



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Desarrollo de ciclos enlatados para torno CNC

TESIS

Que como parte de los requisitos para obtener el título de:

Ingeniero Electromecánico

Presenta

Víctor Hugo Dávalos Martínez

Director de tesis

M. en C. Mario Mejía Ugalde

San Juan del Río, Qro. Octubre del 2011

RESUMEN

En la universidad existe una línea de investigación para el desarrollo de mejoras en los sistemas de control CNC de maquinas-herramienta. Donde se han realizado trabajos sobre la construcción, control e instrumentación de estos sistemas. Para la realización de dichos trabajos se han diseñado diferentes formas de reconstruir las trayectorias generando su propio código máquina.

El objetivo principal de esta investigación, es el de optimizar el código G para los diferentes tipos de maquinados en torno CNC llámense (conicidad, cilindrado, desbaste transversal, etc.), con la finalidad de tener una alta productividad a la hora de maquinar las piezas.

**A mi familia, a mi hermano y a mi madre
por sus grandes enseñanzas**

AGRADECIMIENTOS

Quiero agradecer a todas aquellas personas que contribuyeron en la realización de la presente tesis en especial al asesor de ésta, el M. en C. Mario Mejía Ugalde por su compromiso permanente, su dedicación y constancia, que me entregó durante toda la realización de la tesis, a mis sinodales: Dr. Miguel Trejo Hernández, Dr. Juan Primo Benítez Rangel, Dr. Aurelio Domínguez González, a mis profesores porque de ellos he recibido mi preparación en la Universidad, a la UAQ por haberme permitido llevar a cabo mis estudios de licenciatura en sus instalaciones, a mis compañeros por su apoyo durante todos estos años de estudio y al CONACYT ya que por medio del proyecto sistema inteligente de tiempo real para la modificación de los parámetros de corte para torno CNC bajo condiciones variantes num. 83810 fue posible la implementación práctica del trabajo realizado en esta tesis.

¡Gracias!

ÍNDICE

	Página
1 Introducción	9
1.1 Descripción del problema.....	10
1.2 Antecedentes históricos.....	11
1.3 Ventajas y desventajas principales del uso de un equipo CNC	12
1.4 Estado del arte	13
1.5 Objetivos y justificación de la tesis.....	16
1.5.1 Objetivos	16
1.5.2 Justificación	17
1.6 La presente tesis	17
2 Fundamentación teórica.....	19
2.1 Control numérico por computadora (CNC)	19
2.2 Torno CNC.....	20
2.2.1 Ventajas y desventajas de los tornos CNC.....	21
2.3 Ciclo básico de mecanizado	22
2.4 Estructura de un programa de torneado	23
2.4.1 Puntos de referencia	24
2.4.2 Funciones preparatorias G	24
2.4.3 Funciones misceláneas	28
2.5 Estructura de un bloque.....	28
2.6 Trigonometría	30
2.6.1 Unidades angulares.....	30
2.6.2 Razones trigonométricas.....	31
2.6.3 Sentido de las funciones trigonometricas.....	32
2.7 Función de un compilador	38
2.7.1 Partes en las que trabaja un compilador.....	38
2.7.2 Clasificación de compiladores	39
2.7.3 Analizador sintáctico.....	41
3 Metodología	42
3.1 Definición de código G y M	42
3.1.1 Códigos G's.....	43
3.1.2 Códigos M's	43
3.2 Restricciones en los Bloques.....	43
3.3 Ciclos enlatados o repetitivos	43

3.4	Ciclo de conicidad.....	44
3.5	Ciclo de cilindrado	48
3.6	Ciclo de arco.....	51
3.7	Ciclo de arco interno	57
3.8	Ciclo de cilindrado interno	61
4	Resultados obtenidos.....	64
4.1	Resultados obtenidos de ciclo G84	65
4.2	Resultados obtenidos de ciclo G87	66
4.3	Resultados obtenidos de ciclo G101	67
4.4	Resultados obtenidos de ciclo G102	68
4.5	Resultados obtenidos de ciclo G84-i	69
5	Conclusiones.....	70
	Referencias	71
	Apéndice A.....	72

ÍNDICE DE FIGURAS

Figura	Página
Figura 2.1 Triángulo rectángulo..	31
Figura 2.2 Triángulo rectángulo..	32
Figura 2.3 Primer cuadrante.	33
Figura 2.4 Segundo cuadrante.	34
Figura 2.5 Tercer cuadrante.	36
Figura 2.6 Cuarto cuadrante.	37
Figura 2.7 programa objeto.	39
Figura 3.1 Formato de programa CNC.	42
Figura 3.2 Ciclo de conicidad.	44
Figura 3.3 Movimiento del cortador para el ciclo G84.	48
Figura 3.4 Ciclo enlatado (Ciclo de arco).	51
Figura 3.5 Ciclo enlatado (Ciclo de arco interno).	57
Figura 3.6 Ciclo enlatado (cilindrado interno).	61
Figura 4.1 Ciclo de Cilindrado.	65
Figura 4.2 Pieza obtenida del Ciclo de Cilindrado.	65
Figura 4.3Ciclo de Conicidad.	66
Figura 4.4 Pieza obtenida del Ciclo de Conicidad.	66
Figura 4.5Ciclo de Arco.	67
Figura 4.6 Pieza obtenida del Ciclo de Arco.	67
Figura 4.7 Ciclo de Arco Interno.	68
Figura 4.8 Pieza obtenida del Ciclo de Arco Interno.	68
Figura 4.9 Ciclo de Cilindrado Interno.	69

ÍNDICE DE TABLAS

Tabla	Página
Tabla 2.1 Relación entre grados y radianes.....	31
Tabla 2.2 Funciones trigonométricas.	32

1 Introducción

La máquina herramienta, aunada con el control numérico por computadora, han jugado un papel fundamental en el desarrollo tecnológico del mundo, hasta el punto que no es una exageración decir que su tasa del desarrollo gobierna directamente el avance industrial. Gracias a la utilización de éstas se ha podido realizar de forma práctica, maquinaria de todo tipo que, aunque concebida y realizada, no podía ser comercializada por no existir medios adecuados para su construcción industrial.

Así, por ejemplo, si para la mecanización total de un número de piezas fuera necesario realizar las operaciones de fresado, mandrilado y taladrado, es lógico que se alcanzaría la mayor eficacia si este grupo de máquinas-herramientas estuvieran agrupadas, pero se lograría una mayor eficacia aún si todas estas operaciones se realizaran en una misma máquina. Esta necesidad, sumada a numerosos y nuevos requerimientos, que día a día aparecen, ha forzado la utilización de nuevas técnicas que reemplazan al operador humano. De esta forma se introdujo el control numérico en los procesos de fabricación, impuesto por varias razones:

Necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficientes sin recurrir a la automatización del proceso de fabricación.

Necesidad de obtener productos hasta entonces imposibles o muy difíciles de fabricar, por ser excesivamente complejos para ser controlados por un operador humano.

Necesidad de fabricar productos a precios suficientemente bajos.

Inicialmente, el factor predominante que condicionó todo automatismo fue el aumento de productividad. Posteriormente, debido a las nuevas necesidades de la

industria aparecieron otros factores importantes como la precisión, rapidez y flexibilidad.

Hacia 1942 surgió lo que se podría llamar el primer control numérico verdadero, debido a una necesidad impuesta por la industria aeronáutica para la realización de hélices de helicópteros de diferentes configuraciones (**Gibbs y Crandell, 1991**).

1.1 Descripción del problema

Actualmente se está automatizando un torno CNC el cual hasta el momento no cuenta con ciclos programados para el maquinado de una pieza, solamente se programa mediante una serie de cálculos previos al maquinado para que este siga las trayectorias programadas e indicadas por cada uno de los bloques que conforman el programa. Una de las prioridades importantes de esta investigación es la de optimizar el código G y a la vez crear ciclos enlatados para el maquinado de algunas piezas, ya que este torno solo cuenta con los códigos básicos como son G01, G02, G03 y movimiento de máquina de manera manual.

La necesidad por mejorar la eficiencia y la precisión en sistemas de control numérico ha sido la mejor razón para el desarrollo de nuevas investigaciones y tecnologías tanto para el ámbito industrial como académico, logrando con esto, un gran avance en cuanto a la generación de nuevos métodos.

En la universidad existe una línea de investigación para el desarrollo de mejoras en los sistemas de control CNC de máquinas-herramienta. Donde se han realizado trabajos sobre la construcción, control e instrumentación de estos sistemas. Para la realización de dichos trabajos se han diseñado diferentes formas de reconstruir las trayectorias generando su propio código máquina.

El tipo de código que se implementa en este caso para el torno CNC, se conoce como código o programación manual. Lo cual resulta un poco anticuado ya que se escribe únicamente por medio de razonamientos y cálculos que realiza un operario.

Al conjunto de informaciones que corresponde a una misma fase del mecanizado se le denomina bloque o secuencia, dicho bloque se encuentra conformado de caracteres y palabras, los cuales forman una instrucción simple dentro del maquinado de la pieza.

1.2 Antecedentes históricos

La generación de trayectoria de referencia juega un papel importante en el control computarizado de la máquina-herramienta. Las trayectorias generadas no sólo deben describir con precisión el camino de la herramienta, también deben considerar una cinemática correcta mediante perfiles suaves para mantener una alta exactitud en el seguimiento.,(Erkormaz y Altintas, 2000).

El CNC que significa "control numérico computarizado" y definido como una operación de máquina-herramienta por medio de instrucciones en formato código G, en sistemas de control (Smid, 2003), tuvo su origen a principios de los años cincuenta en el Instituto de Tecnología de Massachusetts (MIT), en donde se automatizó por primera vez una fresadora. (Londoño, 2005).

Control numérico o CN es el término usado para describir el control de movimiento de una máquina y varias otras funciones por medio de instrucciones expresadas como una serie de números generados por un sistema de control electrónico. Pero se tienen antecedentes de la evolución del CN desde 1880 con la introducción de una variedad de herramientas para el maquinado de metales y el comienzo del énfasis en la producción a gran escala, en 1940 se hace la introducción de los controles hidráulicos, neumáticos y electrónicos generando un aumento del maquinado automático, para 1945 comienza la investigación y desarrollo del control numérico realizando experimentos de producción a gran escala con control numérico. En 1955, las herramientas automatizadas comenzaron a aparecer en las plantas de producción para la fuerza aérea de producción de los Estados Unidos y un año más tarde, en 1956, se activa la concentración en la investigación y el desarrollo del control numérico más definido, para 1972 se introduce la micro computadora (Gibbs y Crandell, 1991).

De 1980 hasta la actualidad, se han creado nuevos y variados sistemas de control numérico perfeccionando los anteriores y las aplicaciones a la producción de una gama más grande de procedimientos de maquinado de metales iniciando aplicaciones a otras actividades diferentes utilizando insumos computarizados de control numérico. Además hacen uso de documentos computarizados de planeación gráfica por control numérico, se establecen centros de maquinado para uso general y se desarrollan procedimientos computarizados de trazo de curvas de nivel por control numérico, a bajo costo.

La diferencia entre un torno convencional y uno de Control Numérico Computarizado esta constituida fundamentalmente por su sistema de movimiento automático, el sistema electrónico que procesa y controla los movimientos mediante software (**Mikell, 1997**).

En una máquina CNC, a diferencia de una convencional o manual, una computadora controla la posición y velocidad de los motores que accionan los ejes gracias a esto, puede hacer movimientos que no se pueden lograr manualmente como círculos, líneas diagonales y figuras complejas tridimensionales.

Existen máquinas CNC capaces de mover la herramienta al mismo tiempo en los tres ejes para ejecutar trayectorias tridimensionales como las que se requieren para el maquinado de complejos moldes y troqueles.

El control numérico computarizado es una forma de automatización programable en la cual, en base a una serie de instrucciones codificadas (programa), se gobiernan todas las acciones de una máquina o mecanismo haciendo que este desarrolle una secuencia de operaciones y movimientos previamente establecidos por el programador (**Ogata, 1996**).

1.3 Ventajas y desventajas principales del uso de un equipo CNC

Las principales ventajas del maquinado en equipo CNC, puede ser resumido en: aumento de productividad; permite una mejor planeación de las operaciones; se incrementa la flexibilidad de maquinado solo para varias piezas; reducción en tiempo de programación; disminución en los costos por herramientas;

adaptabilidad y menor costo de mantenimiento; reducción de necesidades de mano de obra y de inventarios; la diversidad de productos, mayor facilidad para crear estructuras más complejas de producción; se incrementa la seguridad para el usuario; reducción del tiempo de flujo de material; reducción del manejo de la pieza de trabajo; menor grado de especialización de personal; aumento en precisión; mejor control del proceso y tiempos de maquinado.

La implementación de CNC implica que los diseños sean cada vez más complejos, la diversidad de productos hace necesario la tendencia a estructuras de producción más flexibles, se tiende a incrementar los tiempos de inspección, alto costo en maquinados de baja producción y se hace necesario minimizar errores, el tiempo de entrega de los productos tiende a ser cada vez más reducido (**Smid, 2003**).

1.4 Estado del arte

Dentro de los trabajos realizados respecto a módulos de control numérico por computadora se han publicado algunos artículos: **Gutiérrez (2004)** desarrolló nuevas metodologías basadas en experimentación para la optimización del proceso de corte, El mismo autor, realiza trabajos de investigación dentro del área de selección y resistencia de materiales, manufactura y pruebas mecánicas en el ITESM campus Toluca. Su principal enfoque es en el área de manufactura, experimentación y manejo de máquinas de control numérico.

En la Universidad Autónoma de Querétaro existe una línea de investigación basada en el control y automatización de máquinas-herramienta, donde se han realizado diversos trabajos que se describen a continuación.

Jiménez (2004) realizó la conversión de un torno convencional a un control numérico por computadora (CNC), seleccionando los elementos para la automatización de la máquina de una forma adecuada, en donde implementó un sistema *CAD/CAM* para la realización de trayectorias, concluyendo que los factores más importantes que afectan el buen funcionamiento de los elementos

son el *ruido*, las altas frecuencias de estática y la longitud de los cables tanto de control como de fuerza.

Muñoz y Traslosheros (2001) en su trabajo estudiaron los aspectos teórico prácticos de los efectos causados por no linealidades como el juego mecánico y la fricción estática en una máquina de control numérico (CNC) de dos grados de libertad, donde diseñan un algoritmo de control el cual su función principal es compensar los problemas de fricción estática y juego mecánico, con lo que intenta demostrar experimentalmente que los algoritmos para compensar el juego mecánico y la fricción estática trabajan mejor que un PID comercial, cabe mencionar que solo lo fue probado para servomotores de CD, donde utilizó el sistema *CAD/CAM* para el desarrollo de este.

García (2006), realizó un sistema de supervisión de ruptura en herramientas de corte para tornos de control numérico computarizado, identificó los principios que rigen la vida útil de las herramientas de corte empleadas en tornos CNC, determinando los principales problemas que éstas presentan al momento del corte y así conocer el tipo de desgaste y daños que sufren. Su control está basado en una PC, utilizando Windows 2000 para generar el control numérico y una tarjeta Galil para el control del movimiento de los servomotores. Desarrolló *parcialmente* un control numérico empleando como plataforma de desarrolló Visual C++, al cual nombró UniCADCAM que le permitió efectuar diversas rutinas de desbaste basadas en interpolaciones lineales y circulares. Determinando que el régimen de desgaste de la herramienta esta determinado por la temperatura, el esfuerzo y la *velocidad relativa* de deslizamiento generado en las áreas de contacto.

Romero (2004) desarrolló una investigación para monitorear el estado de ruptura de la herramienta de corte en donde midió la corriente de los servoamplificadores de los motores de una fresadora para después obtener las señales de fuerza de

corte de la pieza. Realizó un acondicionador de señal para obtener la señal de fuerza a partir de la corriente. También utilizó una tarjeta de adquisición de datos para obtener la señal digital de la fuerza de corte para después procesar la señal y relacionarla con el estado de la herramienta de corte.

Internacionalmente también se han realizado investigaciones para mejorar trayectorias en los maquinados como **Lin y Tai (1999)** quienes proponen una metodología para generar el código de maquinado a partir de los datos tridimensionales provenientes de aparatos de medición de rugosidad sobre superficies maquinadas. Diversos algoritmos se desarrollan para automatizar la generación de los planes de proceso, que contienen información detallada necesaria para el maquinado de control numérico.

García (2004), realiza el primer paso en proceso de automatización al efectuar el diseño de celdas de manufactura basados en conceptos de tecnología de grupos. En ésta tecnología, el objetivo es procesar varias familias de partes en un grupo de máquinas CNC dentro de una celda, en la que el esfuerzo en transporte de material es minimizado. La manufactura de celdas es una aplicación de la tecnología de grupos a los procesos industriales, su principal objetivo es reducir tiempos de montaje y de flujo minimizando así; inventarios, costos en proceso y optimizando tiempos de respuesta en máquinas CNC.

Toro y Arizkuren (2005), que presentan un enfoque distinto en el desarrollo de un *kernel* de simulación para CNC cuya representación interna está basada en niveles, la característica básica es que permite la realización de operaciones booleanas entre modelos 3D de una forma rápida, permitiendo así ser utilizada en simulaciones donde no existe conocimiento *a priori* del programa CN que se está ejecutando (en línea).

Valerio y Anitelle (2005), describe un procedimiento para la implementación de un control numérico por computadora usando un prototipo de software virtual aplicado a torno CN. Este procedimiento se enfoca principalmente sobre el sistema de funcionamiento y la geometría del modelo (diseño físico) de la pieza a maquinar en el torno, además este trabajo apunta hacia el conocimiento potencial y limitaciones de herramientas de interfaz grafica sobre todo en su fabricación y ensamble.

Trejo (2007), describe una metodología para al ajuste en línea de las condiciones de corte en una operación de torneado. El sistema presentado consiste en el monitoreo y el ajuste óptimo de las condiciones de corte con visualización del comportamiento de las corrientes de los servoamplificadores y del husillo, en un módulo de software desarrollado en la UAQ, todo esto, aunado obviamente al previo procesamiento y análisis de las señales con métodos adecuados.

Gómez (2007), dice que para reducir el costo de perfeccionamiento e investigación, es necesario implementar una plataforma de desarrollo que haga factible la integración de todas las tecnologías desarrolladas en la Universidad Autónoma de Querétaro con relación al control numérico por computadora para ser utilizadas en un mismo producto, que se pueda emplear tanto como solución de automatización industrial como herramienta de experimentación e investigación. De esta forma será posible experimentar y desarrollar con las mismas versiones del producto que se encuentran operando en las industrias y se reducirá el tiempo necesario para llevar a la industria los nuevos resultados de la investigación.

1.5 Objetivos y justificación de la tesis

1.5.1 Objetivos

- Implementar ciclos enlatados basados en código G y probarlos experimentalmente en un torno CNC.

1.5.2 Justificación

Día con día la tecnología dentro de la industria manufacturera evoluciona cada vez más rápido. Lo cual da paso a nuevas innovaciones tecnológicas y esto se ve claramente reflejado en la tecnología CNC.

Ya que hoy en día se a logrado minimizar los tiempos de maquinados y se a aumentado la calidad de los procesos de manufactura.

Dentro de esta investigación la justificación mas importante es generar ciclos de maquinado (ciclos enlatados) los cuales realicen alguna tarea especifica. Todo esto basado en código G, el cual nos ayudara a hacer mas fácil y sencillo el maquinado de cualquier pieza. El software utilizado para programar estos ciclos de maquinado será "Visual C++". También con esto se pretende reducir la estructura del programa de maquinado, contando únicamente con uno o dos renglones, los cuales contengan únicamente los parámetros esenciales para realizar el maquinado que se desee.

Todo esto con la finalidad de agilizar el maquinado de la pieza deseada y por consiguiente hacer más fácil la comunicación operador-maquina, y así poder tener:

- Una alta productividad

1.6 La presente tesis

El presente trabajo de investigación fue dividida en cinco capítulos, además de un apéndice al final de la misma. En el capítulo uno, se presentan los antecedentes acerca del CNC a nivel estatal, nacional y mundial. Así mismo hace referencia al estado del arte del control numérico por computadora y ciclos de maquinado, y se menciona la justificación y la descripción del problema a tratar. En el capítulo dos, se da a conocer la fundamentación teórica sobre la cual se basa la investigación

(trigonometría, unidades angulares, estructura de un programa de torneado, códigos G, códigos M etc.). En el capítulo tres, se describe la metodología usada para desarrollar los cálculos matemáticos de los ciclos de maquinado, tales como (conicidad, cilindrado, desbaste transversal, etc.). En el capítulo cuatro, se dan a conocer los resultados de los ciclos de maquinado. En el capítulo cinco, se plantean las conclusiones y referencias relacionadas con la investigación.

2 Fundamentación teórica

2.1 Control numérico por computadora (CNC)

Se considera control numérico a todo dispositivo capaz de dirigir posicionamientos de un órgano mecánico móvil, en el que las órdenes relativas a los desplazamientos del móvil son elaboradas en forma totalmente automática a partir de informaciones numéricas definidas, bien manualmente o por medio de un programa. El término "control numérico" se debe a que las órdenes dadas a la máquina son indicadas mediante códigos numéricos (código G y M). El control numérico puede definirse como un funcionamiento de herramientas de la máquina por los medios de instrucciones específicamente codificado al sistema de control de máquina (**Smid 2000**).

Al principio, las máquinas-herramienta, tales como tornos, fresadoras, taladradoras, cepillos, etc., eran operadas por artesanos especializados, cuyo resultado en la calidad, y en la velocidad de entrega, dependía mucho de la pericia y la predisposición del individuo. Hoy en día, la calidad y la velocidad está dada principalmente por el tipo de máquina, y la tecnología que se utiliza, y no depende mucho del operador; salvo en rubros como la matricería, donde el mecanizado de piezas únicas obliga a una pericia determinante, para aplicar un proceso eficiente, confiable en calidad y entrega a tiempo (**Hernández, 2004**).

Inicialmente, el factor predominante que condicionó todo automatismo fue el aumento de la productividad. Posteriormente, debido a las nuevas necesidades de la industria, aparecieron otros factores no menos importantes como la precisión, la rapidez, la entrega a tiempo y la flexibilidad.

El CNC tuvo su origen a principios de los años cincuenta en el Instituto de Tecnología de Massachusetts (MIT), en donde se automatizó por primera vez una

gran fresadora. En esta época las computadoras estaban en sus inicios y eran tan grandes que el espacio ocupado por la computadora era mayor que el de la máquina. Hoy en día las computadoras son cada vez más pequeñas y económicas, con lo que el uso del CNC se ha extendido a todo tipo de maquinaria: tornos, rectificadoras, máquinas de coser, etc.

En una máquina CNC, a diferencia de una máquina convencional o manual, una computadora controla la posición y velocidad de los motores que accionan los ejes de la máquina. Gracias a esto, puede hacer movimientos que no se pueden lograr manualmente como círculos, líneas diagonales y figuras complejas tridimensionales. Son capaces de mover la herramienta al mismo tiempo en los tres ejes para ejecutar trayectorias tridimensionales como las que se requieren para el maquinado de complejos moldes y troqueles.

En una máquina CNC una computadora controla el movimiento de la mesa, el carro y el husillo. Una vez programada la máquina, ésta ejecuta todas las operaciones por sí sola, sin necesidad de que el operador este manejándola. Esto permite aprovechar mejor el tiempo del personal para que sea más productivo **(Hernández, 2004)**.

2.2 Torno CNC

El torno fue inventado en 1817 por Henry Maudsley. En principio, el torno se utilizaba para torneear, refrentar y perforar piezas de trabajo cilíndricas. También se puede efectuar taladrado, escariado, machuelado, moleteado, esmerilado, fresado, roscado y biselado. Se clasifican por su mecanismo de impulsión, de avance y capacidad de producción.

El torno tiene 5 componentes. Las partes principales del torno son el cabezal principal, bancada, contrapunta, carro y unidad de avance. El cabezal principal contiene los engranes, poleas lo cual impulsan la pieza de trabajo y las unidades de avance. El cabezal incluye el motor, husillo, selector de velocidad, selector de unidad de avance y selector de sentido de avance. Además sirve para soporte y

rotación de la pieza de trabajo que soporta el husillo. La bancada sirve de soporte para las otras unidades del torno (**Flores y Avendaño, 2005**).

Un torno de CNC normalmente es una herramienta de la máquina con dos ejes, el X vertical y el eje de Z horizontal. El rasgo principal de torno que lo distingue de una fresadora es que la parte está rodando sobre la línea de centro de máquina. Además, la herramienta cortante es normalmente estacionaria, montada en un torreón corredizo. La herramienta cortante sigue el contorno del camino programado de la herramienta.

Un torno CNC es un equipo de trabajo que permite manufacturar piezas de distintos materiales y en repetidas ocasiones a través de corte. Este equipo se diferencia de un torno convencional porque es posible programar la secuencia de fabricación de una pieza. Esto significa que el operador sólo hace una vez el trabajo de definir cómo hacer una pieza y el control de la máquina produce la cantidad de piezas que se necesite. A diferencia del torno convencional, el torno CNC consta de dos cuerpos principales: el control y el torno mismo. El control es el computador que procesa y guarda toda la información ingresada por el operador. Una vez realizado esto, el control comunica al torno (mediante señales eléctricas) cómo, cuándo, dónde y qué hacer para elaborar la pieza desea (**Smid, 2000**).

En un sentido amplio se puede decir que un torno CNC, puede hacer todos los trabajos que normalmente se realizan mediante diferentes tipos de torno como paralelos, copiadores, revólver, automáticos e incluso los verticales pueden actuar con control numérico. Su rentabilidad depende del tipo de pieza que se mecanice y de la cantidad de piezas que se tengan que mecanizar en una serie.

2.2.1 Ventajas y desventajas de los tornos CNC

Ventajas:

- Permiten obtener mayor precisión en el mecanizado
- Permiten mecanizar piezas más complejas

- Se puede cambiar fácilmente de mecanizar una pieza a otra
- Se reducen los errores de los operarios
- Cada vez son más baratos los tornos CNC

Como desventajas se pueden indicar las siguientes:

- Necesidad de realizar un programa previo al mecanizado de la primera pieza.
- Costo elevado de herramientas y accesorios
- Conveniencia de tener una gran ocupación para la máquina debido a su alto costo.

2.3 Ciclo básico de mecanizado

Antes de empezar a confeccionar un programa de mecanizado se tiene que conocer bien el mecanizado que se va a realizar en el torno y las dimensiones y características del material de partida, así como la cantidad de piezas que componen la serie que hay que mecanizar. Con estos conocimientos previos, se establece el sistema de fijación de la pieza en el torno, las condiciones tecnológicas del mecanizado en cuanto a velocidad de corte, avance y número de pasadas.

Igualmente se establecen los parámetros geométricos del mecanizado señalando las cotas de llegada y partida de las herramientas, así mismo se selecciona las herramientas que se van a utilizar y las calidades de las mismas.

Velocidad de giro del cabezal. Este dato está en función de las características del material, del grado de mecanizado que se desee y del tipo de herramienta que se utilice. El programa permite adaptar cada momento la velocidad de giro a la velocidad más conveniente. Se representa por la letra (S) y puede expresarse como velocidad de corte o revoluciones por minuto del cabezal.

Avance de trabajo. Hay dos tipos de avance para los carros, uno de ellos muy rápido, que es el avance de aproximación o retroceso al punto de partida, y otro que es el avance de trabajo. Este también está en función del tipo de material, calidad de mecanizado y grado de acabado superficial. El programa permite adaptar cada momento el avance que sea más conveniente. Se representa por la letra (F) y puede expresarse en milímetros por revolución o milímetros de avance por minuto.

Otro factor importante a determinar es que todo programa debe indicar el lugar de posición que se ha elegido para referenciar la pieza que se llama "cero pieza". A partir del cero pieza se establece toda la geometría del programa de mecanizado.

2.4 Estructura de un programa de torneado

La estructura de un programa de torneado esta conformada por una serie de secuencias y funciones donde se van programando las tareas que debe realizar la maquina, de acuerdo con los parámetros de la pieza y las condiciones tecnológicas de su mecanizado.

Existen cuatro términos básicos usados en programación CNC. Ellos se encuentran en artículos técnicos, libros, revistas, etc. Esas palabras son la llave del entendimiento de la terminología CNC. (**Smid 2000**).

Carácter: puede ser un dígito (numero) de 0 a 9, una de las 26 letras programadas o un símbolo como punto, paréntesis, signo menos o porcentaje, entre otros.

Palabra: es la combinación de caracteres alfanuméricos que crean una instrucción simple. Generalmente empezada por una letra seguida de un numero. Por ejemplo: G23 F200.

Bloque: un bloque esta compuesto de un grupo de palabras que forman una instrucción simple (renglón).

Programa: es una lista de bloques que siguen un orden lógico.

2.4.1 Puntos de referencia

Todos los tornos de control numérico están dotados de tres puntos de referencia que hay que tener en cuenta a la hora de iniciar un programa. Estos puntos son conocidos como **cero máquina, cero pieza y punto de referencia**.

El cero máquina es el punto sobre el que se sitúa el sistema de coordenadas inicial de las máquinas, este origen viene dado por el fabricante. Normalmente se sitúa en la cara frontal del plato de garras y sobre el eje X-0.

Se conoce como cero pieza el sistema de referencia que el programador adopta por razones prácticas, teniendo en cuenta la geometría y acotación de la pieza a mecanizar, para facilitar la programación CNC. El cero pieza se programa como la primera función a realizar en cada modelo de pieza nueva que se mecaniza. Este punto tiene como coordenadas X-0 y el valor de Z es la distancia que hay del cero máquina a la cota elegida del plano de la pieza como cero pieza que es el que el programador considere oportuno.

Punto de referencia es aquél donde se realiza la sincronización del sistema de referencia de la máquina. Este punto es hacia donde se dirige el carro portaherramientas cuando cambia de herramienta o cuando se acaba el mecanizado de la pieza. Este punto está prefijado por el fabricante de la máquina.

2.4.2 Funciones preparatorias G

Bajo la letra G acompañada de una cifra se agrupan una gran variedad de funciones que permiten al torno realizar las tareas adecuadas y necesarias para su trabajo.

Hay cuatro tipos básicos de funciones preparatorias:

- Funciones de movilidad

- Funciones tecnológicas
- Funciones de conversión
- Funciones de mecanizado especiales

Funciones de movilidad Las funciones de movilidad más importantes son las siguientes:

G00. Desplazamiento rápido. Indica el desplazamiento más rápido posible del carro portaherramientas, desde el punto de referencia al punto donde inicia el trabajo cada herramienta. Actúa al inicio del programa, cada vez que se produce un cambio de herramienta, y al final del programa en el retorno al punto de referencia.

Durante este movimiento no se mecaniza. El desplazamiento rápido se programa mediante la información de desplazamiento G00 y mediante la indicación del punto de destino. Este punto es alcanzado introduciendo cotas absolutas (G90) o cotas incrementales (G91). El desplazamiento rápido puede implicar una interpolación lineal o bien ser escalonado. La trayectoria programada con G00 se recorre con la máxima velocidad, el desplazamiento rápido, en una línea recta, sin mecanizar la pieza (interpolación lineal). Para esto, el control supervisa la máxima velocidad permitida del eje. Esta velocidad se fija para cada eje como dato de máquina. Si el movimiento de desplazamiento se determina a través del menor valor de las velocidades del eje que han sido fijadas como dato de máquina (**Smid, 2000**).

G01. Interpolación lineal. Indica que la herramienta se está desplazando al avance de trabajo programado, permitiendo las operaciones clásicas de cilindrado y refrentado así como el mecanizado de conos.

G02 Interpolación circular a derechas. Se utiliza cuando es necesario mecanizar zonas esféricas o radiales.

G03. Interpolación circular a izquierdas. Se utiliza cuando es necesario mecanizar zonas esféricas vacías, o radios a izquierdos.

Hay otras funciones de movilidad G, menos importantes y que están en función del equipo que se instale en la máquina.

Funciones tecnológicas. Las funciones tecnológicas son las que se refieren a la forma de programar la velocidad del cabezal y el avance de trabajo. La velocidad de rotación del cabezal se puede programar a las revoluciones por minuto que se desee, para lo cual se antepone la función G97, o se puede programar para que gire a una velocidad de corte constante en m/min. En tal caso se indica con la función G96. Igual sucede con el avance de trabajo, si se desea programar el avance en mm/rev, se antepone la función G95 y si se desea trabajar en mm/min se antepone la función G94.

Funciones de conversión. La función más importante de este grupo es la que corresponde al traslado de origen para situar el cero pieza que se realiza mediante la función G59, también existen funciones si el acotado está en pulgadas o en milímetros. Si bien ya tiene preestablecida la que se va a usar normalmente. Otro caso de conversión es si se programa con cotas absolutas o cotas incrementales.

Funciones de mecanizados especiales. La más popular de estas funciones es la que corresponde a un ciclo de roscado representada por la función G33. Otras funciones de este tipo son las de refrentados, taladrados, roscado con macho, escariado, etc.

Funciones modales. En los programas de CNC, existen funciones que, una vez programadas, permanecen activas hasta que se programa una función contraria, o el programa se termina. Estas funciones son las llamadas funciones modales. En un bloque se pueden programar tantas funciones como se desee, siempre que no sean incompatibles entre ellas. Por ejemplo no se pueden programar en un bloque las funciones G00 y G01.

A continuación se muestra una tabla con las funciones G utilizadas para el torneado.

FUNCIONES G

Grupo 0	G00:	Marcha rápida
	G01:	Interpolación lineal
	G02:	Interpolación circular en el sentido horario
	G03:	Interpolación circular en el sentido antihorario
	G04:	Tiempo de espera
	G33:	Roscado en el registro
	G84:	Ciclo de cilindrado / Ciclo de refrentado
	G85:	Ciclo de roscado
	G86:	Ciclo de Careado – Conicidad
	G87:	Ciclo de Conicidad
	G88:	Ciclo de taladrado con evacuación
Grupo 1	G96:	Velocidad de giro constante
	* G97:	Programación directa de la velocidad de giro
Grupo 2	G94:	Indicación del avance en mm/min (1/100 inch/min)
	* G95:	Indicación del avance en mm/rev (1/100 inch/rev)
Grupo 3	* G53:	Desactivación de desplazamiento 1 y 2
	G54:	Llamada de desplazamiento 1
	G55:	Llamada de desplazamiento 2
Grupo 4	G92:	1. Limitación de la velocidad de giro
		2. Aplicar desplazamiento 5
Grupo 5	* G56:	Desactivación de desplazamiento 3,4,5
	G57:	Llamada de desplazamiento 3
	G58:	Llamada de desplazamiento 4
	G59:	Llamada de desplazamiento 5
Grupo 6	G25:	Llamada de subprograma
	G26:	Llamada de programas de polígonos
	G27:	Salto incondicionado
Grupo 7	G70:	Indicaciones de medidas en pulgadas

	G71:	Indicaciones de medidas en mm
Grupo 8	* G40:	Supresión de la compensación del radio de la herramienta
	G41:	Corrección del radio de la herramienta a la izquierda
	G42:	Corrección del radio de la herramienta a la derecha
Adicional	G50:	Desactivar escalamiento
	G51:	Activar escalamiento
	G90	Programación en absolutas.
	G91	Programación en incrementales.

2.4.3 Funciones misceláneas

La dirección M en un programa CNC identifica las funciones misceláneas, algunas veces llamadas funciones maquina, por ejemplo: prender o apagar el refrigerante, etc. Básicamente equipos auxiliares. No todas las funciones misceláneas son relacionadas a la operación de una maquina CNC, algunas son relacionadas al procesamiento del mismo programa.

2.5 Estructura de un bloque

Un bloque representa una “línea” del programa y se identifica mediante la letra N seguida de un número que sirve para diferenciar ese registro de los demás que componen un programa. El control lee los registros en orden ascendente, pero no tiene restricciones en cuanto a la diferencia numérica de dos registros consecutivos. Esto quiere decir que es posible numerar de cinco en cinco o de veinte en veinte (o una vez aumentar 11 y luego 19). Sin embargo en general se recomienda numerar (y así lo propone automáticamente el control) de diez en diez. Hay dos razones para esto:

Primero, se deja la posibilidad de introducir algún registro posteriormente, ya sea por olvido o por alguna modificación, que se quiera hacer al programa.

Segundo, se es sistemático en la estructuración del programa y se evita tener que cambiar el código que el control pondrá automáticamente. En el caso que se introduzca un registro al final del programa cuya numeración ya haya pasado, el control lo pondrá en el lugar que corresponda.

Existen diferentes tipos de frase de programación, pero son muy similares, a continuación se muestra la estructura de las frases de programación para torno, FANUC 8025.

N----	G----	X----	Z----	F----	S----	T----	M----	(----)
-------	-------	-------	-------	-------	-------	-------	-------	----------

Donde:

N----. Numero de frase o bloque

G----. Funciones preparatorias

X----. Coordenada eje X o R

Z----. Coordenada eje Z o A

F----. Velocidad de avance

S----. Velocidad de giro

T----. Selección

M----. Funciones auxiliares

(----). Comentario.

La única restricción es que no pueden ser usados dos o mas palabras (funciones o comandos) en el mismo bloque con excepción de los códigos G.

2.6 Trigonometría

La trigonometría fue inventada hace más de 2000 años por los griegos, quienes necesitaban métodos precisos para medir ángulos y lados de triángulos. De hecho, la palabra *trigonometría* se deriva de las palabras griegas *trigonon* (triángulo) y *metria* (medición).

La trigonometría en principio es la rama de las matemáticas que estudia las relaciones entre los ángulos y los lados de los triángulos. Para esto se vale de las razones trigonométricas, las cuales son utilizadas frecuentemente en cálculos técnicos. En términos generales, la trigonometría es el estudio de las funciones seno, coseno, tangente, cotangente, secante y cosecante (**Swokowski 2007**).

2.6.1 Unidades angulares.

Un ángulo está formado por dos semirrectas que tienen el mismo origen. Este origen común se llama vértice del ángulo y las dos semirrectas se llaman lados del ángulo. En la medida de ángulos, y por tanto en trigonometría, se emplean tres unidades, si bien la más utilizada en la vida cotidiana es el Grado sexagesimal, en matemáticas es el Radián la más utilizada, y se define como la unidad natural para medir ángulos, el Grado centesimal se desarrolló como la unidad más próxima al sistema decimal, se usa en topografía, arquitectura o en construcción (**Peterson 2001**).

Radián: es la medida del ángulo central de un círculo subtendido por un arco igual en longitud al radio del círculo, en una circunferencia completa hay 2π radianes.

Grado sexagesimal: unidad angular que divide una circunferencia en 360° .

Grado centesimal: unidad angular que divide la circunferencia en 400 grados centesimales.

Tabla 2.1 Relación entre grados y radianes.

(1) $180^\circ = \pi$ radianes
(2) $1^\circ = (\pi/180)$ radianes ≈ 0.0175 radianes
(3) 1 radian = $(180^\circ/\pi) \approx 57.2958^\circ$

Cuando se usa la medida angular en radianes, no se deben indicar unidades (Swokowski 2007).

2.6.2 Razones trigonométricas.

Para cada α , las razones están determinadas de manera única y, por tanto, son funciones de α . Se conocen como *funciones trigonométricas* y se designan como *seno*, *coseno*, *tangente*, *cotangente*, *secante* y *cosecante* y se abrevian como sin, cos, tan, cot, sec y csc, respectivamente. El símbolo $\sin \alpha$, o $\sin(\alpha)$, se usa para la razón a/c , que la función seno asocia con α . Los valores de las otras cinco funciones se denotan de manera semejante.

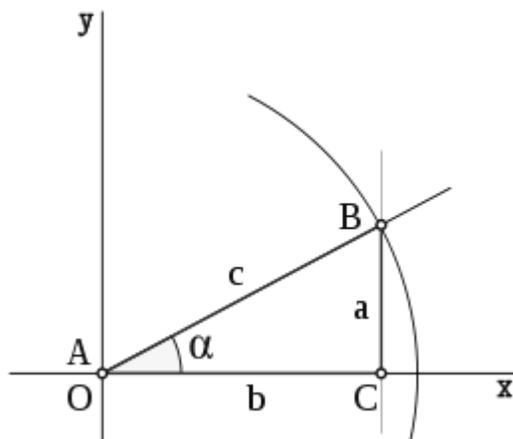


Figura 2.1 Triángulo rectángulo..

Si α es el ángulo en la Figura 2.2 se refiere a los lados b , a y c como el *lado adyacente*, el *lado opuesto* y la *hipotenusa*, respectivamente. Para denotar las longitudes de los lados se usa usualmente los términos *ady*, *op* e *hip*. Con esta

notación, las funciones trigonométricas pueden expresarse como sigue (Swokowski 2007).

Tabla 2.2 Funciones trigonométricas.

$\sin(\alpha) = \frac{op}{hip}$	$\cos(\alpha) = \frac{ady}{hip}$	$\tan(\alpha) = \frac{op}{ady}$
$\csc(\alpha) = \frac{hip}{op}$	$\sec(\alpha) = \frac{hip}{ady}$	$\cot(\alpha) = \frac{ady}{op}$

2.6.3 Sentido de las funciones trigonométricas.

Dados los ejes de coordenadas cartesianas X-Y, de centro O, y una circunferencia goniométrica (circunferencia de radio la unidad) con centro en O; el punto de corte de la circunferencia con el lado positivo de las X, lo señalamos como punto E (véase Figura 2.2)

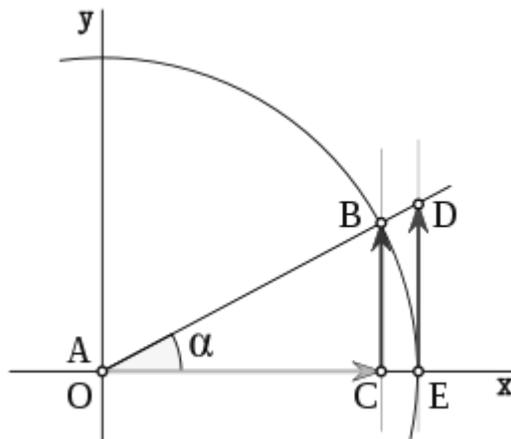


Figura 2.2 Triángulo rectángulo..

La recta r, que pasa por O y forma un ángulo α sobre el eje de las X, corta a la circunferencia en el punto B, la vertical que pasa por B, corta al eje X en C, la vertical que pasa por E corta a la recta r en el punto D.

Por semejanza de triángulos: $\frac{\overline{CB}}{\overline{OC}} = \frac{\overline{ED}}{\overline{OE}}$

Los puntos E y B están en la circunferencia de centro O, por eso la distancia \overline{OE} y \overline{OB} son el radio de la circunferencia, en este caso al ser una circunferencia de radio = 1, y dadas las definiciones de las funciones trigonométricas: $\sin(\alpha) = \overline{CB}$; $\cos(\alpha) = \overline{OC}$; $\tan(\alpha) = \overline{ED}$

$$\text{Tenemos: } \frac{\sin(\alpha)}{\cos(\alpha)} = \frac{\tan(\alpha)}{1}$$

La tangente es la relación del seno entre el coseno.

Primer cuadrante

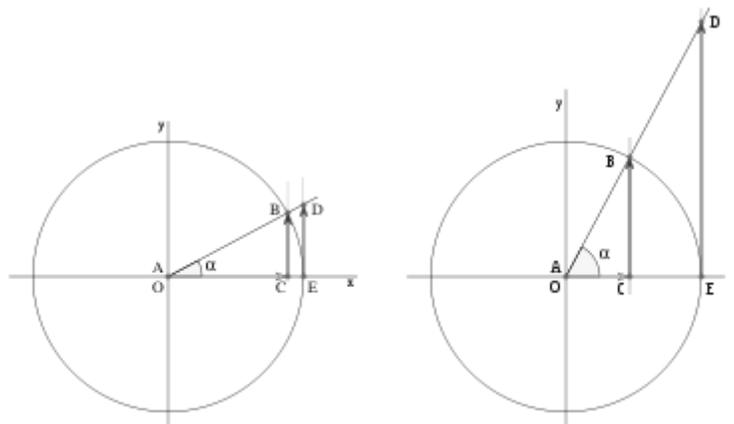


Figura 2.3 Primer cuadrante.

Partiendo de la representación geométrica mostrada en la Figura 2.3 de las funciones trigonométricas, se puede observar las variaciones de las funciones a medida que aumenta el ángulo α .

Para $\alpha = 0$, tenemos que A, C, y D coinciden en B, por tanto: $\sin(0) = 0$; $\cos(0) = 1$; $\tan(0) = 0$;

Si aumenta progresivamente el valor de α , las distancias \overline{CB} y \overline{ED} aumentaran progresivamente, mientras que \overline{OC} disminuirá.

Percatarse que \overline{OC} y \overline{CB} están limitados por la circunferencia y por tanto su máximo valor absoluto será 1, pero \overline{ED} no está limitado, dado que D es el punto de corte de la recta r que pasa por O, y la vertical que pasa por E, en el momento en el que el ángulo $\alpha = 0.5\pi \text{ rad}$, la recta r será la vertical que pasa por O. Dos rectas verticales no se cortan, o lo que es lo mismo la distancia \overline{ED} será infinita.

La tangente toma valor infinito cuando $\alpha = \frac{\pi}{2} \text{ rad}$, el seno vale 1 y el coseno 0.

Segundo cuadrante

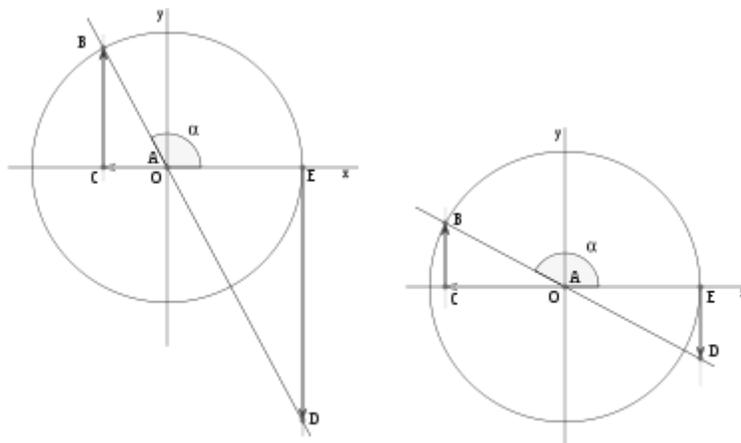


Figura 2.4 Segundo cuadrante.

En la Figura 2.4 se muestra que cuando el ángulo α supera el ángulo recto, el valor del seno empieza a disminuir según el segmento \overline{CB} , el coseno aumenta según el segmento \overline{OC} , pero en el sentido negativo de las X, el valor del coseno toma sentido negativo, si bien su valor absoluto aumenta cuando el ángulo sigue creciendo.

La tangente para un ángulo α inferior a $0.5\pi \text{ rad}$ se hace infinita en el sentido positivo de las Y, para el ángulo recto la recta vertical r que pasa por O y la vertical que pasa por E no se cortan, por lo tanto la tangente no toma ningún valor real, cuando el ángulo supera los $0.5\pi \text{ rad}$ y pasa al segundo cuadrante la prolongación de r corta a la vertical que pasa por E en el punto D real, en el lado negativo de las Y, la tangente \overline{ED} por tanto toma valor negativo en el sentido de las Y, y su valor absoluto disminuye a medida que el ángulo α aumenta progresivamente hasta los $\pi \text{ rad}$.

Resumiendo: en el segundo cuadrante el seno de α , \overline{CB} , disminuye progresivamente su valor desde 1, que toma para $\alpha = 0.5\pi \text{ rad}$, hasta que valga

0, para $\alpha = \pi \text{ rad}$, el coseno, \overline{OC} , toma valor negativo y su valor varia desde 0 para $\alpha = 0.5\pi \text{ rad}$, hasta -1 , para $\alpha = \pi \text{ rad}$. La tangente conserva la relación:

$$\tan(\alpha) = \frac{\sin(\alpha)}{\cos(\alpha)} \text{ incluyendo el signo de estos valores.}$$

Tercer cuadrante

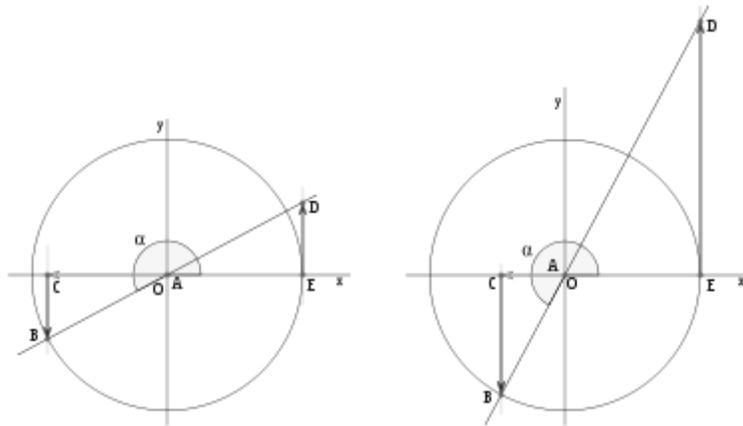


Figura 2.5 Tercer cuadrante.

En el tercer cuadrante, comprendido entre los valores del ángulo $\alpha = \pi \text{ rad}$ a $\alpha = 1.5\pi \text{ rad}$, se produce un cambio de los valores del seno el coseno y la tangente, desde los que toman para π : $\sin(\pi) = 0$; $\cos(\pi) = -1$; $\tan(\pi) = 0$.

Cuando el ángulo α aumenta progresivamente, el seno aumenta en valor absoluto en el sentido negativo de las Y, el coseno disminuye en valor absoluto en el lado negativo de las X, y la tangente aumenta del mismo modo que lo hacia en el primer cuadrante (véase Figura 2.5).

A medida que el ángulo crece el punto C se acerca a O, y el segmento \overline{OC} , el coseno, se hace más pequeño en el lado negativo de las X. El punto B, intersección de la circunferencia y la vertical que pasa por C, se aleja del eje de las X, en el sentido negativo de las Y, el seno, \overline{CB} .

Y el punto D, intersección de la prolongación de la recta r y la vertical que pasa por E, se aleja del eje las X en el sentido positivo de las Y, con lo que la tangente, \overline{ED} , aumenta igual que en el primer cuadrante

Cuando el ángulo α alcance $1.5\pi \text{ rad}$, el punto C coincide con O y el coseno valdrá cero, el segmento \overline{CB} será igual al radio de la circunferencia, en el lado

negativo de las Y, y el seno valdrá -1 , la recta r del ángulo y la vertical que pasa por E serán paralelas y la tangente tomara valor infinito por el lado positivo de las Y.

El seno el coseno y la tangente siguen conservando la misma relación, tanto en valores como en signo, nótese que cuando el coseno vale cero, la tangente se hace infinito.

Cuarto cuadrante

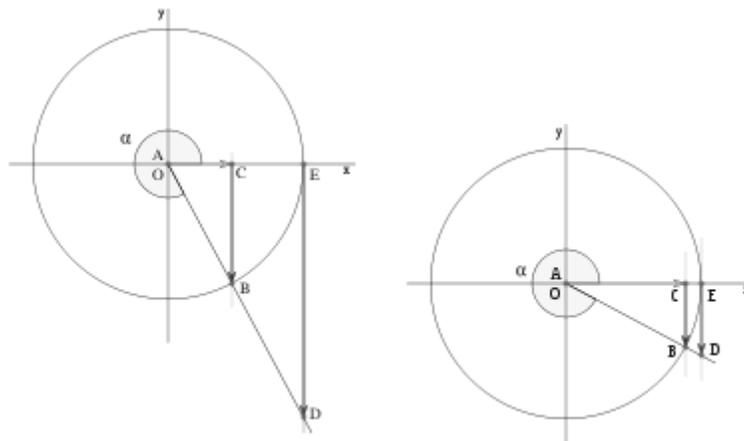


Figura 2.6 Cuarto cuadrante.

En el cuarto cuadrante, que comprende los valores del ángulo α entre $1.5\pi \text{ rad}$ y $2\pi \text{ rad}$ las variables trigonométricas varían desde los valores que toman para $1.5\pi \text{ rad}$ hasta los que toman para $2\pi \text{ rad}$ pasando al primer cuadrante, completando una rotación:

$$\sin(1.5\pi) = -1 \quad \sin(2\pi) = 0$$

$$\cos(1.5\pi) = 0 \quad \cos(2\pi) = 1$$

$$\tan(1.5\pi) = \infty \quad \tan(2\pi) = 0$$

Como puede verse en la Figura 2.6 a medida que el ángulo α aumenta, aumenta el coseno \overline{OC} en el lado positivo de las X, el seno \overline{CB} disminuye en el lado negativo de las Y, y la tangente \overline{ED} también disminuye en el lado negativo de las Y.

Cuando α , vale 2π ó 0π al completar una rotación completa los puntos B, C y D, coinciden en E, haciendo que el seno y la tangente valga cero, y el coseno uno, del mismo modo que al comenzarse el primer cuadrante.

2.7 Función de un compilador

A grandes rasgos un compilador es un programa que lee un programa escrito en un lenguaje, el lenguaje fuente, y lo traduce a un programa equivalente en otro lenguaje, el lenguaje objeto. Como parte importante de este proceso de traducción, el compilador informa a su usuario de la presencia de errores en el programa fuente. Los primeros compiladores a principios de los años cincuenta. Es difícil dar una fecha exacta de la aparición del primer compilador, porque en un principio gran parte del trabajo de experimentación y aplicación se realizó de manera independiente por varios grupos. Gran parte de los primeros trabajos de compilación estaba relacionada con la traducción de fórmulas aritméticas a código de máquina.

2.7.1 Partes en las que trabaja un compilador

Conceptualmente un compilador opera en fases. Cada una de las cuales transforma el programa fuente de una representación en otra. En la Figura 2.7 se muestra una descomposición típica de un compilador. En la práctica se pueden agrupar fases y las representaciones intermedias entre las fases agrupadas no necesitan ser construidas explícitamente.

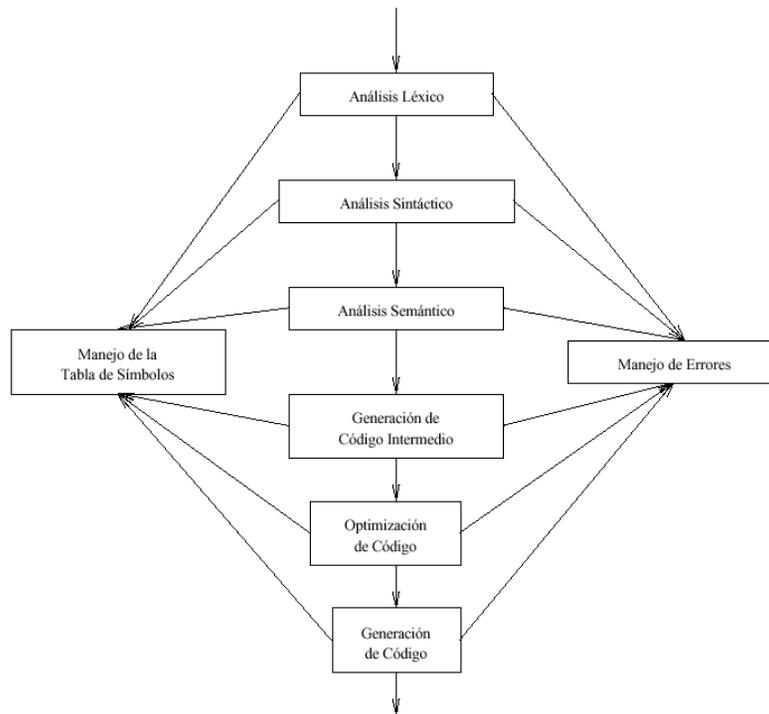


Figura 2.7 programa objeto.

2.7.2 Clasificación de compiladores

El programa compilador traduce las instrucciones en un lenguaje de alto nivel a instrucciones que la computadora puede interpretar y ejecutar. Para cada lenguaje de programación se requiere un compilador separado. El compilador traduce todo el programa antes de ejecutarlo. Los compiladores son programas de traducción insertados en la memoria por el sistema operativo para convertir programas de cómputo en pulsaciones electrónicas ejecutables (lenguaje de máquina). Los compiladores pueden ser de:

- Una sola pasada: examina el código fuente una vez, generando el código o programa objeto.

- Pasadas múltiples: requieren pasos intermedios para producir un código en otro lenguaje, y una pasada final para producir y optimizar el código producido durante los pasos anteriores.
- Optimización: lee un código fuente, lo analiza y descubre errores potenciales sin ejecutar el programa.
- Compiladores incrementales: generan un código objeto instrucción por instrucción (en vez de hacerlo para todo el programa) cuando el usuario teclea cada orden individual. El otro tipo de compiladores requiere que todos los enunciados o instrucciones se compilen conjuntamente.
- Ensamblador: el lenguaje fuente es lenguaje ensamblador y posee una estructura sencilla.
- Compilador cruzado: se genera código en lenguaje objeto para una máquina diferente de la que se está utilizando para compilar. Es perfectamente normal construir un compilador de Pascal que genere código para MS-DOS y que el compilador funcione en Linux y se haya escrito en C++.
- Compilador con montador: compilador que compila distintos módulos de forma independiente y después es capaz de enlazarlos.
- Autocompilador: compilador que está escrito en el mismo lenguaje que va a compilar. Evidentemente, no se puede ejecutar la primera vez. Sirve para hacer ampliaciones al lenguaje, mejorar el código generado, etc.
- Metacompilador: es sinónimo de compilador de compiladores y se refiere a un programa que recibe como entrada las especificaciones del lenguaje para el que se desea obtener un compilador y genera como salida el compilador para ese lenguaje. El desarrollo de los metacompiladores se encuentra con la dificultad de unir la generación de código con la parte de análisis. Lo que sí se han desarrollado son generadores de analizadores léxicos y sintácticos.
- Descompilador: es un programa que acepta como entrada código máquina y lo traduce a un lenguaje de alto nivel, realizando el proceso inverso a la compilación.

2.7.3 Analizador sintáctico

Es una de las partes de un compilador que transforma su entrada en un árbol de derivación.

El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada. Un analizador léxico crea tokens de una secuencia de caracteres de entrada y son estos tokens los que son procesados por el analizador sintáctico para construir la estructura de datos, por ejemplo un árbol de análisis o árboles de sintaxis abstracta.

El análisis sintáctico también es un estado inicial del análisis de frases de lenguaje natural. Es usado para generar diagramas de lenguajes que usan flexión gramatical, como los idiomas romances o el latín. Los lenguajes habitualmente reconocidos por los analizadores sintácticos son los lenguajes libres de contexto. Cabe notar que existe una justificación formal que establece que los lenguajes libres de contexto son aquellos reconocibles por un autómata de pila, de modo que todo analizador sintáctico que reconozca un lenguaje libre de contexto es equivalente en capacidad computacional a un autómata de pila.

3 Metodología

En este capítulo se desarrollarán los cálculos matemáticos necesarios para ir creando nuestro ciclo de maquinado. Tales ciclos como lo son (cilindrado, conicidad, desbaste transversal, etc.) cuentan con una serie de parámetros los cuales nos dan una idea de cómo se puede descifrar el ciclo. Esto con la finalidad de optimizar el código G y hacer de esto un proceso más rápido y sencillo. Otro punto importante es como quedaría nuestro programa de maquinado, lo cual se especifica a continuación.

3.1 Definición de código G y M

En la Figura 3.1 se muestra la estructura de un programa que es la llave del entendimiento de la terminología CNC, esta debe contener la siguiente información; dirección, dato, palabra, bloque y programa, los cuales se describen a continuación.

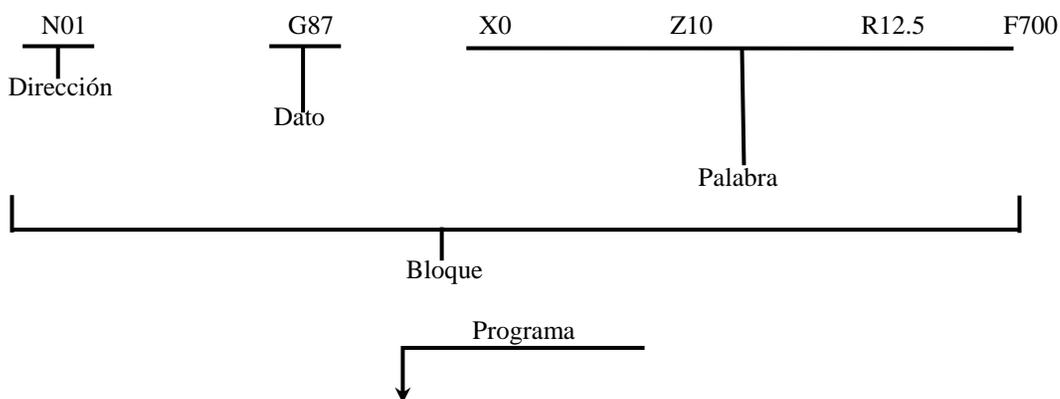


Figura 3.1 Formato de programa CNC.

3.1.1 Códigos G's

Funciones de movimiento de máquina que incluyen movimientos rápidos, avances radiales, pausas, ciclos. (Smid, 2003). Se les conoce también como comandos preparatorios y requeridos para el maquinado de piezas.

3.1.2 Códigos M's

Funciones misceláneas que se requieren para el maquinado de piezas, pero que no son funciones de movimiento de la máquina. Algunos de ellos pueden con el arranque y paro del husillo, cambio de herramienta, encendido y apagado del refrigerante, paro de programa, etc. (Smid, 2003).

3.2 Restricciones en los Bloques

- Deben contener únicamente un solo bloque, el cual contenga los parámetros necesarios para el maquinado de la pieza
- Debe contener únicamente una velocidad de corte
- Debe contener únicamente un parámetro en Z
- Debe contener únicamente un parámetro en X
- Debe contener R total de la pieza
- Y únicamente debe de contar con una sola instrucción G__.

3.3 Ciclos enlatados o repetitivos

Estos ciclos tienen la particularidad de agrupar en una o varias líneas una cantidad considerable de operaciones facilitando así la programación CNC tal como se muestra. En el presente trabajo solo se programará un ciclo enlatado que sirva de ejemplo a la posterior inclusión de más ciclos como son los de desbaste transversal, conicidad, cilindrado, etc.

G84: Cilindrado (G84 X... Z... R... P...F)

G87: Conicidad (G87 X... Z... R... P...F)

Donde.

X: Es la posición final de corte, para G84 y G87

Z: Es la posición final de corte, para G84 y G87

R: Siempre va ha ser negativo (cuadro de corte -X)

F: Velocidad de avance del cortador

P: Es la profundidad de corte

La función G84 o 87 son ciclos enlatados, una línea de información del programa capacitará a la herramienta para ejecutar un cierto número de movimientos distintos.

R: Distancia incremental del comienzo de corte a la posición final del corte para determinar su distribución.

Ya tomando en cuenta todos los parámetros los cuales ya se definieron, se puede proseguir con todo el proceso que conlleva un ciclo de maquinado.

3.4 Ciclo de conicidad

Ejemplo 1: Ciclo enlatado (conicidad)

G87-----Conicidad

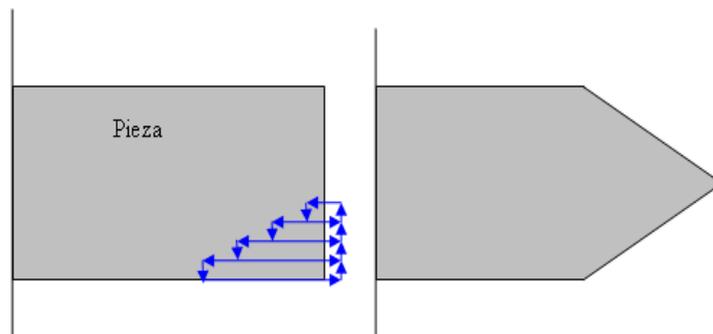


Figura 3.2 Ciclo de conicidad.

La Figura 3.2 muestra las trayectorias para llevar acabo este ejemplo, además se tendrá que tomar en cuenta ciertos parámetros, los cuales nos ayudaran a descifrar las características particulares de este comando. Tales parámetros como lo son G87, X, Z, R, F y P son esenciales para este ejemplo en particular.

Donde:

G87, es el comando el cual se utiliza para llevar acabo la conicidad de la pieza.

X, lo llamaremos como el diámetro al cual se desea dejar la pieza.

Z, será la longitud a la cual se desea hacer la conicidad.

R, será el radio total de la pieza a maquinar.

F, será el avance de la herramienta de corte.

P: Es la profundidad de corte

En este ejemplo en particular se esta tomando el maquinado de conicidad el cual se identifica como un G87 (**Smid 2000**).

Con un diámetro X de 0 mm.

Una longitud a maquinar de 80 mm.

Un radio total de la barra de 12.5 mm.

Un avance de 5200.

Y una profundidad de corte de 1 mm

Ya teniendo los parámetros, se lleva acabo el cálculo matemático el cual nos ayudara a conocer los subparametros del maquinado, como lo son L (indica que tanto entra la herramienta de corte), y esta se determina de la siguiente manera:

$$L = R - \frac{x}{2}$$

Sustituyendo valores se tiene:

$$L = 12.5\text{mm} - \frac{0\text{mm}}{2}$$

$$L = 12.5\text{mm}$$

Después se determina el número de pasadas totales, el cual esta dado por:

$$\text{Num_pasadas} = \frac{L}{P}$$

Donde:

P= 1mm (profundidad de corte).

Sustituyendo valores en la ecuación tenemos:

$$\text{Num_pasadas} = \frac{12.5}{1}$$

$$\text{Num_pasadas} = 12.5 \text{ pasadas}$$

En todos los casos cuando existan decimales, el número de pasadas se redondeara como por ejemplo 12.5 pasadas, (se llevaran acabo 12 pasadas con una profundidad de 1 mm por pasada y una última de 0.5 mm), dando como resultado:

$$\text{Num_pasadas totales} = 13 \text{ pasadas.}$$

Posteriormente se sigue con el cálculo matemático para determinar θ_g .

$$\theta_g = \tan^{-1} \left(\frac{L}{Z} \right)$$

Donde:

θ_g = ángulo de referencia o ángulo general.

Sustituyendo valores tenemos:

$$\theta_g = \tan^{-1} \left(\frac{12.5}{80} \right)$$
$$\theta_g = 8.88^\circ$$

Ya obtenido el ángulo (θ_g) se pueden determinar los catetos adyacentes restantes de cada triángulo, esto se expresa de la siguiente manera:

$$Z_1 = \frac{12.5 - 1}{\tan 8.88} = 73.6 \text{ mm}$$

$$Z_2 = \frac{11.5 - 1}{\tan 8.88} = 67.2 \text{ mm}$$

$$Z_3 = \frac{10.5 - 1}{\tan 8.88} = 60.8 \text{ mm}$$

$$Z_4 = \frac{9.5 - 1}{\tan 8.88} = 54.4 \text{ mm}$$

$$Z_5 = \frac{8.5 - 1}{\tan 8.88} = 48 \text{ mm}$$

$$Z_6 = \frac{7.5 - 1}{\tan 8.88} = 41.6 \text{ mm}$$

$$Z_7 = \frac{6.5 - 1}{\tan 8.88} = 35.2 \text{ mm}$$

$$Z_8 = \frac{5.5 - 1}{\tan 8.88} = 28.8 \text{ mm}$$

$$Z_9 = \frac{4.5 - 1}{\tan 8.88} = 22.4 \text{ mm}$$

$$Z_{10} = \frac{3.5 - 1}{\tan 8.88} = 16 \text{ mm}$$

$$Z_{11} = \frac{2.5 - 1}{\tan 8.88} = 9.6 \text{ mm}$$

$$Z_{12} = \frac{1.5 - 1}{\tan 8.88} = 3.2 \text{ mm}$$

El -1 indica el mm que entra la herramienta de corte por cada pasada que se realiza, todo esto, obviamente en el eje (-x).

Ya por ultimo Z_{13} solo entra en el eje (-x) 0.5 mm y de este punto se desplaza a las coordenadas $-Z=10$, $X=0$, esto con un comando G01 el cual realiza la acción de una interpolación lineal. Todo esto con la finalidad de tumbar los escalonamientos que se fueron formando durante el desbaste de la pieza, lo cual se conoce también como ciclo de “acabado”.

3.5 Ciclo de cilindrado

Ejemplo 2: Ciclo enlatado (cilindrado)

G84-----Cilindrado

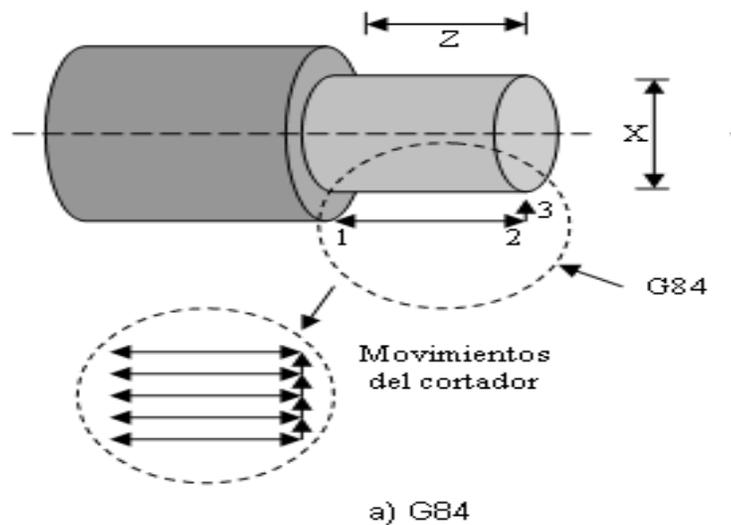


Figura 3.3 Movimiento del cortador para el ciclo G84.

El Ciclo de Cilindrado G84 es mostrado en la Figura 3.3 y para llevar acabo este ejemplo se tendrá que tomar en cuenta ciertos parámetros, los cuales nos ayudaran a descifrar las características particulares de este comando. Tales parámetros como lo son G84, X, Z, R, F y P son esenciales para este ejemplo en particular.

En este ejemplo en particular se esta tomando el maquinado de careado longitudinal el cual se identifica como un G84 (Smid 2000).

Con un diámetro X de 6 mm.

Una longitud a maquinar de 30 mm.

Un radio total de la barra de 9.525 mm.

Un avance de 3200.

Y una profundidad de corte de 1 mm.

Ya teniendo los parámetros, se lleva acabo el cálculo matemático el cual nos ayudara a conocer los subparametros del maquinado, como lo son L (indica que tanto entra la herramienta de corte), y esta se determina de la siguiente manera:

$$L = R - \frac{x}{2}$$

Sustituyendo valores se tiene:

$$L = 9.525\text{mm} - \frac{6\text{mm}}{2}$$

$$L = 6.525\text{mm}$$

Después se determina el número de pasadas totales, el cual esta dado por:

$$\text{Num_pasadas} = \frac{L}{P}$$

Donde:

P= 1mm (profundidad de corte).

Sustituyendo valores en la ecuación tenemos:

$$\text{Num_pasadas} = \frac{6.525}{1}$$

$$\text{Num_pasadas} = 6.525 \text{ pasadas}$$

En todos los casos cuando existan decimales, el número de pasadas se redondeara como por ejemplo 6.525 pasadas, (se llevaran acabo 6 pasadas con una profundidad de 1 mm por pasada y una última de 0.525 mm), dando como resultado:

$$\text{Num_pasadas totales} = 7 \text{ pasadas.}$$

En los primeros seis desbastes de este ciclo de maquinado se va avanzando 1 mm en el eje -x, y se llega a la misma posición (-z+1) y regresa a la posición (z-1), esto con la finalidad de ir dejando un milímetro en cada desbaste que se realiza, para así poder darle un buen acabado a la pieza que se esta maquinando.

Ya por ultimo Z₇ solo entra en el eje (-x) 0.525 mm y de este punto se desplaza a las coordenadas -Z=30, X=0 y posteriormente Z=0, X=6.525 esto con un comando G01 el cual realiza la acción de una interpolación lineal. Todo esto con la finalidad de darle un mejor acabado a la pieza.

3.6 Ciclo de arco

Ejemplo 3: Ciclo enlatado (Ciclo de arco)

G101-----Ciclo de arco

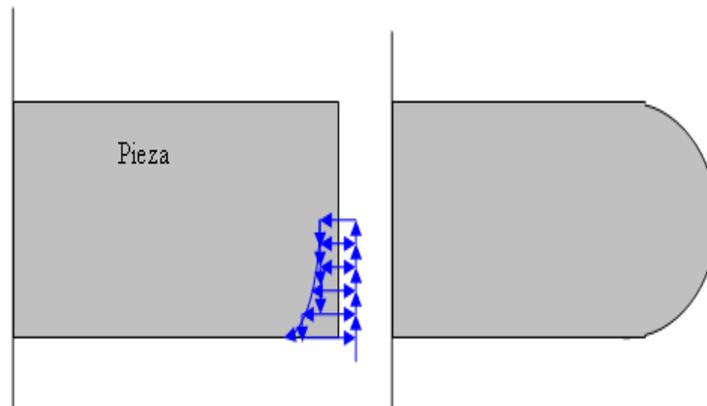


Figura 3.4 Ciclo enlatado (Ciclo de arco).

Aquí se desarrolla el Ciclo para generar una media esfera, cierto maquinado no encontrado en CNC convencionales por tal motivo se le dio el nombre de G101 mostrado en la Figura 3.4 y para llevar acabo este ejemplo se tendrán que tomar en cuenta ciertos parámetros, los cuales nos ayudaran a descifrar las características particulares de este comando. Tales parámetros como lo son G101, X, Z, R, F y P son esenciales para este ejemplo en particular.

En este ejemplo en particular se esta tomando el maquinado de ciclo de arco el cual se identifica como un G101. Este ciclo de maquinado se esta implementando y creando ya que es un ciclo de maquinado con el cual no se contaba.

Con un diámetro X de 0 mm.

Una longitud a maquinar de 9.525 mm.

Un radio total de la barra de 9.525 mm.

Un avance de 3200.

Y una profundidad de corte de 1 mm.

Ya teniendo los parámetros, se lleva a cabo el cálculo matemático el cual nos ayudara a conocer los subparametros del maquinado, como lo son L (indica que tanto entra la herramienta de corte), y esta se determina de la siguiente manera:

$$L = R - \frac{x}{2}$$

Sustituyendo valores se tiene:

$$L = 9.525\text{mm} - \frac{0\text{mm}}{2}$$

$$L = 9.525\text{mm}$$

Después se determina el número de pasadas totales, el cual esta dado por:

$$\text{Num_pasadas} = \frac{L}{P}$$

Donde:

P= 1mm (profundidad de corte).

Sustituyendo valores en la ecuación tenemos:

$$\text{Num_pasadas} = \frac{9.525}{1}$$

$$\text{Num_pasadas} = 9.525 \text{ pasadas}$$

En todos los casos cuando existan decimales, el número de pasadas se redondeara como por ejemplo 9.525 pasadas, (se llevaran a cabo 9 pasadas con

una profundidad de 1 mm por pasada y una última de 0.525 mm), dando como resultado:

Num_pasadas totales= 10 pasadas.

Posteriormente se sigue con el cálculo matemático para determinar la “-z” para cada uno de los 10 desbastes que se llevaran acabo. Para llevar acabo estos cálculos se tendrá que tomar en cuenta la ecuación de la circunferencia la cual esta dada por:

$$r^2 = x^2 + y^2$$

Como se quiere saber en cada uno de los 10 desbastes cuanto es que se va a avanzar en el eje (-z), de la formula se tiene que despejar X para que se le vayan dando valores a Y, y así podamos conocer las distancias deseadas. Despejando X en la formula anterior se tiene:

$$x = (r^2 - y^2)^{1/2}$$

Donde:

X= Distancia en el eje (-z)

r= Radio de la barra

y= Distancia en el eje (-x)

Sustituyendo valores tenemos:

$$Z_0 = [(9.525)^2 - (0)^2]^{1/2}$$

$$Z_0 = 9.525 \text{ mm}$$

Esta pasada no la realiza, ya que primero entra 1 mm, pero cabe mencionarla para ir viendo el comportamiento del cortador en cuanto al desbaste.

$$Z_1 = [(9.525)^2 - (1)^2]^{1/2}$$

$$Z_1 = 9.472 \text{ mm}$$

$$Z_2 = [(9.525)^2 - (2)^2]^{1/2}$$
$$Z_2 = 9.312 \text{ mm}$$

$$Z_3 = [(9.525)^2 - (3)^2]^{1/2}$$
$$Z_3 = 9.041 \text{ mm}$$

$$Z_4 = [(9.525)^2 - (4)^2]^{1/2}$$
$$Z_4 = 8.644 \text{ mm}$$

$$Z_5 = [(9.525)^2 - (5)^2]^{1/2}$$
$$Z_5 = 8.107 \text{ mm}$$

$$Z_6 = [(9.525)^2 - (6)^2]^{1/2}$$
$$Z_6 = 7.397 \text{ mm}$$

$$Z_7 = [(9.525)^2 - (7)^2]^{1/2}$$
$$Z_7 = 6.459 \text{ mm}$$

$$Z_8 = [(9.525)^2 - (8)^2]^{1/2}$$
$$Z_8 = 5.169 \text{ mm}$$

$$Z_9 = [(9.525)^2 - (9)^2]^{1/2}$$
$$Z_9 = 3.118 \text{ mm}$$

$$Z_{10} = [(9.525)^2 - (9.525)^2]^{1/2}$$
$$Z_{10} = 0.0 \text{ mm}$$

Por lo tanto para saber cuanto es el avance en $-z$ se tienen que sustituir los valores obtenidos de Z_{10} a Z_1 en la siguiente formula:

$$Z_{\text{real}} = r - (r^2 - y^2)^{1/2}$$

$$Z_{\text{real}} = 9.525 - Z_{10}$$

$$Z_{\text{real}} = 9.525 - 0.0$$

$$Z_{\text{real}} = 9.525 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_9$$

$$Z_{\text{real}} = 9.525 - 3.118$$

$$Z_{\text{real}} = 6.407 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_8$$

$$Z_{\text{real}} = 9.525 - 5.169$$

$$Z_{\text{real}} = 4.356 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_7$$

$$Z_{\text{real}} = 9.525 - 6.459$$

$$Z_{\text{real}} = 3.066 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_6$$

$$Z_{\text{real}} = 9.525 - 7.397$$

$$Z_{\text{real}} = 2.128 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_5$$

$$Z_{\text{real}} = 9.525 - 8.107$$

$$Z_{\text{real}} = 1.418 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_4$$

$$Z_{\text{real}} = 9.525 - 8.644$$

$$Z_{\text{real}} = 0.881 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_3$$

$$Z_{\text{real}} = 9.525 - 9.041$$

$$Z_{\text{real}} = 0.484 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_2$$
$$Z_{\text{real}} = 9.525 - 9.312$$
$$Z_{\text{real}} = 0.215 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_1$$
$$Z_{\text{real}} = 9.525 - 9.472$$
$$Z_{\text{real}} = 0.052 \text{ mm}$$

$$Z_{\text{real}} = 9.525 - Z_0$$
$$Z_{\text{real}} = 9.525 - 9.525$$
$$Z_{\text{real}} = 0.0 \text{ mm}$$

Al ir aumentado de Z_{10} a Z_0 se puede notar claramente que la circunferencia se va formando, dando como resultado cero, cuando el radio toma el mismo valor de Z_0 .

Ya por ultimo Z_{real} entra en el eje (-x) 9.525 mm y de este punto $Z=0$, $-X=9.525$ se desplaza al punto $-Z=9.525$, $X=0$, esto con un comando G02 el cual realiza la acción de una interpolación circular en sentido horario. Todo esto con la finalidad de tumbar los escalonamientos que se fueron formando durante el desbaste de la pieza, lo cual se conoce también como ciclo de "acabado".

3.7 Ciclo de arco interno

Ejemplo 3: Ciclo enlatado (Ciclo de arco interno)

G102-----Ciclo de arco interno

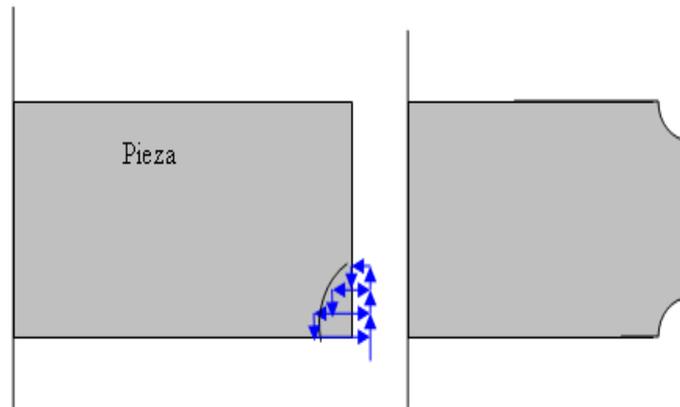


Figura 3.5 Ciclo enlatado (Ciclo de arco interno).

El ejemplo es mostrado en la Figura 3.5 y para llevar acabo este se tendrán que tomar en cuenta ciertos parámetros, los cuales nos ayudaran a descifrar las características particulares de este comando. Tales parámetros como lo son G102, X, Z, R, F y P son esenciales para este ejemplo en particular.

En este ejemplo en particular se esta tomando el maquinado de ciclo de arco interno el cual se identifica como un G102. Este ciclo de maquinado se esta implementando y creando ya que es un ciclo de maquinado con el cual no se contaba.

Con un diámetro X de 0 mm.

Una longitud a maquinar de 9.525 mm.

Un radio total de la barra de 9.525 mm.

Un avance de 3200.

Y una profundidad de corte de 1 mm.

Ya teniendo los parámetros, se lleva a cabo el cálculo matemático el cual nos ayudara a conocer los subparametros del maquinado, como lo son L (indica que tanto entra la herramienta de corte), y esta se determina de la siguiente manera:

$$L = R - \frac{x}{2}$$

Sustituyendo valores se tiene:

$$L = 9.525\text{mm} - \frac{0\text{mm}}{2}$$

$$L = 9.525\text{mm}$$

Después se determina el número de pasadas totales, el cual esta dado por:

$$\text{Num_pasadas} = \frac{L}{P}$$

Donde:

P= 1mm (profundidad de corte).

Sustituyendo valores en la ecuación tenemos:

$$\text{Num_pasadas} = \frac{9.525}{1}$$

$$\text{Num_pasadas} = 9.525 \text{ pasadas}$$

En todos los casos cuando existan decimales, el número de pasadas se redondeara como por ejemplo 9.525 pasadas, (se llevaran a cabo 9 pasadas con una profundidad de 1 mm por pasada y una última de 0.525 mm), dando como resultado:

Num_pasadas totales= 10 pasadas.

Posteriormente se sigue con el cálculo matemático para determinar la “-z” para cada uno de los 10 desbastes que se llevaran acabo. Para llevar acabo estos cálculos se tendrá que tomar en cuenta la ecuación de la circunferencia la cual esta dada por:

$$r^2 = x^2 + y^2$$

Como se quiere saber en cada uno de los 10 desbastes cuanto es que se va a avanzar en el eje (-z), de la formula se tiene que despejar X para que se le vayan dando valores a Y, y así podamos conocer las distancias deseadas. Despejando X en la formula anterior se tiene:

$$x = (r^2 - y^2)^{1/2}$$

Donde:

X= Distancia en el eje (-z)

r= Radio de la barra

y= Distancia en el eje (-x)

Sustituyendo valores tenemos:

$$Z_1 = [(9.525)^2 - (1)^2]^{1/2}$$

$$Z_1 = 9.472 \text{ mm}$$

$$Z_2 = [(9.525)^2 - (2)^2]^{1/2}$$

$$Z_2 = 9.312 \text{ mm}$$

$$Z_3 = [(9.525)^2 - (3)^2]^{1/2}$$

$$Z_3 = 9.041 \text{ mm}$$

$$Z_4 = [(9.525)^2 - (4)^2]^{1/2}$$

$$Z_4 = 8.644 \text{ mm}$$

$$Z_5 = [(9.525)^2 - (5)^2]^{1/2}$$
$$Z_5 = 8.107 \text{ mm}$$

$$Z_6 = [(9.525)^2 - (6)^2]^{1/2}$$
$$Z_6 = 7.397 \text{ mm}$$

$$Z_7 = [(9.525)^2 - (7)^2]^{1/2}$$
$$Z_7 = 6.459 \text{ mm}$$

$$Z_8 = [(9.525)^2 - (8)^2]^{1/2}$$
$$Z_8 = 5.169 \text{ mm}$$

$$Z_9 = [(9.525)^2 - (9)^2]^{1/2}$$
$$Z_9 = 3.118 \text{ mm}$$

$$Z_{10} = [(9.525)^2 - (9.525)^2]^{1/2}$$
$$Z_{10} = 0.0 \text{ mm}$$

Al ir aumentando de uno en uno los valores de Y se puede notar claramente que la circunferencia se va formando, dando como resultado cero, cuando el radio al cuadrado toma el mismo valor de Y al cuadrado.

Ya por ultimo Z_{10} entra en el eje (-x) 9.525 mm y de este punto $Z=0$, $-X=9.525$ se desplaza al punto $-Z=9.525$, $X=0$, esto con un comando G03 el cual realiza la acción de una interpolación circular en sentido antihorario. Todo esto con la finalidad de tumbar los escalonamientos que se fueron formando durante el desbaste de la pieza, lo cual se conoce también como ciclo de “acabado”.

3.8 Ciclo de cilindrado interno

Ejemplo 2: Ciclo enlatado (cilindrado interno)

G84-----Cilindrado interno

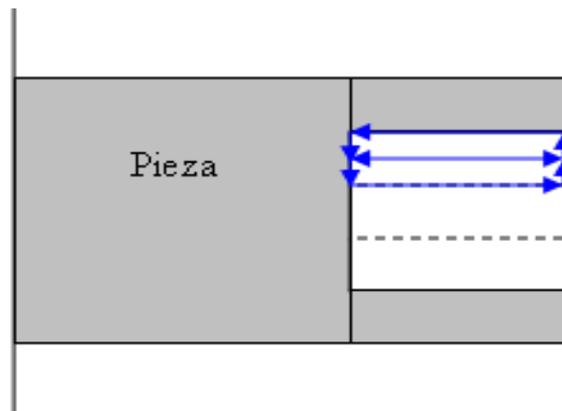


Figura 3.6 Ciclo enlatado (cilindrado interno).

En la Figura 3.6 se muestra el ejemplo y para llevar acabo este se tendrán que tomar en cuenta ciertos parámetros, los cuales nos ayudaran a descifrar las características particulares de este comando. Tales parámetros como lo son G84, X, Z, D, F y P son esenciales para este ejemplo en particular.

En este ejemplo en particular se esta tomando el maquinado de careado longitudinal el cual se identifica como un G84 (**Smid 2000**).

Con un diámetro X de 15.05 mm.

Una longitud a maquinar de 30 mm.

Un diámetro de la broca utilizada de 12.7 mm.

Un avance de 3200.

Y una profundidad de corte de 1 mm.

Ya teniendo los parámetros, se hace el barrenado en la pieza a la longitud de 30 mm con la broca indicada, posteriormente se lleva acabo el cálculo matemático el cual nos ayudara a conocer los subparametros del maquinado, como lo son L (indica que tanto entra la herramienta de corte), y esta se determina de la siguiente manera:

$$L = \frac{X - D}{2}$$

Sustituyendo valores se tiene:

$$L = \frac{15.05 - 12.7}{2}$$

$$L = 1.175\text{mm}$$

Después se determina el número de pasadas totales, el cual esta dado por:

$$\text{Num_pasadas} = \frac{L}{P}$$

Donde:

P= 1mm (profundidad de corte).

Sustituyendo valores en la ecuación tenemos:

$$\text{Num_pasadas} = \frac{1.175}{1}$$

$$\text{Num_pasadas} = 1.175 \text{ pasadas}$$

En todos los casos cuando existan decimales, el número de pasadas se redondeara como por ejemplo 1.175 pasadas, (se llevara acabo una pasada con una profundidad de 1 mm y una última de 0.175 mm), dando como resultado:

Num_pasadas totales= 2 pasadas.

En el primer desbastes de este ciclo de maquinado se va avanzando 1 mm en el eje $-x$, y se llega a la misma posición ($-z+1$) y regresa a la posición ($z-1$), esto con la finalidad de ir dejando un milímetro en cada desbaste que se realiza, para así poder darle un buen acabado a la pieza que se esta maquinando.

Ya por ultimo Z solo entra en el eje ($-x$) 0.175 mm y de este punto se desplaza a las coordenadas $-Z=30$, $X=0$ y posteriormente $Z=0$, $X=1.175$ esto con un comando G01 el cual realiza la acción de una interpolación lineal. Todo esto con la finalidad de darle un mejor acabado a la pieza.

4 Resultados obtenidos

Con la elaboración de estos ciclos de maquinado se pudo observar que los cálculos matemáticos que se realizaron y se programaron, siguen correctamente las trayectorias que se deseaban, dando como resultado el correcto maquinado de la pieza. Por lo cual se puede decir que se optimizó el código G para la realización de los diferentes tipos de maquinado que se tienen, tales como (conicidad G87, cilindrado G84, arco G101 “este ciclo de maquinado no existía por lo cual se le propuso este nombre o código”, arco interno G102 y cilindrado interno G84-i).

Por lo tanto se logró el objetivo de hacer más fácil y amena la comunicación usuario-máquina, esto con la finalidad de realizar diferentes tipos de maquinado de una forma mucho más sencilla, sin tener que hacer una serie de cálculos previos al maquinado, simplemente capturar los parámetros necesarios los cuales pide el maquinado y listo.

También se logró la generación de tecnología propia CNC aplicable a torno, lo cual es muy importante para nuestra facultad y para nosotros mismos, ya que uno se va dando cuenta de que tan importante es la innovación hoy en día. Todo esto también con la finalidad de ir mejorando las trayectorias de los maquinados lo más que se pueda para así poder contar con ciclos de maquinados eficientes y de muy buena calidad.

Y por supuesto se llevaron a cabo las pruebas de funcionalidad para cada uno de estos ciclos de maquinado de manera satisfactoria. Dando como resultado las trayectorias deseadas.

4.1 Resultados obtenidos de ciclo G84

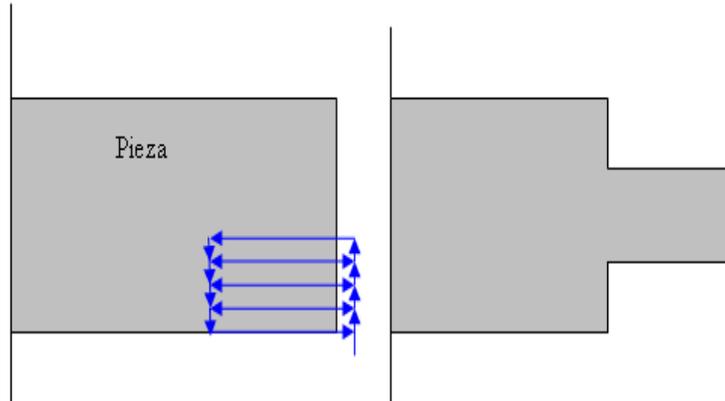


Figura 4.1 Ciclo de Cilindrado.

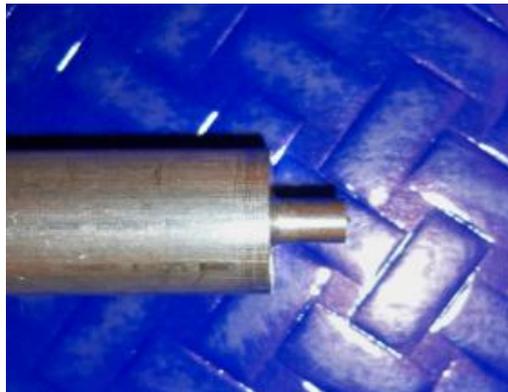


Figura 4.2 Pieza obtenida del Ciclo de Cilindrado.

Como se puede observar en la Figura 4.2 y Figura 4.3, las trayectorias que se calcularon fueron correctamente las que siguió la herramienta de corte a la hora del maquinado, dando como resultado las trayectorias deseadas para este ciclo G84 (cilindrado).

4.2 Resultados obtenidos de ciclo G87

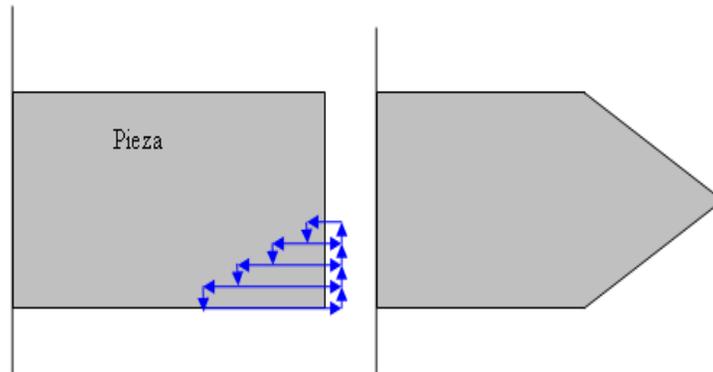


Figura 4.3Ciclo de Conicidad.

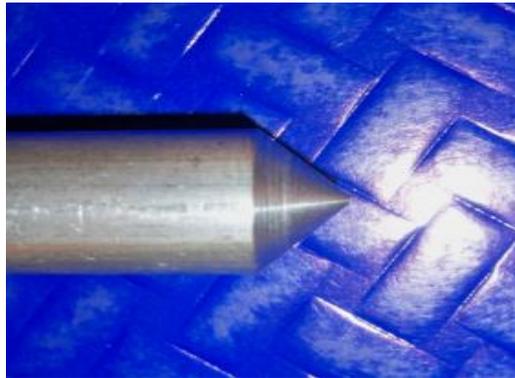


Figura 4.4 Pieza obtenida del Ciclo de Conicidad.

Como se puede observar en la Figura 4.3 y Figura 4.4, las trayectorias que se calcularon fueron correctamente las que siguió la herramienta de corte a la hora del maquinado, dando como resultado las trayectorias deseadas para este ciclo G87 (conicidad).

4.3 Resultados obtenidos de ciclo G101

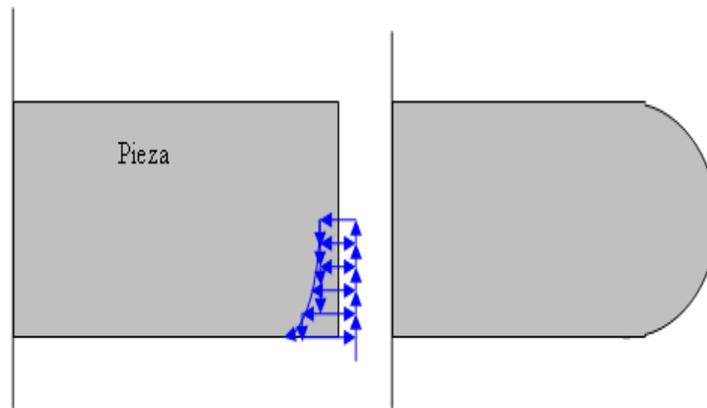


Figura 4.5 Ciclo de Arco.



Figura 4.6 Pieza obtenida del Ciclo de Arco.

Como se puede observar en la Figura 4.5 y Figura 4.6, las trayectorias que se calcularon fueron correctamente las que siguió la herramienta de corte a la hora del maquinado, dando como resultado las trayectorias deseadas para este ciclo G101 (arco).

4.4 Resultados obtenidos de ciclo G102

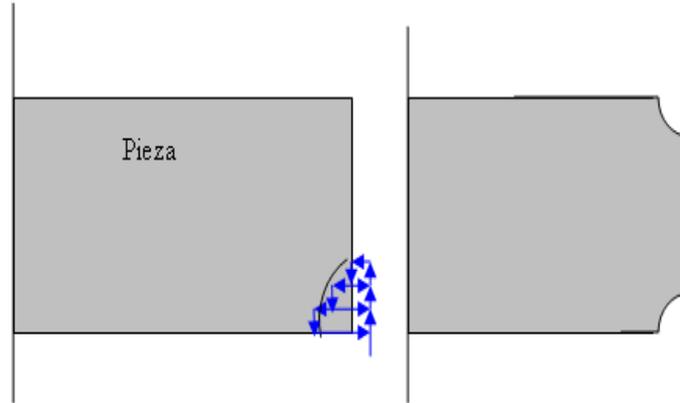


Figura 4.7 Ciclo de Arco Interno.

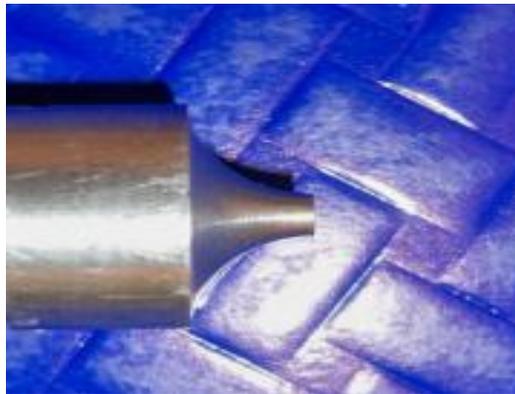


Figura 4.8 Pieza obtenida del Ciclo de Arco Interno.

Como se puede observar en la Figura 4.7 y Figura 4.8, las trayectorias que se calcularon fueron correctamente las que siguió la herramienta de corte a la hora del maquinado, dando como resultado las trayectorias deseadas para este ciclo G102 (arco interno).

4.5 Resultados obtenidos de ciclo G84-i

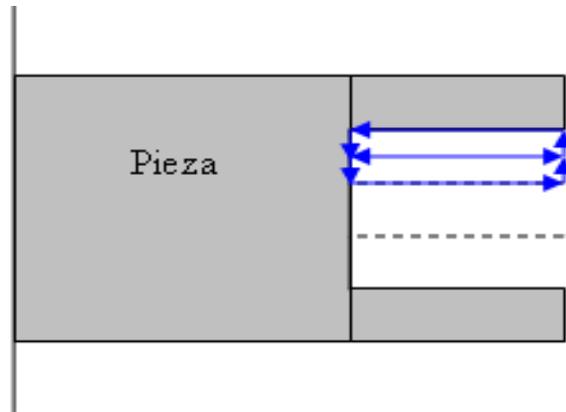


Figura 4.9 Ciclo de Cilindrado Interno.

Como se puede observar en la Figura 4.9, las trayectorias que se calcularon fueron correctamente las que siguió la herramienta de corte a la hora del maquinado, dando como resultado las trayectorias deseadas para este ciclo G84-i (cilindrado interno).

5 Conclusiones

Con la filmación de los videos y las piezas maquinadas se puede concluir que los ciclos de maquinado siguen correctamente las trayectorias calculadas y programadas para cada uno de estos ciclos, dando como resultado las trayectorias deseadas.

Al aplicar los ciclos de maquinado (ciclos enlatados) se reduce considerablemente el tamaño del archivo, así como el número de instrucciones necesarias para el maquinado de la pieza. Evitando cálculos innecesarios previos al maquinado. Estas ventajas en el ámbito laboral se traducen en ganancias ya que se logra minimizar los tiempos de maquinado considerablemente y se aumenta la producción en serie.

El programa realizado cumple con las expectativas esperadas, ya que cuenta con todos los cálculos matemáticos e instrucciones bien definidas para cada uno de los ciclos de maquinado. Dando como resultado una buena estructura de programa.

Referencias

- Andrade Flores Roberto. 1996. Máquinas herramientas de control numérico computarizado.
- García Barbosa Andrés Jorge, Córdoba Ernesto Nieto. 2004. Inicio del proceso de automatización utilizando tecnología de grupos.
- Gibbs David, Crandell Thomas M. 1991. An Introduction to CNC Maching and Programming. Second Edition.
- Gómez Loenzo Roberto Augusto. 2007. Sistema de control numérico en tiempo real de arquitectura abierta reconfigurable.
- Gutiérrez Emmanuel. 2004. Modelos de comportamiento y sistema de control adaptable para la industria automotriz en el área de optimización de procesos de maquinado en torno CNC. ITESM Toluca.
- Londoño Ospina Nelson, Álvarez Díaz Jairo, León Simanca Pedro y Zapata Edwin Marín. Septiembre de 2005. Descripción del diseño y construcción de un torno de control numérico.
- Mikell P Groover. 1997. Fundamentos de manufactura moderna, materiales, procesos y sistemas, México. Ed. Prentice may.
- Nieves Antonio y Domínguez Federico C. 2002. Métodos numéricos aplicados a la ingeniería, Editorial Continental, Segunda edición.
- Ogata Katsuhiko. 2005. Ingeniería de control moderna. cuarta edición Pearson Prentice Hall.
- Smid Peter 2006. CNC Programming Techniques. First Edition.
- Smid Peter. 2003. CNC Programming Handbook. Second Edition.
- Trejo Hernández Miguel. 2007. Métodos y aportaciones en la selección óptima de las condiciones de corte en procesos de torneado CNC.
- Valerio Netto Antonio y Anitelle Tarcisio Anésio. 2005. Procedure for the implementation of a virtual cnc lathe.

Apéndice A

Las órdenes de desplazamiento que corresponden a las funciones M y se presentan a continuación.

FUNCIONES M

Grupo 0	M03:	Husillo CONECTADO en sentido horario
	M04:	Husillo CONECTADO en sentido antihorario
	M05:	PARO del husillo
	M19:	PARO exacto del husillo
Grupo 1	M38:	Paro exacto CONECTADO
	* M39:	Paro exacto DESCONECTADO
Grupo 2	M00:	Paro programado
	M17:	Final de subprograma
	M30:	Final del programa con regreso al principio del programa
Grupo 3	M08:	Refrigerante CONECTADO
	* M09:	Refrigerante DESCONECTADO
Grupo 5	M25:	Abrir el elemento de sujeción
	M26:	Cerrar el elemento de sujeción
Grupo 6	M20:	Pinola de contrapunto retrasada
	M21:	Pinola de contrapunto adelantada
Grupo 7	* M23:	Cubeta de recogida retrasada
	M24:	Cubeta de recogida adelantada
Grupo 8	M50:	Desactivación de la lógica de giro de la torreta revólver
	M51:	Activación de la lógica de giro de la torreta revólver
Grupo 9	M52:	Desactivación del automatismo de la puerta
	M53:	Activación del automatismo de la puerta

* Se activa al encender el equipo

Esquema del programa en Visual C++:

```

/*-----
edit_sim.c -- Funciones de simulacion
-----*/
#include <windows.h>
#include <stdio.h> //para imprimir en pantalla (sprintf)
#include <math.h>
#include <time.h>
#define pi 3.141516
#include "cbw.h"

BOOL CALLBACK Preguntas (HWND hDlg, UINT iMsg, WPARAM wParam, LPARAM lParam);
void procesaMensajes2(HWND hwnd,HACCEL hAccelerator);
void OkMessage1 (HWND hwnd, char *szMessage);
void Errores (HWND hwnd, HWND hwndTab);
//Ramiro
void sleep(unsigned int mseconds);
FILE *DATOS;
//R
extern HINSTANCE hInst;
char b_stop;
int vr;
//Ramiro
USHORT DataValue = 0;
int BoardNum = 1;
int UDStat=0;
int Comparador=0;
int contador=1;
//R

LRESULT APIENTRY TextSim (HWND hwnd,HWND hwndTab,HWND hwndEdit,HWND mm3,HWND mm8,
                          HWND mm9,HACCEL hAccelerator,int ZoomP)
{
    POINT pt;
    HPEN hLapiz;

    HDC hdc;
    double czMover, cxMover,zfin, xfin, zini, xini;
    int iLength; //long. en el doc.
    char xx[20],yy[20],zz[20],ff[20],AA[20],BB[20],RR[20],FF[20],b,rr[20];
    char dd,e,f1,g,c='a',ss[3];
    int mens=1, i;
    int pos,posx,posy,posz,posf,posr;
    double x,y,z,f;
    int salto,co,ds,inc=-1,Zoom_x,Zoom_z;
    double realz,realx,pos_z,pos_x;
    char ach[80];
    double num;
    double l,Num_pasadas,Num_pasadasT,Num_pasadas1,Num_pasadas2,pasada_extra=0;
    double teta_g, z_prima;
    double prof=1; // profundidad de un 1mm

    ////////////para g002
    double z1,z2,z3,x3,x2,x1,x1_z1_x2_z2_x3_z3_;
    double alfa,beta,A,B,radio,r,r_1;
    ////////////

    PSTR a; //palabra x palabra en todo el documento(hwndedit)
    ////////////funcion par obtener las coordenadas de la pantalla
    RECT rc;
    static SIZE gMinSize;
    int az,ax;

    GetWindowRect(hwnd,&rc);
    gMinSize.cx = (rc.right - rc.left);
    gMinSize.cy = (rc.bottom - rc.top);
    az=gMinSize.cx;
    ax=gMinSize.cy;

    if ((az<=700)
        {
            az=700;
        }
    if ((ax<=500)
        {
            ax=500;
        }
    }

    ////////////
    iLength = GetWindowTextLength (hwndEdit) ;
    if (NULL == (a = (PSTR) malloc (iLength + 1)))
        return FALSE ;
    GetWindowText (hwndEdit, a, iLength+1) ;
    for(i=0;i<20;i++) //inicializacion de x
    {
        xx[i]=0;
        yy[i]=0;
        zz[i]=0;
        ff[i]=0;
    }
}

```



```

case 0x0a: //salto de linea
SendMessage (hwndEdit, EM_SETSEL, vr, vr+2) ;//selecciona la palabra
SendMessage (hwndEdit, EM_SCROLLCARET, 0, 0) ;

break;
case 0x00: //fin de archivo
SendMessage (hwndEdit, EM_SETSEL, vr-1, vr) ;//selecciona la palabra
SendMessage (hwndEdit, EM_SCROLLCARET, 0, 0) ;
if(mens==1)
{
wsprintf (ach, "Simulacion terminada", a) ;
MessageBox (hwndEdit, ach,ach , MB_OK) ;
vr=-1;
}

break;

case 'G': //Comandos G

ss[0]=a[vr+1];
ss[1]=a[vr+2];
ss[2]=a[vr+3];

num=atof(ss);

////////////////////////////////////hugo

if (num==87) //comando G87 ciclo de conicidad
{
double retiro=0;

////////////////////////////////////
sprintf (ach,"Ciclo enlatado de conicidad G87", 1) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
pos=vr; //guardo la pocicion del vector de texto

c='a'; //inicializacion de variable
while(cl=0x0a) //salto de linea
{
for(i=0;i<20;i++)
{
xx[i]=0;
yy[i]=0;
zz[i]=0;
ff[i]=0;
}

//borrando el vector

c=a[pos];

if(c=='X')

{

co=0;

posx=pos;

dd='a'; //inicializacion de variable

while ((dd!=0x20) & (dd!=0x0a)) //espacio

{

dd=a[posx];

posx=posx+1;

if(dd!=0x0a)

{

if(dd!=0x20)

{

xx[co]=a[posx];

co=co+1;

}

}

}

}

pos=posx-1;//u

vr=pos;

x=atof(xx);

// x=ZoomP*1000*x;

////////////////////////////////////
sprintf (ach,"X=%f", x) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////

}

```

```

                                                                    if(c=='R')
                                                                    {
co=0;
posr=pos;
e='a'; //inicializacion de variable
while((e!=0x20) & (e!=0x0a))
                                                                    {
e=a[posr];
posr=posr+1;
if(e!=0x0a)
{
if(e!=0x20)
{
rr[co]=a[posr];
co=co+1;
}
}
                                                                    }

pos=posr-1;
vr=pos;
r=atof(rr);
                                                                    }

                                                                    if(c=='Z')
                                                                    {
co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))
                                                                    {
f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
zz[co]=a[posz];
co=co+1;
}
}
                                                                    }

pos=posz-1;
vr=pos;
                                                                    }

                                                                    z=atof(zz);
                                                                    //z=ZoomP*1000*z;
                                                                    ///////////////////////////////////////////////////////////////////
                                                                    sprintf (ach,"Z=%f", z) ;
                                                                    MessageBox (hwnd, ach,ach , MB_OK) ;
                                                                    ///////////////////////////////////////////////////////////////////

```

```

}

if(c=='F')
{
    co=0;
    posf=pos;
//inicializacion de variable

    g=a[posf];
    posf=posf+1;
    if(g!=0x0a)
    {
        ff[co]=a[posf];
        co=co+1;
    }

    pos=posf-1;
    vr=pos;

        f=atof(ff);
        //////////////////////////////////////
        sprintf (ach,"F=%f", f) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////
}

(c!='Y') & (c!='Z') & (c!='F')

    pos=pos+1;

}

////////////////////////////////////calculos matematicos
l=r/(x/2);

        //////////////////////////////////////
        sprintf (ach,"l=%f", l) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

Num_pasadas=l/prof;

        //////////////////////////////////////
        sprintf (ach,"Num_pasadas=%f", Num_pasadas) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

Num_pasadas1=(int)Num_pasadas;
        //conversion a entero
        //////////////////////////////////////
        sprintf (ach,"Num_pasadas1=%f", Num_pasadas1) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

Num_pasadas2=Num_pasadas;

        //////////////////////////////////////
        sprintf (ach,"Num_pasadas2=%f", Num_pasadas2) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

Num_pasadasT=Num_pasadas2-Num_pasadas1;

        //////////////////////////////////////
        sprintf (ach,"Num_pasadasT=%f", Num_pasadasT) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

if((Num_pasadasT>0) & (Num_pasadasT<1))
{
    Num_pasadas=Num_pasadas1+1;
    pasada_extra=1;
}
else
{
    Num_pasadas=Num_pasadas;
    pasada_extra=0;
}

        //////////////////////////////////////
        sprintf (ach,"Num_pasadas=%f", Num_pasadas) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

        //////////////////////////////////////
        sprintf (ach,"pasada_extra=%f", pasada_extra) ;
        MessageBox (hwnd, ach,ach , MB_OK) ;
        //////////////////////////////////////

```

```

teta_g=atan(l/z);
for(i=0;i<Num_pasadas;i++)
{
////////////////////////////////////primera linea (desbaste eje x)
////////////////////////////////////
    sprintf (ach,"Desvaste_transversa=%i", (i+1));
    MessageBox (hwnd, ach,ach , MB_OK);
    //////////////////////////////////////

    hdc = GetDC (hwndTab);

    // x=0.0000006317*az*x;
        // z=0.0000006317*az*z;
        czMover =(0);

        if((i+1)==Num_pasadas)
        // 0 //para Z
            if(pasada_extra==1)
            {
                cxMover =(-(0.0000006317*az*Num_pasadasT))*ZoomP*1000; // -1mm //para X

                //////////////////////////////////////
                sprintf (ach,"Num_pasadasT_ultima=%f", Num_pasadasT);
                MessageBox (hwnd, ach,ach , MB_OK);
                //////////////////////////////////////
            }
            else{}
        else
        {
            cxMover =(-(0.0000006317*az*prof))*ZoomP*1000; // -1mm //para X
        }

        zfin = zini + czMover;
        xfin = xini + cxMover;
        hLapiz = CreatePen(PS_SOLID, 1, RGB(255, 0, 0));
        SelectObject(hdc, hLapiz);

        MoveToEx(hdc, (int)zini, (int)xini, &pt);
        LineTo(hdc,(int)zfin, (int)xfin);

        ReleaseDC (hwndTab, hdc);
        DeleteObject(hLapiz);

        //ir al programa del editor
        zini = zfin;
        xini = xfin;
        //////////////////////////////////////
////////////////////////////////////

////////////////////////////////////segunda linea (desbaste eje z)
////////////////////////////////////
    sprintf (ach,"Desvaste_longitudinal=%i", (i+1));
    MessageBox (hwnd, ach,ach , MB_OK);
    //////////////////////////////////////

    if((i+1)==Num_pasadas)
    // 0 //para Z
        if(pasada_extra==1)
        {
            //z_prima=(Num_pasadasT)/(tan(teta_g));
            z_prima=0;
            //////////////////////////////////////
            sprintf (ach,"pasada_fraccionaria=%f", Num_pasadasT);
            MessageBox (hwnd, ach,ach , MB_OK);
            //////////////////////////////////////
        }
        else{}
    else
    {
        z_prima=-((i+1)*prof)/(tan(teta_g));

        compensacion para el acabado final
        z_prima=z_prima-prof; //z-

    }

    hdc = GetDC (hwndTab);
    // x=0.0000006317*az*x;
        // z=0.0000006317*az*z;

        czMover =(-(0.0000006317*az*z_prima))*ZoomP*1000; // -z' //para Z
        cxMover =(0);
    // 0 //para X

        zfin = zini + czMover;

        xfin = xini + cxMover;

```



```

                                                    {
                                                    //retiro
                                                    double retiro=0;
                                                    ////////////////////////////////////////////////////
                                                    sprintf (ach,"Ciclo enlatado de conicidad G84", 1) ;
                                                    MessageBox (hwnd, ach,ach , MB_OK) ;
                                                    ////////////////////////////////////////////////////
                                                    pos=vr; //guardo la pocion del vector de texto

                                                    c='a'; //inicializacion de variable
                                                    while(cl=0x0a) //salto de linea
                                                    {
                                                        for(i=0;j<20;i++)
                                                        {
                                                            xx[i]=0;
                                                            yy[i]=0;
                                                            zz[i]=0;
                                                            ff[i]=0;
                                                        }
                                                    }

//borrando el vector
c=a[pos];
if(c=='X')
{
co=0;
posx=pos;
dd='a'; //inicializacion de variable
while ((dd!=0x20) & (dd!=0x0a)) //espacio
{
dd=a[posx];
posx=posx+1;
if(dd!=0x0a)
{
if(dd!=0x20)
{
xx[co]=a[posx];
co=co+1;
}
}
}
pos=posx-1;//u
vr=pos;
x=atof(xx);
// x=ZoomP*1000*x;
////////////////////////////////////////////////////
sprintf (ach,"X=%f", x) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////////////////////
}

if(c=='R')
{
co=0;
posr=pos;
e='a'; //inicializacion de variable
while((e!=0x20) & (e!=0x0a))
{
e=a[posr];
posr=posr+1;
if(e!=0x0a)
{
if(e!=0x20)

```

```

        {
            rr[co]=a[posr];
            co=co+1;
        }
    }

    pos=posr-1;
    vr=pos;
    r=atof(rr);

    //////////////////////////////////////
    sprintf (ach,"R=%f", r) ;
    MessageBox (hwnd, ach,ach , MB_OK) ;
    //////////////////////////////////////
}
if(c=='Z')
{

co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))
{

f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
                zz[co]=a[posz];
                co=co+1;
            }
        }
    }

    pos=posz-1;
    vr=pos;

    z=atof(zz);

    //z=ZoomP*1000*z;
    //////////////////////////////////////
    sprintf (ach,"Z=%f", z) ;
    MessageBox (hwnd, ach,ach , MB_OK) ;
    //////////////////////////////////////
}

if(c=='F')
{

co=0;
posf=pos;
//inicializacion de variable
g='a';
while(g!=0x0a)
{

g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
                ff[co]=a[posf];

```

```

co=co+1;

}

pos=posf-1;

vr=pos;

f=atof(ff);
////////////////////////////////////
sprintf (ach,"F=%f", f);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

}

else if ((cl='X') &

(cl='Y') & (cl='Z') & (cl='F'))

{

pos=pos+1;

}

////////////////////////////////////calculos matematicos
l=r*(x/2);

////////////////////////////////////
sprintf (ach,"l=%f", l);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

Num_pasadas=l/prof;

////////////////////////////////////
sprintf (ach,"Num_pasadas=%f", Num_pasadas);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

Num_pasadas1=(int)Num_pasadas;
//conversion a entero
////////////////////////////////////
sprintf (ach,"Num_pasadas1=%f", Num_pasadas1);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

Num_pasadas2=Num_pasadas;

////////////////////////////////////
sprintf (ach,"Num_pasadas2=%f", Num_pasadas2);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

Num_pasadasT=Num_pasadas2-Num_pasadas1;

////////////////////////////////////
sprintf (ach,"Num_pasadasT=%f", Num_pasadasT);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

if((Num_pasadasT>0) & (Num_pasadasT<1))
{
Num_pasadas=Num_pasadas1+1; //se incrementa una pasada
pasada_extra=1; //bandera ON para el ultimo desbaste
}
else
{
Num_pasadas=Num_pasadas; //se deja el numero de pasada
pasada_extra=0;
}

////////////////////////////////////
sprintf (ach,"Num_pasadas=%f", Num_pasadas);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

////////////////////////////////////
sprintf (ach,"pasada_extra=%f", pasada_extra);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

for(i=0;i<Num_pasadas;i++)
{
////////////////////////////////////primera linea (desbaste eje x)
////////////////////////////////////
sprintf (ach,"Desvaste_transversa=%i", (i+1));
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

hdc = GetDC (hwndTab);

// x=0.0000006317*az*x;
// z=0.0000006317*az*z;
czMover =(0);

if((i+1)==Num_pasadas)
// para Z
if(pasada_extra==1)
{
cxMover =-(0.0000006317*az*Num_pasadasT)*ZoomP*1000; // -1mm //para X
}

////////////////////////////////////
sprintf (ach,"2009=%f", Num_pasadasT);
MessageBox (hwnd, ach,ach , MB_OK);
}

```



```

////////////////////////////////////
sprintf (ach,"retiro=%i", (i+1) );
MessageBox (hwnd, ach,ach , MB_OK );
////////////////////////////////////
hdc = GetDC (hwndTab) ;
// x=0.000006317*az*x;
// z=0.000006317*az*z;

//retiro
retiro=prof;

// czMover =((0.000006317*az*z_prima))*ZoomP*1000; // +z' //para Z
czMover =(0);
// 0 //para X
cxMover =(+ (0.000006317*az*retiro))*ZoomP*1000; // -1mm //para X

zfin = zini + czMover ;
xfin = xini + cxMover ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
SelectObject(hdc, hLapiz);

MoveToEx(hdc, (int)zini, (int)xini, &pt);
LineTo(hdc,(int)zfin, (int)xfin);

ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

//ir al programa del editor
zini = zfin;
xini = xfin;
////////////////////////////////////

////////////////////////////////////
////////////////////////////////////Tercera linea (de regreso)
////////////////////////////////////
sprintf (ach,"regreso=%i", (i+1) );
MessageBox (hwnd, ach,ach , MB_OK );
////////////////////////////////////

hdc = GetDC (hwndTab) ;
// x=0.000006317*az*x;
// z=0.000006317*az*z;

czMover =((0.000006317*az*z_prima))*ZoomP*1000; // +z' //para Z
cxMover =(0);
// 0 //para X

zfin = zini + czMover ;
xfin = xini + cxMover ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
SelectObject(hdc, hLapiz);

MoveToEx(hdc, (int)zini, (int)xini, &pt);
LineTo(hdc,(int)zfin, (int)xfin);

ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

//ir al programa del editor
zini = zfin;
xini = xfin;
////////////////////////////////////

////////////////////////////////////regreso a posicion
////////////////////////////////////
sprintf (ach,"retiro=%i", (i+1) );
MessageBox (hwnd, ach,ach , MB_OK );
////////////////////////////////////

hdc = GetDC (hwndTab) ;
// x=0.000006317*az*x;
// z=0.000006317*az*z;

//retiro
retiro=prof;

czMover =(0);
// 0 //para X
cxMover =(- (0.000006317*az*retiro))*ZoomP*1000; // -1mm //para X

zfin = zini + czMover ;
xfin = xini + cxMover ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
SelectObject(hdc, hLapiz);

MoveToEx(hdc, (int)zini, (int)xini, &pt);
LineTo(hdc,(int)zfin, (int)xfin);

```



```

        zfin = zini + czMover ;
    xfin = xini + cxMover ;
    hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
    SelectObject(hdc, hLapiz);

    MoveToEx(hdc, (int)zini, (int)xini, &pt);
    LineTo(hdc,(int)zfin, (int)xfin);

        ReleaseDC (hwndTab, hdc) ;
        DeleteObject(hLapiz);

        //ir al programa del editor
        zini = zfin;
        xini = xfin;
        ////////////
    }

    ////////////////////////////////////////
    ////////////////////////////////////////hugo2_final

    ////////////////////////////////////////hugo3_inicio

        if (num==101) //comando G101 ciclo de cilindrado
        {
            //retiro
            double retiro=0;
            double var1=1;

            ////////////////////////////////////////
            sprintf (ach,"Ciclo enlatado de conicidad G101", 1) ;
            MessageBox (hwnd, ach,ach , MB_OK) ;
            ////////////////////////////////////////
            pos=vr; //guardo la pocion del vector de texto

            c='a'; //inicializacion de variable
            while(cl=0x0a) //salto de linea
            {
                for(i=0;i<20;i++)
                {
                    xx[i]=0;
                    yy[i]=0;
                    zz[i]=0;
                    ff[i]=0;
                }

//borrando el vector
c=a[pos];
if(c=='X')
{
co=0;
posx=pos;
dd='a'; //inicializacion de variable
while ((dd!=0x20) & (dd!=0x0a)) //espacio
{
dd=a[posx];
posx=posx+1;
if(dd!=0x0a)
{
if(dd!=0x20)
{
xx[co]=a[posx];
co=co+1;
}
}
}
}
pos=posx-1;//u
vr=pos;

x=atof(xx);

// x=ZoomP*1000*x;
//////////////////////////////////////
sprintf (ach,"X=%f", x) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
//////////////////////////////////////

```

```

}
if(c=='R')
{
co=0;
posr=pos;
e='a'; //inicializacion de variable
while((e!=0x20) & (e!=0x0a))
{
e=a[posr];
posr=posr+1;
if(e!=0x0a)
{
if(e!=0x20)
{
rr[co]=a[posr];
co=co+1;
}
}
}
}
pos=posr-1;
vr=pos;
r=atof(rr);
radio=atof(rr);
radio=ZoomP*1000*radio;//DUDA 1000
////////////////////////////////////
sprintf (ach,"R=%f", r);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////
}
if(c=='Z')
{
co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))
{
f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
zz[co]=a[posz];
co=co+1;
}
}
}
}
pos=posz-1;
vr=pos;
z=atof(zz);

```

```

//z=ZoomP*1000*z;
////////////////////////////////////
sprintf (ach,"Z=%f", z) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
}

if(c=='F')
{
co=0;
posf=pos;
//inicializacion de variable
g='a';
while(g!=0x0a)
{
g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
ff[co]=a[posf];
co=co+1;
}
}
pos=posf-1;
vr=pos;

f=atof(ff);
////////////////////////////////////
sprintf (ach,"F=%f", f) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
}
else if ((c=='X') &
{
}

}

////////////////////////////////////calculos matematicos
l=r-(x/2);
////////////////////////////////////
sprintf (ach,"l=%f", l) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
Num_pasadas=l/prof;
////////////////////////////////////
sprintf (ach,"Num_pasadas=%f", Num_pasadas) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
Num_pasadas1=(int)Num_pasadas; //conversion a entero
////////////////////////////////////
sprintf (ach,"Num_pasadas1=%f", Num_pasadas1) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
Num_pasadas2=Num_pasadas;
////////////////////////////////////
sprintf (ach,"Num_pasadas2=%f", Num_pasadas2) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
Num_pasadasT=Num_pasadas2-Num_pasadas1;
////////////////////////////////////
sprintf (ach,"Num_pasadasT=%f", Num_pasadasT) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
if((Num_pasadasT>0) & (Num_pasadasT<1))
{
Num_pasadas=Num_pasadas1+1; //se incrementa una pasada
pasada_extra=1; //bandera ON para el ultimo desbaste
}
else
{
Num_pasadas=Num_pasadas; //se deja el numero de pasada
pasada_extra=0;
}
////////////////////////////////////
sprintf (ach,"Num_pasadas=%f", Num_pasadas) ;

```

```

        MessageBox (hwnd, ach,ach , MB_OK ) ;
        //////////////////////////////////////
        //////////////////////////////////////
        sprintf (ach,"pasada_extra=%f", pasada_extra) ;
        MessageBox (hwnd, ach,ach , MB_OK ) ;
        //////////////////////////////////////
////////////////////////////////////
//teta_g=atan(l/z);
for(i=0;i<Num_pasadas;i++)
{
////////////////////////////////////primera linea (desbaste eje x)
////////////////////////////////////
        sprintf (ach,"Desvaste_transversa=%i", (i+1)) ;
        MessageBox (hwnd, ach,ach , MB_OK ) ;
        //////////////////////////////////////

        hdc = GetDC (hwndTab) ;
        // x=0.0000006317*az*x;
        // z=0.0000006317*az*z;
        czMover =(0);

        if((i+1)==Num_pasadas)
//      0 //para Z
        {
                if(pasada_extra==1)
                {
////////////////////////////////////
                        sprintf (ach,"2009=%f", Num_pasadasT) ;
                        MessageBox (hwnd, ach,ach , MB_OK ) ;
                        //////////////////////////////////////
                }
                else{
////////////////////////////////////
                        sprintf (ach,"hhh=%f", Num_pasadasT) ;
                        MessageBox (hwnd, ach,ach , MB_OK ) ;
                        //////////////////////////////////////
                        cxMover =(-(0.0000006317*az*prof))*ZoomP*1000; // -1mm //para X
                }
                else
                {
                        cxMover =(-(0.0000006317*az*prof))*ZoomP*1000; // -1mm //para X
                }
        }

        zfin = zini + czMover ;
        xfin = xini + cxMover ;
        hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
        SelectObject(hdc, hLapiz);

        MoveToEx(hdc, (int)zini, (int)xini, &pt);
        LineTo(hdc,(int)zfin, (int)xfin);

        ReleaseDC (hwndTab, hdc) ;
        DeleteObject(hLapiz);

        //ir al programa del editor
        zini = zfin;
        xini = xfin;
        //////////////////////////////////////
////////////////////////////////////
////////////////////////////////////segunda linea (desbaste eje z)
////////////////////////////////////
        sprintf (ach,"Desvaste_longitudinal=%i", (i+1)) ;
        MessageBox (hwnd, ach,ach , MB_OK ) ;
        //////////////////////////////////////

        if((i+1)==Num_pasadas)
//      0 //para Z
        {
                if(pasada_extra==1)
                {
                        //z_prima=(Num_pasadasT)/(tan(teta_g));
                        z_prima=0;

////////////////////////////////////
                        sprintf (ach,"pasada_fraccionaria=%f", Num_pasadasT) ;
                        MessageBox (hwnd, ach,ach , MB_OK ) ;
                        //////////////////////////////////////
                }
                else{
                }
                else
                {
                        z_prima=r-(sqrt((r*r)- (((r-(var1*(prof))))*(r-(var1*(prof)))) ));

                if ((z_prima>1) || (z_prima==1))
                { z_prima=z_prima-prof;}
        }
        }
}
var1=var1+1;

```



```

////////////////////////////////////
ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

////////////////////////////////////
//ir al programa del editor

////////////////////////////////////
}

////////////////////////////////////
////////////////////////////////////hugo3_final

////////////////////////////////////hugo4_inicio

if (num==102) //comando G102 ciclo
{
//retiro
double retiro=0;
double var1=1;

////////////////////////////////////
sprintf (ach,"Ciclo enlatado de conicidad G102", 1) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
pos=vr; //guardo la pocicion del vector de texto

c='a'; //inicializacion de variable
while(c!=0x0a) //salto de linea
{
for(i=0;i<20;i++)
{
xx[i]=0;
yy[i]=0;
zz[i]=0;
ff[i]=0;
}

//borrando el vector
c=a[pos];
if(c=='X')
{
co=0;
posx=pos;
dd='a'; //inicializacion de variable
while ((dd!=0x20) & (dd!=0x0a)) //espacio
{
dd=a[posx];
posx=posx+1;
if(dd!=0x0a)
{
if(dd!=0x20)
{
xx[co]=a[posx];
co=co+1;
}
}
}
}
pos=posx-1;//u
vr=pos;

x=atof(xx);
// x=ZoomP*1000*x;
////////////////////////////////////
sprintf (ach,"X=%f", x) ;
MessageBox (hwnd, ach,ach , MB_OK) ;
////////////////////////////////////
}

if(c=='R')
{
co=0;

```

```

posr=pos;
e='a'; //inicializacion de variable
while((e!=0x20) & (e!=0x0a))
{
e=a[posr];
posr=posr+1;
if(e!=0x0a)
{
if(e!=0x20)
{
rr[co]=a[posr];
co=co+1;
}
}
}

pos=posr-1;
vr=pos;
r=atof(rr);
radio=ZoomP*1000*radio;//DUDA 1000

////////////////////////////////////
sprintf (ach,"R=%f", r);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

}
if(c=='Z')
{

co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))
{
f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
zz[co]=a[posz];
co=co+1;
}
}
}

pos=posz-1;
vr=pos;

z=atof(zz);
//z=ZoomP*1000*z;
////////////////////////////////////
sprintf (ach,"Z=%f", z);
MessageBox (hwnd, ach,ach , MB_OK);
////////////////////////////////////

}

```

```

if(c=='F')
{
co=0;
posf=pos;
//inicializacion de variable
g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
ff[co]=a[posf];
co=co+1;
}
}
pos=posf-1;
vr=pos;
f=atof(ff);
////////////////////////////////////
sprintf(ach,"F=%f", f);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
}
else if ((cl=='X') &
{
pos=pos+1;
}
}

////////////////////////////////////calculos matematicos
l=r*(x/2);
////////////////////////////////////
sprintf(ach,"l=%f", l);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
Num_pasadas=l/prof;
////////////////////////////////////
sprintf(ach,"Num_pasadas=%f", Num_pasadas);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
Num_pasadas1=(int)Num_pasadas;
//conversion a entero
////////////////////////////////////
sprintf(ach,"Num_pasadas1=%f", Num_pasadas1);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
Num_pasadas2=Num_pasadas;
////////////////////////////////////
sprintf(ach,"Num_pasadas2=%f", Num_pasadas2);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
Num_pasadasT=Num_pasadas2-Num_pasadas1;
////////////////////////////////////
sprintf(ach,"Num_pasadasT=%f", Num_pasadasT);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
if((Num_pasadasT>0) & (Num_pasadasT<1))
{
Num_pasadas=Num_pasadas1+1; //se incrementa una pasada
pasada_extra=1; //bandera ON para el ultimo desbaste
}
else
{
Num_pasadas=Num_pasadas; //se deja el numero de pasada
pasada_extra=0;
}
////////////////////////////////////
sprintf(ach,"Num_pasadas=%f", Num_pasadas);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
////////////////////////////////////
sprintf(ach,"pasada_extra=%f", pasada_extra);
MessageBox(hwnd, ach,ach , MB_OK);
////////////////////////////////////
}

//teta_g=atan(l/z);

```

```

for(i=0;i<Num_pasadas;i++)
{
//////////////////////////////////////primera linea (desbaste eje x)
//////////////////////////////////////
    sprintf (ach,"Desvaste_transversa=%i", (i+1));
    MessageBox (hwnd, ach,ach , MB_OK);
    ////////////////////////////////////////

    hdc = GetDC (hwndTab);

    // x=0.0000006317*az*x;
        // z=0.0000006317*az*z;
        czMover =(0);

        if((i+1)==Num_pasadas)
        // 0 //para Z
            if(pasada_extra==1)
            {
                cxMover =(-(0.0000006317*az*Num_pasadasT))*ZoomP*1000; // -1mm //para X

                ////////////////////////////////////////
                sprintf (ach,"2009=%f", Num_pasadasT);
                MessageBox (hwnd, ach,ach , MB_OK);
                ////////////////////////////////////////
            }
            else{

                ////////////////////////////////////////
                sprintf (ach,"hhh=%f", Num_pasadasT);
                MessageBox (hwnd, ach,ach , MB_OK);
                ////////////////////////////////////////
                cxMover =(-(0.0000006317*az*prof))*ZoomP*1000; // -1mm //para X

            }
            else
            {
                cxMover =(-(0.0000006317*az*prof))*ZoomP*1000; // -1mm //para X
            }

        zfin = zini + czMover;

        xfin = xini + cxMover;
        hLapiz = CreatePen(PS_SOLID, 1, RGB(255, 0, 0));
        SelectObject(hdc, hLapiz);

        MoveToEx(hdc, (int)zini, (int)xini, &pt);
        LineTo(hdc,(int)zfin, (int)xfin);

        ReleaseDC (hwndTab, hdc);
        DeleteObject(hLapiz);

        //ir al programa del editor
        zini = zfin;
        xini = xfin;
        ////////////////////////////////////////
//////////////////////////////////////

//////////////////////////////////////segunda linea (desbaste eje z)
//////////////////////////////////////
    sprintf (ach,"Desvaste_longitudinal=%i", (i+1));
    MessageBox (hwnd, ach,ach , MB_OK);
    ////////////////////////////////////////

    if((i+1)==Num_pasadas)
    // 0 //para Z
        if(pasada_extra==1)
        {
            //z_prima=(Num_pasadasT)/(tan(teta_g));
            z_prima=0;

            ////////////////////////////////////////
            sprintf (ach,"pasada_fraccionaria=%f", Num_pasadasT);
            MessageBox (hwnd, ach,ach , MB_OK);
            ////////////////////////////////////////
        }
        else{

            ////////////////////////////////////////
            //z_prima=r-(sqrt((r*r)- (((r-(var1*(prof))))*(r-(var1*(prof)))) ));
            //z_prima=sqrt((r*r)-(var1*var1*prof*prof));

            ////////////////////////////////////////
            //if ((z_prima>1) || (z_prima==1))
            //{ z_prima=z_prima-prof;}

            var1=var1+1;

        }

}

hdc = GetDC (hwndTab);
// x=0.0000006317*az*x;

```



```
////
```

```
//ir al programa del editor
```

```
////////////////////////////////////  
////////////////////////////////////hugo4_final
```

```
if (num==0) {  
    //comando G00  
    pos=vr; //guardo la pocion del vector de texto  
    c='a'; //inicializacion de variable  
    while(c!=0x0a) //salto de linea  
    {  
        for(i=0;i<20;i++)  
        {  
            xx[i]=0;  
            yy[i]=0;  
            zz[i]=0;  
            ff[i]=0;  
        }  
    }  
}
```

```
//borrando el vector  
c=a[pos];  
if(c=='X')  
{  
    co=0;  
    posx=pos;  
    dd='a'; //inicializacion de variable  
    while ((dd!=0x20) & (dd!=0x0a)) //espacio  
    {  
        dd=a[posx];  
        posx=posx+1;  
        if(dd!=0x0a)  
        {  
            if(dd!=0x20)  
            {  
                xx[co]=a[posx];  
                co=co+1;  
            }  
        }  
    }  
    pos=posx-1;//u  
    vr=pos;  
    x=atof(xx);  
    x=ZoomP*1000*x;  
}
```

```
if(c=='Y')  
{
```

```
co=0;  
posy=pos;  
e='a'; //inicializacion de variable  
while((e!=0x20) & (e!=0x0a))  
  
e=a[posy];  
posy=posy+1;
```

```
{
```

```

if(e!=0x0a)
{
if(e!=0x20)
{
yy[co]=a[posy];
co=co+1;
}
}

}

pos=posy-1;
vr=pos;
y=atof(yy);

}
if(c=='Z')
{

co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))

{

f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
zz[co]=a[posz];
co=co+1;
}
}
}

pos=posz-1;
vr=pos;
z=atof(zz);
z=ZoomP*1000*z;

}

if(c=='F')
{

co=0;
posf=pos;
//inicializacion de variable
g='a';
while(g!=0x0a)
{

g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
ff[co]=a[posf];
co=co+1;
}
}
}
}

```

```

    }
}

pos=posf-1;
vr=pos;

f=atof(ff);

}

else if ((c!='X') &
{
}

}

(c!='Y') & (c!='Z') & (c!='F'))

pos=pos+1;

////////////////////////////////////
hdc = GetDC (hwndTab) ;
x=0.0000006317*az*x;

z=0.0000006317*az*z;

czMover =z; //para Z
cxMover =x; //para X

zfin = zini + czMover ;

xfin = xini + cxMover ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255,0,0));
SelectObject(hdc, hLapiz);

MoveToEx(hdc, (int)zini, (int)xini, &pt);
LineTo(hdc,(int)zfin, (int)xfin);

ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

////////////////////////////////////Condiciones para el E.T
if ((zfin > (realz*az) ) || (zfin < (realz*az+0.0000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
b=0x00;
Errores(hwnd, hwndTab);
}

if ((xfin > ( 0.2025*ax+realx*az)) || ( xfin < (( 0.2025*ax+realx*az)+0.0000006317*az*(Zoom_x))))
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
b=0x00;
Errores(hwnd, hwndTab);
}

////////////////////////////////////
////////////////////////////////////
//ir al programa del editor
zini = zfin;
xini = xfin;
////////////////////////////////////

if (num==1) } //comando G01
{
pos=vr; //guardo la pocion del vector de texto

c='a'; //inicializacion de variable
while(c!=0x0a) //salto de linea
{
for(i=0;i<20;i++)
{
xx[i]=0;
yy[i]=0;
zz[i]=0;
ff[i]=0;
}

//borrando el vector
c=a[pos];
if(c=='X')
{
co=0;
posx=pos;
dd='a'; //inicializacion de variable
while ((dd!=0x20) & (dd!=0x0a)) //espacio
{
dd=a[posx];
}
}
}
}

```

```

        posx=posx+1;
        if(dd!=0x0a)
        {
            if(dd!=0x20)
            {
                xx[co]=a[posx];
                co=co+1;
            }
        }
    }
    pos=posx-1;//u
    vr=pos;
    x=atof(xx);
    x=ZoomP*1000*x;

}

                                                                    if(c=='Y')
                                                                    {

co=0;
posy=pos;
e='a'; //inicializacion de variable
while((e!=0x20) & (e!=0x0a))
                                                                    {

e=a[posy];
posy=posy+1;
if(e!=0x0a)
{
    if(e!=0x20)
    {
        yy[co]=a[posy];
        co=co+1;
    }
}

                                                                    }

pos=posy-1;
vr=pos;
                                                                    y=atof(yy);
                                                                    }
                                                                    if(c=='Z')
                                                                    {

co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))
                                                                    {

f1=a[posz];
posz=posz+1;
if(f1!=0x0a)

```

```

{
if(ff1=0x20)
{
zz[co]=a[posz];
co=co+1;
}
}

pos=posz-1;
vr=pos;

z=atof(zz);
z=ZoomP*1000*z;

if(c=='F')

co=0;
posf=pos;

//inicializacion de variable

g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
ff[co]=a[posf];
co=co+1;
}

pos=posf-1;
vr=pos;

f=atof(ff);

(c=='Y') & (c=='Z') & (c=='F')

pos=pos+1;

}

else if ((c=='X') &
{
}

}

////////////////////////////////////
hdc = GetDC (hwndTab) ;
x=0.0000006317*az*x;

z=0.0000006317*az*z;

czMover =z; //para Z
cxMover =x; //para X

zfin = zini + czMover ;
xfin = xini + cxMover ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
SelectObject(hdc, hLapiz);

MoveToEx(hdc, (int)zini, (int)xini, &pt);
LineTo(hdc,(int)zfin, (int)xfin);

ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

//////////////////////////////////////Condiciones para el E.T
if ((zfin > (realz*az) ) || (zfin < (realz*az+0.0000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
b=0x00;
}

```

```

Erroses(hwnd, hwndTab);
}
if ((xfin > ( 0.2025*ax+realx*az)) || ( xfin < (( 0.2025*ax+realx*az)+0.0000006317*az*(Zoom_x))))
{
    OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
    b=0x00;
    Erroses(hwnd, hwndTab);
}
////////////////////////////////////
//ir al programa del editor
zini = zfin;
xini = xfin;
////////////////////////////////////

if (num==2) } //comando G02
{
    pos=vr; //guardo la pocicion del vector de texto
    c='a'; //inicializacion de variable
    while(cl=0x0a) //salto de linea
    {
        for(i=0;i<20;i++)
        {
            xx[i]=0;
            yy[i]=0;
            zz[i]=0;
            ff[i]=0;
        }
    }

//borrando el vector
c=a[pos];
if(c=='A')
{
    co=0;
    posx=pos;
    dd='a'; //inicializacion de variable
    while ((dd!=0x20) & (dd!=0x0a)) //espacio
    {
        dd=a[posx];
        posx=posx+1;
        if(dd!=0x0a)
        {
            if(dd!=0x20)
            {
                AA[co]=a[posx];
                co=co+1;
            }
        }
    }
    pos=posx-1;//u
    vr=pos;
    A=atof(AA);
}

if(c=='B')
{
    co=0;
    posy=pos;
    e='a'; //inicializacion de variable
    while((e!=0x20) & (e!=0x0a))
    {
        e=a[posy];
        posy=posy+1;
    }
}

```

```

if(e!=0x0a)
{
if(e!=0x20)
{
BB[co]=a[posy];
co=co+1;
}
}

}

pos=posy-1;
vr=pos;

B=atof(BB);

}
if(c=='R')
{

co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))

{

f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
RR[co]=a[posz];
co=co+1;
}
}

}

}

pos=posz-1;
vr=pos;

radio=atof(RR);
radio=ZoomP*1000*radio//DUDA 1000

}

if(c=='F')
{

co=0;
posf=pos;

//inicializacion de variable

g='a';
while(g!=0x0a)
{

g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
FF[co]=a[posf];
co=co+1;
}
}
}
}

```

```

    }
}

pos=posf-1;
vr=pos;

f=atof(F);

} else if ((c!=A) &
{
}

(c!=B) & (c!=R) & (c!=F))
pos=pos+1;

}

////////////////////////////////////
hdc = GetDC (hwndTab) ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
SelectObject(hdc, hLapiz);

alfa=A;// ángulo de inicio del arco
beta=B;// ángulo de avance del arco
r=radio;// radio del círculo

z1=zini;// coordenada z del centro del círculo
x1=xini;// coordenada x del centro del círculo

z2=z1+(0.0000006317*az*r)*cos((alfa*pi)/180);//cuadrante 2
x2=x1-(0.0000006317*az*r)*sin((alfa*pi)/180);

z3=z1+(0.0000006317*az*r)*cos((alfa-beta)*pi/180);
x3=x1-(0.0000006317*az*r)*sin((alfa-beta)*pi/180);

////////////////////////////////////Traslacion de ejes
//para P
z2_=z1;
x2_=x1;
//para C
z1_=z1-(0.0000006317*az*r)*cos((alfa*pi)/180);//cuadrante 2
x1_=x1+(0.0000006317*az*r)*sin((alfa*pi)/180);
//para Q
z3_=z1_+(0.0000006317*az*r)*cos((alfa-beta)*pi/180);
x3_=x1_-(0.0000006317*az*r)*sin((alfa-beta)*pi/180);
////////////////////////////////////

MoveToEx(hdc,(int)z2_,(int)x2_,&pt);//cambia el handle de C //inter G00
//LineTo(hdc,(int)z2,(int)x2);//cambia el handle a P

r=sqrt(pow((x3-x1),2)+pow((z3-z1),2));//distancia de P a C (radio)

AngleArc( hdc, // manipulador a contexto de dispositivo
(int)z1_ // coordenada z del centro de la circunferencia
(int)x1_ // coordenada x del centro de la circunferencia
(int)r, // radio del círculo
(float)alfa, // ángulo de inicio del arco
(float)-beta// ángulo de avance del arco
);

ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

////////////////////////////////////
/*
radio=radio/(ZoomP*1000);

sprintf(ach,"radio=%f",radio);
MessageBox (hwndEdit, ach,ach, MB_OK) ;

sprintf(ach,"alfa=%f",alfa);
MessageBox (hwndEdit, ach,ach, MB_OK) ;

sprintf(ach,"beta=%f",beta);
MessageBox (hwndEdit, ach,ach, MB_OK) ;
*/

////////////////////////////////////
/*
////////////////////////////////////Condiciones para el E.T Punto C
if ((z1_ > (realz*az) ) || (z1_ < (realz*az+0.0000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
b=0x00;
Errores(hwnd, hwndTab);
}

if ((x1_ > ( 0.2025*ax+realx*az) ) || ( x1_ < (( 0.2025*ax+realx*az)+0.0000006317*az*(Zoom_x))))
{
}
}

```

```

OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
b=0x00;
Errores(hwnd, hwndTab);
}
////////////////////////////////////
////////////////////////////////////Condiciones para el E.T Punto P
if ((z2_ > (realz*az) ) || (z2_ < (realz*az+0.000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
b=0x00;
Errores(hwnd, hwndTab);
}
if ((x2_ > ( 0.2025*ax+realx*az) ) || ( x2_ < (( 0.2025*ax+realx*az)+0.000006317*az*(Zoom_x))))
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
b=0x00;
Errores(hwnd, hwndTab);
}
}
////////////////////////////////////
////////////////////////////////////Condiciones para el E.T Punto Q
if ((z3_ > (realz*az) ) || (z3_ < (realz*az+0.000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
b=0x00;
Errores(hwnd, hwndTab);
}
if ((x3_ > ( 0.2025*ax+realx*az) ) || ( x3_ < (( 0.2025*ax+realx*az)+0.000006317*az*(Zoom_x))))
{
OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
b=0x00;
Errores(hwnd, hwndTab);
}
}
*/
////////////////////////////////////

//ir al programa del editor
zini=z3_;// coordenada z del punto final del arco
xini=x3_;// coordenada x del punto final del arco
////////////////////////////////////

if (num==3) //comando G03
{
pos=vr; //guardo la pocision del vector de texto
c='a'; //inicializacion de variable
while(c!=0x0a) //salto de linea
{
for(i=0;i<20;i++)
{
xx[i]=0;
yy[i]=0;
zz[i]=0;
ff[i]=0;
}
}

//borrando el vector
c=a[pos];
if(c=='A')
{
co=0;
posx=pos;
dd='a'; //inicializacion de variable
while ((dd!=0x20) & (dd!=0x0a)) //espacio
{
dd=a[posx];
posx=posx+1;
if(dd!=0x0a)
{
if(dd!=0x20)
{
AA[co]=a[posx];
co=co+1;
}
}
}
}
}

```

```

pos=posx-1;//u
vr=pos;
    A=atof(AA);
}

                                                                    if(c=='B')
                                                                    {

co=0;
posy=pos;
e='a'; //inicializacion de variable
while((e!=0x20) & (e!=0x0a))
                                                                    {

e=a[posy];
posy=posy+1;
if(e!=0x0a)
{
if(e!=0x20)
{
BB[co]=a[posy];
co=co+1;
}
}
}

                                                                    }

pos=posy-1;
vr=pos;
    B=atof(BB);

                                                                    }
                                                                    if(c=='R')
                                                                    {

co=0;
posz=pos;
f1='a'; //inicializacion de variable
while((f1!=0x20) & (f1!=0x0a))
                                                                    {

f1=a[posz];
posz=posz+1;
if(f1!=0x0a)
{
if(f1!=0x20)
{
RR[co]=a[posz];
co=co+1;
}
}
}

                                                                    }

pos=posz-1;
vr=pos;
    radio=atof(RR);
    radio=ZoomP*1000*radio;//DUDA 1000
                                                                    }

```

```

if(c=='F')
{
co=0;
posf=pos;
//inicializacion de variable
g=a[posf];
posf=posf+1;
if(g!=0x0a)
{
FF[co]=a[posf];
co=co+1;
}
}
pos=posf-1;
vr=pos;
f=atof(FF);
}
else if ((c=='A') &
{
pos=pos+1;
}
}
}

////////////////////////////////////
hdc = GetDC (hwndTab) ;
hLapiz = CreatePen(PS_SOLID , 1, RGB(255, 0, 0));
SelectObject(hdc, hLapiz);

alfa=A; // ángulo de inicio del arco //180
beta=B; // ángulo de avance del arco //90
r=radio; // radio del círculo
z1=zini; // coordenada z del centro del círculo
x1=xini; // coordenada x del centro del círculo

z2=z1+(0.0000006317*az*r)*cos((alfa*pi)/180); // punto P
x2=x1-(0.0000006317*az*r)*sin((alfa*pi)/180);

z3=z1+(0.0000006317*az*r)*cos(((360-alfa-beta)*pi)/180); //punto Q
x3=x1-(0.0000006317*az*r)*sin(((360-alfa-beta)*pi)/180);

//////////////////////////////////////Traslacion de ejes
//para P
z2_ =z1;
x2_ =x1;
//para C
z1_ =z1-(0.0000006317*az*r)*cos((alfa*pi)/180);
x1_ =x1+(0.0000006317*az*r)*sin((alfa*pi)/180);
//para Q
z3_ =z1_+(0.0000006317*az*r)*cos(((360-alfa-beta)*pi)/180);
x3_ =x1_+(0.0000006317*az*r)*sin(((360-alfa-beta)*pi)/180);
//////////////////////////////////////

MoveToEx(hdc,(int)z2_,(int)x2_,&pt); //cambia el handle de C //inter G00
r=sqrt(pow((x2-x1),2)+pow((z2-z1),2)); //distancia de P a C (radio)

AngleArc( hdc, // manipulador a contexto de dispositivo
(int)z1_ , // coordenada z del centro de la circunferencia
(int)x1_ , // coordenada x del centro de la circunferencia
(int)r, // radio del círculo
(float)alfa, // ángulo de inicio del arco
(float)beta // ángulo de avance del arco
);

ReleaseDC (hwndTab, hdc) ;
DeleteObject(hLapiz);

////////////////////////////////////
//*/
radio=radio/(ZoomP*1000);

sprintf(ach,"radio=%f",radio);
MessageBox (hwndEdit, ach,ach, MB_OK) ;

```

```

        sprintf(ach,"alfa=%f",alfa);
        MessageBox (hwndEdit, ach,ach, MB_OK) ;

        sprintf(ach,"beta=%f",beta);
        MessageBox (hwndEdit, ach,ach, MB_OK) ;
    */

    //////////////////////////////////////
    /*
    //////////////////////////////////////Condiciones para el E.T Punto C
    if ((z1_ > (realz*az) ) || (z1_ < (realz*az+0.000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
    {
        OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
        b=0x00;
        Errores(hwnd, hwndTab);
    }
    if ((x1_ > ( 0.2025*ax+realx*az) ) || ( x1_ < (( 0.2025*ax+realx*az)+0.000006317*az*(Zoom_x))))
    {
        OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
        b=0x00;
        Errores(hwnd, hwndTab);
    }
    //////////////////////////////////////
    //////////////////////////////////////Condiciones para el E.T Punto P
    if ((z2_ > (realz*az) ) || (z2_ < (realz*az+0.000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
    {
        OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
        b=0x00;
        Errores(hwnd, hwndTab);
    }
    if ((x2_ > ( 0.2025*ax+realx*az) ) || ( x2_ < (( 0.2025*ax+realx*az)+0.000006317*az*(Zoom_x))))
    {
        OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
        b=0x00;
        Errores(hwnd, hwndTab);
    }
    //////////////////////////////////////
    //////////////////////////////////////Condiciones para el E.T Punto Q
    if ((z3_ > (realz*az) ) || (z3_ < (realz*az+0.000006317*az*(Zoom_z))))//condiciones del espacio de trabajo
    {
        OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje Z [ 0 a -520 ]");
        b=0x00;
        Errores(hwnd, hwndTab);
    }
    if ((x3_ > ( 0.2025*ax+realx*az) ) || ( x3_ < (( 0.2025*ax+realx*az)+0.000006317*az*(Zoom_x))))
    {
        OkMessage1 (hwnd, "Parametros fuera del \n espacio de trabajo \n \n Eje X [ 0 a -82 ]");
        b=0x00;
        Errores(hwnd, hwndTab);
    }
    */
    //////////////////////////////////////
    //ir al programa del editor
    zini=z3_;// coordenada finales del circulo
    xini=x3_;// coordenada finales del circulo
    //////////////////////////////////////
    }
    num=-1; //inicializacion de variable
    break;

    //////////////////////////////////////ini_Ramiro
    case 'T': //comandos T

    palabra
    SendMessage (hwndEdit, EM_SETSEL, vr, vr+3) ;//selecciona la
    SendMessage (hwndEdit, EM_SCROLLCARET, 0, 0) ;

    //Ramiro
    /* -- Signal Direction */
    //#define SIGNAL_IN 2
    //#define SIGNAL_OUT 4

    //
    cbDConfigPort (BoardNum, FIRSTPORTA,DIGITALIN);
    cbDConfigPort (BoardNum, FIRSTPORTB,DIGITALOUT);
    cbDConfigBit(BoardNum, FIRSTPORTA, 7, SIGNAL_IN);
    cbDOut(BoardNum, FIRSTPORTB,0);
    DATOS=fopen("dato.txt","w");

    /*
    cbDOut(BoardNum, FIRSTPORTB,0);
    sleep(100);
    cbDOut(BoardNum, FIRSTPORTB,1);
    sleep(100);
    cbDOut(BoardNum, FIRSTPORTB,3);
    sleep(100);
    cbDOut(BoardNum, FIRSTPORTB,0);

    */
    //Ramiro

```

```

ss[0]=a[vr+1];
ss[1]=a[vr+2];

num=atof(ss);
if (num==1)
{
    &DataValue);

    la posicion a la que requiere mover torreta!");

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,6);
    }while(DataValue!=num+32+128);
    sleep(12);

    //*****//=====>

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,14);

    }while(DataValue!=num+16+64);

    cbDOut(BoardNum, FIRSTPORTB,12);
        sleep(30);
    //*****//=====>160 ms

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,13);
    }while(DataValue!=num+16+64+128);

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,13);
    }while(DataValue!=num+16+64);

    cbDOut(BoardNum, FIRSTPORTB,8);
    sleep(70);
    cbDOut(BoardNum, FIRSTPORTB,0);

    sleep(1);
    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

    //////////LECTURA DE LA POSICION ACTUAL////////
    UDStat=cbDIn(BoardNum,FIRSTPORTA,

    Comparador=DataValue-16;
    fprintf(DATOS,"%i",Comparador);
    fclose(DATOS);

    if(Comparador==num)
    {
        OkMessage1 (hwnd, "Revise
    }
    else
    {
        if(Comparador>=5)
            {
                do
                {

                //
                //*****//=====>

                do
                {

                do
                {

                do
                {

```

```

}while(DataValue!=num+16);

//fprintf(DATOS,"%i",DataValue);
//fclose(DATOS);
//////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
OkMessage1 (hwnd, "Posicion 1. Listo para cortar!");

else
}
{
do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

do
{

sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

fprintf(DATOS,"%i",DataValue);
fclose(DATOS);
//////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
OkMessage1 (hwnd, "Posicion 1. Listo para cortar!");

}
}
}

```

```

if (num==2)
{
    &DataValue);

    la posicion a la que requiere mover torreta!);

    if((Comparador>=6)||((Comparador==1))

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,6);
    }while(DataValue!=num+32+128);

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,14);

    }while(DataValue!=num+16+64);

    cbDOut(BoardNum, FIRSTPORTB,12);
    sleep(30);

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,13);
    }while(DataValue!=num+16+64+128);

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,13);
    }while(DataValue!=num+16+64);

    cbDOut(BoardNum, FIRSTPORTB,8);
    sleep(70);
    cbDOut(BoardNum, FIRSTPORTB,0);

    sleep(1);
    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    }while(DataValue!=num+16);

    fprintf(DATOS,"%i",DataValue);
    fclose(DATOS);
    ///////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
    OkMessage1 (hwnd, "Posicion 2. Listo para cortar!");
}
}

/////////LECTURA DE LA POSICION ACTUAL/////////
UDStat=cbDIn(BoardNum,FIRSTPORTA,
    Comparador=DataValue-16;
    fprintf(DATOS,"%i",Comparador);
    fclose(DATOS);

if(Comparador==num)
{
    OkMessage1 (hwnd, "Revise
}
else
{
do
{
do
{
do
{
do
{
do
{

```

TORRETA EN SENTIDO CONTRARIO DE MANECILLAS DEL RELIJ//////////

```
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);
```

```
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
```

```
}while(DataValue!=num+16+64);
```

```
cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);
```

```
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64+128);
```

```
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64);
```

```
cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);
```

```
sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);
```

```
fprintf(DATOS,"%i",DataValue);
fclose(DATOS);
```

```
//////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
```

```
OkMessage1 (hwnd, "Posicion 2. Listo para cortar!");
```

&DataValue);

```
if (num==3)
```

```
}
{
```

```
//////////LECTURA DE LA POSICION ACTUAL//////////
```

```
UDStat=cbDIn(BoardNum,FIRSTPORTA,
```

```
Comparador=DataValue-16;
fprintf(DATOS,"%i",Comparador);
fclose(DATOS);
```

```
else
}
{
//////////GIRO DE
do
{
```

```
}
```

```
}
```

la posicion a la que requiere mover torreta!!");

if((Comparador>=7)||(Comparador<=2))

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,6);
}while(DataValue!=num+32+128);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
}while(DataValue!=num+16+64+128);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

sleep(1);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

fprintf(DATOS,"%i",DataValue);
fclose(DATOS);

/////////ENVIO DE MENSAJE DE "READY TO CUT"/////////

OkMessage1 (hwnd, "Posicion 3. Listo para cortar!!");

TORRETA EN SENTIDO CONTRARIO DE MANECILLAS DEL RELIJ/////////

```
if(Comparador==num)
{
    OkMessage1 (hwnd, "Revise
}
else
{
```

```
{
do
{
```

```
else
{
/////////GIRO DE
```

```
do
{
```

```

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

do
{

sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

fprintf(DATOS,"%i",DataValue);
fclose(DATOS);
/////////ENVIO DE MENSAJE DE "READY TO CUT"/////////
OkMessage1 (hwnd, "Posicion 3. Listo para cortar!");

}

}

}

if (num==4)
{
/////////LECTURA DE LA POSICION ACTUAL/////////
UDStat=cbDIn(BoardNum,FIRSTPORTA,
Comparador=DataValue-16;
fprintf(DATOS,"%i",Comparador);
fclose(DATOS);

if(Comparador==num)
{
OkMessage1 (hwnd, "Revise

else
{

```

```

if((Comparador==8)||(Comparador<=3))
{
do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,6);
}while(DataValue!=num+32+128);

do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
}while(DataValue!=num+16+64+128);

do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

do
{
sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

//fprintf(DATOS,"%i",DataValue);
//fclose(DATOS);
//////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
OkMessage1 (hwnd, "Posicion 4. Listo para cortar!");
}
else
{
//////////GIRO DE
do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);

if(Comparador>5)

```


TORRETA EN SENTIDO CONTRARIO DE MANECILLAS DEL RELIJ//////////

{
//////////GIRO DE

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

do
{

sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

fprintf(DATOS,"%i",DataValue);
fclose(DATOS);

//////////ENVIO DE MENSAJE DE "READY TO CUT"//////////

OkMessage1 (hwnd, "Posicion 5. Listo para cortar!");

}

}

}

if (num==6)
{

&DataValue);

//////////LECTURA DE LA POSICION ACTUAL//////////

UDStat=cbDIn(BoardNum,FIRSTPORTA,

Comparador=DataValue-16;
fprintf(DATOS,"%i",Comparador);
fclose(DATOS);

la posicion a la que requiere mover torreta!!");

if((Comparador<=5)&&(Comparador>=2))

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

cbDOut(BoardNum, FIRSTPORTB,6);

}while(DataValue!=num+32+128);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

cbDOut(BoardNum, FIRSTPORTB,14);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);

sleep(30);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

cbDOut(BoardNum, FIRSTPORTB,13);

}while(DataValue!=num+16+64+128);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

cbDOut(BoardNum, FIRSTPORTB,13);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);

sleep(70);

cbDOut(BoardNum, FIRSTPORTB,0);

sleep(1);

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

}while(DataValue!=num+16);

//fprintf(DATOS,"%i",DataValue);

//fclose(DATOS);

/////////ENVIO DE MENSAJE DE "READY TO CUT"/////////

OkMessage1 (hwnd, "Posicion 6. Listo para cortar!");

TORRETA EN SENTIDO CONTRARIO DE MANECILLAS DEL RELIJI/////////

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);

if(Comparador==num)

{

OkMessage1 (hwnd, "Revise

}

else

{

{

do
{

do
{

do
{

do
{

do
{

else

}

{
/////////GIRO DE

do
{

```

cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64+128);

do
{

UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

do
{

sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

fprintf(DATOS,"%i",DataValue);
fclose(DATOS);
/////////ENVIO DE MENSAJE DE "READY TO CUT"/////////
OkMessage1 (hwnd, "Posicion 6. Listo para cortar!");

}

}

if (num==7)
{
/////////LECTURA DE LA POSICION ACTUAL/////////
UDStat=cbDIn(BoardNum,FIRSTPORTA,
Comparador=DataValue-16;
fprintf(DATOS,"%i",Comparador);
fclose(DATOS);

if(Comparador==num)
{
OkMessage1 (hwnd, "Revise
la posicion a la que requiere mover torreta!");
}
else

```

```

{
if((Comparador<=6)&&(Comparador>=3))
{
do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,6);
}while(DataValue!=num+32+128);

do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,14);

}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,12);
sleep(30);

do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
}while(DataValue!=num+16+64+128);

do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,13);
}while(DataValue!=num+16+64);

cbDOut(BoardNum, FIRSTPORTB,8);
sleep(70);
cbDOut(BoardNum, FIRSTPORTB,0);

do
{
sleep(1);
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

fprintf(DATOS,"%i",DataValue);
fclose(DATOS);
//////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
OkMessage1 (hwnd, "Posicion 7. Listo para cortar!");

}
else
{
//////////GIRO DE
do
{
UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
cbDOut(BoardNum, FIRSTPORTB,5);
}while(DataValue!=num+32+128);

do

```

TORRETA EN SENTIDO CONTRARIO DE MANECILLAS DEL RELIJI//////////


```

}while(DataValue!=num+16+64);

    cbDOut(BoardNum, FIRSTPORTB,12);
        sleep(30);

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64+128);

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,14);
}while(DataValue!=num+16+64);

    cbDOut(BoardNum, FIRSTPORTB,8);
    sleep(70);
    cbDOut(BoardNum, FIRSTPORTB,0);

    sleep(1);
    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
}while(DataValue!=num+16);

    fprintf(DATOS,"%i",DataValue);
    fclose(DATOS);
    ///////////ENVIO DE MENSAJE DE "READY TO CUT"//////////
    OkMessage1 (hwnd, "Posicion 8. Listo para cortar!");

}

}

if (num==9)
{
    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    fprintf(DATOS,"%i",DataValue);
    fclose(DATOS);
    cbDOut(BoardNum, FIRSTPORTB,0);

    OkMessage1 (hwnd, "Mueve la torreta a la posición 9");

}

    UDStat=cbDIn(BoardNum,FIRSTPORTA, &DataValue);
    cbDOut(BoardNum, FIRSTPORTB,1);
    sleep(300);
    cbDOut(BoardNum, FIRSTPORTB,2);
    sleep(300);
    cbDOut(BoardNum, FIRSTPORTB,4);
    sleep(300);
    cbDOut(BoardNum, FIRSTPORTB,8);
    sleep(300);
    cbDOut(BoardNum, FIRSTPORTB,8);
}

```

```

sleep(500);
cbDOut(BoardNum, FIRSTPORTB,0);
sleep(300);
}while(DataValue!=7+16);*/
    OkMessage1 (hwnd, "Mueve la torreta a la posición 9");
                                cbDOut(BoardNum, FIRSTPORTB,0);
                                }

////////////////////////////////////fin_Ramiro
                                }
                                break;
                                }
                                }while(b!=0x00);
return TRUE ;
}

void sleep(unsigned int mseconds)
{
    clock_t goal=mseconds+clock();
    while(goal>clock());
}

```