

Necesito simular representaciones espaciales y animarlas,
¿Cómo lograrlo?

Autor: MC. Carlos Gutiérrez Barea.

ABSTRACT:

In The following article is approached the application of the ellipse in spatial representations of the geometric elements (the point, the stright line and the plan) and of some simple geometric solids (the pyramid and the prism). Furthermore, the offers the possibility to give animation to such objets represented in third dimension. The effects that are achieved through this flat figures are of much usefulness in the use of computer as a means of teaching, fundamentally in the effect of animation simulation of given processes and phenomena.

RESUMEN:

En el siguiente artículo se aborda la aplicación de la Elipse en representaciones espaciales de los elementos geométricos (el Punto, la línea Recta y el Plano) y de algunos sólidos geométricos sencillos (la pirámide y el prisma), además, la posibilidad que ésta ofrece para dar animación a dichos objetos representados en tercera dimensión. Los efectos que se logran a través de esta figura plana son de mucha utilidad en el uso de computadora como medio de enseñanza, fundamentalmente en el efecto de simulación de animación de determinados procesos y fenómenos.

Introducción.

Sin dudas, el ordenador como medio de enseñanza supera a los anteriores medios empleados con fines docentes en la posibilidad de interacción hombre-máquina (alumno-medio de enseñanza), por sus potencialidades, en la simulación y en la implementación de animaciones, ajustadas a determinados datos, lo que nos acerca aún más a los procesos y fenómenos de estudio de la realidad objetiva.

Hoy el desarrollo de los equipos de computo es extremadamente acelerado en todas las vertientes y aplicaciones de la vida social (en las ciencias, el arte, el deporte, la industria y el comercio), por las enormes ventajas que éstos ofrecen en cuanto a posibilidades (gráficas, de textos, de cálculos, operaciones matemáticas y estadísticas, sonoras, etc.), confiabilidad, seguridad, velocidad operacional, entre otras, encontrando un campo virgen fundamentalmente en la actividad docente.

Mucho se ha avanzado en las representaciones espaciales en estos equipos donde la pantalla de nuestros monitores es un plano más. Las multimedia, la realidad virtual, etc., entre otras son tecnologías de avanzada que influyen en nuestra percepción sensorial y por consiguiente en nuestro conocimiento. La gama de colores (cromatismo), sonidos, efectos, y animaciones, entre otras, son cualidades que cobran vigencia en estas nuevas tecnologías.

Las herramientas diseñadas para el logro de determinados efectos, en estos equipos de cómputo, vienen ajustadas a determinados lenguajes, las que se ofertan empaquetadas y/o enmascaradas fundamentalmente en lenguajes de versiones avanzadas, donde conociendo su sintaxis, convocándolas y transfiriéndoles determinados parámetros se logra el efecto deseado; pero, ¿cómo se lleva a cabo?, y si no las tenemos a nuestra disposición ¿cómo implementar el mismo efecto?.

Ante la necesidad, por ejemplo, de representar un trabajo donde se materialicen determinados efectos (de animación) para demostrar cómo se lleva a cabo el abatimiento de los planos de proyección, en el estudio sobre la teoría de las proyecciones, y de elementos proyectados sobre éstos, si no se dispone de las herramientas a las que hacíamos referencia anteriormente o de bibliografía al respecto, se hace necesario tomar en cuenta el estudio sobre la trayectoria de elementos y cuerpos que se mueven en el espacio, hasta llegar a una observación:

Si hacemos rotar un punto A alrededor de un eje inmóvil o-ó, a partir de un punto fijo K, llamado centro, éste lo hará en el plano T perpendicular al eje de giro y el mismo generará una figura plana llamada circunferencia con centro en K y radio KA (Fig. 1), en la cual todos los puntos de su trayectoria equidistarán del centro de rotación. Esta figura plana tendrá características proyectivas muy bien definidas, en dependencia de su posición en el espacio o del plano donde se encuentre o donde se proyecte. Por ejemplo:

Si la circunferencia es perpendicular a un plano, entonces se proyectará, en éste, cual un segmento de recta.

Si la circunferencia es paralela a un plano, ésta se proyectará en dicho plano idéntica, en verdadera magnitud, como una circunferencia.

Si la circunferencia está inclinada respecto a un plano, ésta se proyectará, en dicho plano, como una Elipse.

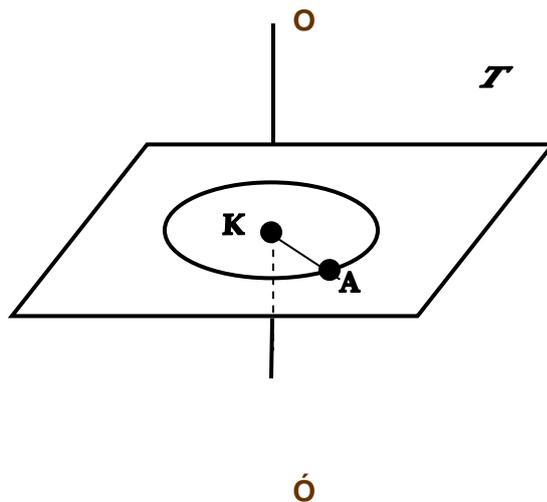


Fig. 1

Tras muchas pruebas y análisis para ver cómo se podía en un plano (pantalla) representar figuras en tercera dimensión, simulando su posición en el espacio, y luego implementarle movimiento, siguiendo una determinada trayectoria lo más cerca posible de la realidad, se llegó a la conclusión que esto era posible gracias a esa figura plana llamada elipse.

La elipse se estudia dentro de las secciones cónicas la cual se genera al cortar un cono circular recto (cono de revolución) con un plano que forma con su eje un ángulo mayor que el que forman con él las generatrices del cono.

Estas curvas han sido objeto de estudio en varias ramas del saber, en las matemáticas (en Geometría Analítica, en Geometría Descriptiva), en Dibujo Técnico y en el taller de pailería, entre otras.

Desarrollo.

La Elipse es una curva plana engendrada por un punto (del plano) que se mueve de tal manera que la suma de sus distancias a dos puntos fijos (F1 y F2), llamados

focos, es constante e igual al diámetro más largo AB, que se llama eje mayor. Es una curva cerrada, plana y simétrica en dos sentidos.

C

o

D

A

B

a

a

b

b

C

MD

o

D

A

B

N

F1

F2

Elementos a considerar en la Elipse. (figura 2)

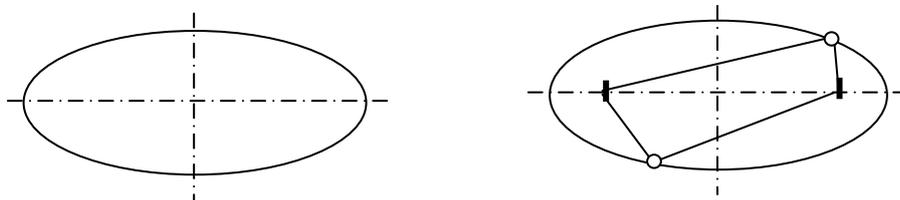


Figura 2

donde:

-  Focos: F1 y F2.
-  Eje mayor: AB y su longitud es 2a (el eje mayor determina su mayor largo).
-  Eje menor: CD y su longitud es 2b (el eje menor determina su mayor ancho).
-  Semieje mayor: oB y su longitud es a.
-  Semieje menor: oC y su longitud es b.
-  Radio vectores: MF1, MF2, NF1, NF2, ...
-  Constantes: MF1+MF2=NF1+NF2= ... =2a.
-  Distancia focal: /F1F2/ = 2c.
-  Diámetro (∅): es todo segmento de la elipse que pasa por su centro (o) y la divide en dos partes iguales.

El círculo es un caso particular de la elipse: es una elipse con excentricidad igual a cero. [Papp D. 1982]

En el estudio de la circunferencia son de suma importancia, tanto para su construcción y desarrollo la constante Pi (Pi=3.1415926535898), el radio (R) de la circunferencia y las funciones trigonométricas seno y coseno. En el caso de la circunferencia R se mantiene constante. En el caso de la Elipse además de los datos indicados, para la circunferencia, se incluyen el factor de dilatación/contracción para los radios r y R (semieje menor y semieje mayor).

Proyección Isométrica rectangular de la circunferencia.

El estudio de las proyecciones axonométricas es de vital importancia para la representación de figuras espaciales en un solo plano, pudiendo representarse en éste las tres dimensiones del espacio, donde cobra interés, por sus características proyectivas, la representación axonométrica de la circunferencia ya que la misma situada sobre cualquiera de los planos axonométricos se proyectará según una elipse.

Tabla 1: valores que como constantes se emplean en la construcción de una Elipse.

deformación	ángulo entre los ejes	Planos	Coeficiente de	
			Eje Mayor	Eje menor
Isométrica (igual medida)	$\alpha=\beta=\delta= 120^\circ$	F, H, L	1.22	0.71

Dimétrica Ortogonal	$\alpha = \beta = 131^{\circ} 25'$	F, H	1.06	0.35
(dos medidas)	$\delta = 97^{\circ} 10'$	L	1.06	0.84
Dimétrica Oblicua	$\alpha = \beta = 135^{\circ}$	F	1.00	1.00
(dos medidas)	$\delta = 90^{\circ}$	H, L	1.06	0.35

En todos los casos $\alpha + \beta + \delta = 360^{\circ}$

Tomando como referencia una línea horizontal, para la representación de la circunferencia en el plano Frontal (F), el eje mayor irá inclinado 60° y a 120° si la vamos a representar en el plano Lateral (L), si la vamos a representar en el plano Horizontal (H) el eje mayor será paralelo respecto a dicha horizontal, en cualquiera de los casos será, el eje menor perpendicular al eje mayor de la elipse.

Otras de las axonometrías utilizadas son la Dimétrica Rectangular y la Dimétrica Oblicua. En la primera el eje mayor de la Elipse irá inclinado a $7^{\circ} 10'$ en todos los planos; en el F y H respecto a una horizontal, en el L, respecto a una vertical. En la segunda, la inclinación del eje mayor será de $22^{\circ} 30'$ para los planos L y H, en el plano F la circunferencia se dibuja sin deformación alguna.

Los valores dados en la tabla 1 son de suma importancia para la construcción geométrica de la Elipse en el cálculo del eje mayor y del eje menor, tanto con instrumentos de dibujo como para su representación lo más exacta posible en los dibujos por computadoras.

nota: Para obtener la longitud del eje mayor de la elipse se multiplica E_{may} por el diámetro de la circunferencia y para obtener la longitud del eje menor se multiplica E_{men} por el diámetro de la circunferencia.

Las Elipses en computación.

La mayoría de los lenguajes de programación tienen definidos procedimientos para la representación de arcos de circunferencia, circunferencias y Elipses. En lenguajes como el BASIC y el QUICK BASIC, para este propósito se utiliza el procedimiento CIRCLE cuya sintaxis es:

CIRCLE (X, Y), radio, color, ángulo inicial, ángulo final, relación altura/anchura.

En dependencia a la relación altura/anchura (V), se podrá obtener o un Círculo (V=1) o una Elipse (si $V > 1$ la Elipse se representará estirada en el eje vertical, si $0 < V < 1$ entonces la Elipse se representará estirada en el eje Horizontal).

En el lenguaje Turbo Pascal la Elipse se dibuja con un procedimiento dado para este fin, y su sintaxis es:

ELLIPSE (X, Y, ángulo inicial, ángulo final, radioX, radioY);

Todos estos lenguajes de programación y la mayoría de los llamados graficadores dan la posibilidad de dibujar elipses, pero están limitados a ofrecer una representación de las mismas, sólo, en el eje horizontal o en el vertical. En ellos se ofrecen los procedimientos dados para las representaciones gráficas (la elipse en nuestro caso), materializando el efecto de una forma cerrada y acabada, no nos es posible aprovechar los puntos que la definen. No obstante, utilizando sencillos recursos que nos brinda la matemática, es posible representar su trayectoria punto a punto y "almacenar" las coordenadas de cada uno de estos puntos en aras de lograr el efecto de animación deseado.

Para lograr este objetivo sólo me son necesario como datos las coordenadas X, Y del centro de rotación, su radio (distancia desde el punto en cuestión al centro de rotación dado) y su ángulo de inclinación, según la axonometría empleada, para calcular las coordenadas (Xo, Yo) de cada punto P[n] de su trayectoria, donde:

$X_o = \text{coordenada X} + \text{Cos}(\text{ángulo}) * \text{Radio} * \text{Factor de formación para uno de sus ejes.}$

$Y_o = \text{coordenada Y} + \text{Sin}(\text{ángulo}) * \text{Radio} * \text{Factor de deformación para el otro eje.}$

-  El centro de coordenadas (X, Y), será el centro a partir del cual se dibujará la elipse.
-  El radio será la longitud constante correspondiente al radio de la circunferencia que se desea representar en axonometría.
-  El ángulo estará en función de la variable de control de ciclo (0 a 360) pudiéndose adicionar o sustraer grado en dependencia del origen y recorrido del punto.
-  El factor de deformación se da en la tabla 1 según la disposición para cada eje (eje mayor o eje menor), y en dependencia de la axonometría utilizada.
-  Los valores empleados como parámetros en las funciones trigonométricas de estos lenguajes de programación deben darse en radianes, pudiendo llevar a cabo la operación de conversión de un sistema a otro (en grados decimales a radianes) mediante la siguiente operación:

$$Ar = Ad * \frac{\pi}{180}$$

donde : Ar = ángulo en radianes,
Ad = ángulo es decimales.
 $\pi = 3.415926535898$

En ninguno de los lenguajes de programación está implementada la posibilidad de representar elipses inclinadas (Para representaciones Axonométricas por ejemplo), matemáticamente es posible mediante las siguientes ecuaciones de la elipse:

Ecuaciones de la elipse:

$X = A + \text{Cos}(\alpha) * \text{Radio} * \text{Emay.}$

$Y = B + \text{Sin}(\alpha) * \text{Radio} * \text{Emen.}$

Rotación:

$X1 = A + (X-A) * \text{Cos}(\beta) + (Y-B) * \text{Sin}(\beta).$

$Y1 = B + (X-A) * \text{Sin}(\beta) + (Y-B) * \text{Cos}(\beta).$

Donde:

A - Coordenada X del centro de la elipse.

B - Coordenada Y del centro de la elipse.

α - Ángulo para un punto dado de la elipse.

Radio - Radio de la circunferencia (Diámetro de la circunferencia / 2).

Emay y Emen - Factor de deformación para los ejes mayor y menor de la elipse (tabla #1).

β - Ángulo de rotación, inclinación, de los ejes de la elipse (según la axonometría aplicada y el plano escogido para su representación).

X, Y - Coordenada de un punto cualquiera de la elipse.

X1, Y1 - Coordenada de un punto de la elipse rotada a un ángulo β .

Dar vida a los elementos.

Si se toma en consideración que cualquier elemento geométrico proyectable (la Recta y el Plano) está formado por puntos y que todo cuerpo poliédrico está limitado por planos que al interceptarse lo hacen según aristas (segmentos), las cuales al interceptarse lo hacen en un vértice (punto) y además, si se conoce el centro de rotación de cada uno de esos puntos, se puede calcular la trayectoria elíptica de los mismos y lograr una representación del objeto en tercera dimensión. Almacenando los valores de dicha trayectoria, es posible dar vida a los elementos, haciéndolos mover en pantalla, a esta operación se le conoce como animación.

En el anexo 1 se propone a modo de ejemplo un programa realizado en Turbo Pascal versión 5.0 donde puede darse animación a un elemento geométrico (Punto, Recta o Plano), en dependencia del número de puntos o vértices que se deseen animar.

X

Y

Z

Estos efectos de animación pueden lograrse implementando la rotación y/o traslación, del elemento, en cualquiera de los ejes del espacio tridimensional X, Y, Z (Fig. 3). En dependencia del algoritmo que se implemente puede resultar muy útil el uso de un ciclo (o bucle) cuya variable de control irá determinando la posición variante de un punto para cada instante dado.

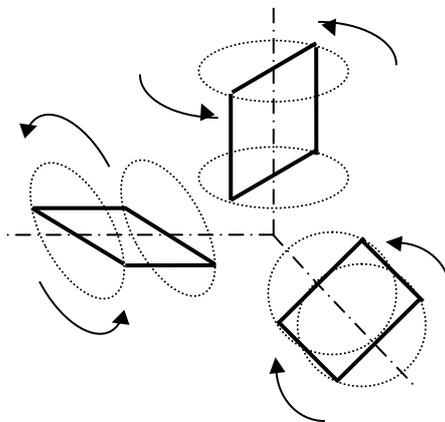


Figura 3.

Veamos algunas indicaciones para la animación de los entes geométricos (el Punto, la Recta y el Plano), así como de cuerpos poliédricos (el Prisma y la Pirámide).

Mover un punto: Para ello solo es necesario seguir la trayectoria de los puntos de la elipse recorriendo la lista almacenada (figura 4). Por ejemplo:

P[n]

0°

360°

180°

270°

90°

Con **PSET (X,Y), Color** en los lenguajes BASIC.

Con **PUTPIXEL (X, Y, Color);** en Turbo Pascal

donde: X, Y son las coordenadas de cada punto de la trayectoria elíptica almacenada.

En la figura 4 P[n] es la posición del punto en un instante dado, de uno de los puntos de la trayectoria de la elipse, con coordenada X,Y predefinida.

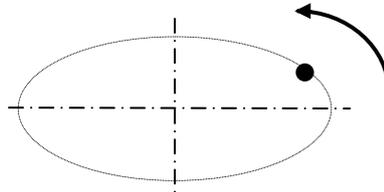


Figura 4.

Mover una recta (Fig. 5): Para lograrlo solo es necesario mover dos puntos de la trayectoria de la elipse, con el mismo incremento y separados a un ángulo de 180°.

P[n]

0°

360°

180°

270°

90°

P[n+180]

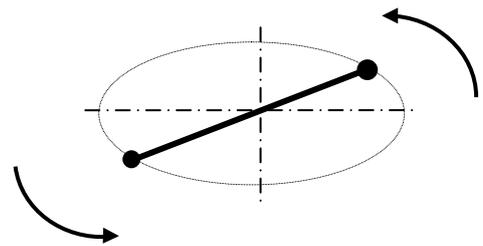


Figura 5.

Mover un plano: Para lograr este efecto se necesitan como datos el número de puntos (vértices), a mover, y la simetricidad del polígono que limita a dicho plano.

Si se trata de un polígono regular el problema resulta más sencillo. En efecto, basta tener en cuenta el número de lados y calcular la amplitud del ángulo entre cada una de sus aristas mediante la siguiente fórmula:

$$\text{Ang} = 360^\circ / \text{Número de lados.}$$

P[a]

P[b]

P[c]

P[d]

90°

180°

270°

0°

360°

Se sugiere que para que le sea viable la representación del polígono el ángulo debe ser un entero (múltiplo de 360°), siendo factible la representación de polígonos de 3, 4, 5, 6, 8, 9, 10 y hasta 12 lados.

Cada vértice del polígono a representar estará ubicado en un múltiplo del valor obtenido en Ang. El primero de los vértices estaría en la posición cero (ángulo cero), si es una poligonal cerrada su último vértice coincidiría con el primero (Figura 6).

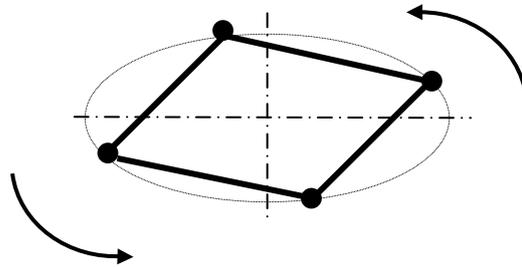


Figura 6.

P[b]

P[a]

P[c]

P[d]

En el caso de un polígono irregular (Fig.7) es necesario hallar primero el centro de rotación del polígono, luego la longitud del radio de cada punto respecto a éste y determinar las trayectorias de dichos puntos (todas manteniendo el mismo centro de rotación).

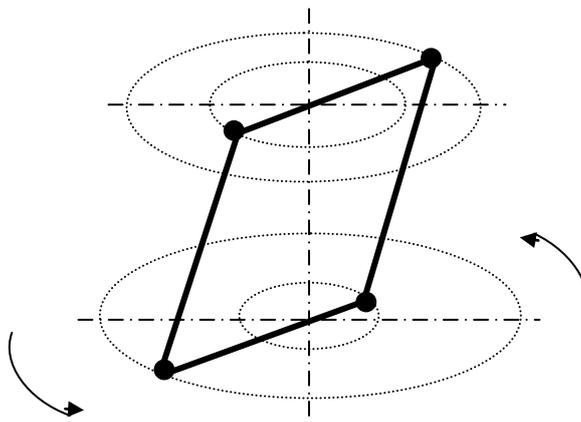


Figura 7.



Mover un sólido geométrico (Poliedro); Para lograr este objetivo es necesario representar primero el polígono de la base inferior, donde quedaría definido el número de lados del poliedro (ver mover un plano), luego se determina, según la altura del cuerpo, el polígono y la trayectoria de la base superior. (Uniendo los puntos correspondientes a ambas bases, mediante segmentos, quedará representado el objeto en sí.

Para la pirámide (Fig. 8) se limita su altura, y dicho vértice (S) permanecerá inmóvil, solo se moverá el polígono de la base y los segmentos laterales correspondientes.

En el prisma (Fig. 9) el problema se resume a mover cada punto de ambas bases (separadas a una distancia igual a la altura) y los segmentos laterales que los unen con el mismo incremento/decremento.

P[a]

P[b]

P[c]

P[d]

S

0°

360°

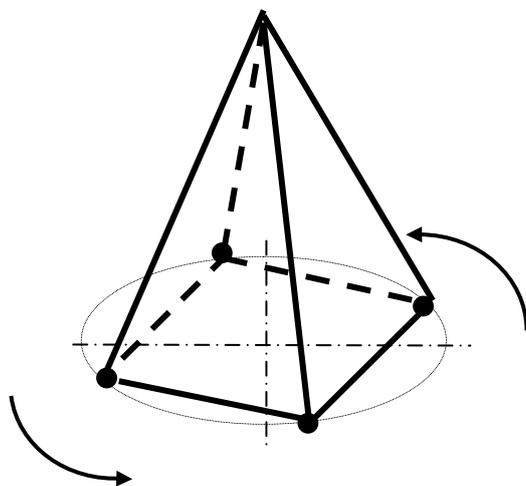
180°

270°

90°

P[a]

P[b]



P[c]

P[d]

K[a]

K[b]

K[c]

K[d]

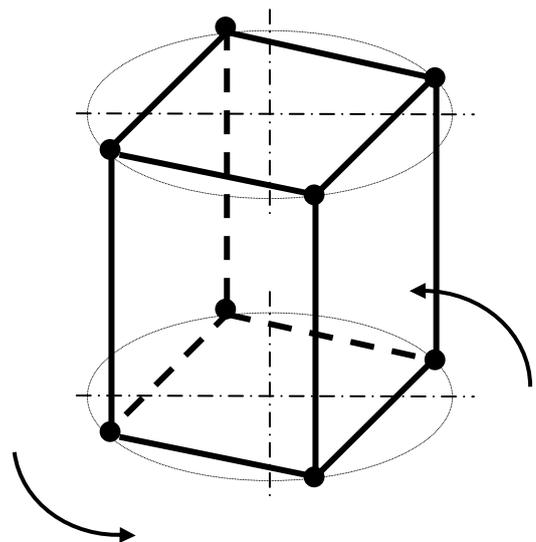


Figura 8.

Figura 9.

Conclusión.

En la industria del software, las tecnologías para las representaciones espaciales han avanzado considerablemente, con técnicas sofisticadas para este fin, donde la Realidad Virtual está tratando de invadir los dominios de la filosofía del conocimiento. En la actualidad los creadores de software ya casi no utilizan las técnicas numéricas, empleando en su lugar el "ray tracing" (trazados de rayos) y el "rendering" (construcción por refracción y/o reflexión de la luz) las cuales han invadido la industria del software en 3D.

Nuestros laboratorios docentes para computación aún por las limitaciones económicas que atraviesa el país, no cuentan con estas altas tecnologías a disposición de nuestros docentes y educandos, de ahí la utilidad de este trabajo. De una manera sencilla se pueden lograr efectos útiles en el uso de la computación como medio de enseñanza, fundamentalmente para alumnos que se inician en la programación y con versiones de lenguajes con los que contamos.

La Elipse es un recurso que nos permite simular representaciones espaciales (en 3D) en pantalla (en un plano) e incluso animarlas. Si se aprovechan los algoritmos expuestos en este trabajo, se puede "construir" nuevos modelos (reales o producto de la imaginación) ya sea de un objeto determinado o simular el proceso constructivo y/u operacional de determinados equipos durante la elaboración mecánica o manual de objetos, aplicando los conocimientos y habilidades desarrolladas en las aulas especializadas de dibujo y las técnicas de computación.

Los procedimientos y las fórmulas brindadas fueron aplicados en el software "Elements" (del autor) el cual es un simulador utilizado como medio de enseñanza en el estudio sobre la teoría de las proyecciones.

Bibliografía.

-  Ballester, S. et al.(1992). Metodología de la Enseñanza de la Matemática. Tomo I. Ministerio de Educación. Editorial Pueblo y Educación. 459 pg.
-  Boajvalov, S. et al.(1969) Introducción a la Geometría Analítica. Editorial Ciencia y Técnica . 246 pg.
-  Bogoliúbov, S. (1988) Dibujo Técnico. Editorial Mir. Moscú.
-  Cherbakov, S. et al.(1991). Geometría Analítica II. Editorial Pueblo y Educación. 283 pg.
-  Domenech T. y Antón R. (1990) Dibujo Básico. Editorial Pueblo y Educación.
-  García, L. (1995). Reflexiones sobre el uso del ordenador en la educación. Revista Educación y Tecnología #117. Sep-Oct. España. pg 48-51.
-  Gutiérrez B. Carlos (1998) Elements: una alternativa en el estudio de las proyecciones. Tesis en opción al título de máster en informática educativa. I.S.P. de Holguín Cuba.
-  Fernández, J.M. (1980) Apuntes sobre Dibujo Técnico. Editorial Científico

Técnica.

-  French y Vierch. (1969) Dibujo de ingeniería. Edición Revolucionaria. 1969.
-  Katrib, M.(1988). Lenguajes de Programación y Técnicas de Computación. Editorial Pueblo y Educación. 421 pg.
-  Larburu, N. (1968) El trazado en el taller de caldería. Editora Ciencia y Técnica.
-  Loktev, O.V. (1987) Curso breve de Geometría Descriptiva. Editorial Mir. Moscú.
-  Marin A. et al. (1982) Geometría Descriptiva. Editorial Pueblo y Educación. La Habana.
-  Papp D. (1982) Descubridores y descubrimientos entre Leonardo y Freud. Editorial de la Universidad de Concepción. Chile.
-  Perelman, Y.I. "Astronomía Recreativa.". Editora Juvenil, La Habana 1996.
-  Pérez, J. et al. (1989). La utilización de la computación en el tratamiento de las secciones cónicas. Pedagogía Cubana. Año I. Julio-Septiembre. 112 pg.
-  Ramírez, A. et al (1995). Paquetes de programa de aplicación a la geometría plana para el estudio independiente de los estudiantes. Pedagogía 95. Cuba.
-  Ramírez, G. (1994). La computadora, herramienta del proceso enseñanza-aprendizaje. Revista Educación #48, México. pg 48-53. Nov.
-  RibniKov, K.(1987). Historia de las Matemáticas. Editorial MIR, Moscú. 487 pg.
-  Rodríguez, A. (1996). Secciones Cónicas. Trabajo de curso en opción al título de Licenciado en la especialidad Matemática-Computación.
-  Tshipkin, A.G. (1989) Manual de matemáticas para la enseñanza media. Editorial Mir. Moscú.

ANEXO 1

-

```
Program Rota_Elementos;
```

```
Uses CRT, Graph;
```

```
Var Gd, Gm, R, C, NI, I : Integer;  
    F: Real, P, Q: PointType;  
    Elipse : Array [0 .. 360] of PointType;
```

```
Procedure GiraElem;
```

```
Var Color: 0..4; Inic, Fin: Integer;
```

```
Begin
```

```
Color := GetColor; Inic := R + I * Gm; Fin := R + (I+1) * Gm;
```

```
If Fin >= 360 Then Fin := Fin - 360;
```

```
SetColor(3); SetWriteMode(XorPut);
```

```
Rectangle(Elipse[Inic].x - 1, Elipse[Inic].y - 1, Elipse[Inic].x + 1, Elipse[Inic].y + 1);
```

```
If NI = 2 Then Rectangle (Elipse[Fin].x - 1, Elipse[Fin].y - 1,
```

```

        Elipse[Fin].x + 1, Elipse[Fin].y + 1);
If NI > 0 Then Line (Elipse[Inic].x, Elipse[Inic].y, Elipse[Fin].x, Elipse[Fin].y);
SetColor(Color); SetWriteMode(NormalPut);
End;

```

```

Begin

```

```

Write('Número de puntos o vértices: ');
Repeat ReadLn(NI); Until NI In [1..6, 8..10, 12];

```

```

Gd := Detect; InitGraph(Gd, Gm, '');

```

```

If GraphResult <> GrOk Then Halt(1);

```

```

SetGraphMode(1);

```

```

F := Pi / 180; C := 0; R:=0,

```

```

While C <= 720 do

```

```

    Begin

```

```

        P.x :=Round(160 + Cos(R * F) * 40 * 1.22);

```

```

        P.y :=Round( 70 - Sin(R * F) * 40 * 0.71 * 0.833);

```

```

        Elipse[R] := P; PutPixel (P.x, P.y, 1);

```

```

        Inc (C, 2); Inc (R);

```

```

    End;

```

```

    OutTextXy (100, 190, '.... LISTO ....'); ReadLn;

```

```

    Gm := 360 Div NI;

```

```

    Repeat

```

```

        For R := 0 To Gm do Begin

```

```

            For I := 0 To NI - 1 do GiraElem;

```

```

            If NI <> 2 Then

```

```

                Begin Delay (24);

```

```

                    For I := 0 To NI - 1 do GiraElem;

```

```

            End;

```

```

        End;

```

```

    Until Keypressed;

```

```

    CloseGraph;

```

```

End.

```