, que han venido a revolucionar el mundo de la imprenta, utilizan un método de impresión llamado PostScript, y es por esto, que es posible imprimir nuestro trabajo realizado en una computadora directamente a un dispositivo, como si fuera a una hoja simple.

7.2 El lenguaje PostScript

PostScript es el nombre de un lenguaje de programación de computadora desarrollado originalmente por Adobe Systems

Incorporated para comunicar información gráfica de alto nivel a Impresoras láser digitales. PostScript puede imprimir cualquier tipo de documento incluyendo texto, gráficas, imágenes y color. Y es compatible con todo tipo de computadoras y la mayoría de los paquetes de aplicación. La seguridad de una Impresora PostScript, es que ésta siempre va a ser compatible, no importa qué hardware y software se esté utilizando, no importa qué se necesite imprimir. Adobe PostScript siempre estará allí ahora y para siempre.

PostScript es ... Cualquiera cosa. Cualquiera lugar. Cualquiera hora. El lenguaje de descripción de página de PostScript es tan simple y tan ilimitado como eso. Y no hace nada más que quitar los límites en cómo estás disponible para expresarte a tí mismo en una impresión. No importa en qué ambiente de computación estés trabajando. O qué dispositivo de salida estés utilizando. O dónde esté localizada. O cuándo lo quieras imprimir.

7.3 Características del lenguaje PostScript

El PostScript es un lenguaje simple de programación interprete con poderosas capacidades gráficas. Su aplicación primordial es para describir la apariencia del texto, formas gráficas, e imágenes simplificadas en páginas impresas o desplegadas. Un programa en este lenguaje puede comunicar una descripción de un documento desde un sistema de composición a un sistema de impresión o controlar la apariencia del texto y gráficas en un despliegue. La descripción es de alto Nivel e independiente del dispositivo.

La descripción de páginas y las capacidades de interacción gráficas del lenguaje PostScript incluye las siguientes características, las cuales pueden ser usadas en cualquier combinación:

- Formas arbitrarias hechas de líneas, arcos, rectángulos y curvas cúbicas. Tales formas pueden interceptarse por sí mismas y tener secciones discontinuas y vacíos.
- Operadores de pintura que permiten a una forma ser delineada con líneas de cualquier tipo, rellena con cualquier color, o usada como una librería para ser usada en otras gráficas. Los colores pueden ser usados en una variedad de caminos: escalas de grises, RGB, CMYK, y CIE. Otras características especiales son inclusive moldeadas como un tipo especial de colores: repetición de patrones, mapas de colores, y separaciones.
- Texto con gráficas completamente integradas. En el modelo gráfico del lenguaje PostScript, los caracteres de texto en tipo de letra estándar y definidas por el usuario son tratados como formas gráficas que pueden ser operados con cualquiera de los operadores normales de gráficos.
- Imágenes simplificadas derivadas de formas naturales (tales como fotografías escaneadas) o sintéticamente generadas. El lenguaje PostScript puede describir imágenes en cualquier resolución y acorde a una variedad de modelos de colores. Esto proporciona una variedad de caminos para reproducir imágenes en un dispositivo de salida.
- Un sistema de coordenadas normal que soporte todas las combinaciones de transformaciones lineales, incluyendo traslación, escala, rotación, reflexión, y simetría. Estas transformaciones se aplican uniformemente a todos los elementos de una página, incluyendo texto, formas gráficas, e imágenes.

Una descripción de página en lenguaje PostScript puede ser ilustrada en una impresora, pantalla, u otro dispositivo que interprete el PostScript. Como un interprete ejecuta comandos para pintar letras, formas gráficas, e imágenes, este convierte la descripción del lenguaje PostScript de alto nivel dentro del formato de datos raster de bajo nivel para el dispositivo electrónico particular.

Normalmente, programas de aplicación tales como sistemas de composición de documentos, ilustradores y sistemas de diseño asistido por computadoras generan descripción de páginas de lenguaje PostScript automáticamente. Los programadores generalmente escriben programas en lenguaje PostScript solamente cuando crean nuevas aplicaciones. Sin embargo, en situaciones especiales un programador puede escribir un programa en lenguaje PostScript para tomar ventaja de las capacidades del lenguaje PostScript que no están accesibles a través de un programa de aplicación.

Las extensas capacidades gráficas del lenguaje PostScript están encerradas en el marco de trabajo de un lenguaje de programación de propósito general. El lenguaje incluye un lote de tipos de datos convencionales, tales como números, arreglos, y cadenas; primitivas de control, tales como condicionales, ciclos, y procedimientos; y algunas características inusuales, tales como diccionarios. Estas características permiten al programador de aplicaciones definir operaciones de alto nivel que se acercan más a las necesidades de la aplicación y entonces generar comandos para invocar estas operaciones de alto nivel. Tales descripciones son más compactas y fáciles de generar que una escritura entera en términos de un lote fijo de operaciones básicas.

Programas en lenguaje PostScript pueden ser creados, transmitidos, e interpretados en la forma de código ASCII (American Standard Code International Interchange) como esta escritura misma. El lenguaje entero puede ser descrito en términos de letras impresas y espacios en blanco. Esta representación les conviene a los programadores para crear, manipular, y comprender. Esto también facilita el almacenamiento y transmisión de archivos a través de diversas computadoras y sistemas operativos, independientemente del equipo en el que fue creado.

También hay formas de código binario del lenguaje para el uso en medios controlados, por ejemplo, cuando el programa está asegurado de una ruta completa de comunicaciones transparentes tales como en el sistema de despliegue de PostScript. Adobe recomienda estricta

inclinación a la representación ASCII de programas en lenguaje PostScript para almacenamiento de archivos o intercambio de documentos.

7.4 Evolución del lenguaje PostScript

Desde su introducción en 1985, el lenguaje PostScript ha sido extendido considerablemente para garantizar poder, eficiencia, y flexibilidad de programación. Típicamente, esta extensión del lenguaje ha sido diseñada para adaptar el lenguaje PostScript a nuevas tecnologías de imágenes o sistemas de medio ambiente. Mientras estas extensiones han introducido significativamente nueva funcionalidad y flexibilidad al lenguaje, el modelo básico de imágenes permanece sin cambios. Las principales extensiones son:

- *Color.* Las extensiones de color proporcionan un modelo de color (CMYK) Cyan-Magenta-Yellow-Black para colores específicos y un operador *colorimage* para iluminar imágenes con colores probados. Ellos incluyen controles adicionales de interpretación para dispositivos de salida de color.
- *Tipos de letra compuestos.* La extensión de tipos de letra compuestos amplían la facilidad básica de tipos de letra para soportar lotes de letras que son muy largas o tienen requerimientos complejos para codificar o posicionamiento de las letras.
- *Despliegue de PostScript.* El sistema de despliegue de PostScript permite a las aplicaciones de estaciones de trabajo usar el lenguaje PostScript como un modelo de imágenes para manejar la apariencia del despliegue. Algunas de las extensiones son especiales para interactuar con las aplicaciones de despliegue, tales como ejecución concurrente y soporte para sistemas de ventanas. Otras extensiones

Capítulo VII. Impresión en Fotocomposición. ■

son más generales y son intencionadas para proporcionar aceleración o conveniencia en la programación.

En adición a las extensiones del lenguaje descritas con anterioridad, existen otras adiciones menores al lenguaje, tales como extensión de sistema de archivos para soportar productos que incluye discos o cartuchos.

Mientras que el lenguaje PostScript es una definición estándar, no todos los interpretes de PostScript incluyen todas las características del lenguaje. Productos que contienen el software de PostScript de Adobe se pueden dividir en las siguientes categorías:

- Nivel 1 esta implementación incluye todos los operadores del Nivel 1. Algunas implementaciones del Nivel 1 incluyen una o más extensiones del lenguaje. Por ejemplo, soporte para impresoras PostScript a color operadores del Nivel 1 más las extensiones de color.
- Nivel 2 esta implementación incluye todos los operadores del Nivel 1 y Nivel 2.
- Sistema de despliegue de PostScript incluye los operadores del despliegue de PostScript y se pueden basar en las implementaciones ya sea del Nivel 1 o Nivel 2. El sistema de despliegue PostScript basado en el Nivel 1 incluye las extensiones de despliegue de PostScript y color mencionadas anteriormente, y en algunos casos otras extensiones también. El sistema de despliegue de PostScript basado en el Nivel 2 incluye todos los operadores del Nivel 2.

7.5 Un vistazo al PostScript Nivel 2

En un esfuerzo por unificar todas las extensiones del lenguaje, PostScript Nivel 2 introduce un pequeño número de características del lenguaje.

- **Diccionarios.** Muchos operadores del Nivel 2 esperan un operador de diccionario que contiene pares de valores clave que especifican parámetros al operador. Estas características del lenguaje controlado de esta forma incluye medios tonos, imágenes, formas, patrones, e instalación de dispositivos. Esta organización permite una parametrización opcional y una extensibilidad a futuro. Por conveniencia en el uso de tales operadores, la sintaxis del lenguaje PostScript incluye nuevas instrucciones, <<and>>, para construir el diccionario que contiene la unión del par de palabras clave.
- **Manejo de Memoria.** Ahora es posible eliminar entradas individuales del diccionario y eliminar definiciones de tipo de letra en diferente orden en el cual ellos fueron definidos. Memoria Virtual (VM) es utilizada automáticamente para componer objetos que no son accesibles por su tamaño. En general, la memoria es más eficiente intercambiando tamaños para diferentes usos y las restricciones arbitrarias de la memoria han sido eliminadas.
- **Recursos.** Un recurso es una colección de objetos nombrados que pueden localizarse ya sea en la memoria virtual o pueden ser localizados y cargados en la memoria virtual dependiendo de la demanda. Allí son separados por categoría de recursos con espacios de nombre independiente, por ejemplo, tipos de letra y formas son categorías de recursos diferentes. El lenguaje incluye facilidades convenientes para la localización y manejo de recursos.
- **Filtros.** Un filtro transforma los datos como si éstos hubieran sido leídos o escritos a un archivo. El lenguaje soporta filtros para códigos ASCII o datos binarios, comprimidos o expandidos, y subarchivos

empotrados. Propiamente usados, estos filtros reducen el costo de almacenamiento y transmisión de la descripción de página, especialmente aquellos que contienen una simple imagen.

- *Decodificación Binaria.* En adición a la decodificación estándar ASCII, la sintaxis del lenguaje incluye dos representaciones de código binario. Estas decodificaciones binarias proporcionan eficiencia de generación, representación, e interpretación. Sin embargo, ellas son menos portables que la decodificación ASCII y su uso está restringido solamente en control de medio ambiente.
- *Rutas de Usuario.* Una ruta de usuario es un procedimiento propio que consiste enteramente en operadores de construcción de rutas y sus operandos coordinados. Los operadores de rutas de usuario perfeccionan la construcción de rutas y pintar conos como una operación simple; ambas son convenientes y eficientes. Hay un espacio de ruta de usuario para optimizar la interpretación que es invocado en repetidas ocasiones. También hay algunos operadores convenientes para pintar rectángulos.
- *Formas.* Una forma es una descripción propia de alguna gráfica, texto e imágenes que van a ser pintadas en múltiples ocasiones en cada una de las páginas o en ocasiones en diferentes localidades en una página simple. Hay un espacio de forma para optimizar los usos repetidos de la misma forma.
- *Espacios de color.* El color puede ser especificado de acuerdo a una gran variedad de sistemas de color, incluyendo escalas de grises, RGB, CMYK, y CIE. Patrones, mapa de colores, y separaciones también son moldeados como espacios de color. El espacio de color es ahora un parámetro explícito del estado gráfico.
- *Espacios de color CIE.* El lenguaje soporta diversos espacios de color de diversos dispositivos independientes basados en el CIE 1931 espacio (XYZ), un sistema para especificar valores de color en la misma forma que es relacionado a la percepción de la vista humana. En una especificación de color dado de base CIE se pueden esperar resultados

Capítulo VII. Impresión en Fotocomposición. ■

consistentes en diferentes dispositivos de salida de color, independientemente de la variación en el mercado de la tecnología, tintas colorantes o pantallas de fósforo.

- *Patrones.* Es posible pintar con patrones también como con colores sólidos. Cuando el color activo es un patrón, operadores de pintura tales como "paint" que es producido por la repetición de pequeñas figuras gráficas, llamado un patrón celda, a intervalos fijos en X y Y para cubrir las áreas que van a ser iluminadas. La apariencia de un patrón celda está definida por un procedimiento de lenguaje PostScript arbitrario; el cual puede incluir gráficas, texto, e imágenes simplificadas. Hay un patrón apartado para optimizar los usos repetidos del mismo patrón.
- *Imágenes.* Hay diversas ampliaciones para facilitar la impresión de imágenes simplificadas: usando cualquier espacio de color, valores compuestos de 12 bits, uso directo de archivos como fuente de datos, y decodificación adicional y opciones de presentación.
- *Otros Operadores de Texto y Gráficas.* Existe otra gama diversa de operadores nuevos que optimizan la velocidad. Objetos de estado gráfico permiten un cambio rápido a través de estados gráficos arbitrarios. El operador "selectfont" optimiza el intercambio de tamaños de tipo de letra.
- *Configuración del dispositivo.* El operador "setpagedevice" proporciona un marco de trabajo independiente del dispositivo para especificar los requerimientos de la descripción de página y para controlar las dos características estándar, tales como el número de copias, y la característica opcional de los dispositivos, tales como impresión dúplex.
- *Parámetros de Interpretación.* Operadores administrativos, tales como configuración del sistema y cambios de la entrada y salida de los parámetros del dispositivo, ahora están organizados en una forma más sistemática. La localidad de memoria y otros recursos para propósitos específicos están bajo el control del software. Por ejemplo, los

Capítulo VII. Impresión en Fotocomposición. ■

parámetros que controlan el tamaño máximo de memoria ha de ser usada por el VM. Apartado para tipo de letra, forma, patrón y pantallas con medios tonos.

CAPITULO VIII. APLICACION

8.1 Procedimiento

En un país como el nuestro, en pleno desarrollo, en donde la construcción de nuevas vías de comunicación; entre otras cosas, es indispensable y necesaria, se incrementa la necesidad de nuevos medios eficientes de comunicación, difusión e información, sobre todo, de los caminos construidos y en proceso de construcción. La elaboración de un Atlas de Carreteras eficiente, vanguardista y de calidad, resuelve parte de las exigencias de hoy en día.

Como aplicación, se describirá el procedimiento para dibujar un Atlas de Carreteras, con el siguiente equipo y material:

- Computadora 486 DX2 66Mhz, 8MB RAM, 500 MB Disco Duro, Sistema de Video Super VGA.
- Tableta digitalizadora de 12"
- Programa AutoCAD
- Programa VIAS.c y variantes.
- Impresora con Sistema de Fotocomposición
- Fotografías aéreas a escala de las carreteras y localidades actuales, en la República Mexicana.

8.2 Paso 1. Definición de Láminas

Definir las láminas (LAYERS) a utilizar, con sus propiedades particulares, mismas que se describen en la siguiente tabla:

Nombre del Layer	Color	Descripción
AUTOP	Verde	Lámina que contendrá todos los trazos que representen a las Autopistas.
CARRPRIN	Cyan	Lámina que contendrá todos los trazos que representen a las carreteras Principales.
CARRSEC	Azul	Lámina que contendrá todos los trazos que representen a las carreteras Secundarias.
TERRA	Rojo	Lámina que contendrá todos los trazos que representen a las carreteras en Terracería.
BRECHA	Amarillo	Lámina que contendrá todos los trazos que representen a las Brechas.
FERRO	Café	Lámina que contendrá todos los trazos que representen a las vías ferroviarias.
POB500	Negro	Lámina que contendrá todos los símbolos que representen a las poblaciones con más de 500,000 habitantes.
POB100	Negro	Lámina que contendrá todos los símbolos que representen a las poblaciones que se

		encuentren en el rango de 100,001 a 500,000 habitantes.
POB40	Negro	Lámina que contendrá todos los símbolos que representen a las poblaciones que se encuentren en el rango de 40,001 a 100,000 habitantes.
POB15	Negro	Lámina que contendrá todos los símbolos que representen a las poblaciones que se encuentren en el rango de 15,001 a 40,000 habitantes.
POB5	Negro	Lámina que contendrá todos los símbolos que representen a las poblaciones que se encuentren en el rango de 5,001 a 15,000 habitantes.
POB0	Negro	Lámina que contendrá todos los símbolos que representen a las poblaciones con el número de habitantes no mayor a 5,000.

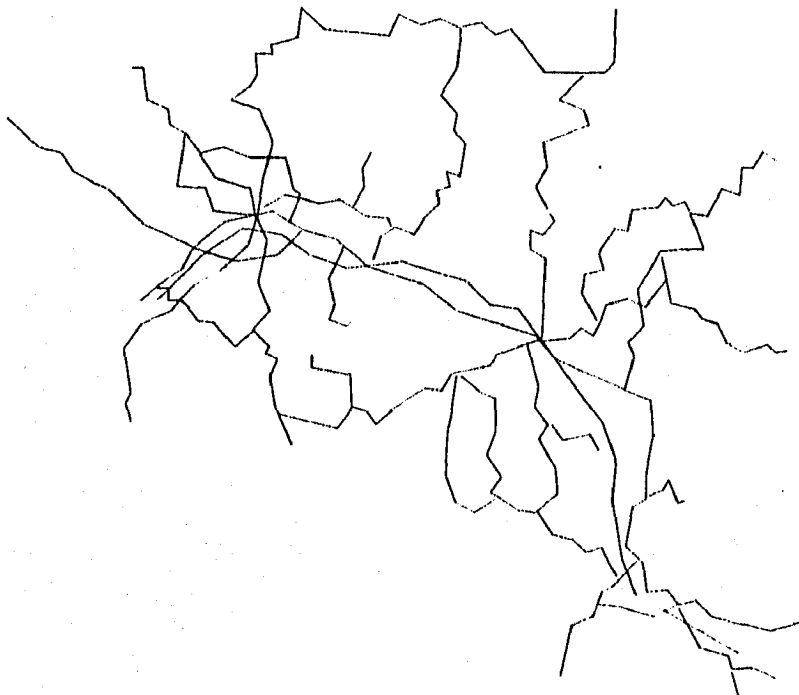
Estas láminas quedarán almacenadas en un archivo de AutoCAD llamado Atlas1 con extensión DWG, a manera de plantilla, para ser utilizado en todas las digitalizaciones subsecuentes.

8.3 Paso 2. Digitalización de Caminos

Utilizando la tableta digitalizadora, se procede a trazar las diferentes carreteras que se presenten en las fotografías aéreas, cargando el archivo Atlas1 para utilizarlo como plantilla, por las definiciones de las láminas que contiene. Las láminas se activarán como actuales, en la

medida en que se solicite por el tipo de trazo que se esté realizando, es decir, si el trazo es de una carretera principal, en AutoCAD se deberá activar como actual la lámina CARRPRIN.

Siguiendo estas instrucciones, un ejemplo de los trazos realizados en la tableta, se muestran como sigue:



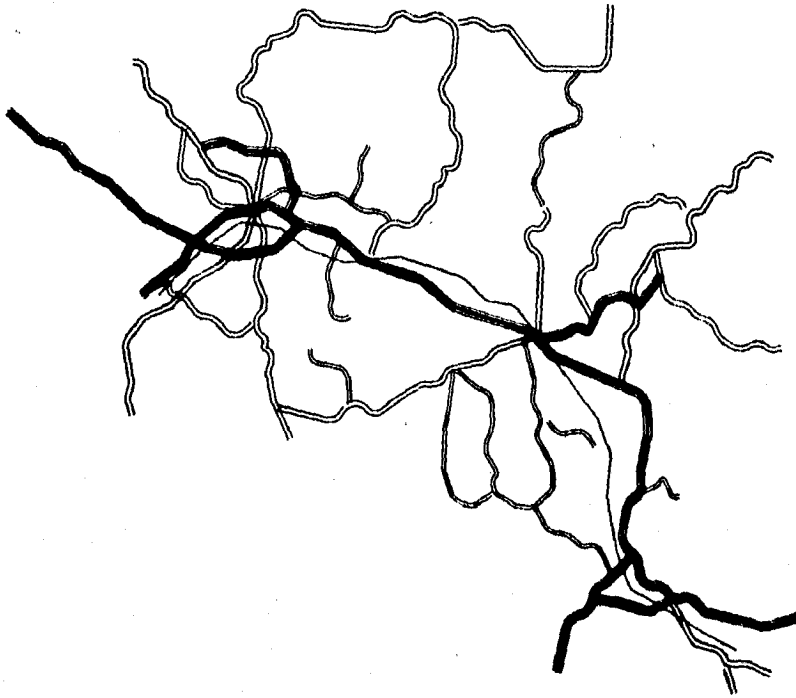
En estos trazos, no se percibe el tipo de carretera que se pretende presentar, esta distinción está reconocida por la lámina que contiene cada uno de los trazos, lámina que está claramente definida en AutoCAD. Esta separación de tipos de carretera, no es suficiente para una interpretación correcta del dibujo, es necesario darle una presentación adecuada, una suavización en los vértices y una calidad en las líneas acorde a la calidad de los caminos, misma que se aplica en el paso siguiente.

8.4 Paso 3. Dar Calidad a los Trazos

Para dar calidad a los trazos digitalizados, se emplean los programas VIAS.c y CurvHor.c, éstos se ejecutarán para cada una de las láminas que contienen carreteras, de esta forma, se genera el archivo Trazo1.DXF con la información de los trazos anteriores y se aplican los programas de la siguiente forma:

vias	trazo1.dxf	a1.dxf	autop	0.05	0.2	0.01
curvhor	trazo1.dxf	a11.dxf	autop		0.2	0.03
vias	trazo1.dxf	a2.dxf	carrprin	0.0325	0.2	0.015
vias	trazo1.dxf	a3.dxf	carrsac	0.03	0.2	0.01
vias	trazo1.dxf	a4.dxf	terra	0.02	0.2	0.01
vias	trazo1.dxf	a5.dxf	brecha	0.02	0.2	0.01
curvhor	trazo1.dxf	a6.dxf	ferro		0.2	0.01

Regresando al programa de AutoCAD, en la plantilla Atlas1, se cargan las transformaciones efectuadas, quedando el dibujo con la siguiente presentación:

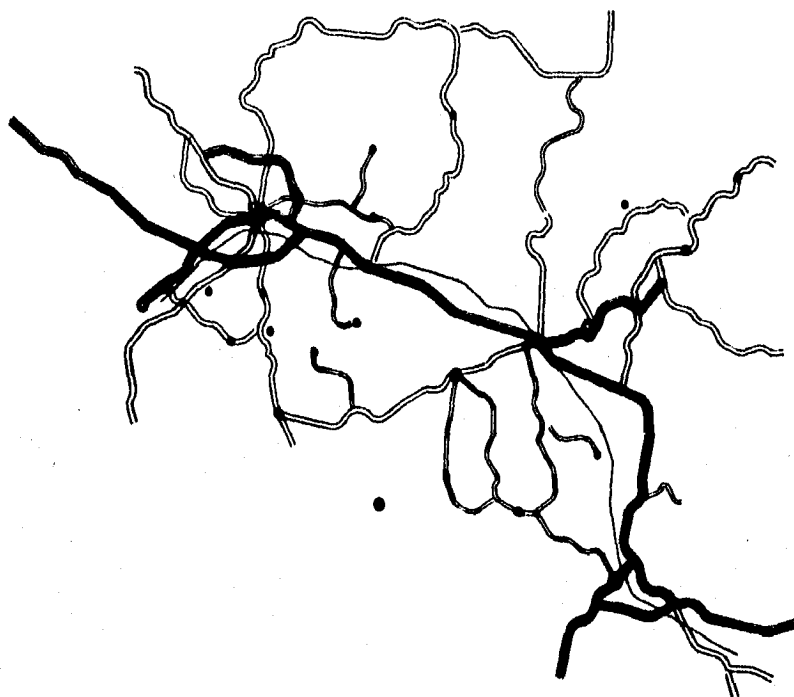


8.5 Paso 4. Colocación de Símbolos

La colocación de símbolos se realiza inmediatamente después de la transformación de los trazos, en este caso, los únicos símbolos que se colocarán serán los representativos de las poblaciones, los cuales tienen la siguiente nomenclatura:

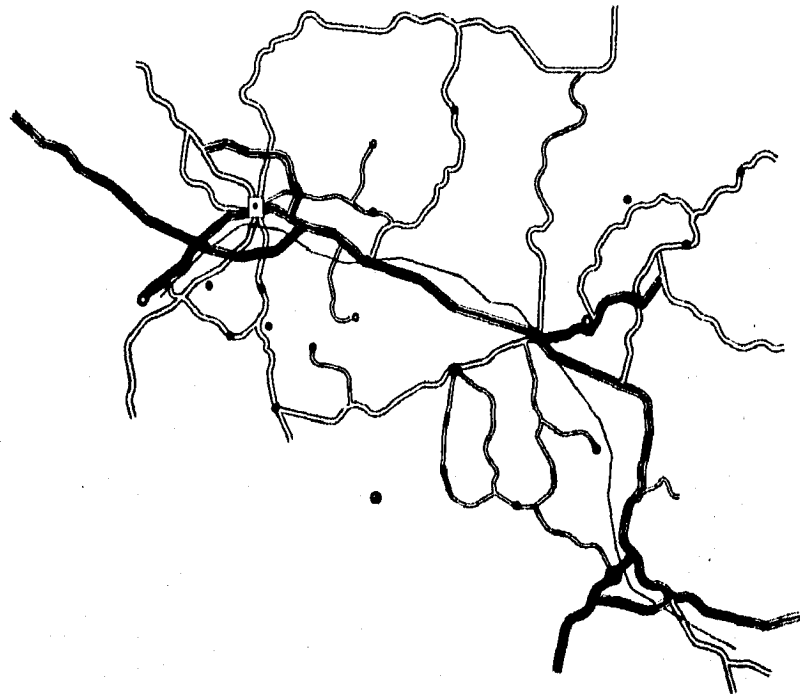
- MAS DE 500,000 HAB.
- DE 100,001 A 500,000 HAB.
- DE 40,001 A 100,000 HAB.
- DE 15,001 A 40,000 HAB.
- DE 5,001 A 15,000 HAB.
- DE 1 A 5,000 HAB.

Una vez dibujados en AutoCAD, utilizando copias múltiples, se colocan cada uno de los símbolos en su respectivo lugar en el dibujo, con la precaución de activar la lámina correspondiente para cada tipo de población. Al concluir, el dibujo quedará como sigue:



8.6 Paso 5. Edición

El trabajo de edición consiste en eliminar los excedentes de los trazos, cortar los cruces en las vías y corregir cualquier mal formación de las transformaciones, para darle al trabajo la presentación definitiva. La presentación final de nuestro ejemplo, se muestra en la siguiente figura:



8.7 Paso 6. Rotulación

La presencia de nombres, textos y notas en el dibujo, es indispensable para la pronta ubicación de cualquier punto en el dibujo. Con la rotulación, el dibujo se presenta de la siguiente forma:

CONCLUSIONES

El acelerado ritmo de crecimiento de la computación, permite y obliga a rediseñar procedimientos y optimizar constantemente los procesos, a la medida de los objetivos, alcances y metas de cada empresa.

Este proceso de dibujo, está finamente diseñado, analizado concienzudamente, pero no descarta la posibilidad de mejorarse, sobre todo cuando el avance de la tecnología lo permita.

En este trabajo se logran describir todos los aspectos que intervienen para la obtención del objetivo principal del proceso, que es generar un dibujo, utilizando como herramienta un equipo especial de cómputo.

El programa de apoyo, está desarrollado en Lenguaje C, y reduce considerablemente el trabajo de edición de un dibujo, aproximadamente a la cuarta parte del tiempo que se le podría dedicar, y es una facilidad que brinda el paquete de AutoCAD, el cual se caracteriza por su amplia versatilidad.

BIBLIOGRAFIA

1. PostScript Language Reference Manual. Adobe Systems Incorporated. Addison-Wesley Publishing Company, Inc. Second Edition, United States of America, July 1991.
2. PostScript Language Program Design. Adobe Systems Incorporated. Addison-Wesley Publishing Company, Inc. United States of America, march 1992.
3. Topografía. Montes de Oca. Alfaomega. Cuarta Edición. México, D.F. Julio 1989.
4. AutoCAD R 10 Reference Manual. Autodesk, Inc. Command Technology Corp. United States of America, february 1989.
5. Borland C++ Ver. 3.0 Reference Manual. Borland International, Inc. United States of America, 1991.
6. Internet. World Wide Web.