



INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación e Innovación Tecnológica

“Desarrollo de un sistema de manufactura asistida por computadora para moldes”

Tesis que para obtener el grado de
Maestría en Tecnología Avanzada

Presenta

Ing. Eduardo Alejandro Ramírez Yáñez

Directores de tesis

M. en C. Agustín Cruz Contreras

Dr. Domitilo Libreros



México D.F. Diciembre 2011



SIP-14-BIS

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 13:00 horas del día 16 del mes de Noviembre del 2011 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CIITEC para examinar la tesis titulada:

"Desarrollo de un sistema de manufactura asistida por computadora para moldes"

Presentada por el alumno:

Ramírez

Yáñez

Eduardo Alejandro

Apellido paterno

Apellido materno

Nombre(s)

Con registro:

B	0	9	1	5	9	2
---	---	---	---	---	---	---

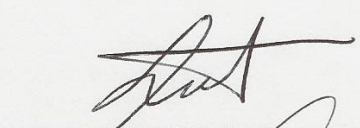
aspirante de:

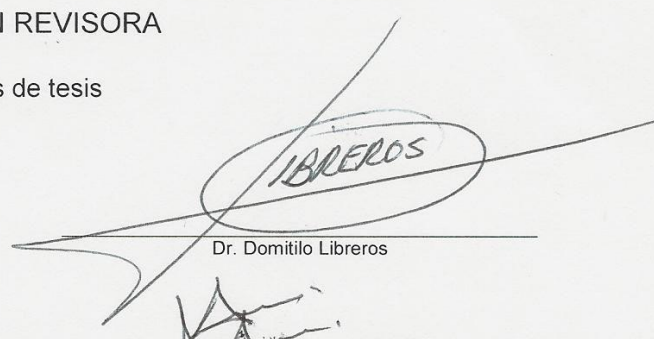
Maestría en Tecnología Avanzada

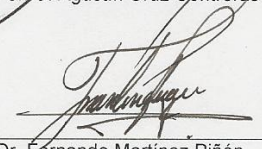
Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

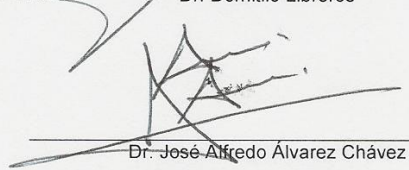
LA COMISIÓN REVISORA

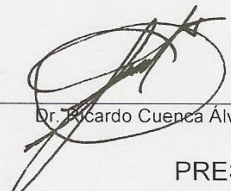
Directores de tesis


M. en C. Agustín Cruz Contreras

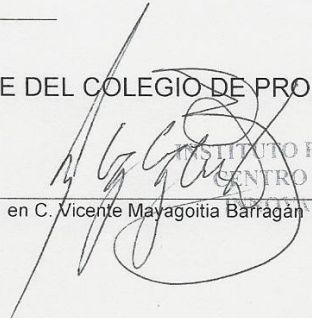

Dr. Domitilo Libreros


Dr. Fernando Martínez Piñón


Dr. José Alfredo Álvarez Chávez


Dr. Ricardo Cuenca Álvarez

PRESIDENTE DEL COLEGIO DE PROFESORES


M. en C. Vicente Mayagoitia Barragán



SEP

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN E
INNOVACIÓN TECNOLÓGICA
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 20 del mes de noviembre del año 2011, el (la) que suscribe Eduardo Alejandro Ramírez Yáñez alumno (a) del Programa de Maestría en Tecnología Avanzada, con número de registro B091592, adscrito al Centro de Investigación e Innovación Tecnológica, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de M. en C. Agustín Cruz Conteras y cede los derechos del trabajo intitulado "Desarrollo de un sistema de manufactura asistida por computadora para moldes", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección lalojandro50@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.


Eduardo Alejandro Ramírez Yáñez
Nombre y firma

Dedico este trabajo a mis padres Mario y Rosario por el apoyo incondicional brindado durante todo este tiempo, por las enseñanzas que día a día muestran con el ejemplo, y sobre todo por demostrarme que no hay límites ni obstáculos que no se puedan superar.

A mis hermanos Ricardo, Mario, Daniel y Diego por la paciencia que me tuvieron y por el apoyo recibido.

A mis abuelos por las experiencias y vividas transmitidas, así también por el ánimo que me siempre me han brindado.

A mi novia Gaby por la paciencia, el apoyo y comprensión que siempre he recibido y por estar conmigo en las buenas y en las malas.

A mi familia y amigos que de alguna u otra manera me han apoyado.

Alejandro

Agradecimientos

Agradezco al Instituto Politécnico Nacional por brindarme la oportunidad de formarme como técnico, ingeniero y maestro.

Al Centro de Investigación e Innovación Tecnológica y al Consejo Nacional de Ciencia y Tecnología por su apoyo y respaldo necesario para el desarrollo de éste trabajo.

Agradezco al maestro Agustín Cruz Contreras por su asesoría y apoyo durante la maestría, por la paciencia y dedicación al dirigir este trabajo de investigación, por sus consejos y observaciones.

Agradezco a los profesores que me instruyeron en sus respectivas asignaturas, por sus enseñanzas académicas y personales.

Índice general

ÍNDICE GENERAL.....	III
ÍNDICE DE FIGURAS.....	VII
RESUMEN.....	XIII
ABSTRACT	XIV

INTRODUCCIÓN

INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA	2
PROPUESTA.....	3
JUSTIFICACIÓN.....	3
OBJETIVO GENERAL	3
OBJETIVOS PARTICULARES	3
METODOLOGÍA	3
APORTACIONES.....	4
ESTRUCTURA DE LA TESIS	4

CAPÍTULO 1: ANTECEDENTES TEÓRICOS

INTRODUCCIÓN	7
1.1 ASPECTOS GENERALES	7
1.1.1 MÁQUINA HERRAMIENTA.....	7
1.1.1.1 Torno.....	7
1.1.1.2 Fresadora.....	8
1.1.1.3 Proceso de Maquinado	9
1.1.1.3.1 Desbaste	9
1.1.1.3.2 Semi-acabado	9
1.1.1.3.3 Acabado	9
1.1.2 CONTROL NUMÉRICO COMPUTARIZADO CNC	9
1.1.2.1 Código G y M	10
1.1.2.2 Programación en el control numérico	12
1.1.2.2.1 Programación manual	12
1.1.2.2.2 Programación por medio de software	12
1.1.3 CENTRO DE MAQUINADO.....	12
1.1.3.1 Características	13
1.1.4 DIBUJO ASISTIDO POR COMPUTADORA.....	13
1.1.5 MANUFACTURA ASISTIDA POR COMPUTADORA	14
1.1.6 SISTEMA CAD/CAM.....	14
1.1.7 IMPRESORAS 3D	14
1.1.8 MOLDES.....	15
1.1.9 MODELOS PARA MOLDES DE FUNDICIÓN.....	16
1.2 ASPECTOS DEL MECANISMO.....	17
1.2.1 HUSILLO	17
1.2.1.1 Husillo embalado	17
1.2.2 MOTORES A PASOS	18

1.2.2.1	Motor de reluctancia variable.....	19
1.2.2.2	Motor de imán permanente	19
1.2.2.3	El motor a pasos híbrido	20
1.2.2.4	Bipolar.....	20
1.2.2.5	Unipolar	20
1.2.2.6	Control del motor a pasos	21
1.2.2.6.1	Paso completo	21
1.2.2.6.2	Paso completo de una bobina activa	22
1.2.2.6.3	Paso completo de dos bobinas activas	23
1.2.2.6.4	Medio paso.....	24
1.2.2.6.5	Velocidad de rotación.....	25
1.2.2.6.6	Control de las bobinas.....	25
1.2.2.7	Puente H.....	26
1.2.3	SENSORES.....	28
1.2.3.1	Sensor de final de carrera.....	28
1.2.3.2	Sensores de inicio o home.....	28
1.3	ASPECTOS DE LA INTERFAZ.....	30
1.3.1	PUERTO PARALELO.....	30
1.3.1.1	El registro de datos.....	32
1.3.1.2	El registro de estado	32
1.3.1.3	El registro de control.....	32
1.3.1.4	Características de Entrada/Salida.....	33
1.3.1.5	Lectura y escritura de datos.....	33
1.3.1.6	Enmascaramiento	34
1.3.2	ACOPLAMIENTO ÓPTICO.....	36
1.3.2.1	Acoplamiento óptico para control de dispositivos eléctricos.....	36
1.3.2.2	Triac	37
1.3.3	FUENTE DE ALIMENTACIÓN.....	37
1.3.3.1	Regulador de voltaje	38
1.4	ASPECTOS DE PROGRAMACIÓN.....	39
1.4.1	FORMATO DE ESTEREOLITOGRAFÍA (STL)	39
1.4.1.1	Formato STL en código ASCII	39
1.4.1.2	Ejemplo	40
1.4.2	IMAGEN DIGITAL	42
1.4.2.1	Vecindad	43
1.4.2.2	Tamaño de vecindad	44
1.4.2.3	Conectividad	44
1.4.2.4	Operadores morfológicos	45
1.4.2.5	Dilatación	45
1.4.2.6	Erosión	46
1.4.2.7	Apertura	47
1.4.2.8	Cierre.....	48
1.4.3	VOXEL	49
1.4.3.1	Conectividad	52

CAPÍTULO 2:DISEÑO E IMPLEMENTACIÓN DEL CAM

INTRODUCCIÓN	55
2.1 DISEÑO.....	55
2.1.1 CONSTRUCCIÓN DE LA SUPERFICIE DEL SÓLIDO	56
2.1.1.1 Extracción de vértices	56
2.1.1.2 Obtención de las aristas del triángulo	58
2.1.1.3 Cálculo de incrementos.....	63
2.1.1.4 Obtención de voxels del área del triángulo	67
2.1.1.5 Redundancias y huecos	72
2.1.1.6 Obtención de los voxels que integran la superficie total del sólido	73
2.1.1.7 Rebanado del sólido	74
2.1.2 OBTENCIÓN DE PERFIL	76
2.1.2.1 Obtención del contorno.....	76
2.1.2.2 Rellenado del interior del contorno.....	77
2.1.2.3 Rellenado de huecos.....	77
2.1.2.4 Identificación de perfil	78
2.1.2.5 Eliminación de protuberancias de un pixel	80
2.1.2.6 Compensación de la herramienta de corte.....	81
2.1.2.7 Seguimiento de perfil	83
2.2 IMPLEMENTACIÓN DEL SOFTWARE	86
2.2.1 CONSTRUCCIÓN DE SUPERFICIE DEL SÓLIDO.....	87
2.2.1.1 Rutina de extracción de vértices	88
2.2.1.2 Obtención de las aristas.....	91
2.2.1.2.1 Rutina de unión de dos puntos	92
2.2.1.2.2 Cálculo de incrementos.....	93
2.2.1.2.3 Cálculo de los voxels	95
2.2.1.2.4 Redondeo	95
2.2.1.3 Cálculo de voxels en la superficie del triángulo	98
2.2.1.3.1 Determinar la arista mayor y el vértice opuesto.....	99
2.2.1.4 Obtención de los voxels que integran la superficie total del sólido	100
2.2.2 OBTENCIÓN DE PERFIL	101
2.2.2.1 Obtención del contorno.....	102
2.2.2.2 Rellenado del interior del contorno.....	102
2.2.2.3 Rellenado de huecos.....	104
2.2.2.4 Eliminación de protuberancias de un pixel	106
2.2.2.5 Identificación del perfil.....	108
2.2.2.6 Compensación de la herramienta de corte.....	110
2.2.2.7 Seguimiento de perfil	110
2.2.3 MOVIMIENTO DE LOS MOTORES	113
2.2.3.1 Obtención de desplazamientos en los ejes Y y Z.....	113
2.2.4 INTERFAZ GRÁFICA	114
2.2.5 INTERFAZ DE USUARIO	120

CAPÍTULO 3: INTERFAZ Y MECANISMO

INTRODUCCIÓN	124
3.1 MECANISMO	124
3.1.1 MECANISMO DEL EJE Z.....	125
3.1.2 MECANISMO DEL EJE Y	126
3.1.3 MECANISMO DEL EJE X	127
3.1.4 SENSORES.....	131
3.1.4.1 Sensor de final de carrera.....	131
3.1.4.2 Sensores de inicio.....	133
3.1.5 FUENTE DE ALIMENTACIÓN.....	133
3.1.6 HERRAMIENTA DE CORTE.....	134
3.2 INTERFAZ.....	135
3.2.1 SOFTWARE.....	135
3.2.1.1 Movimientos para los ejes x, y, z.....	135
3.2.2 HARDWARE	137
3.2.2.1 Controlador de motor a pasos	140
3.2.2.2 Control de encendido y apagado de la fuente de alimentación	141

CAPÍTULO 4: APLICACIÓN

INTRODUCCIÓN	145
4.1 MOLDE	145
4.1.1 CAVIDAD	147
4.1.1.1 Obtención de la pieza en formato STL.....	147
4.1.1.2 Inicio del programa	150
4.1.1.3 Imágenes obtenidas del CAM	150
4.1.2 CORAZÓN DE MOLDE.....	156
4.1.2.1 Pieza en formato stl.....	156
4.1.2.2 Imágenes obtenidas del CAM	157
4.2 RELIEVE.....	162
4.3 EJEMPLOS.....	164

CONCLUSIONES Y TRABAJO A FUTURO

CONCLUSIONES	167
SUGERENCIAS PARA TRABAJO FUTURO	167
REFERENCIAS.....	169

Índice de figuras

Capítulo 1: Antecedentes teóricos

Figura 1-1. Torno paralelo convencional	8
Figura 1-2. Fresadora vertical.....	9
Figura 1-3. Molde con corte transversal de dos piezas para botella.....	15
Figura 1-4. a) Molde de inyección de plástico de 2 piezas, b) Molde de inyección de plástico de 3 piezas	16
Figura 1-5. Modelo de molde de fundición	16
Figura 1-6. Tornillo cuerda ACME	17
Figura 1-7. Husillo embalado	18
Figura 1-8. Visualización de balines	18
Figura 1-9. Rotor de un motor a pasos	20
Figura 1-10. Imagen de un estator.....	20
Figura 1-11. Motor bipolar de cuatro cables de salida	20
Figura 1-12. Motor unipolar de cinco cables.....	21
Figura 1-13. Motor unipolar de seis cables	21
Figura 1-14. Bobinas conectadas a dos fuentes de alimentación.	25
Figura 1-15. Diagrama del puente H	26
Figura 1-16. Un par de puentes H para el control de las bobinas del motor bipolar	26
Figura 1-17. Diagrama de conexión entre el L297 y el L298.....	27
Figura 1-18. Interruptor de una vía.....	28
Figura 1-19. Interruptor de dos vías	28
Figura 1-20. Sensor de posición de inicio	29
Figura 1-21. Especificación técnica del sensor	29
Figura 1-22. Imagen de distribución del puerto paralelo.....	30
Figura 1-23. Arreglo del dispositivo 6N137.....	36
Figura 1-24. Arreglo electrónico del MOC 3010	37
Figura 1-25. Símbolo del tiristor	37
Figura 1-26. Arreglo del TRIAC con tiristores	37
Figura 1-27. Símbolo del TRIAC.....	37
Figura 1-28. Fuente conmutada.....	38
Figura 1-29. Diagrama eléctrico del regulador de voltaje.....	38
Figura 1-30. Superficie de un cubo en formato STL	40
Figura 1-31. Imagen digital	43
Figura 1-32. Vecindad de píxeles	43
Figura 1-33. Vecindad de dos píxeles	44
Figura 1-34. Conectividad entre píxeles.....	44
Figura 1-35. Imágenes del EE sobre una imagen digital.....	45
Figura 1-36. Descripción del elemento estructurante.....	45
Figura 1-37. Proceso de dilatación en una imagen binaria.....	46
Figura 1-38. Proceso de erosión en una imagen binaria.....	47
Figura 1-39. Proceso de apertura.....	48
Figura 1-40. Proceso de cierre.....	49
Figura 1-41. Vecindad de voxels por caras	50
Figura 1-42. Vecindad de voxels por aristas	50

Figura 1-43. Vecindad de voxels por vértice.....	51
Figura 1-44. 26-Adyacente si comparten vértice, arista o cara	51
Figura 1-45. 18-Adyacente de un voxel	52
Figura 1-46. Conectividad entre voxels.....	52

Capítulo 2: Diseño e implementación del CAM

Figura 2-1. Diagrama del proceso para maquinar a partir del CAD.....	55
Figura 2-2. Procedimiento para la obtención de los voxels de la superficie del sólido	56
Figura 2-3. Ubicación del triángulo del bloque de descripción.....	57
Figura 2-4. Ubicación de los voxels de los vértices de los triángulos.....	58
Figura 2-5. Ubicación de los voxels que conforman la superficie del cubo.....	58
Figura 2-6. Proceso de obtención de los voxels de la superficie	58
Figura 2-7. Ubicación de dos puntos en un plano cartesiano	59
Figura 2-8. Ubicación del primer y segundo pixel calculado.....	61
Figura 2-9. Línea formada con pixeles	62
Figura 2-10. Ubicación espacial de los vértices extraídos.....	62
Figura 2-11. Ubicación espacial del voxel (V1+1)	64
Figura 2-12. Construcción de voxels de una arista.....	65
Figura 2-13. Construcción de las aristas del triángulo	67
Figura 2-14. Método radial para obtención de voxels del área	68
Figura 2-15. Imagen de los voxels a unirse por medio de una línea	69
Figura 2-16. Imagen de la línea que une el vértice opuesto con el voxel "a"	69
Figura 2-17. Muestra de los voxels calculados hasta el momento	70
Figura 2-18. Ubicación de los voxels del área del triángulo	72
Figura 2-19. Duplicidad y huecos en el cálculo de obtención de voxels.....	73
Figura 2-20. Pieza dibujada en CAD.....	73
Figura 2-21. Pieza dibujada en CAD en formato STL	73
Figura 2-22. Imágenes de correspondencia de rebanada.....	75
Figura 2-23. Imagen binaria con tamaño de pixel mayor.....	76
Figura 2-24. Procedimiento para el relleno del interior del contorno	77
Figura 2-25. Pixeles internos rellenos con secuelas de los huecos.....	77
Figura 2-26. Elemento estructurante para el llenado de huecos	78
Figura 2-27. Imagen binaria después de rellenar huecos.....	78
Figura 2-28. Contorno de la figura	78
Figura 2-29. Perfil del contorno	79
Figura 2-30. Proceso de marcar el contorno, con un barrido vertical hacia arriba.....	79
Figura 2-31. Proceso de marcar el contorno con un EE en barrido horizontal.....	80
Figura 2-32. Proceso de marcado del contorno terminado	80
Figura 2-33. Imagen con protuberancia vertical.....	81
Figura 2-34. Imagen después del proceso de cierre y protuberancia dilatada.....	81
Figura 2-35. Cortador vertical con punto de corte: a) con avance hacia la derecha b) y la izquierda	81
Figura 2-36. Imagen binaria con el punto de corte del cortador.....	82
Figura 2-37. Cortede la herramienta sin compensación	82
Figura 2-38. EE para la dilatación de imagen	82

Figura 2-39. Imagen binaria con compensación de la herramienta	83
Figura 2-40. Compensación de la herramienta de corte	83
Figura 2-41. Contorno marcado con compensación de herramienta	84
Figura 2-42. Dirección de búsqueda de perfil hacia la derecha (a) e izquierda (b) respectivamente	84
Figura 2-43. Recorrido del EE sobre el perfil hacia la derecha.....	84
Figura 2-44. Recorrido del EE sobre del perfil hacia la izquierda	85
Figura 2-45. Construcción de la superficie con voxels	87
Figura 2-46. Proceso de extracción de vértices	88
Figura 2-47. Proceso para obtención de voxels de la superficie de un triángulo	91
Figura 2-48. Proceso de la extracción de vértices	91
Figura 2-49. Ubicación espacial de los vértices del triángulo.....	92
Figura 2-50. Proceso de cálculo de voxels de una línea.....	92
Figura 2-51. Ubicación espacial del voxel (V1+1)	96
Figura 2-52. Construcción de voxels de una arista.....	97
Figura 2-53. Algoritmo del cálculo de coordenadas de los voxels siguientes.....	97
Figura 2-54. Construcción de las aristas del triángulo	98
Figura 2-55. Proceso para obtener el área de un triángulo	98
Figura 2-56. Proceso de determinar la arista mayor.....	99
Figura 2-57. Método radial para obtención de voxels del área	99
Figura 2-58. Ubicación de voxels del área del triángulo.....	100
Figura 2-59. Construcción virtual de la superficie	100
Figura 2-60. Ejemplo de rebanado de la superficie virtual.....	101
Figura 2-61. Proceso de la obtención del perfil.....	101
Figura 2-62. Imagen de contorno con huecos y doble espesor.....	102
Figura 2-63. Diagrama de la rutina para el rellenado del contorno	103
Figura 2-64. Imagen con el interior llenado y con secuelas de los huecos.....	104
Figura 2-65. Imagen binaria después de rellenar huecos.....	104
Figura 2-66. Proceso para la identificación del perfil	108
Figura 2-67. Imagen binaria con compensación de la herramienta.....	110
Figura 2-68. Recorrido del perfil hacia la derecha	111
Figura 2-69. Recorrido del perfil hacia la izquierda.....	111
Figura 2-70. Dirección de búsqueda del perfil	113
Figura 2-71. Desplazamiento simultáneo entre el EE sobre el perfil y de la herramienta de corte	114
Figura 2-72. Interfaz de desarrollador	115
Figura 2-73. Botones de movimiento de ejes	116
Figura 2-74. Botones de inicio de los ejes.....	116
Figura 2-75. Displays de la ubicación en cada eje de la herramienta.....	116
Figura 2-76. Botón de inicio de la ubicación local de la herramienta	117
Figura 2-77. Inicio de cálculo de voxels de la superficie	117
Figura 2-78. Maquinado hacia la derecha	117
Figura 2-79. Rebanada obtenida de la impresión de voxels	117
Figura 2-80. Imagen del llenado interno de la pieza	118
Figura 2-81. Imagen sin huecos internos.....	118
Figura 2-82. Perfil a seguir	118
Figura 2-83. Imagen de la trayectoria descrita por la herramienta	119

Figura 2-84. Botón de maquinado hacia la izquierda	119
Figura 2-85. Botón de maquinado hacia la izquierda	119
Figura 2-86. Identificación de las partes de la interfaz de diseñador	120
Figura 2-87. Pantalla de usuario	121

Capítulo 3: Interfaz y mecanismo

Figura 3-1. Diseño del prototipo en CAD	124
Figura 3-2. Vista explosionada del mecanismo del eje "z"	125
Figura 3-3. Imagen del diseño del mecanismo del eje Z	126
Figura 3-4. Diseño del mecanismo del eje Y	126
Figura 3-5. Vista explosionada del mecanismo del eje Y	127
Figura 3-6. Mesa de trabajo donde se sujeta el bloque a maquinar	127
Figura 3-7. Vista del mecanismo del eje X	128
Figura 3-8. Vista explosionada del mecanismo del eje X	128
Figura 3-9. Sujeción de los motores al mecanismo	129
Figura 3-10. Vista explosionada	129
Figura 3-11. Prototipo terminado	130
Figura 3-12. Fotografía del prototipo	131
Figura 3-13. Fotografía de la ubicación de un sensor de final de carrera	132
Figura 3-14. Circuito eléctrico de los sensores de finales de carrera	132
Figura 3-15. Fotografía de la ubicación de un sensor de inicio del eje Y	133
Figura 3-16. Fuente conmutada de 12 volts a 8.5 Amperes	133
Figura 3-17. Fotografía del mototool y la herramienta de corte montado en la máquina ..	134
Figura 3-18. Diagrama de proceso de la interfaz	136
Figura 3-19. Conexión electrónica del puerto a los aisladores y a los drivers de los pines de dirección y paso del L297	138
Figura 3-20. Conexión de las entradas al puerto de los sensores de inicio	139
Figura 3-21. Circuito electrónico del driver del motor	140
Figura 3-22. Fotografía del driver de un motor	140
Figura 3-23. Conexión electrónica del encendido y apagado del mototool y la fuente de alimentación	141
Figura 3-24. Diagrama de comunicación entre el puerto paralelo con los controladores, sensores de inicio y el control de encendido	141
Figura 3-25. Circuito de interfaz entre PC y máquina	142
Figura 3-26. Fotografía de la tarjeta de interfaz y de controladores de los motores.	143

Capítulo 4: Aplicación

Figura 4-1. Ejemplos de empaques con forma del producto	145
Figura 4-2. Empaques de hilo dental	145
Figura 4-3. Molde para blíster de producto de hilo dental	146
Figura 4-4. Ensamble con la cavidad en forma transparente	146
Figura 4-5. Cavidad del molde de empaque	147
Figura 4-6. Selección del tipo de formato	148
Figura 4-7. Ventana donde se selecciona el tipo de código, la resolución, unidades, etc. .	148
Figura 4-8. Ventana mostrada al guardar la pieza en código ASCII	149
Figura 4-9. Ventana mostrada al guardar la pieza en código binario	149

Figura 4-10. Formato stl de la pieza dibujada	150
Figura 4-11. Sección de anotación de las características de la herramienta de corte	150
Figura 4-12. Distribución de las pantallas de la interfaz de desarrollador	151
Figura 4-13. Pantallas tomadas de la rebanada 1.....	151
Figura 4-14. Maquinado de rebanada	152
Figura 4-15. Pantallas tomadas de la rebanada 25.....	152
Figura 4-16. Pantallas tomadas de la rebanada 50.....	153
Figura 4-17. Pantallas tomadas de la rebanada 75.....	154
Figura 4-18. Pantallas tomadas de la rebanada 120.....	154
Figura 4-19. Fotografía del maquinado de la rebanada en acabado	155
Figura 4-20. Fotografía de la pieza terminada.....	155
Figura 4-21. Corazón del molde	156
Figura 4-22. Ventada de características del formato stl	156
Figura 4-23. Superficie de la pieza en formato stl.....	157
Figura 4-24. Imagen de la rebanada 1 del corazón del molde.....	157
Figura 4-25. Fotografía de maquinado	158
Figura 4-26. Imagen capturada de la rebanada 25.....	158
Figura 4-27. Fotografía del mecanismo maquinando siguiendo el perfil en la computadora	159
Figura 4-28. Fotografía del maquinado de las últimas rebanadas en el desbaste	160
Figura 4-29. Fotografía del maquinado en la etapa de acabado	160
Figura 4-30. Renderizado de las piezas en CAD.....	161
Figura 4-31. Fotografía del molde maquinado	161
Figura 4-32. Dibujo del modelo a fabricar	162
Figura 4-33. Superficie formada por triángulos en el formato stl	162
Figura 4-34. Pantallas tomadas de la rebanada número 75	163
Figura 4-35. Fotografía isométrica del modelo terminado	163
Figura 4-36. Fotografía del modelo terminado.....	164
Figura 4-37. Fotografía del maquinado de las letras IPN.....	164
Figura 4-38. Fotografía del logo de "Hecho en México"	165

Índice de diagrama de bloques

Diagrama de bloques 1-1. Conexión de los dispositivos electrónicos	36
Diagrama de bloques 2-1. Extracción de vértices	56
Diagrama de bloques 3-1. Conexión entre los dispositivos para el control de motores.....	135

Índice de diagramas de flujo

Diagrama de flujo 2-1. Algoritmo para obtención de coordenada de vértice.....	90
Diagrama de flujo 2-2. Algoritmo de cálculo de incrementos	94
Diagrama de flujo 2-3. Algoritmo de relleno interno del contorno	103
Diagrama de flujo 2-4. Algoritmo para el llenado de huecos.....	105
Diagrama de flujo 2-5. Algoritmo de dilatación.....	106
Diagrama de flujo 2-6. Algoritmo de erosión.....	107
Diagrama de flujo 2-7. Algoritmo de marcar contorno	109
Diagrama de flujo 2-8. Algoritmo de seguimiento de perfil	112

Resumen

Se desarrolló un sistema de manufactura asistida por computadora para piezas en 3 dimensiones. El método desarrollado proporciona una traducción exenta de errores, por lo cual es posible pasar directamente del dibujo en CAD al maquinado, de esta forma es costeable fabricar desde una pieza. El método consiste en conocer toda la geometría de la pieza, por lo cual se determinan todos los puntos de la superficie a partir de un archivo en formato STL, para la fabricación de la pieza se secciona su superficie en rebanadas y se maquina el perfil de cada rebanada.

Para la prueba del CAM se construyó un mecanismo cartesiano (x, y, z), con estos tres grados de libertad sólo es posible la fabricación de piezas en 3D con las características de los moldes. Como dispositivos de transmisión se usaron husillos embalados, para el posicionamiento de los ejes se emplearon motores a pasos. El control del mecanismo y la interfaz con el usuario se realiza a través de una computadora personal.

Abstract

A computer-aided manufacturing system for pieces has been developed. The developed method provides a free-error translation, thus it is possible to change directly from CAD drawing to the piece manufacturing, this way is feasible to make one piece. The method consists in knowing the full geometry of the piece; therefore, all the surface points were determined based on a STL format file.

In order to manufacture the piece, it is necessary to slice its surface and each surface profile has to be manufactured. For the CAM test a Cartesian (x, y, z) mechanism was built. Due this system have only three-freedom degrees, it is just possible to build pieces with the mold characteristics.

As transmission devices, packed spindle were used, as axis positioning devices, stepper motors were used. The control mechanism and the user interface are made with a personal computer.

Introducción

Introducción

A lo largo de la historia el hombre ha inventado herramientas para facilitar tareas de la vida cotidiana y en su búsqueda de mejores materiales nació la metalurgia y a partir de ella surgieron las máquinas herramientas, cuya función principal es darle forma al material.

Las máquinas tuvieron mucho auge en la revolución industrial para la fabricación de diversas maquinarias y herramientas de distintas industrias como: la textil, la alimentaria, la fundidora, la minera, la petrolera, la automotriz, etc.

Dos de las máquinas más utilizadas en la industria son el torno y la fresadora, la primera realiza piezas de revolución y la segunda piezas con caras planas y ranuras.

El proceso de fabricación empieza con el dibujo de la pieza a fabricar, después el operador de la máquina revisa el plano para que el maquinado que está realizando cumpla con las especificaciones del diseño y para esto debe tener instrumentos de medición. La rapidez con que se fabrique y la calidad del terminado de la pieza dependen de la habilidad, destreza y conocimiento del operador.

Una de las piezas que se fabrican son los modelos para la realización de moldes para fundición y los moldes de inyección.

El proceso de fundición consiste en calentar un material como aluminio, bronce, latón, hierro hasta convertirlo en un estado líquido y vaciarlo en un molde hecho de arena, donde posteriormente se solidifica al enfriarse y finalmente se rompe el molde para obtener la pieza.

Con la llegada del plástico, muchas piezas que se realizaban en metal fueron sustituidas por el polímero ya que su punto de fusión es mucho menor que el de los metales, por ello los moldes de arena fueron sustituidos por el metal y al no deshacer el molde la producción aumentó considerablemente.

Es por esto que se han hecho moldes en acero para realizar el moldeo por inyección, que es un proceso semi-continuo que consiste en inyectar un polímero, vidrio, etc., en estado fundido en un molde cerrado a presión, a través de un orificio pequeño llamado compuerta, para que el material se distribuya dentro del molde y posteriormente se solidifica. La pieza o parte final se obtiene al abrir el molde y sacar de la cavidad la pieza moldeada.

La fabricación de estos moldes es tan compleja como la pieza misma, y en la actualidad la mayoría son maquinadas en fresadoras de CNC.

Con el paso del tiempo y con la necesidad de tener mayor producción, las máquinas cambiaron de ser operadas manualmente a ser máquinas totalmente autónomas. Esto ha sido gracias a la automatización de los procesos de operación de la máquina, que en un principio fueron por medio de sistemas mecánicos y después fueron sustituidos o complementados por sistemas eléctricos, neumáticos e hidráulicos, y recientemente por sistemas electrónicos y computacionales [1].

La ventaja de automatizar las máquinas ha impactado en la disminución de accidentes y también en el aumento de la velocidad de maquinado, en la facilidad en su manejo y control, así como la capacidad de fabricar piezas complejas como esferas o relieves, consecuentemente aumentando la eficiencia y la precisión del maquinado.

El Control Numérico Computarizado (CNC) controla a la máquina mediante comandos programados de manera eficiente y exacta en todo momento. Utiliza un lenguaje formado por números, letras y símbolos normalizados conocidos como códigos G & M, dirigidos, supervisados y ejecutados por medio de una computadora integrada a la máquina.

La máquina realiza los movimientos de la pieza o de la herramienta de acuerdo a las instrucciones mandadas desde la computadora para la fabricación, dando lugar a que existan tornos y fresadoras CNC's.

Las máquinas CNC tienen la capacidad de fabricar piezas iguales a gran escala, sin embargo, la necesidad por generar mayor producción, excitó al desarrollo y al perfeccionamiento de los procesos de fabricación de moldes y modelos para fundición, pues es posible realizar una cantidad mucho mayor de piezas en diferentes materiales a un menor costo y mayor rapidez.

De la misma manera para facilitar el diseño de una pieza, surgió el dibujo asistido por computadora (CAD) que es el conjunto de aplicaciones informáticas que permiten definir el producto a fabricar.

La manufactura asistida por computadora (CAM) es el sistema que agrupa las aplicaciones encargadas de traducir las especificaciones de diseño a especificaciones de maquinado, esto para facilitar la generación de códigos.

De esta forma se tienen sistemas CAD/CAM/CNC que son parte del proceso de manufactura de una pieza, desde el dibujo hasta la pieza terminada. Además existen otros sistemas asistidos por computadora que complementan la administración de las máquinas y el proceso de la producción y otros aún más especializados.

Sin embargo, a pesar de que los sistemas CAD/CAM aminoran el trabajo del operador, sigue existiendo la intervención del hombre, en este caso de un especialista para la depuración y corrección del programa que contiene los códigos G y M, que describen la trayectoria que debe seguir la herramienta para obtener la pieza deseada, y más aún si la pieza es demasiado compleja, es decir, si contiene una geometría como las utilizadas en los moldes, ya que el número de líneas de código crece [2].

Con el uso de estas tecnologías es posible tener una nave industrial con varias máquinas CNC y estar siendo operadas incluso por una sola persona.

Para realizar de una manera más efectiva la traducción de las especificaciones del diseño a las especificaciones del maquinado, se desarrolló un sistema de manufactura que contempla en todo momento la geometría de la pieza y utiliza el barrido como estrategia de maquinado.

Planteamiento del problema

Dado que en los sistemas CAD/CAM/CNC utilizan como estrategia de maquinado la trayectoria que debe seguir la herramienta de corte e ignoran la geometría de la pieza, existen errores inherentes a la traducción, como errores de trayectoria y colisiones entre la herramienta de corte con el material, por lo que los programas generados deben revisarse a base de simulaciones y al ser bastante extensos, su depuración es muy tardada, lo cual hace que estos sistemas solo sean costables para producción en serie [1] [3].

Propuesta

La propuesta es desarrollar un sistema de manufactura asistida por computadora, exenta de errores, la cual emplea el barrido como estrategia de maquinado, conociendo en todo momento la geometría de la pieza a fabricar y a partir de ella se genere los movimientos de la herramienta sin la intervención de algún especialista.

Justificación

Hacer más sencillo el uso de tecnologías CAD/CAM, sin la necesidad de ser expertos en programación de CNC, además de eliminar el tiempo destinado a depurar el programa generado por el CAM, ya que no se genera alguno. Lográndose de esta manera pasar directamente del dibujo al maquinado.

Objetivo general

Implementar un sistema de manufactura asistida por computadora exento de errores, empleando el barrido como estrategia de maquinado.

Objetivos particulares

- Generar un algoritmo para obtener las especificaciones de maquinado, que contemple en todo momento la geometría de la pieza.
- Implementar el algoritmo desarrollado con bajo costo computacional.

Metodología

La metodología que se siguió para el desarrollo del trabajo es la que a continuación se describe.

- Interpretar las especificaciones de diseño de los archivos STL
- Desarrollar el algoritmo para obtención de voxels que integran el área de cada triángulo que forma la superficie del sólido
- Desarrollar el algoritmo para el rebanado de la superficie, la obtención y el seguimiento del perfil de cada rebanada.
- Programación del CAM.
- Diseñar y construir interfaz PC-máquina.
- Diseñar y construir la tarjeta electrónica para el movimiento de los motores.
- Construcción del mecanismo de tres ejes x, y, z.
- Desarrollar la interfaz para usuario.
- Escalar los pixeles de la imagen a la resolución de la máquina.
- Realizar pruebas en el mecanismo

Aportaciones

A continuación se enlistan las aportaciones realizadas de este trabajo.

- Se desarrolló un CAM exento de errores de traducción
- Se desarrolló un algoritmo para la obtención de puntos de la superficie de un sólido a partir de archivos STL.
- Se desarrolló una plataforma para el uso del láser como herramienta de corte para la fabricación de piezas en 3 dimensiones.

Estructura de la tesis

La forma en que se estructura el escrito es por capítulos, y a continuación se explican de forma general sobre los asuntos que se tratan en cada capítulo de esta tesis, para la familiarización del tema y del trabajo desarrollado.

En el Capítulo 1: “Antecedentes teóricos”, se explican las bases teóricas utilizadas para el desarrollo del trabajo realizado, tanto en la parte mecánica, electrónica y de programación. Así como referencias de la evolución de las máquinas y de las tecnologías actuales para la manufactura de piezas tan complejas como los moldes.

En el Capítulo 2: “Diseño e implementación del CADiseño e implementación del CAM”, se abordan los aspectos de diseño de los algoritmos desarrollados para la construcción de la superficie del sólido dibujado en CAD, desde como extraer los valores de los vértices del archivo STL, de calcular las coordenadas de los voxels que forman las aristas y el área de cada triángulo, así como de la obtención del perfil que debe describir la herramienta para el maquinado. Se aborda también sobre la programación de dichos algoritmos y de las rutinas para el movimiento de los motores, así como la descripción de la interfaz de diseñador y de usuario.

En el Capítulo 3: “Interfaz y mecanismo”, se habla de la forma de comunicación física que existe entre la PC y la máquina de 3 ejes, de la implementación del mismo y de cómo se mandan los datos de la imagen obtenida a los controladores de los motores, que mueven el mecanismo y a su vez a la herramienta de corte.

En el Capítulo 4: “Aplicación”, se muestra paso a paso la forma en que el CAM desarrollado realiza las tareas de los algoritmos desarrollados, dando a lugar a la comprobación del software en un medio físico y funcional.

En el apartado de las conclusiones y trabajo a futuro, se hace una breve reseña de las conclusiones obtenidas del trabajo realizado, así como de algunas recomendaciones de trabajos a futuros.

Finalmente hace la mención de las referencias utilizadas para esta tesis.

Capítulo 1:

Antecedentes

teóricos

Introducción

Este capítulo aborda aspectos teóricos necesarios para el desarrollo de este trabajo que se describen en los capítulos 2 y 3. Debido a que es un trabajo en el área de la mecatrónica, es decir que engloba tres áreas de ingeniería: la mecánica, electrónica y de computación, el capítulo se ha dividido en 4 secciones generales para su comprensión en los siguientes aspectos: generales, mecanismo, interface y cómputo.

En la sección de aspectos generales se habla sobre las máquinas y su evolución a través de los años, de cómo han surgido nuevas tecnologías que complementan la manufactura de piezas facilitando dicho proceso y aumentando la eficiencia del maquinado.

En la sección de aspectos del mecanismo, se desarrolla el tema sobre los mecanismos utilizados en las máquinas para el movimiento de la pieza o herramienta de corte para la manufactura.

En la sección de los aspectos de la interface se explica la forma en la que se comunica la computadora con la máquina, es decir, es un medio físico de convertir los mandos computacionales en señales que controlan los elementos eléctricos y electromecánicos.

En la sección de los aspectos computacionales se presentan las características del formato de archivo a manipular, así como de información de la imagen binaria y de los algoritmos de procesamiento de imagen.

1. 1 Aspectos generales

En la actualidad, en la industria metal mecánica se cuentan con máquinas sofisticadas de control automático en donde la intervención del usuario es mínima, y que además pueden ser complementados con otros sistemas asistidos por computadora para tener una mejor administración y una eficiente producción. Sin embargo, aún en la mayoría de las empresas todavía se trabajan con máquinas convencionales, dado que aún están en condiciones de trabajar, además de ser más económicas que las máquinas CNC y no requiere de una capacitación tan sofisticada. A continuación se da una breve explicación de las máquinas y tecnologías que ayudan al proceso de manufactura.

1.1.1 Máquina herramienta

La máquina herramienta es un tipo de máquina que se utiliza para dar forma a materiales sólidos, principalmente metales. El modelado de la pieza se realiza por la eliminación de una parte del material, que se puede realizar por arranque de viruta, estampado, corte o electroerosión o con nuevas tecnologías como el láser. Las dos máquinas herramientas más utilizadas en la industria son las siguientes:

1.1.1.1 Torno

Esta máquina permite fabricar piezas de forma geométrica de revolución y operan haciendo girar la pieza, sujeta en el cabezal o entre los puntos de centros, mientras una o varias herramientas de corte son empujadas en un movimiento regulado de avance contra la superficie de la pieza, cortando la viruta de acuerdo con las condiciones tecnológicas de maquinado adecuados, como se muestra en la Figura 1-1.

La herramienta de corte está montada sobre un carro que se desplaza sobre unas guías o rieles paralelos al eje de giro de la pieza que se torne, llamado eje Z; sobre este carro hay otro que se mueve según el eje X, en dirección radial a la pieza, y puede haber un tercer carro llamado auxiliar que se puede inclinar, para hacer conos, y donde se apoya la torreta portaherramientas. Cuando el carro principal desplaza la herramienta a lo largo del eje de rotación, produce el cilindrado de la pieza, y cuando el carro transversal se desplaza de forma perpendicular al eje de simetría de la pieza se realiza la operación denominada refrentado.



Figura 1-1. Torno paralelo convencional

1.1.1.2 Fresadora

Es una máquina herramienta utilizada para realizar maquinados por arranque de viruta mediante el movimiento de una herramienta rotativa de varios filos de corte denominada fresa (Figura 1-2). La pieza se desplaza en los tres ejes (x, y, z) acercando las zonas a maquinar a la herramienta, permitiendo obtener formas diversas como superficies planas, ranuras rectas y de forma de acuerdo al cortador como: ranuras de media luna, dientes de engrane y de catarinas.

Las fresadoras pueden clasificarse según el número de ejes de operación:

- Ejes x, y, z.
- Ejes x, y, z, y giro de pieza.
- Eje x, y, z, giro de pieza y giro de herramienta de corte en un eje.
- Eje x, y, z, giro de pieza y giro de herramienta en dos ejes.

y por la orientación del eje de giro:

- Fresadora horizontal
- Fresadora vertical
- Fresadora universal



Figura 1-2. Fresadora vertical

1.1.1.3 Proceso de Maquinado

Tanto en las máquinas convencionales como las de control numérico computarizado, existe un proceso de fabricación, la mayoría de los maquinados pasan por tres etapas, llevadas a cabo por una variedad de estrategias básicas y sofisticadas de maquinado, dependiendo del material. Las etapas son:

1.1.1.3.1 Desbaste

Comienza con material en bruto, conocido como tejo, y se maquina con grandes cortes y avance medio para aproximarse a grandes rasgos a la forma del modelo final. En el fresado, las estrategias comunes son en zig-zag o barrido, por ruta, por penetración. En el torneado, son el cilindrado y el careo.

1.1.1.3.2 Semi-acabado

El semi-acabado tiene que dejar una pequeña cantidad de material para que la herramienta pueda cortar con precisión, se aproxima al modelo final, teniendo una tolerancia de unos milímetros arriba de la medida nominal.

1.1.1.3.3 Acabado

Acabado implica un paso lento a través del material en cortes muy finos, la velocidad del giro de la herramienta o del material debe ser rápido y su avance lento.

1.1.2 Control Numérico Computarizado CNC

En el afán de eliminar o disminuir el error y la limitación humana en el maquinado de piezas complejas, se ha sustituido el control humano con el control automático, dentro de los cuales se encuentra el control numérico.

Poco después de la Segunda Guerra Mundial se desarrollaron los sistemas de control numérico (NC), que utilizaban una serie de números perforados en una cinta de papel o tarjetas para controlar el movimiento de la máquina y consistía en planificar cada movimiento que el operario realizaría, para posteriormente ser ejecutados secuencialmente de una manera rápida, evitando las imprecisiones que se comenten en un proceso manual.

Posteriormente, con el auge de la computación surgió el Control Numérico Computarizado (CNC), que es una forma de automatización programable y controla a la máquina mediante comandos programados en un medio de almacenamiento. Es numérico por que utiliza un lenguaje formado por números, letras y símbolos normalizados, conocidos como códigos G & M, para definir un programa de instrucciones para desarrollar una tarea específica y están dirigidos, supervisados y ejecutados por medio de una computadora integrada a la máquina [4][5].

El objetivo del CNC es lograr la automatización en la fabricación de piezas y el de las máquinas automáticas es reproducir las piezas diseñadas la cantidad de veces necesarias y disminuir al máximo la intervención del hombre en la operación de la misma [5].

La capacidad de cambiar el programa hace que el CNC sea apropiado para volúmenes de producción medios y altos, dado que es más fácil escribir nuevos programas que realizar cambios en los equipos de procesado.

El principio de operación del CNC es el control de la posición relativa de una herramienta o elemento de procesado con respecto al objeto a procesar, en base a un sistema de coordenadas que especifican el movimiento de la herramienta de corte con relación a los ejes de la máquina.

En un torno se controlan los movimientos de la herramienta en dos ejes: el X y el Z, para los desplazamientos longitudinales y transversales de la torreta respectivamente.

En las fresadoras se controlan los desplazamientos: longitudinal y transversal de la mesa, y el vertical del cabezal donde se encuentra el porta- herramienta, correspondientes a los ejes X, Y, Z respectivamente, pudiendo ser de más ejes.

Para realizar los movimientos se incorporan servomotores en los mecanismos de desplazamiento de la torreta, en el caso del torno; la mesa y el cabezal, para la fresadora.

Se considera control numérico a todo dispositivo capaz de dirigir posicionamiento de una pieza mecánica móvil, en el que las órdenes relativas a los desplazamientos del móvil son elaboradas en forma totalmente automática a partir de informaciones numéricas definidas, ya sea de manera manual o por medio de un programa.

1.1.2.1 Código G y M

La programación nativa de la mayoría de las máquinas de Control Numérico Computarizado se efectúa mediante un lenguaje de bajo nivel llamado G & M.

Se trata de un lenguaje de programación vectorial mediante el cual se describen acciones simples y entidades geométricas sencillas (básicamente segmentos de recta y arcos de circunferencia) junto con sus parámetros de maquinado (velocidades de husillo y de avance de herramienta).

El nombre G & M viene del hecho de que el programa está constituido por instrucciones Generales y Misceláneas.

Si bien en el mundo existen aún diferentes lenguajes de programación con códigos G&M, se dio un gran paso adelante a través de la estandarización que promovió la ISO (International Organization for Standardization) [1].

Esta estandarización fue adoptada por la totalidad de los fabricantes industriales serios de CNC y permite utilizar los mismos programas en distintas máquinas CNC de manera directa o con adaptaciones menores.

Los códigos de programación están normalizados para que un mismo programa pueda servir en diversas máquinas. Los caracteres más usados son:

- N: es la dirección correspondiente al número de bloque o secuencia. Esta dirección va seguida normalmente de un número de tres o cuatro cifras.
- X, Y, Z: son las direcciones correspondientes a las cotas según los ejes X, Y, Z de la máquina herramienta. Dichas cotas se pueden programar en forma absoluta o relativa, es decir, con respecto al cero de la pieza o con respecto a la última cota respectivamente.
- G: es la dirección correspondiente a las funciones preparatorias. Se utilizan para informar al control de las características de las funciones de maquinado, como por ejemplo, forma de la trayectoria, tipo de corrección de herramienta, parada temporizada, ciclos automáticos, programación absoluta y relativa, etc. La función G va seguida de un número de dos cifras que permite programar hasta 100 funciones preparatorias diferentes.

Ejemplos:

- G00: El trayecto programado se realiza a la máxima velocidad posible.
 - G01: Los ejes se gobiernan de tal forma que la herramienta se mueve a lo largo de una línea recta.
 - G02: Interpolación circular en sentido horario.
 - G03: Interpolación circular en sentido antihorario.
 - G33: Indica ciclo automático de roscado.
 - G40: Cancela compensación.
 - G41: Compensación de corte hacia la izquierda.
 - G42: Compensación de corte a la derecha.
-
- M: es la dirección correspondiente a las funciones auxiliares o complementarias. Se usan para indicar a la máquina que se deben realizar operaciones tales como parada programada, rotación del husillo a la derecha o a la izquierda, cambio de herramienta. La dirección M va seguida de un número de dos cifras que permite programar hasta 100 funciones auxiliares diferentes.

Ejemplos:

- ❖ M00: Provoca una parada incondicional del programa, detiene el husillo y la refrigeración.
- ❖ M02: Indica el fin del programa. Se debe escribir en el último bloque del programa y posibilita la parada del control una vez ejecutadas el resto de las operaciones contenidas en el mismo bloque.
- ❖ M03: Activa la rotación del husillo en sentido horario.
- ❖ M04: Activa la rotación del husillo en sentido antihorario.
- ❖ M05: Parada del cabezal
- ❖ M06: Cambio de herramienta
- F: Corresponde a la velocidad de avance, seguida de cuatro cifras en mm/min.
- S: Correspondiente a la velocidad de rotación del husillo principal (r.p.m.), usando cuatro dígitos.

Al conjunto de información que corresponde a una misma fase del maquinado se le denomina bloque o secuencia, que se numeran para facilitar su búsqueda. Este conjunto de informaciones es interpretado por la computadora. Una secuencia o bloque de programa debe contener todas las funciones geométricas, funciones máquina y funciones tecnológicas del maquinado. De tal modo, un bloque de programa consta de varias instrucciones.

1.1.2.2 Programación en el control numérico

Se pueden utilizar dos métodos para la programación de los códigos, la primera y la que más se ha utilizado es la programación manual y la segunda, que está teniendo mayor auge es la programación por medio del software [2][6].

El programa de maquinado comprende todo el conjunto de datos que el control necesita para la fabricación de la pieza.

1.1.2.2.1 Programación manual

El programa se escribe por medio de razonamientos y cálculos que realiza un operario, y conforme anota los comandos, revisa la programación por medio de la simulación y puede detectar las fallas o errores que surjan. Sin embargo, este tipo de programación es cansada y más si la pieza a fabricar es compleja, y las posibilidades de tener error son muchas

1.1.2.2.2 Programación por medio de software

Los cálculos los realiza una computadora, por medio de un software conocido como CAM, que traduce en códigos G y M, a partir de un dibujo hecho en CAD. Sin embargo hay que revisar el programa para asegurarse de que no existan fallas y que la pieza simulada sea idéntica a la diseñada en CAD.

1.1.3 Centro de maquinado

El centro de maquinado, es una máquina de gran automatización que es capaz de realizar diversas operaciones de maquinado con el CNC, con una mínima intervención humana. Las operaciones más comunes en las que se usa esta máquina son aquellas que usan

herramientas de cortes rotatorias como brocas y cortadores. A comparación de este sistema de maquinado con los sistemas más tradicionales, se destaca como ventaja la velocidad de producción y como desventaja la inversión necesaria.

1.1.3.1 Características

Estos centros de maquinado cuentan con las siguientes características:

- La versatilidad y flexibilidad debido al alto grado de automatización, son capaces de ejecutar diferentes operaciones de maquinado en una sola pieza.
- Brinda un buen acabado superficial.
- Estas máquinas son reconfigurables.
- Uniformidad en la producción, condición necesaria para la producción en serie.
- Alta velocidad de producción.

Algunas de las virtudes que demuestran la flexibilidad y reconfigurabilidad mencionadas son, el cambio automático de herramientas, las utilidades de paletas transportadoras y el posicionado automático de la pieza de trabajo.

1.1.4 Dibujo Asistido por Computadora

Para facilitar el diseño de las piezas a fabricar, han surgido programas de dibujo que muestran ventajas sobre el tradicional.

El CAD es el uso de un amplio rango de herramientas computacionales que asisten a diseñadores de ingeniería, arquitectura, etc. También se puede llegar a encontrar denotado con las siglas CADD, es decir, dibujo y diseño asistido por computadora (Computer Assisted Drawing and Design) [7].

Permite al diseñador crear imágenes de partes, circuitos integrados, ensambles y modelos de prácticamente todo lo que se imagine en una estación gráfica conectada a una computadora. Estas imágenes se transforman en la base de un nuevo diseño, o en la modificación de uno previamente existente [8][9].

Estas herramientas se pueden dividir básicamente en programas de dibujo en dos (2D) y tres dimensiones (3D). Las herramientas de dibujo en 2D se basan en entidades geométricas vectoriales como puntos, líneas, arcos y polígonos, con las que se puede operar a través de una interfaz gráfica. Los modeladores en 3D añaden superficies y sólidos.

Pueden asociarse a cada entidad una serie de propiedades como color, usuario, capa, estilo de línea, nombre, definición geométrica, material, etc., que permiten manejar la información de forma lógica y enlazar el CAD a los sistemas de gestión y producción.

De los modelos 3D pueden obtenerse planos con cotas y anotaciones para generar la documentación técnica específica de cada proyecto, además, producir pre-visualizaciones foto-realistas del producto, aunque se prefiere exportar los modelos a programas especializados en visualización y animación [10].

1.1.5 Manufactura Asistida por Computadora

El CAM es el uso de programas informáticos para el control de máquinas herramientas y maquinaria relacionada en la fabricación de piezas de trabajo, el CAM no desplaza a profesionales calificados, solo es una herramienta que facilita su trabajo.

El CAM es por lo general un archivo de texto de código G y M, que obtiene a partir de un archivo de un dibujo realizado por computadora, a veces cientos de comandos de largo, que se transfiere entonces a una máquina CNC mediante un control numérico directo (DNC) del programa.

El DNC o distribuidor de control numérico, es un término para la creación de redes de fabricación de máquinas CNC, ya que en algunos controladores, la memoria de las máquinas es demasiado pequeña para un programa por lo que el programa se almacena en un equipo independiente y se envía directamente a la máquina, un bloque a la vez.

Las aplicaciones en la industria del CAM son: generación de programas de control numérico, simulación de estrategias y trayectorias de herramientas para maquinado de piezas diseñadas a partir de un modelo CAD, programación de soldadura y ensamblajes robotizados.

En casos muy particulares el CAM también se involucra en: Inspección asistida por computadora (CAI) y Ensayo asistido por computadora (CAT).

1.1.6 Sistema CAD/CAM

Una vez descritas las ventajas del CAD y del CAM, existe la posibilidad de combinarlas para optimizar la rapidez del diseño y la manufactura, conocido como sistema CAD/CAM [11], es decir, que a partir del Diseño Asistido por Computadora se realiza la Manufactura Asistida por Computadora [12], esto por medio de un software que interpreta el dibujo hecho en CAD y lo traduce en lenguaje de códigos G y M, para manufacturarse en una máquina CNC [2][13].

Pueden obtenerse piezas complejas, con excelente calidad en un menor tiempo, siendo óptimo para tener producción en bajo y mediano volumen, tan flexibles que se pueden manufacturar piezas totalmente diferentes una tras de otra, solo con el cambio del dibujo electrónico.

Aunado a esto, existen otras tecnologías que complementan este sistema, para optimizar y hacer más eficientes los sistemas de producción, como los que se mencionan a continuación.

1.1.7 Impresoras 3D

A pesar de que existen máquinas que fabrican casi cualquier pieza y que veces es necesario tener un prototipo del mismo para realizar algunas pruebas ergonómicas o análisis dinámico, etc., y dado que maquinar piezas complejas es costoso, surgió otra manera de fabricar piezas a un precio relativamente menor, haciendo el procedimiento contrario a la remoción, que es añadir material. Esto es posible con la impresión en 3D [14].

La necesidad de poder tener un modelo real de la pieza, mecanismo o producto, antes de maquinarlo, ha originado el desarrollo de impresoras 3D. Las máquinas de estereolitografía pueden construir cualquier forma el volumen con una serie de rebanadas [15]. En última

instancia estas máquinas requieren una serie de curvas cerradas en 2D que se rellenan con material solidificado [16][17].

La estereolitografía es un proceso de fabricación aditiva de resina (fotopolímero líquido de curado UV), y un láser UV para construir partes del modelo de una capa a la vez. En cada capa, el rayo láser traza una parte de la sección transversal del modelo en la superficie de la resina líquida. La exposición a la luz UV solidifica el trazo del láser de la resina y se adhiere a la capa de abajo [18].

Posteriormente, la plataforma desciende cierta cantidad lineal (0.05-0.15 mm) y después, se barre una capa de resina en toda la superficie recubriendo la capa anterior con material fresco, y el proceso se repite hasta tener el modelo completo. La pieza terminada se limpia de exceso de resina por inmersión en un baño químico y luego en un horno de curado UV.

1.1.8 Moldes

Dado que el prototipo desarrollado cuenta con tres grados de libertad, las características físicas de los moldes son el tipo adecuado de piezas que se pueden fabricar, por esta razón se explica brevemente acerca de ellos.

El molde es un ensamble que está formado de dos bloques (principalmente) de metal, que sirven para darle forma a un material fundido (plástico, vidrio, metal, cerámica), el cual se distribuye en las cavidades formadas por los dos bloques y se enfría hasta solidificarse. Posteriormente el molde se abre y se extrae el producto moldeado [19][20].

Una de las características importantes de un molde es el ángulo de salida, esto se refiere a que en un molde de dos piezas se debe de respetar el ángulo mínimo en la dirección de desmolde [21] (la dirección en la que se extrae el producto) con objeto que pueda salir la pieza, como se muestra en la Figura 1-3.

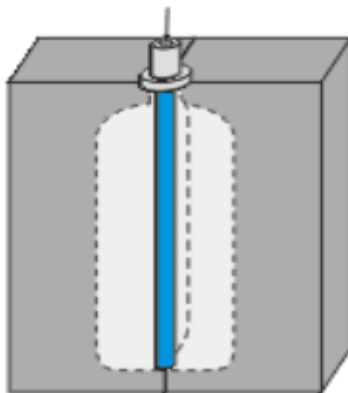


Figura 1-3. Molde con corte transversal de dos piezas para botella

Ya que si por el contrario la pieza necesita tener una forma como se muestra en el inciso a) de la Figura 1-4 no podrá ser desmoldada con el molde de 2 piezas, a menos que sufra una modificación en la orientación de la pieza o que deba estar como se muestra en el inciso b) de la Figura 1-4[22].

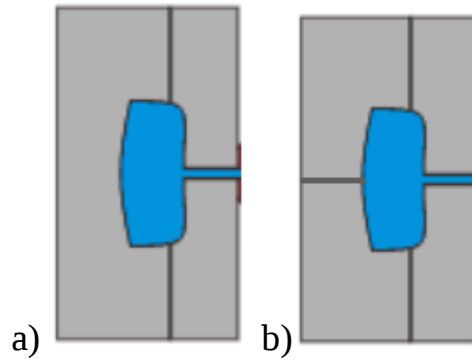


Figura 1-4. a) Molde de inyección de plástico de 2 piezas, b) Molde de inyección de plástico de 3 piezas

Es importante señalar que la mayoría de los moldes no son fabricados en su totalidad por una sola máquina, si no que requieren de varias especializadas en un proceso y complementarias unas de otras para fabricar en su totalidad el molde y más aún cuando son multipiezas para productos más elaborados.

En este trabajo la parte del molde a realizar es lo que corresponde a la cavidad y el corazón, lo que le da la forma al material fundido.

1.1.9 Modelos para moldes de fundición

Dado que las características geométricas del modelo que se usa para realizar un molde de arena son las adecuadas para la fabricación en el prototipo de tres ejes desarrollado (ver capítulo 3), se habla brevemente de los modelos para los moldes de arena.

Otro tipo de molde es el realizado con arena para el proceso de fundición, que es el proceso de fabricación de piezas comúnmente metálicas, y consiste en fundir el material e introducirlo en el molde donde se solidifica.

La fundición en arena requiere un modelo a tamaño natural de madera, plástico o metal que define la forma externa de la pieza que se pretende reproducir y que formará la cavidad interna en el molde, como se muestra en la Figura 1-5.

Las superficies del modelo deberán respetar unos ángulos mínimos con la dirección de desmolde (la dirección en la que se extraerá el modelo), con objeto de no dañar el molde de arena durante su extracción. Este ángulo se denomina ángulo de salida, y se recomiendan ángulos entre 0.5° y 2° .



Figura 1-5. Modelo de molde de fundición

1. 2 Aspectos del mecanismo

En esta sección se habla de los aspectos teóricos y componentes que intervienen en el mecanismo desarrollado, esto incluye todo lo que respecta al movimiento de la herramienta.

El mecanismo, por definición es el conjunto de sólidos, móviles unos respecto de otros, unidos entre sí mediante diferentes tipos de uniones, cuyo propósito es la transmisión de las máquinas. En este caso es aquel que le va a dar movimiento a la herramienta de corte, y se hará utilizando el principio del husillo y tuerca.

1.2.1 Husillo

El husillo es un tipo de tornillo pero más largo y sin cabeza, que al girar produce el desplazamiento longitudinal de la tuerca en la que va enroscado.

El tipo de cuerda más utilizado en un husillo para transmitir movimiento en máquinas es el Acme, que tiene un ángulo de rosca de 29° y una cara plana grande en la cresta y en la raíz (Figura 1-6).

Existen tres clases de rosca Acme, 2G, 3G y 4G, y cada una tiene holguras en todas dimensiones para permitir movimiento libre, la clase 2G es la más común, las clases 3G y 4G permiten menos juego u holgura, y se usan en el husillo de un torno o de la mesa de una maquina fresadora, con una tuerca, usualmente de bronce.



Figura 1-6. Tornillo cuerda ACME

Sin embargo, como en cualquier tipo de cuerda existe la holgura o juego mecánico entre el tornillo y la tuerca, ya que no es posible darles la misma medida puesto que no se deslizarían debido a la fricción, además de que con el tiempo y el uso el acoplamiento se desgasta, aumentando el juego y esto genera problemas de precisión y en consecuencia al error acumulativo del cambio de sentido de desplazamiento, para evitar esto se han utilizado los husillos embalados [23][24].

1.2.1.1 Husillo embalado

Con el fin de atacar el problema del juego mecánico, surgieron los tornillos embalados o de bolas. Estos husillos tienen uno o más circuitos de balines espaciados entre el tornillo y la tuerca (Figura 1-7). Cuando el tornillo o la tuerca son rotados, las bolas ruedan en los surcos formados por la cuerda del tornillo. Cada circuito tiene un mecanismo de retorno que hace circular continuamente los balines formando un circuito cerrado, como se muestra en la Figura 1-8.



Figura 1-7. Husillo embalado

Los tornillos embalados son de alta eficiencia y ofrecen la oportunidad de desplazar cargas al transformar el movimiento de rotación en movimiento lineal. Tienen elementos rodantes (balines) entre el tornillo y la tuerca, permitiendo un juego u holgura entre el husillo y la tuerca sea prácticamente nula, por lo que el error acumulativo que presentan los husillos convencionales al cambiar de sentido de desplazamiento se reduce al máximo²⁵.

Al reemplazar la fricción de deslizamiento por la fricción de rodamiento, la eficiencia del tornillo aumenta de un 30% a un 90%, haciéndolos mucho más durables y generando menor esfuerzo para los componentes motrices.



Figura 1-8. Visualización de balines

1.2.2 Motores a pasos

Para transmitir el movimiento al husillo es necesaria la ayuda de un motor, y específicamente uno que su control de posición sea sencillo. Existen motores de CA y de CD, sin embargo los de CD tienen la capacidad de poder ser controlado de una forma sencilla no solamente la velocidad si no también su posición, es decir, si se requiere un movimiento de 3.25 revoluciones se puede obtener y a una velocidad deseada [26].

Los motores de CD trabajan con corriente directa, y su fácil control de posición, par y velocidad los han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos.

El motor paso a paso es un tipo de motor de CD que convierte los pulsos eléctricos en desplazamientos angulares discretos, es decir, se mueven un paso a la vez por cada pulso que se le aplique. Dichos pasos varían desde 90° hasta 1.8°, por lo que se necesitan 4 y 200 pasos respectivamente para completar una revolución y pueden ser controlados por impulsos procedentes de sistemas lógicos [27].

La ventaja con la que cuenta es tener alta precisión y repetitividad en cuanto al posicionamiento, lo que los hace ideales para la construcción de mecanismos en donde se requieren movimientos muy exactos.

Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas.

Existen 3 tipos fundamentales de motores paso a paso: el motor de reluctancia variable, el motor de magnetización permanente, y el motor híbrido.

1.2.2.1 Motor de reluctancia variable

Tiene un rotor multipolar de hierro y un estator devanado laminado, rota cuando los dientes del rotor son atraídos a los dientes del estator electromagnéticamente energizados. La inercia del rotor de un motor de paso de reluctancia variable es pequeña y la respuesta es muy rápida, pero la inercia permitida de la carga es pequeña. Cuando los devanados no están energizados, el par estático de este tipo de motor es cero, generalmente, el paso angular de este motor es de 15°

La reluctancia magnética de un material, es la resistencia al flujo magnético. Se define como la relación entre la fuerza magnetomotriz (*f.m.m.*) [A] (ampere, comúnmente llamado *amper-vuelta*) y el flujo magnético (weber).

La reluctancia **R** de un circuito magnético uniforme se puede calcular como:

$$R = \frac{l}{\mu A} \dots \dots \dots (1)$$

Donde:

- R: reluctancia, medida en amperio vuelta por weber ($\frac{A}{\text{Weber}}$). Esta unidad es equivalente al inverso del Henrio (H^{-1}) multiplicado por el número de espiras.
- l: longitud del circuito, medida en metros.
- μ : permeabilidad magnética del material, medida en H/m (henrio/metro).
- A: Área de la sección del circuito (sección del núcleo magnético), en metros cuadrados.

1.2.2.2 Motor de imán permanente

Conocido también como PMSM (permanent magnet synchronous motor). Permite mantener un par diferente de cero cuando el motor no está energizado y el ángulo de rotación se determina por el número de polos en el estator (Figura 1-10).

La construcción de los rotores de los servomotores sincrónicos de imán permanente (Figura 1-9) pueden adoptar una forma cilíndrica con un bajo diámetro y gran longitud (cilinder rotor) llamados de flujo radial, o pueden tener un rotor en forma de disco más liviano rotor de disco (disk rotor), también llamadas máquinas de flujo axial, resultando así en ambos casos un bajo momento de inercia y una constante de tiempo mecánica baja.



Figura 1-9. Rotor de un motor a pasos



Figura 1-10. Imagen de un estator

Existen dos tipos de motores de imán permanente: el unipolar y el bipolar, los cuales se detallan más adelante.

1.2.2.3 El motor a pasos híbrido

Este motor tiene una configuración que es una mezcla de los dos tipos de motores a pasos descritos anteriormente, el de reluctancia variable y el de imán permanente. El rotor suele estar constituido por anillos de acero dulce dentado en un número ligeramente distinto al del estator y dichos anillos montados sobre un imán permanente dispuesto axialmente.

El rotor de apilado simple contiene dos piezas de polos separados por un magneto permanente magnetizado, con los dientes opuestos desplazados en una mitad de un salto de diente para permitir una alta resolución de pasos. Este tipo de motor tiene una alta precisión y alto par y se puede configurar para suministrar un paso angular tan pequeño como 1.8° [28].

1.2.2.4 Bipolar

Por lo general presentan cuatro cables de salida. Su control para el movimiento requiere del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento, como se muestra en la Figura 1-11.

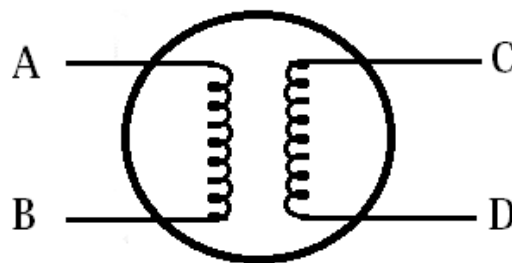


Figura 1-11. Motor bipolar de cuatro cables de salida

1.2.2.5 Unipolar

Estos motores suelen tener cinco o seis cables de salida, dependiendo de la conexión interna. En el caso del primero, cada par de bobinas tiene un solo común (Figura 1-12) y el

segundo, cada par de bobinas tiene un común separado (Figura 1-13), sin embargo, un motor unipolar de 6 cables puede ser usado como un motor bipolar si se deja las líneas del común al aire.

Para controlar su movimiento se utiliza un cable común a la fuente de alimentación y posteriormente se van colocando los otros cables a tierra en un orden específico para generar cada paso.

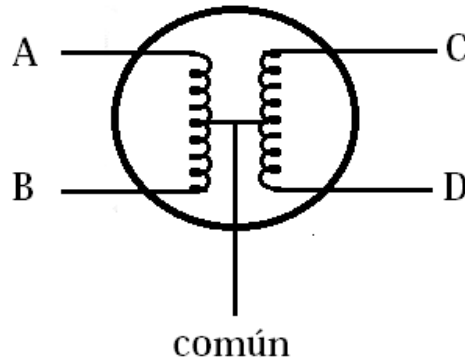


Figura 1-12. Motor unipolar de cinco cables

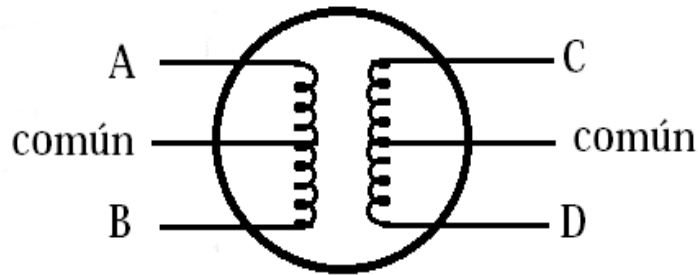


Figura 1-13. Motor unipolar de seis cables

1.2.2.6 Control del motor a pasos

Los motores a pasos necesitan la inversión de la corriente que circula en sus bobinas en una secuencia determinada para girar, y cada inversión de la polaridad provoca el movimiento del eje en un paso, cuyo sentido de giro está determinado por la secuencia seguida [29].

La variación de la dirección del campo magnético creado en el estator produce un movimiento de seguimiento por parte del rotor de imán permanente, el cual intentará alinearse con el campo magnético inducido por las bobinas que excitan los electroimanes.

Existen dos modos de controlar este tipo de motor, una conocida como paso completo y la otra como medio paso.

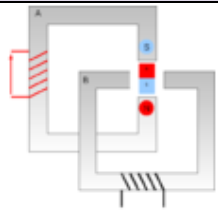
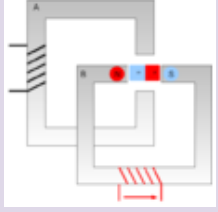
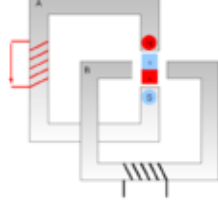
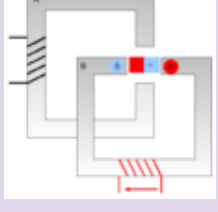
1.2.2.6.1 Paso completo

El paso completo consiste en energizar las bobinas de tal forma que el rotor se desplace un paso, es decir, si un motor de 200 pasos al recibir el pulso eléctrico el rotor gira 1.8° . Existen dos formas de realizarlo, y se describen a continuación³⁰

1.2.2.6.2 Paso completo de una bobina activa

En esta secuencia se activa solo una bobina a la vez, esto brinda un funcionamiento más suave y más rápido dado que la bobina que no está energizada disipa su energía y cuando se energiza con la polaridad invertida, su respuesta es mejor. Sin embargo, al estar solo una bobina activa, el torque de paso y retención es la mitad en comparación con la del paso completo y dos bobinas activas que se describe más adelante en la Tabla 1-1.

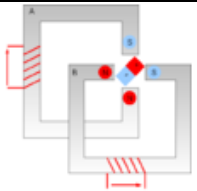
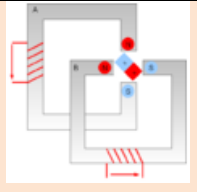
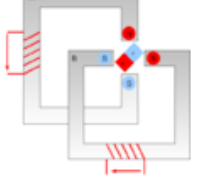
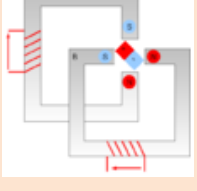
Tabla 1-1. Tabla de orden de fases de paso completo de un motor de paso angular de 90°

Paso	Bobina A		Bobina B		Imágenes
	Terminal 1	Terminal 2	Terminal 1	Terminal 2	
Paso 1	+Vcc	-Vcc	-	-	
Paso 2	-	-	+Vcc	-Vcc	
Paso 3	-Vcc	+Vcc	-	-	
Paso 4	-	-	-Vcc	+Vcc	

1.2.2.6.3 Paso completo de dos bobinas activas

Con la secuencia que se muestra en la Tabla 1-2, el motor avanza un paso por vez y debido a que las dos bobinas siempre están activas, se obtiene el máximo torque de paso y de retención, sin embargo la velocidad de secuencias de pulsos se limita, dado que no existe tiempo para disipar la energía en cada bobina, además de que en retención el motor tiende a calentarse más rápido dado que es cuando más corriente consume.

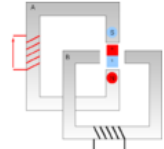
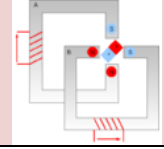
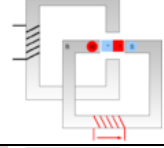
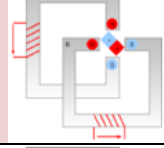
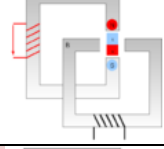
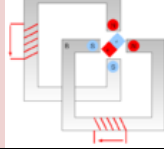
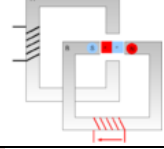
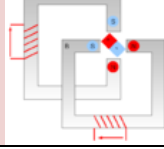
Tabla 1-2. Tabla de orden de fases de paso completo de un motor de paso angular de 90°

Paso	Bobina A		Bobina B		Imagen
	Terminal 1	Terminal 2	Terminal 1	Terminal 2	
Paso 1	+Vcc	-Vcc	+Vcc	-Vcc	
Paso 2	-Vcc	+Vcc	+Vcc	-Vcc	
Paso 3	-Vcc	+Vcc	-Vcc	+Vcc	
Paso 4	+Vcc	-Vcc	-Vcc	+Vcc	

1.2.2.6.4 Medio paso

Con este tipo de control se activan las bobinas de tal forma que se obtiene un movimiento angular igual a la mitad del paso real. Para ello se activan primero 2 bobinas y luego solo 1, con la secuencia que se muestra en la Tabla 1-3, donde también se aprecia que se conforma de 8 movimientos.

Tabla 1-3. Tabla de orden de fases de motor a pasos en medio paso

Paso	Bobina A		Bobina B		Imagen
	Terminal 1	Terminal 2	Terminal 1	Terminal 2	
Paso 1	+Vcc	-Vcc	-	-	
(Semi-)Paso 2	+Vcc	-Vcc	+Vcc	-Vcc	
Paso 3	-	-	+Vcc	-Vcc	
(Semi-)Paso 4	-Vcc	+Vcc	+Vcc	-Vcc	
Paso 5	-Vcc	+Vcc	-	-	
(Semi-)Paso 6	-Vcc	+Vcc	-Vcc	+Vcc	
Paso 7	-	-	-Vcc	+Vcc	
(Semi-)Paso 8	+Vcc	-Vcc	-Vcc	+Vcc	

En esta secuencia, se obtiene la ventaja de que la velocidad de respuesta es similar al control de paso completo de una bobina activa y en cuanto al torque combina la máxima con la media, además el número de pasos aumenta al doble, dándole mayor resolución.

1.2.2.6.5 Velocidad de rotación

La velocidad de rotación se define por la ecuación:

$$V = 60 * \frac{f}{n} \dots\dots\dots(2)$$

Donde:

- f: frecuencia del tren de impulsos
- n: número de polos que forman el motor

Cabe recalcar que debido a que los motores paso a paso son dispositivos mecánicos y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los pulsos aplicados es un punto muy importante a tener en cuenta.

Si bien hay que decir que para los motores a pasos la máxima frecuencia admisible suele estar alrededor de los 625 Hz. En cierto sentido el motor debe alcanzar el paso antes que la próxima secuencia de pulsos comience. Si la frecuencia de pulsos es muy elevada el motor puede reaccionar en alguna de las siguientes formas:

- Puede que no realice ningún movimiento en absoluto.
- Puede comenzar a vibrar pero sin llegar a girar.
- Puede girar erráticamente.
- Puede llegar a girar en sentido opuesto.

1.2.2.6.6 Control de las bobinas

Anteriormente se explicó la secuencia para invertir la polaridad de las bobinas en un motor a pasos para generar movimiento en ambos sentidos, y para esto se requieren de dos fuentes de energía para energizar las bobinas, como se muestra a continuación:

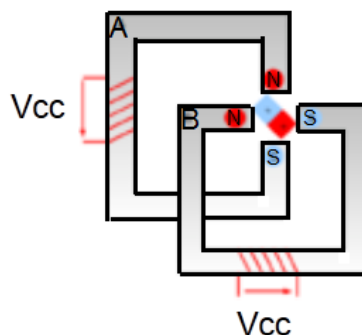


Figura 1-14. Bobinas conectadas a dos fuentes de alimentación.

Sin embargo, en cuestiones prácticas y económicas tener dos fuentes no es viable, por lo que se recurre al siguiente arreglo, llamado puente H.

1.2.2.7 Puente H

El puente H es un circuito electrónico que permite a un motor eléctrico de CD girar en ambos sentidos, con una sola fuente de alimentación.

El puente H está constituido de 4 interruptores abiertos, conectados a las terminales del motor y por el otro extremo a la fuente de alimentación. El funcionamiento para que el motor gire hacia un lado, se cierran únicamente los interruptores S1 y S4, y para que gire a la inversa, se desconectan S1 y S4 y se conectan S3 y S2.

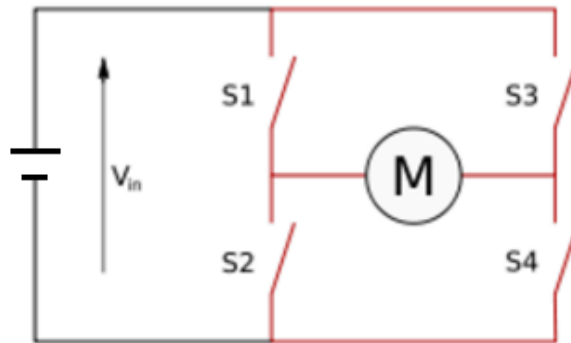


Figura 1-15. Diagrama del puente H

Para el control de los motores Bipolares se utiliza el Puente H para cada bobina, como se muestra en la Figura 1-16. Si se cierran los interruptores S1 y S4 se activa la bobina A y el rotor da un paso, y si además se cierran los interruptores S5 y S8, da medio paso [31].

Con este circuito y con la secuencia necesaria para el control de polaridad de las bobinas se obtiene el tipo de paso deseado: Paso completo y Medio paso.

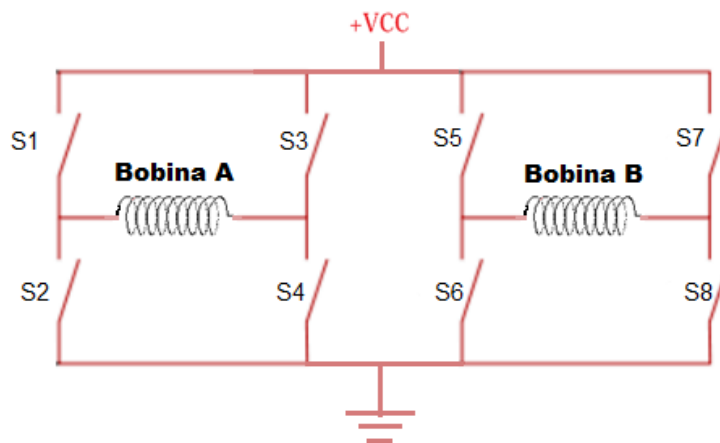


Figura 1-16. Un par de puentes H para el control de las bobinas del motor bipolar

El arreglo del puente H se puede realizar con componentes electrónicos, sin embargo, en el mercado existen circuitos integrados que realizan esta función. Uno de ellos y el utilizado en este trabajo es el L298.

La activación de los interruptores puede hacerse de manera manual y por medio de la computadora a través del puerto paralelo, sin embargo en el mercado existen dispositivos electrónicos que guardan esta configuración internamente. El utilizado para este trabajo es el circuito integrado L297, el cual tiene entradas que activan el sentido de giro del motor y del tipo de paso deseado: medio y paso completo [29].

En combinación de estos dos circuitos integrados, se realiza la conexión de la siguiente manera

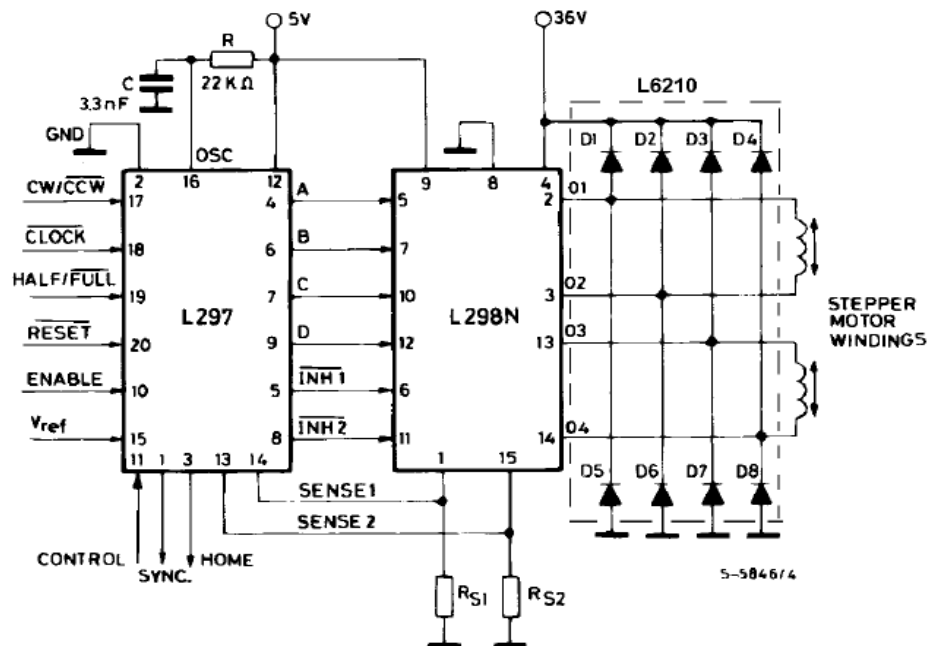


Figura 1-17. Diagrama de conexión entre el L297 y el L298

El dispositivo L297 recibe señales eléctricas para indicar las siguientes funciones:

Si en el pin 17 recibe 5 volts (un “1” lógico), el motor girará en sentido horario, si recibe 0 volts (un “0” lógico), en sentido anti-horario.

En el pin 18 recibe los pulsos en una frecuencia determinada, que genera el movimiento del motor.

El pin 19 indica si es el motor se mueve con paso completo o medio paso, si recibe 0 Volts (0 lógico) o 5Volts (1 lógico) respectivamente.

El pin 20 indica un reset o inicio con 5 Volts (1 lógico).

El pin 10 habilita o deshabilita el funcionamiento del L297, si recibe 5 volts o 0 volts respectivamente.

El pin 15 recibe un voltaje de referencia.

EL dispositivo L298 como se mencionó anteriormente, es un par de puentes H que realizan la activación y desactivación de las bobinas secuencialmente para el movimiento del motor a pasos.

1.2.3 Sensores

Para este trabajo se requieren usar sensores de final de carrera y de inicio de la herramienta, la primera por razones de seguridad y la segunda por la necesidad de saber la ubicación espacial de la herramienta

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, etc. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

A continuación se mencionan los sensores que se han utilizado en el mecanismo desarrollado.

1.2.3.1 Sensor de final de carrera

El mecanismo desarrollado, al igual que muchas máquinas, requieren de un sensor para indicar los límites físicos que tienen en el desplazamiento. Los sensores de finales de carrera, son dispositivos que indican el límite de la carrera o del espacio lineal máximo que tiene un mecanismo para desplazarse, con el fin de evitar una colisión, que indudablemente dañaría a la infraestructura a tal grado de romperse.

Existen sensores de proximidad que indican un límite de seguridad pero que por razones de que exista alguna falla en este sensor, se utiliza uno inmediatamente después que actúa directamente sobre la alimentación, es decir, que corta la corriente al ser activado o inhibe algún driver para parar el movimiento.

El microswitch es un interruptor que en vez de tener una entrada y una salida como se muestra en la Figura 1-18, tiene una entrada o común y dos salidas (Figura 1-19), es decir, tiene dos posiciones y por ende dos estados; el normalmente abierto (“N.O.”, por sus siglas en inglés) y el normalmente cerrado (N.C.). Y de acuerdo al estado del microswitch en su posición original, cuando no está activado, se les conoce como el normalmente abierto N.O. y normalmente cerrado N.C.

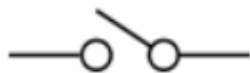


Figura 1-18. Interruptor de una vía

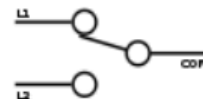


Figura 1-19. Interruptor de dos vías

1.2.3.2 Sensores de inicio o home

Para saber la ubicación de la herramienta, es necesario posicionarla inicialmente en un punto espacial conocido en las tres coordenadas, para ello se ocupan sensores de posición y se explican a continuación.

El dispositivo consiste en una conexión de diodo emisor infrarrojo frente a un detector de óptico encerrado en una caja de termoplástico negro (Figura 1-20). La conmutación del

detector se lleva a cabo cada vez que un objeto opaco pasa a través de la ranura entre el emisor y el detector.

Cuando no hay nada que interrumpa la luz entre el diodo emisor y el receptor, la salida Vo tiene un nivel lógico de '0', y cuando es interrumpido por algún objeto, y el receptor no recibe luz, el estado lógico cambia a 1 (Figura 1-21).

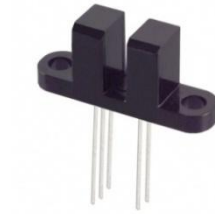


Figura 1-20. Sensor de posición de inicio

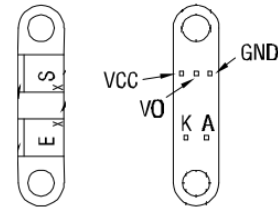


Figura 1-21. Especificación técnica del sensor

1.3 Aspectos de la interfaz

Para tener comunicación entre la computadora y los circuitos que controlan la máquina, es necesaria una interfaz. La interfaz es la conexión entre dos computadoras o máquinas de cualquier tipo dando una comunicación. El puerto es una interfaz a través de la cual diferentes tipos de datos se pueden enviar y recibir. Dicha interfaz puede ser de tipo físico (paralelo, USB, serie, etc.), o puede ser a nivel de software en cuyo caso se usa frecuentemente el término puerto lógico.

1.3.1 Puerto paralelo

Se ha seleccionado este tipo de interfaz debido a que es mucho más sencillo y económico de implementar a diferencia del puerto USB, que requiere de una serie de componentes electrónicos para poder ser usados; y del puerto serial ya que la forma de comunicación no es la adecuada para este trabajo.

El puerto paralelo es una interfaz entre la computadora y un periférico, cuya característica principal es que los bits de datos viajan juntos enviando un paquete de byte a la vez, es decir, se implementa un cable o una vía física para cada bit de datos formando un bus.

Existen tres tipos de conectores definidos por el estándar IEEE 1284, el primero llamado 1284 tipo A es un conector hembra de 25 pines de tipo D; el 1284 tipo B es un conector de 36 pines y se encuentran en la mayoría de las impresoras; el tercero se denomina 1284 tipo C que se trata de un conector similar al 1284 tipo B pero más pequeño.

El puerto paralelo está compuesto por un bus de comunicación bidireccional de 8 bits de datos además de un conjunto de líneas de protocolo. Las líneas de comunicación cuentan con un retenedor que mantiene el último valor que les fue escrito hasta que se escribe un nuevo dato, las características eléctricas son:

- Tensión de nivel alto: 3 o 5 V.
- Tensión de nivel bajo: 0 V.
- Corriente de salida máxima: 2.6 mA.
- Corriente de entrada máxima: 24 mA.

El puerto paralelo 1284 tipo A de una PC, utiliza un conector hembra de tipo D de 25 pines (DB-25 S). El orden de las pines del conector es como se muestra en la Figura 1-22.

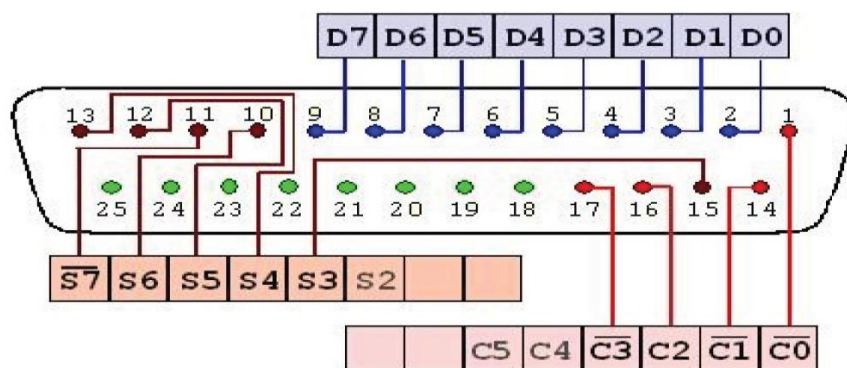


Figura 1-22. Imagen de distribución del puerto paralelo

La Tabla 1-4 describe la función de cada pin del conector 1284 tipo A.

Tabla 1-4. Configuración del puerto paralelo estándar

Pin	E/S	Polaridad activa	Descripción
1	Salida	0	Strobe
2 ~ 9	Salida	-	Líneas de datos (bit 0/patita 2, bit 7/patita 9)
10	Entrada	0	Línea acknowledge (activa cuando el sistema remoto toma datos)
11	Entrada	0	Línea busy (si está activa, el sistema remoto no acepta datos)
12	Entrada	1	Línea Falta de papel (si está activa, falta papel en la impresora)
13	Entrada	1	Línea Select (si está activa, la impresora se ha seleccionado)
14	Salida	0	Línea Autofeed (si está activa, la impresora inserta una nueva línea por cada retorno de carro)
15	Entrada	0	Línea Error (si está activa, hay un error en la impresora)
16	Salida	0	Línea Init (Si se mantiene activa por al menos 50 micro-segundos, ésta señal auto inicializa la impresora)
17	Salida	0	Línea Select input (Cuando está inactiva, obliga a la impresora a salir de línea)
18 ~ 25	-	-	Tierra eléctrica

Como sistema operativo, el DOS puede gestionar hasta cuatro interfaces de puertos paralelos: LPT1, LPT2, LPT 3 y LPT4, además reserva las siglas PRN como sinónimo del LPT 1, de modo que puede ser tratado como un archivo genérico. La dirección de entrada/salida de cada uno de los puertos paralelos y el número de puertos instalados en un PC se muestra en la pantalla inicial de arranque del equipo es frecuente, casi estándar que las direcciones de los dos primeros puertos paralelos sean las siguientes:

- LPT1 = 0x378 Hexadecimal
- LPT2 = 0x278 Hexadecimal

Las tarjetas del puerto paralelo tiene una estructura muy simple, consta de tres registros: de control, de estado y de datos. Todas las señales que intervienen en el puerto tienen asociado un bit en uno de esos registros de acuerdo con las funciones asignadas a cada línea en particular.

1.3.1.1 El registro de datos

Es de tipo latch de 8 bits que puede ser leído y escrito desde el procesador. Es el registro donde el procesador en operaciones de salida (OUT), pone el dato que se quiere enviar a la impresora y su dirección coincide con la dirección base del puerto paralelo (0x378 en LPT 1). En la Tabla 1-5 se muestra la distribución de los bits de este registro y los pines asociados a cada uno de ellos en el conector DB-25.

Tabla 1-5. Registro de datos del puerto paralelo

REGISTRO DE DATOS (D)								
BIT	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
PIN	9	8	7	6	5	4	3	2

1.3.1.2 El registro de estado

El registro de estado indica la situación actual de la impresora conectada al puerto, de acuerdo con los niveles de tensión que tengan las líneas ACK, BSY, PAP y OF/ON lo que permite controlar el comportamiento de la impresora.

Se trata de un registro de entrada (Lectura) de información, su dirección se obtiene sumando 1 a la dirección base del puerto (0x379 en LPT1).

Tabla 1-6. Registro de estado del puerto paralelo

REGISTRO DE ESTADO								
BIT	7	6	5	4	3	2	1	0
	/BSY	/ACK	PAP	ON-OFF	ERR	x	x	x
PIN	10	11	12	13	15			

Los bits de este registro se designan según se muestran en la Tabla 1-6, en la que el símbolo «/» adelante del nombre del bit indica que es activo a nivel bajo.

Pero el bit 7 además (/ BSY) del registro de estado (bit 7) es invertido por el hardware y por tanto la línea tiene un nivel complementado al que aparece en ese bit.

1.3.1.3 El registro de control

El registro de control permite controlar las transferencias de información con la impresora, y puede ser escrito y leído desde el microprocesador. Es un registro de entrada/salida cuya dirección se obtiene sumando 2 a la dirección base del puerto (0x37A en LPT1). Los bits de

este registro se designan en la Tabla 1-7, donde el símbolo «/» delante del nombre del bit indica que es activo a nivel bajo.

Tabla 1-7. Registro de control del puerto paralelo

REGISTRO DE CONTROL								
BIT	7	6	5	4	3	2	1	0
	X	X	X	IRQ	DSL	INI/	ALF	STR
PIN					17	16	14	1

El puerto paralelo se utiliza para manejar impresoras, sin embargo, dado que éste puerto tiene un conjunto de entradas y salidas digitales, se puede emplear para lectura de datos y control de dispositivos.

1.3.1.4 Características de Entrada/Salida

Cuando se usa el puerto paralelo para un propósito distinto al original, se puede hablar de 12 líneas de salida de información desde la computadora:

- Pines del 2 al 9 (8 pines) → registro de datos
- Pines 1,14, 16 y 17(4 pines) → registro de control

13 líneas de entrada al mismo:

- Pines del 2 al 9 (8 pines) → registro de datos
- Pines 10, 11, 12,13 y 15 (5 pines) → registro de estado

Esto hace del puerto paralelo una interfaz de comunicación con el exterior bastante flexible.

El registro de estado es de sólo lectura y lo que se recibe es el estado lógico de los pines 10, 11, 12, 13y 15 del conector DB-25 (el bit S 7 contiene el complemento del estado de la línea). Los tres bits de menor peso (SO-S2) no se utilizan y habitualmente se encuentran a nivel alto.

El registro de control es parcialmente bidireccional. Cuando se escribe en los cuatro bits de menor peso (C0 - C3) lo que se hace es establecer el nivel lógico de los pines C2 de forma directa y C0, C1 y C3 de forma complementada. Los tres bits de mayor peso (C5-C7) no se utilizan.

En algunas tarjetas de interface paralelo, el bit C6 del registro de control influye en la configuración de modo que si C6 = 0 las líneas de datos se configuran como entradas y si C6 =1 las líneas de datos se configuran como salidas.

1.3.1.5 Lectura y escritura de datos

Se requiere hablar un poco sobre la lectura y escritura de datos a través del puerto para mandar señales para el movimiento de los motores.

Para poder leer y escribir datos por el puerto desde un programa de lenguaje C, por ejemplo mandar un “1” al pin 8 se escribe como sigue:

Inport (0x378);

Outport(0x378, 0x40);

Donde:

- Inport: Es la instrucción para leer el dato del puerto
- Outport: Es la instrucción para escribir en el puerto
- 0x378: Es la dirección del puerto
- 0x40: Indica que el valor escrito es en valor hexadecimal 40 → '0100 0000' en binario

Pero si además se quisiera mandar un “1” al bit 7(pin9) tendría que escribirse como sigue:

Outport(0x378,0x80);

Donde:

0x80 → '1000 0000'

Con esta instrucción se mandado un 1 al bit 7, sin embargo, se ha mandado un '0' al bit 6 lo que ha afectado el valor. Para corregir esto habría que mandar un '1' a los bits 6 y 7 como se muestra:

Outport(0x378,0xC0);

Donde:

0xC0 → '1100 0000'

Hasta ahorita únicamente se han mandado dos datos y se han hecho de forma manual, además de que se realizó el cambio de dato para lograr el objetivo de mandar un '1' al bit 6 y 7 en eventos diferentes, pero para evitar esto se recurre al proceso del enmascaramiento.

1.3.1.6 Enmascaramiento

Dado que el tipo de dato que se lee y escribe en el puerto paralelo es de 1 byte (8 bits) no es posible manipular un bit a la vez. Es por eso que se recurre al enmascaramiento que consiste en modificar uno o varios bits sin alterar los otros valores del dato.

Para el caso de querer mandar el valor “1” al bit 7 sin afectar los demás bits se hace el siguiente procedimiento.

Primeramente se lee lo que hay escrito en el puerto:

Inport (0x378);

El dato que se tiene es:

0x40 → '0100 000'

Ahora, se escribe el dato “1” en el bit deseado y los demás en “0”:

(0x80) → '1000 0000'

Se utiliza el operador “or” que es lo mismo que la suma ‘+’ y se realiza la operación con los dos datos:

0x40 →		0100 000
0x80 →	+	1000 0000
0xC0 ←		1100 0000

La instrucción en lenguaje C se escribe de la siguiente manera:

```
unsigned char Dir;
Dir= Inport (0x378);
Dir=Dir | 0x80 ;
Outport(0x378,Dir);
```

Donde:

- Dir: es la variable de tipo unsigned char, donde se almacena el valor de actual de puerto
- |: es el operador “or”

Ahora supóngase que se quiere poner en “0” el bit 6. Primeramente se lee lo que hay escrito en el puerto

```
Inport (0x378);
```

0xC0 → 1100 0000

Y se escribe el dato “0” en el bit deseado, y los demás en “1”:

(0xEF) ← 1011 1111

Se utiliza el operador “and” que es lo mismo que la multiplicación (x) y se realiza la operación, igual que en el proceso de la suma que es bit por bit:

0xC0 →		1100 0000
0xEF →	x	1011 1111
0x80 ←		1000 0000

En lenguaje C, esta operación se hace de la siguiente manera

```
unsigned char Dir;
Dir= Inport (0x378);
Dir=Dir & 0xEF ;
Outport(0x378,Dir);
```

Donde:

&: es el operador “and”

1.3.2 Acoplamiento óptico

Para tener comunicación entre la computadora y el mecanismo de una forma segura, se utiliza un aislamiento con un acoplamiento sin contacto físico para proteger la parte de control con la de potencia ya que el riesgo de un corto circuito sin el aislamiento dañaría gravemente a la computadora. A continuación se muestra el diagrama de bloques de la conexión entre los dispositivos.



Diagrama de bloques 1-1. Conexión de los dispositivos electrónicos

El optoacoplador, optoaislador o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor excitado mediante la luz emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un foto-receptor cuya conexión entre ambos es óptica.

Uno de estos optoacopladores es el 6N137 que está constituido de un LED (Light Emitter Diode) compuesto AlGaAs que emite a 850nm acoplado ópticamente a un foto-detector de alta velocidad de compuerta lógica.

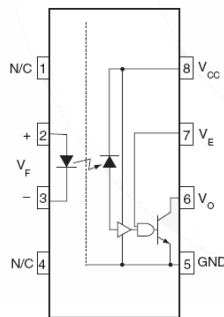


Figura 1-23. Arreglo del dispositivo 6N137

1.3.2.1 Acoplamiento óptico para control de dispositivos eléctricos

Para encender y apagar dispositivos eléctricos de corriente alterna con dispositivos electrónicos con aislamiento para poder ser controlado por la PC, se utiliza un circuito integrado especializado para dicha tarea.

El MOC 3010 es un optoacoplador compuesto por un LED emisor de infrarrojo y un TRIAC receptor diseñado para baja corriente de conmutación de corriente alterna de alto aislamiento eléctrico (7500 Vac pico) Figura 1-24.

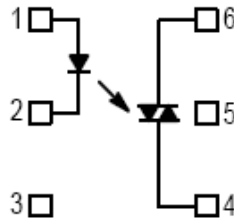


Figura 1-24. Arreglo electrónico del MOC 3010

1.3.2.2 Triac

Un TRIAC (Triode for Alternative Current o Triodo para Corriente Alterna) es un dispositivo semiconductor de la familia de los transistores, que es un interruptor capaz de conmutar la corriente alterna.

Su estructura interna se asemeja en cierto modo a la disposición que formarían dos SCR (Figura 1-25). (Silicon Controlled Rectifier o Rectificador Controlado de Silicio) en antiparalelo (Figura 1-26).

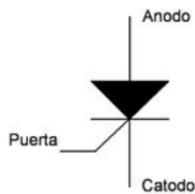


Figura 1-25. Símbolo del tiristor

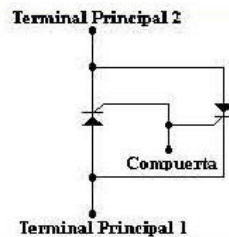


Figura 1-26. Arreglo del TRIAC con tiristores



Figura 1-27. Símbolo del TRIAC

El TRIAC posee tres electrodos: A1, A2 (en este caso pierden la denominación de ánodo y cátodo) y puerta (Figura 1-27). El disparo del TRIAC se realiza aplicando una corriente al electrodo puerta.

Este dispositivo se usará para el encendido y apagado de la máquina y de la herramienta de corte, en combinación con el dispositivo MOC 3010.

1.3.3 Fuente de alimentación

Para energizar todo el sistema con corriente directa es necesario una fuente de alimentación, a continuación se describe la que se ha utilizado en el proyecto.

Se ha ocupado una fuente conmutada de 12 Volts 8 amperes (Figura 1-28). Una fuente conmutada es un dispositivo electrónico que transforma energía eléctrica de C.A. a C.D. mediante transistores en conmutación a altas frecuencias (20-100 Kilociclos típicamente) entre corte (abiertos) y saturación (Cerrados).

La forma de onda cuadrada resultante es aplicada a transformadores con núcleo de ferrita para obtener uno o varios voltajes de salida de corriente alterna (CA) que luego son rectificadas (con diodos rápidos) y filtradas (Inductores y condensadores) para obtener los voltajes de salida de corriente continua (CC).

Las ventajas de este método incluyen menor tamaño y peso del núcleo, mayor eficiencia por lo tanto menor calentamiento. Las desventajas comparándolas con fuentes lineales es que son más complejas y generan ruido eléctrico de alta frecuencia que debe ser cuidadosamente minimizado para no causar interferencias a equipos próximos a estas fuentes.



Figura 1-28. Fuente conmutada

1.3.3.1 Regulador de voltaje

Debido a que es necesario utilizar 5 volts de corriente directa para el funcionamiento de algunos dispositivos electrónicos como el L297, y al no tener una fuente que provea de este valor de tensión, se hace uso de un circuito integrado regulador de voltaje como el 7805, cuya función consiste en entregar un voltaje de 5 V. a partir de 12 Volts para este caso, que se obtiene de la fuente conmutada. A continuación se muestra la Figura 1-29 el diagrama eléctrico para el funcionamiento del regulador de voltaje.

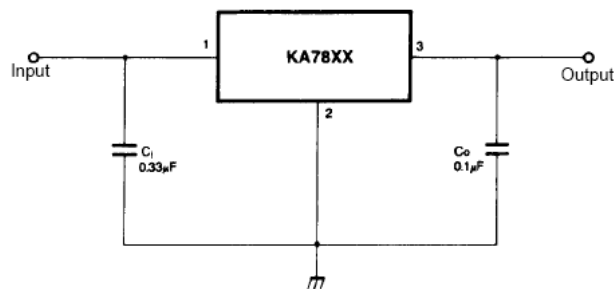


Figura 1-29. Diagrama eléctrico del regulador de voltaje

1. 4 Aspectos de programación

En esta sección se incluye información con aspectos teóricos relacionados al formato de estereolitografía (STL) y sobre el procesamiento de imagen necesarios para el desarrollo del software.

1.4.1 Formato de estereolitografía (stl)

Este es el tipo de formato que se utiliza para poder reconstruir la superficie de la pieza dibujada dada las características para representar la superficie del sólido.

El STL es un formato de archivo nativo de la estereolitografía CAD, software creado por 3D System. Este formato de archivo es el apoyo de muchos otros paquetes de software, es ampliamente utilizado para la creación de prototipos rápidos. Los archivos STL sólo describen la geometría de la superficie de un objeto tridimensional sin ningún tipo de representación del color, textura u otros atributos comunes modelo CAD [32][33]. El formato STL se especifica tanto en representaciones con códigos ASCII como en binario, siendo éste último el más común ya que son más compactos.

Un archivo STL forma la superficie de la pieza que se representa con una serie de “n” cantidad de triángulos no estructurados, y de cada triángulo se describe la ubicación espacial de sus vértices y del vector perpendicular a la superficie formada (ordenado por la regla de la mano derecha) [34].

1.4.1.1 Formato STL en código ASCII

Un archivo ASCII STL inicia con la frase:

solid name

Donde *name* es el nombre de la pieza y es una cadena opcional (tenga en cuenta que si el nombre se omite debe haber un espacio después de “*solid*”). El archivo continúa con un número “n” de bloques, que es el número de triángulos que forman la superficie, donde cada bloque describe tanto la ubicación del vector normal al plano como la de los vértices de solo un triángulo y ésta descripción se representa de la siguiente manera:

```
facet normal n i n j n k
outer loop lazo externo
  vertex v1 x e v1 y e v1 z e
  vertex v2 x e v2 y e v2 z e
  vertex v3 x e v3 y e v3 z e
endloop
endfacet
```

Donde: n_i - n_k y $v1x$ - $v3z$ son coeficientes de punto flotante 'e' es el signo de la notación exponencial (10x). El archivo termina con:

endsolid name

En la mayoría de los programas de diseño que tienen este tipo de formato es posible seleccionar el tipo de código: ASCII ó binario, además la calidad de la resolución depende de la cantidad de triángulos que conforman la superficie, y mientras mayor sea el número consecuentemente el tamaño del archivo aumenta.

1.4.1.2 Ejemplo

Un cubo de 10 x 10 x10 milímetros generado por el CAD al ser guardado en un archivo de estereolitografía (STL), la superficie del cubo es formado por una serie finita de triángulos y por cada cara existen dos triángulos, obteniendo un total de 12 triángulos, como se muestra en la Figura 1-30.

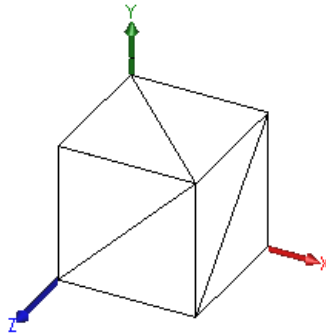


Figura 1-30. Superficie de un cubo en formato STL

De acuerdo al formato anteriormente descrito en archivo ASCII se obtiene la siguiente información:

```
solid cubo unitario
1 facet normal -1.000000e+000 0.000000e+000 0.000000e+000
  outer loop
    vertex 0.000000e+000 0.000000e+000 1.000000e+001
    vertex 0.000000e+000 1.000000e+001 1.000000e+001
    vertex 0.000000e+000 0.000000e+000 0.000000e+000
  endloop
endfacet
2 facet normal -1.000000e+000 0.000000e+000 0.000000e+000
  outer loop
    vertex 0.000000e+000 0.000000e+000 0.000000e+000
    vertex 0.000000e+000 1.000000e+001 1.000000e+001
    vertex 0.000000e+000 1.000000e+001 0.000000e+000
  endloop
endfacet
3 facet normal 0.000000e+000 -1.000000e+000 0.000000e+000
  outer loop
    vertex 1.000000e+001 0.000000e+000 1.000000e+001
    vertex 0.000000e+000 0.000000e+000 1.000000e+001
    vertex 1.000000e+001 0.000000e+000 0.000000e+000
  endloop
endfacet
4 facet normal 0.000000e+000 -1.000000e+000 0.000000e+000
  outer loop
    vertex 1.000000e+001 0.000000e+000 0.000000e+000
    vertex 0.000000e+000 0.000000e+000 1.000000e+001
    vertex 0.000000e+000 0.000000e+000 0.000000e+000
  endloop
```

```

endfacet
5 facet normal 1.000000e+000 0.000000e+000 0.000000e+000
  outer loop
    vertex 1.000000e+001 1.000000e+001 1.000000e+001
    vertex 1.000000e+001 0.000000e+000 1.000000e+001
    vertex 1.000000e+001 1.000000e+001 0.000000e+000
  endloop
endfacet
6 facet normal 1.000000e+000 0.000000e+000 0.000000e+000
  outer loop
    vertex 1.000000e+001 1.000000e+001 0.000000e+000
    vertex 1.000000e+001 0.000000e+000 1.000000e+001
    vertex 1.000000e+001 0.000000e+000 0.000000e+000
  endloop
endfacet
7 facet normal 0.000000e+000 1.000000e+000 0.000000e+000
  outer loop
    vertex 0.000000e+000 1.000000e+001 1.000000e+001
    vertex 1.000000e+001 1.000000e+001 1.000000e+001
    vertex 0.000000e+000 1.000000e+001 0.000000e+000
  endloop
endfacet
8 facet normal 0.000000e+000 1.000000e+000 0.000000e+000
  outer loop
    vertex 0.000000e+000 1.000000e+001 0.000000e+000
    vertex 1.000000e+001 1.000000e+001 1.000000e+001
    vertex 1.000000e+001 1.000000e+001 0.000000e+000
  endloop
endfacet
9 facet normal 0.000000e+000 0.000000e+000 1.000000e+000
  outer loop
    vertex 1.000000e+001 0.000000e+000 1.000000e+001
    vertex 1.000000e+001 1.000000e+001 1.000000e+001
    vertex 0.000000e+000 0.000000e+000 1.000000e+001
  endloop
endfacet
10 facet normal 0.000000e+000 0.000000e+000 1.000000e+000
  outer loop
    vertex 0.000000e+000 0.000000e+000 1.000000e+001
    vertex 1.000000e+001 1.000000e+001 1.000000e+001
    vertex 0.000000e+000 1.000000e+001 1.000000e+001
  endloop
endfacet
11 facet normal 0.000000e+000 0.000000e+000 -1.000000e+000
  outer loop
    vertex 1.000000e+001 1.000000e+001 0.000000e+000
    vertex 1.000000e+001 0.000000e+000 0.000000e+000
    vertex 0.000000e+000 1.000000e+001 0.000000e+000
  endloop
endfacet
12 facet normal 0.000000e+000 0.000000e+000 -1.000000e+000

```

```

outer loop
  vertex 0.000000e+000 1.000000e+001 0.000000e+000
  vertex 1.000000e+001 0.000000e+000 0.000000e+000
  vertex 0.000000e+000 0.000000e+000 0.000000e+000
endloop
endfacet
endsolid

```

De esta información se extraen los vértices de los triángulos que forman la superficie para la obtención de las coordenadas de los voxels que forman la superficie del sólido, explicado a detalle en el Capítulo 2:.

1.4.2 Imagen digital

En este apartado se dan aspectos teóricos de una imagen binaria, ya que una vez que se ha construido la superficie del sólido en el Capítulo 2: se procede al rebanado, donde cada rebanada pasa a ser una imagen binaria, la cual requiere ciertas operaciones de procesamiento de imagen para obtener un perfil con ciertas características para mandar la información a la máquina.

Una imagen digital es una representación bidimensional de una imagen utilizando bits (unos y ceros) Figura 1-31. Si la resolución de la imagen es estática o dinámica, se trata de un gráfico rasterizado y de un gráfico vectorial respectivamente. En general una imagen digital es de gráfico rasterizado.

Una imagen rasterizada, también llamada mapa de bits, imagen matricial o bitmap, es una estructura o fichero de datos que representa una rejilla rectangular de píxeles o puntos de color denominado “raster” que se puede visualizar en un monitor de computadora, papel u otro dispositivo de representación. Son caracterizadas por su altura y anchura (en píxeles) y por su profundidad de color (en bits por pixel) que determina el número de colores distintos que se pueden almacenar en cada pixel.

Una imagen vectorial está formada por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, posición, color, etc. Por ejemplo un círculo de color rojo quedaría definido por la posición de su centro, su radio, el grosor de línea y su color.

En infografía¹ 3D el concepto de una rejilla plana de píxeles se extiende a un espacio tridimensional formado por ladrillos cúbicos llamados "voxels". En este caso, existe una reja tridimensional con elementos (cubitos) que contienen la información del color. A pesar de que los voxels son un concepto potente para tratar cuerpos con formas complejas exigen mucha memoria para ser almacenados. En consecuencia, a la hora de producir imágenes en tres dimensiones se utilizan más a menudo imágenes vectoriales 3D (ver Voxel).

INSTITUTO POLITÉCNICO NACIONAL _____

¹ Información gráfica (representación visual de descripciones presentadas de manera gráfica normalmente figurativa)

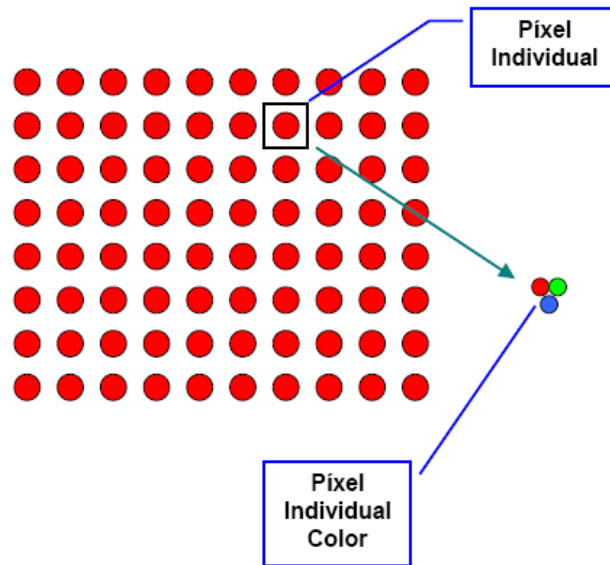


Figura 1-31. Imagen digital

1.4.2.1 Vecindad

Una imagen binaria es un arreglo matricial de píxeles donde cada píxel puede tener sólo dos valores 0 y 1, siendo los colores más comunes el blanco y el negro. Los píxeles tienen vecinos contiguos y como su nombre lo indica la vecindad-4, es cuando se contempla únicamente los píxeles ubicados arriba, abajo, a la izquierda y a la derecha del píxel, o los 4 que están en diagonal. Los de vecindad-8, cuando se contemplan los dos anteriores, como se muestra en la Figura 1-32

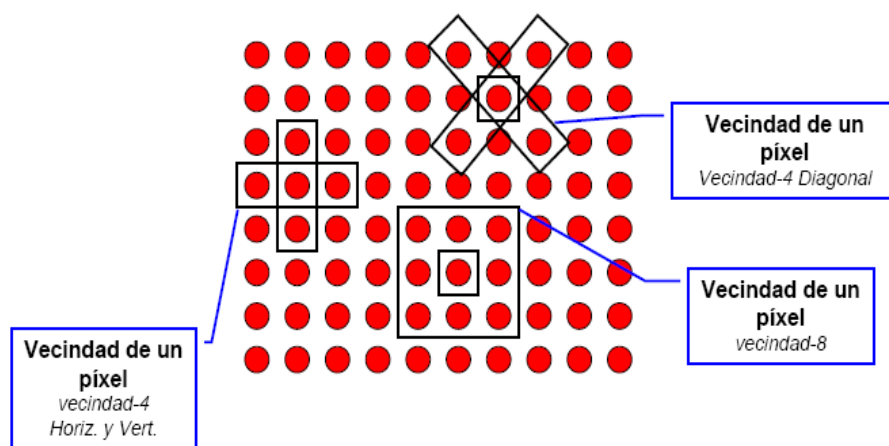


Figura 1-32. Vecindad de píxeles

Matemáticamente, la vecindad se representa de la siguiente manera:

Un píxel p con coordenadas (x,y)

1. Vecindad-4
 - a. $V4(p) = \{ (x+1,y), (x-1,y), (x,y+1), (x,y-1) \}$
 - b. $VD(p) = \{ (x+1,y+1), (x-1,y-1), (x-1,y+1), (x+1,y-1) \}$
2. Vecindad-8
 - a. $V8(p) = \{ V4(p) \cup VD(p) \}$

1.4.2.2 Tamaño de vecindad

El tamaño de vecindad es la cantidad de píxeles que se desplaza dentro de la vecindad, que en el caso anterior fue de 1 y en el que se muestra a continuación en la Figura 1-33 de vecindad dos.

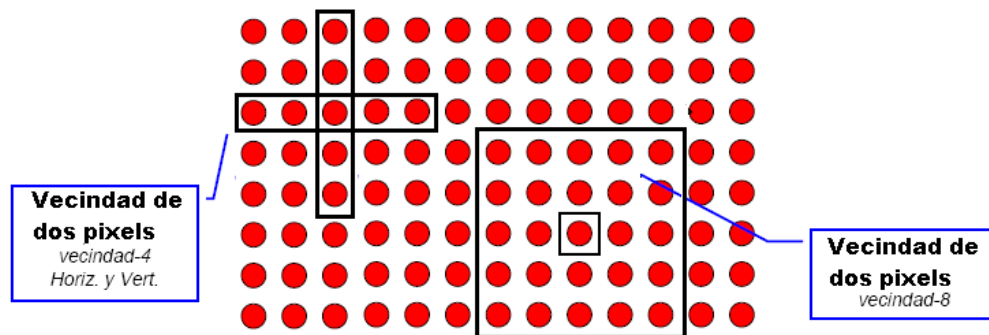


Figura 1-33. Vecindad de dos píxeles

1.4.2.3 Conectividad

Como su nombre lo indica es la forma en que un píxel se conecta con otro, y está muy ligado al concepto de vecindad específicamente al de un píxel (Figura 1-34).

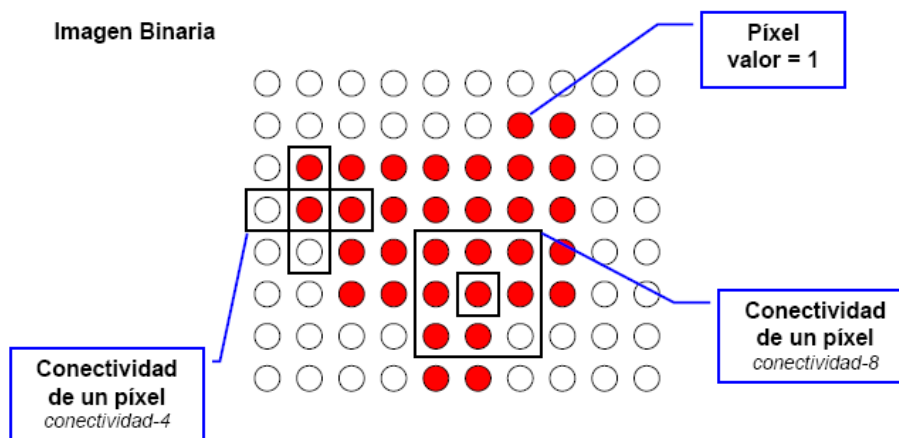


Figura 1-34. Conectividad entre píxeles

1.4.2.4 Operadores morfológicos

La morfología matemática es un método no lineal de procesar imágenes digitales basándose en la forma [35][36]. Los operadores morfológicos analizan la estructura geométrica de una imagen usando como sonda un patrón de ajuste que se denomina Elemento Estructural (EE). Desplazando el EE sobre la imagen, el operador analiza típicamente su posición en relación al primer plano y al fondo de la misma [37].

El EE es un conjunto de píxeles formando una pequeña imagen que determinará la estructura de la imagen sobre la que se aplicará la operación morfológica (Figura 1-35). Este será el responsable de la forma y el tamaño de los objetos que forman la imagen. Todo EE se expresa con respecto a un origen local (O) llamado elemento director [38].



Figura 1-35. Imágenes del EE sobre una imagen digital

El EE puede tener cualquier tamaño y forma (horizontal, vertical, cuadrado, etc.). Su centro se sitúa en cada píxel de la imagen original, aplicando la operación morfológica sobre los puntos situados bajo el EE (Figura 1-35)[39]. Las operaciones morfológicas se explican a continuación.

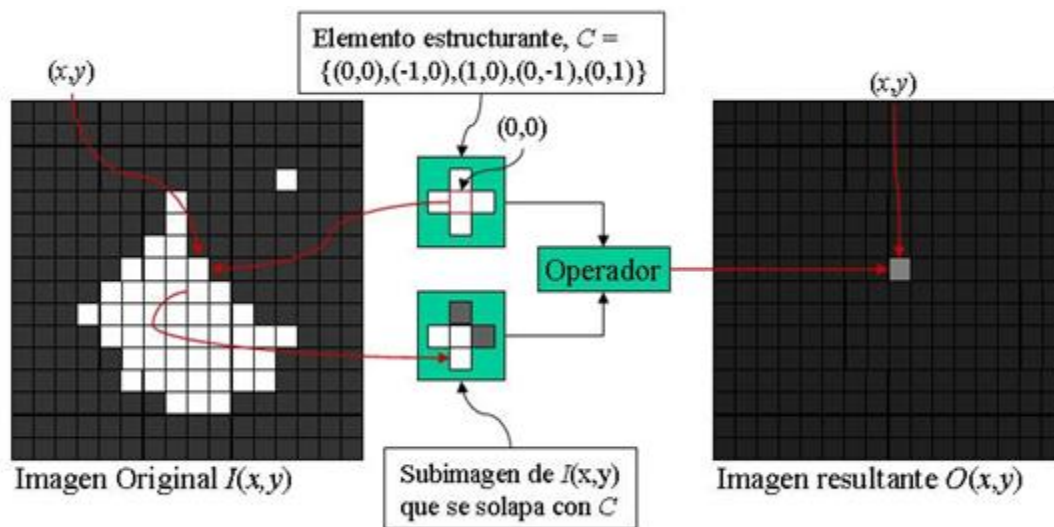


Figura 1-36. Descripción del elemento estructurante

1.4.2.5 Dilatación

Este operador es también conocido como 'relleno', 'expansión' o 'crecimiento', es usado para rellenar 'huecos' de tamaño igual o menor que el elemento estructural con la que se opera la dilatación [35][36][40].

Sobre cada píxel de la imagen se superpone el origen del elemento estructural. Si el píxel de la imagen no es cero, cada píxel que se encuentra en la estructura es añadido al resultado aplicando el operador 'or' (Figura 1-37).

Con la notación:

$$A \oplus B \dots\dots\dots(3)$$

Representando la dilatación de una imagen A por un elemento estructural B, se puede escribir:

$$C = A \oplus B = \bigcup_{b \in B} (A)_b \dots\dots\dots(4)$$

Donde $(A)_b$ representa la traslación de A por b. Intuitivamente para cada elemento no cero b_i de B, A es trasladado i,j y sumado a C usando el operador 'or'.

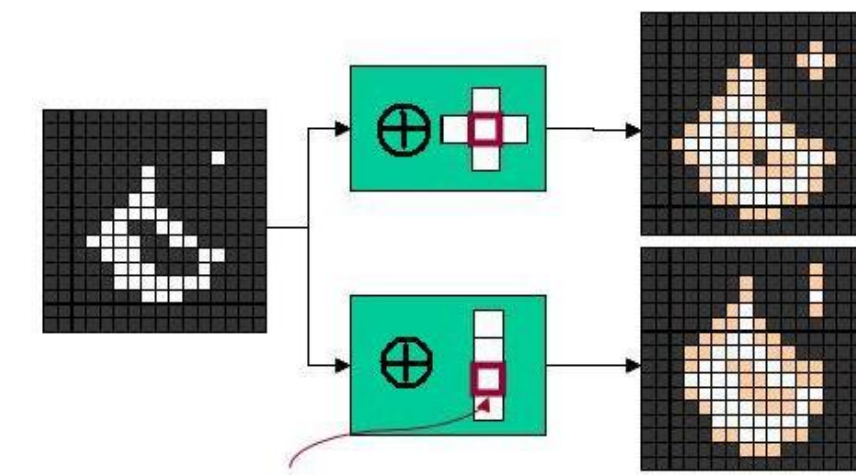


Figura 1-37. Proceso de dilatación en una imagen binaria

Matemáticamente se expresa de la siguiente manera:

$$O(x,y) = \Phi(Cx, A) \dots\dots\dots(5)$$

En algunos casos:

$$O(x,y) = \Phi((C^{\wedge})x, A) \dots\dots\dots(6)$$

La operación morfológica $\Phi(A)$ aplicada a una imagen X consiste en desplazar el EE por toda la imagen. Si se posiciona el EE sobre algún punto de la imagen, el píxel de la imagen sobre el que se superpone O se denomina píxel actual. El resultado de la relación entre el EE y la imagen X en la posición actual se almacena en el píxel actual de la imagen. Toda transformación Φ aplicada sobre una imagen X da como resultado otra imagen.

1.4.2.6 Erosión

La erosión es lo opuesto a la dilatación y sus efectos son de 'encogimiento', 'contracción', o 'reducción'. Puede ser utilizado para eliminar islas menores en tamaño que el elemento estructurante [38][39].

Sobre cada píxel de la imagen se superpone el origen del elemento estructurante. Si cada elemento no cero de dicho elemento está contenido en la imagen, entonces el píxel de salida es puesto a “1” como se muestra Figura 1-38[36].

Como notación:

$$A \otimes B \dots\dots\dots (7)$$

Como representación de la erosión de una imagen A por el elemento estructurante B, se puede definir:

$$C = A \otimes B \bigcap_{b \in B} (A)_{-b} \dots\dots\dots (8)$$

Donde $(A)_{-b}$ representa la traslación de A por b. B puede ser visto como una 'sonda' que se desliza a lo largo de toda la imagen A, testando la naturaleza espacial de A en cada punto. Si B trasladado i, j puede ser contenido en A (poniendo el origen de B en i, j), entonces i, j pertenece a la erosión de A por B[35][36].

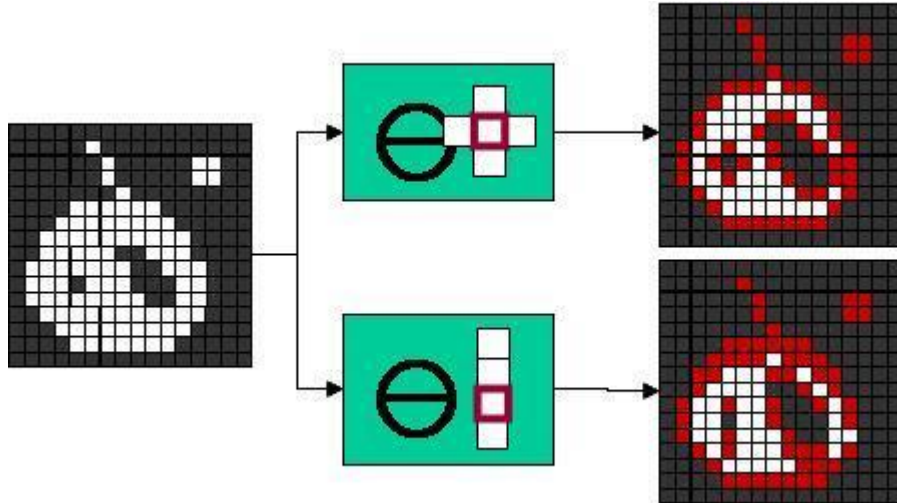


Figura 1-38. Proceso de erosión en una imagen binaria

1.4.2.7 Apertura

La apertura de A por B se obtiene por la erosión de A por B, seguida por la dilatación de la imagen resultante por B:

$$A \circ B = (A \ominus B) \oplus B \dots\dots\dots (9)$$

La apertura también está dada por:

$$A \circ B = \bigcup_{Bx \subseteq A} B_x \dots\dots\dots (10)$$

Lo que significa que es el lugar geométrico de las traslaciones del elemento estructurante B dentro de la imagen A (Figura 1-39). Los efectos de la apertura en una región de interés dada se pueden resumir en tres aspectos:

- Se eliminan islas de tamaño menor al elemento de estructura.
- Se eliminan picos o cabos más delgados que el elemento de estructura.
- Se rompen istmos cuya anchura sea menor al diámetro del elemento de estructura.

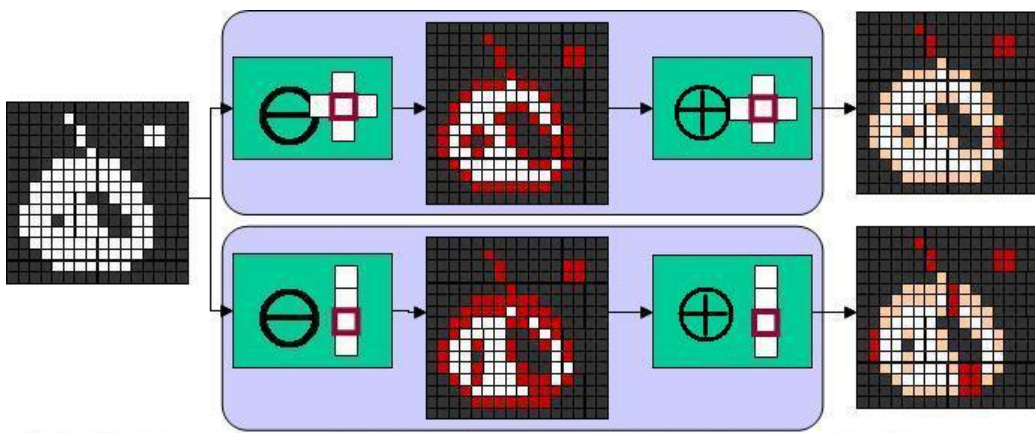


Figura 1-39. Proceso de apertura

Supóngase que se ha escrito una nota en un papel no muy absorbente y parece que están creciendo pequeñas raíces peludas por todas partes de la escritura. La apertura esencialmente elimina las pequeñas "rayitas" exteriores derramadas y restaura el texto. El efecto secundario es que redondea las cosas. Los bordes afilados comienzan a desaparecer [41].

1.4.2.8 Cierre

El cierre de A por B se obtiene por la dilatación de A por B , seguida por la erosión de la estructura resultante por B :

$$A \bullet B = (A \oplus B) \ominus B \dots \dots \dots (11)$$

El cierre también se puede obtener por

$$A \bullet B = (A^c \circ B^s)^c \dots \dots \dots (12)$$

Donde X^c denota el complemento de X respecto a E , esto es:

$$X^c = \{x \in E \mid x \notin X\} \dots \dots \dots (13)$$

Lo anterior significa que el cierre es el complemento del lugar geométrico de las traslaciones de la simetría del elemento estructurante fuera de la imagen A (Figura 1-40).

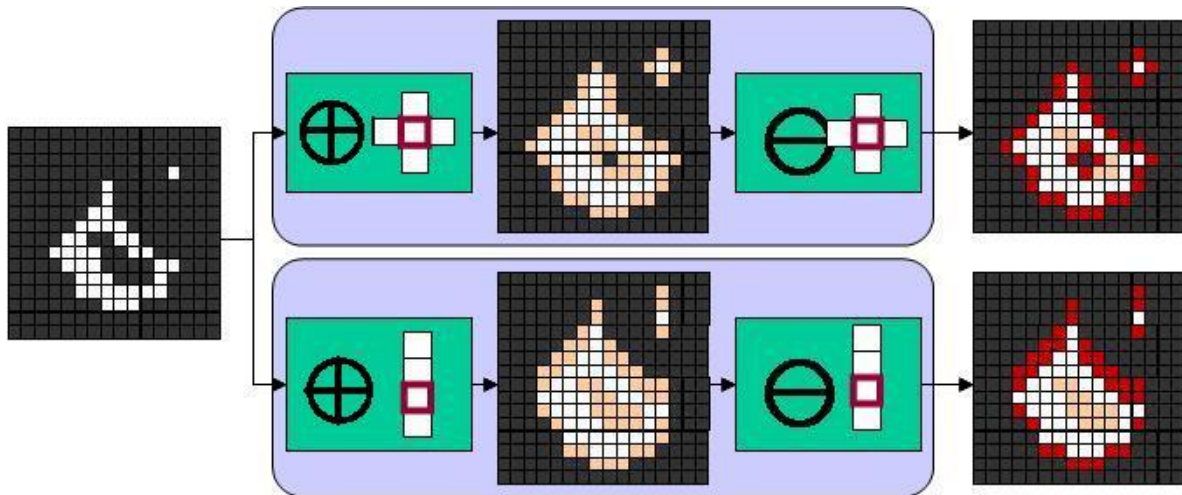


Figura 1-40. Proceso de cierre

1.4.3 Voxel

Dado que el CAM propuesto genera de manera virtual los puntos que forman la superficie de la pieza en tres dimensiones, es necesario hablar de los voxels.

El voxel (del inglés *volumetric pixel*, o *Volumetric Picture Element*) es la unidad cúbica que compone un objeto tridimensional. Constituye la unidad mínima procesable de una matriz tridimensional y es, por tanto el equivalente del píxel en un objeto en 3D.

Dado que en la imagen binaria un pixel tiene una vecindad, sin embargo en la terminología en 2D no define satisfactoriamente en 3D. La teoría que se ocupa de estas cuestiones topológicas se llama topología discreta en 3D. en el caso de los voxels también es posible este concepto y se explican a continuación [42].

El espacio discreto 3D es un conjunto de puntos de la red integral en el espacio euclidiano 3D definido por sus coordenadas cartesianas (x, y, z). Un voxel es la unidad de volumen cúbico centrada en el punto de la red integral. El valor del voxel se asigna son {0,1}, los voxels asignados con "1" son los "negros" que representan objetos opacos, y los asignados con "0" son los "blancos" que representan el fondo transparente.

Dos voxels son 26-adyacentes si comparten o bien un vértice, una arista o una cara (Figura 1-44), es decir que cada voxel tiene 26 voxels adyacentes de los cuales: ocho comparten un vértice con el voxel central (Figura 1-43), doce comparten una arista (Figura 1-42), y comparten seis una cara (Figura 1-41). En consecuencia, los voxels que comparten caras se definen como 6-adyacentes, y los voxels que comparten aristas y caras se definen como 18-adyacentes.

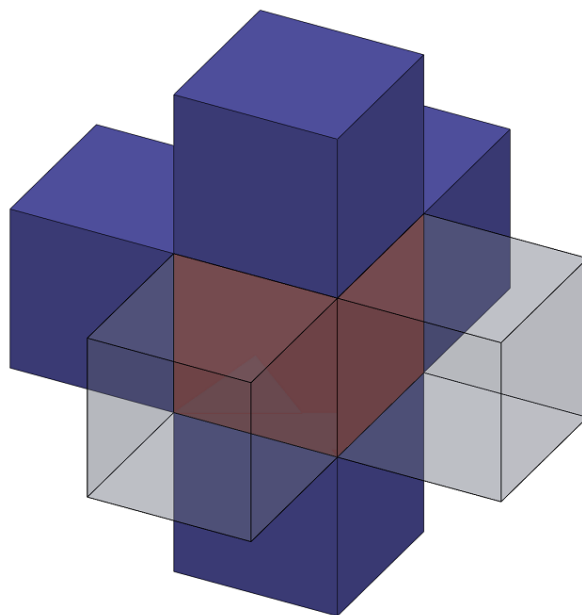


Figura 1-41. Vecindad de voxels por caras

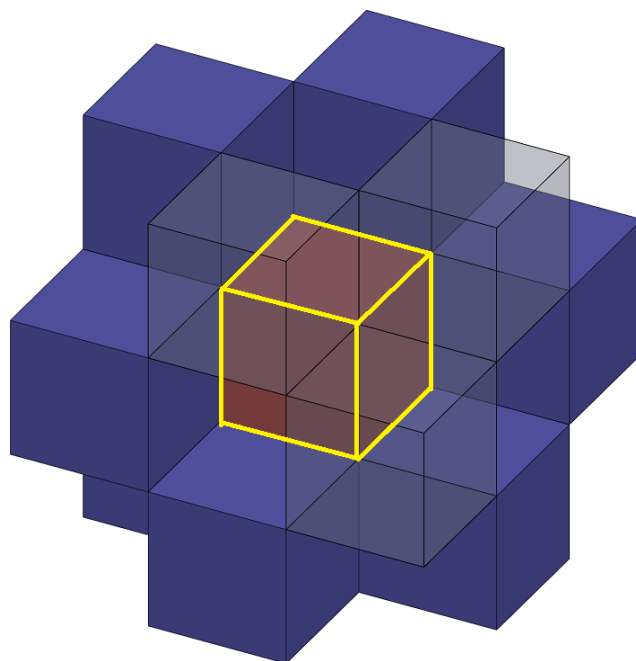


Figura 1-42. Vecindad de voxels por aristas

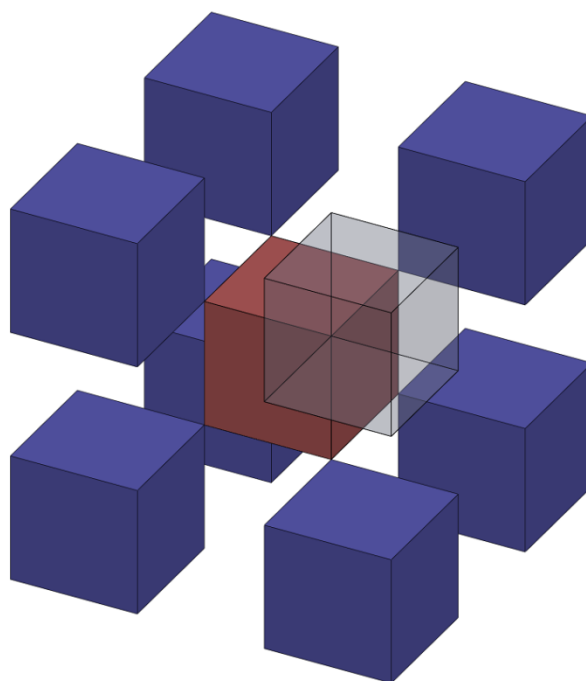


Figura 1-43. Vecindad de voxels por vértice

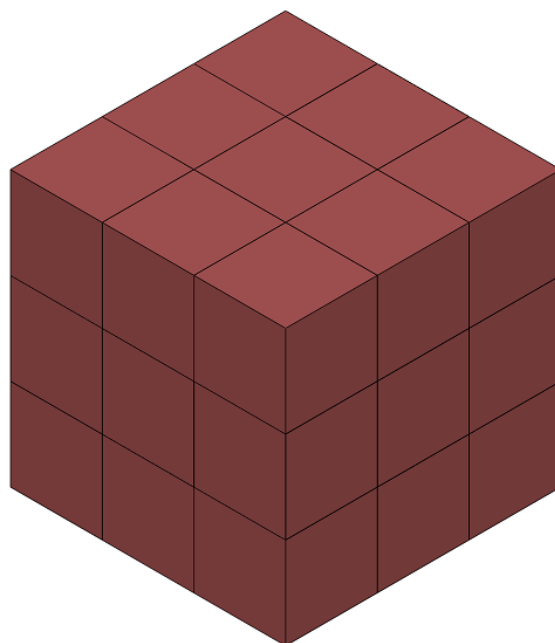


Figura 1-44. 26-Adyacente si comparten vértice, arista o cara

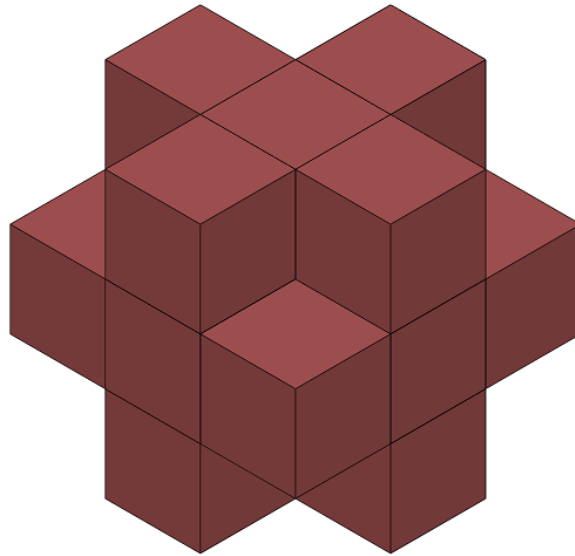


Figura 1-45. 18-Adyacente de un voxel

1.4.3.1 Conectividad

La conectividad entre un voxel y otro está dada por caras (6), por aristas (12) y por vértices (8), como se muestra en la Figura 1-46, existiendo en total 26 direcciones en la que se puede conectar un voxel.

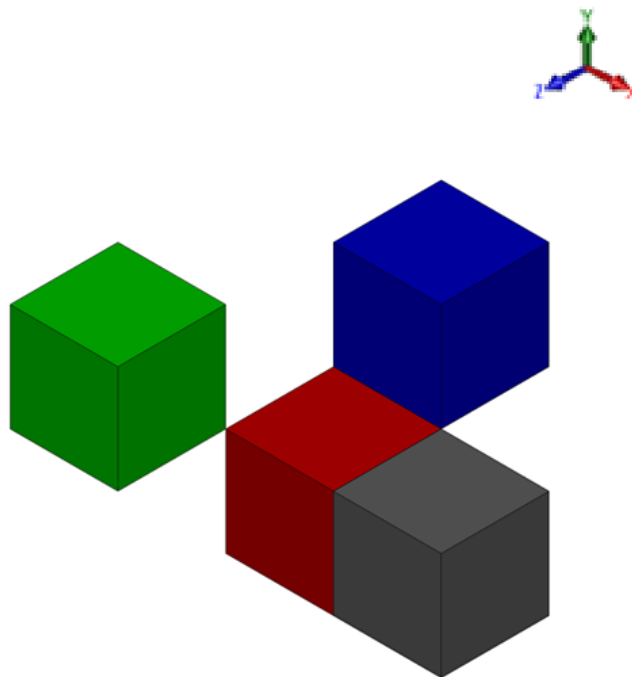


Figura 1-46. Conectividad entre voxels

Capítulo 2:

Diseño e

implementación

del CAM

Introducción

Este capítulo se ha dividido en dos secciones: la parte de diseño y la de implementación del CAM. En la primera sección se habla de la forma en que fueron desarrollados los algoritmos para la obtención de las coordenadas (x, y, z) de los voxels que integran las aristas y las áreas de los triángulos de los que está conformado la superficie virtual de la pieza dibujada en CAD. Así también, de cómo se rebana la pieza virtual y se logra obtener el perfil en cada rebanada con operaciones morfológicas en la imagen binaria, para que la herramienta de corte describa dicha trayectoria y así tener la pieza real dibujada.

En la sección de implementación, se explica la forma en la que se programaron los algoritmos para lograr el maquinado de la pieza.

2.1 Diseño

En este apartado se explica la forma de construir con voxels la superficie virtual de un sólido dibujado en CAD, a partir de la ubicación de los vértices de los triángulos que forman dicha superficie, cuya información se encuentra en el archivo STL.

Para lograr la construcción de la superficie, primeramente se extraen los valores de las coordenadas (x, y, z) de los tres vértices de un triángulo, después se calculan los puntos que forman las aristas y posteriormente se calcula la superficie del triángulo. Repitiendo esta operación para cada uno de los triángulos que forman la superficie del sólido, se obtienen todos los puntos que la forman.

Con la ubicación espacial de los puntos que forman la superficie del sólido, se procede a seccionar el sólido en rebanadas y de cada una obtener una imagen en dos dimensiones y obtener así un perfil, el cual será seguido para que la herramienta de corte describa la misma trayectoria, esto se hace dado que la estrategia de maquinado lo requiere así.

Para la construcción de la superficie del sólido y la obtención del perfil a seguir, se realizan los siguientes pasos a seguir como se muestra en la Figura 2-1

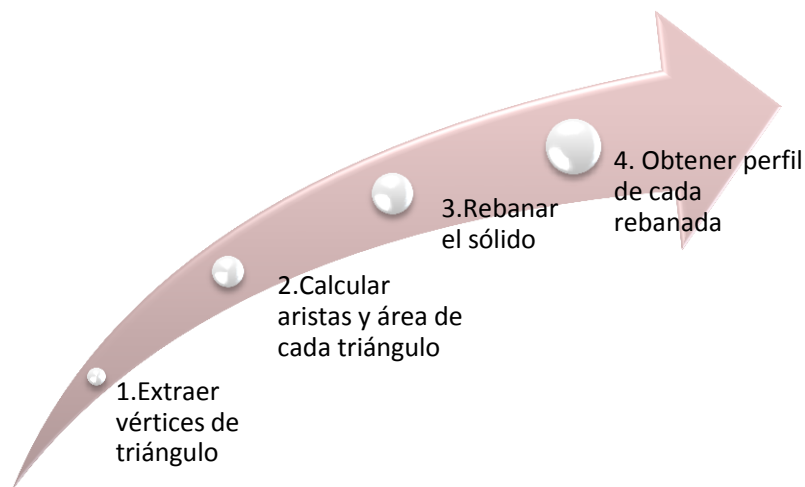


Figura 2-1. Diagrama del proceso para maquinar a partil del CAD

2.1.1 Construcción de la superficie del sólido

A continuación se describen los procesos que se desarrollan para obtener la ubicación espacial de los voxels que conforman la superficie de la pieza dibujada en CAD.

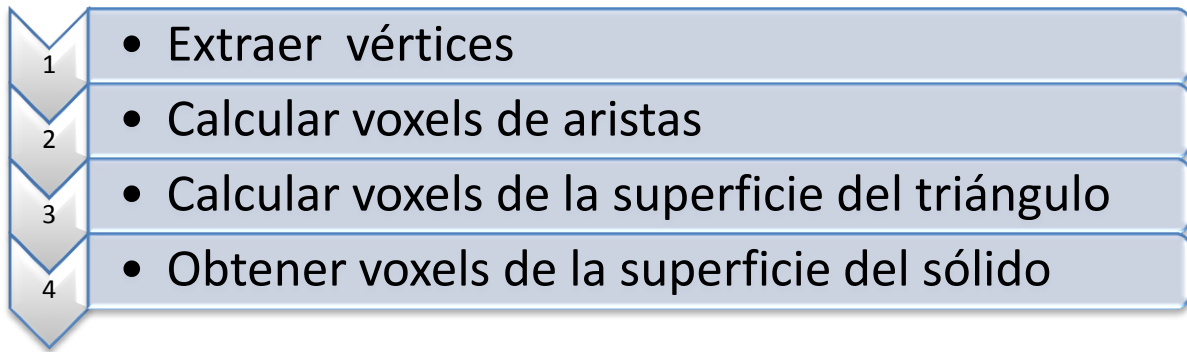


Figura 2-2. Procedimiento para la obtención de los voxels de la superficie del sólido

Como se puede observar, el primer paso consiste en extraer los vértices de los triángulos que conforman la superficie del sólido, para posteriormente calcular la ubicación de los voxels que integran las aristas e inmediatamente después calcular los voxels del área y finalmente realizar dicho procedimiento con todos los triángulos que forman la superficie para así obtener los voxels que forman la superficie.

2.1.1.1 Extracción de vértices

El procedimiento de extraer vértices consiste en buscar en el archivo STL los valores de las coordenadas (x,y,z) de los 3 vértices que forman un triángulo y capturarlos en variables, como lo muestra el Diagrama de bloques 2-1.

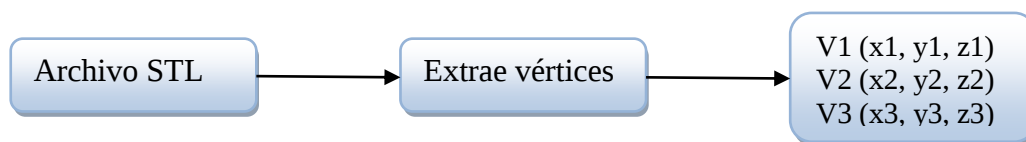


Diagrama de bloques 2-1. Extracción de vértices

Se muestra a continuación un bloque del listado de la descripción en la ubicación de la información obtenida del archivo CAD de un cubo de 10 milímetros por lado en formato STL, cuya representación del triángulo se representa en la Figura 2-3.

```

facet normal 1.000000e+000 0.000000e+000 0.000000e+000
  outer loop
    vertex 1.000000e+001 1.000000e+001 1.000000e+001
    vertex 1.000000e+001 0.000000e+000 1.000000e+001
    vertex 1.000000e+001 1.000000e+001 0.000000e+000
  endloop
  
```

endfacet

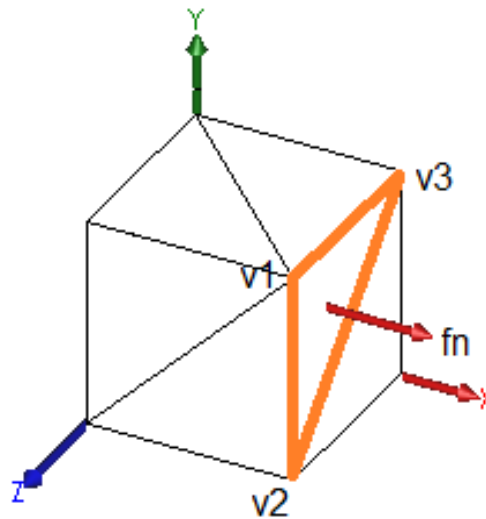


Figura 2-3. Ubicación del triángulo del bloque de descripción

El primer renglón del “facet normal”, se refiere a la ubicación en vectores del vector normal al triángulo, pero que no se requieren para el cálculo de los voxels, por lo que se ignora esta información.

El primer vértice V1 tiene los siguientes valores:

- 1.000000e+001

Donde:

- 1.000000: es el valor de la base
- e: representa la notación científica que equivale a 10^x
- +: representa el signo del exponente
- 001: es el valor del exponente

Por lo que el valor de la coordenada en x es:

- $1 \times 10^1 = 10$ [mm]

De la misma manera se calculan los valores en “y” y en “z”, lo que da por resultado que las coordenadas de V1 son: (10, 10, 10).

Para el caso del vértice V2 las coordenadas son: (10, 0, 10), y el vértice V3 (10, 10, 0)

De esta manera, se han extraído las coordenadas de los voxels que representan los vértices V1, V2, V3, del primer triángulo y están representadas en la Figura 2-4.

Con esta información extraída es suficiente para poder conocer la ubicación de todos los voxels que conforman las aristas y el área del primer triángulo. Y al repetir lo anteriormente descrito se obtienen los voxels de todos los triángulos y consecuentemente los que forman la superficie del cubo.

Es decir, que de un cubo donde únicamente se sabe la ubicación espacial de los vértices Figura 2-4, se obtenga un cubo con todos los voxels que integran su superficie como se muestra en la Figura 2-5.

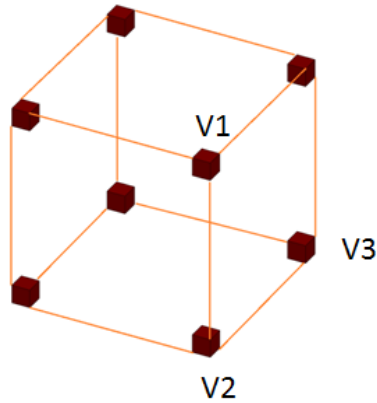


Figura 2-4. Ubicación de los voxels de los vértices de los triángulos

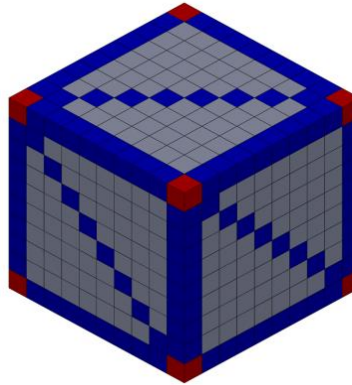


Figura 2-5. Ubicación de los voxels que conforman la superficie del cubo

2.1.1.2 Obtención de las aristas del triángulo

Una vez extraídos los vértices del triángulo, se procede a calcular las coordenadas que integran las aristas, el cual consta de las siguientes etapas:

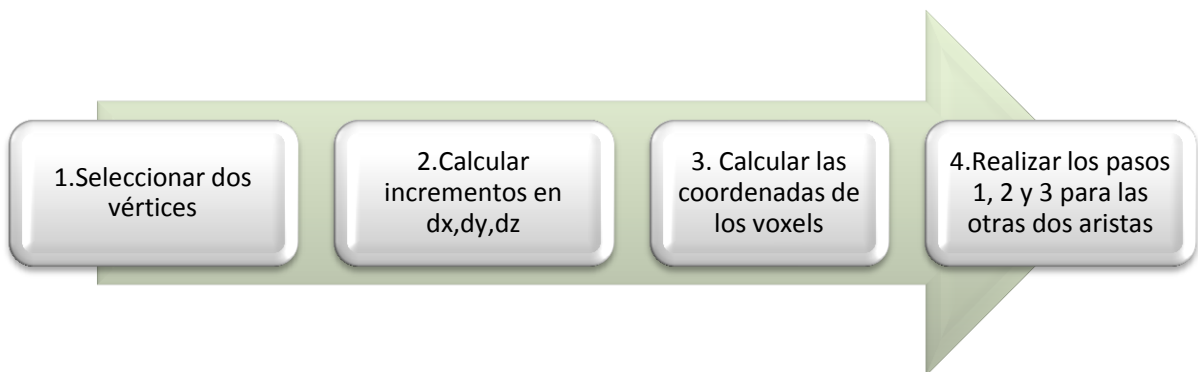


Figura 2-6. Proceso de obtención de los voxels de la superficie

Primeramente se seleccionan dos de los tres vértices que forman el triángulo para calcular los voxels que integran la arista, a partir de operaciones que calculan los incrementos en las tres coordenadas y obtener los valores enteros en (x,y,z), repitiendo el procedimiento para las otras dos aristas.

Dado que en la literatura sólo se encontraron algoritmos para construir figuras en 2 dimensiones, se tuvo la necesidad de desarrollar algoritmos basados en los de 2D, para la generación de líneas discretas en el espacio por medio de voxels.

Partiendo de la idea básica de unir dos puntos en el plano como se muestra en la Figura 2-7, se calcula la pendiente con la siguiente ecuación:

$$m = (y_2 - y_1)/(x_2 - x_1) = \Delta y / \Delta x \dots \dots \dots (11)$$

Posteriormente se realiza el cálculo para obtener los valores en “y” dados los de “x” o viceversa realizando el despeje correspondiente de la siguiente ecuación de la recta.

$$y = m x + b \dots \dots \dots (12)$$

Puesto que en una imagen digital la línea está formada por una sucesión finita de puntos discretos llamados pixeles y dado que la pendiente puede obtener valores fraccionales (en consecuencia los valores de “y”), los valores de las coordenadas tendrían valores fraccionales, por lo que es necesario determinar qué pixeles están más cerca de la trayectoria de la línea ideal, por lo que se tiene que realizar un redondeo para obtener un valor entero en ambas coordenadas (x, y) [43].

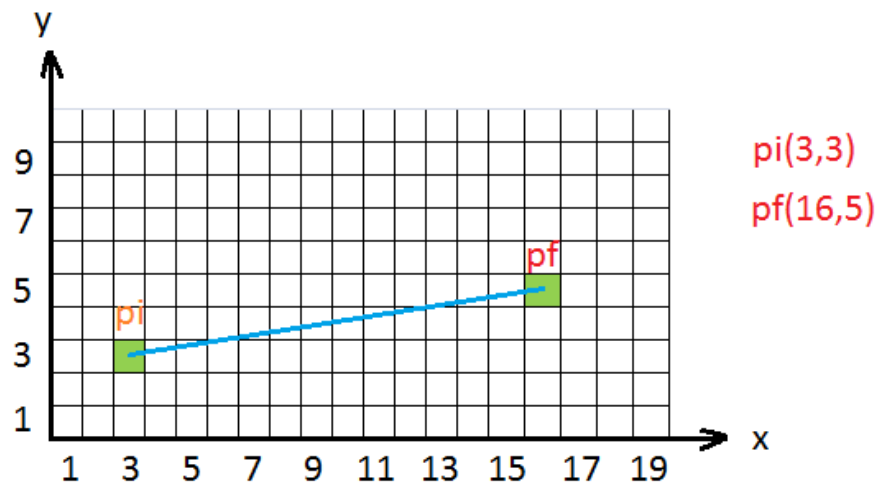


Figura 2-7. Ubicación de dos puntos en un plano cartesiano

Para obtener las coordenadas de los pixeles que forman la línea se utiliza la ecuación (11) y los siguientes criterios.

Si la pendiente $|m| < 1$, es decir que $\Delta x > \Delta y$, el incremento en “x” es unitario ($\Delta x = 1$ y $\Delta y = m$, dado que $m = \Delta y / \Delta x$) y los valores sucesivos de “y” se calculan con la siguiente ecuación:

$$x_{i+1} = x_i + 1 \dots\dots\dots (13)$$

$$y_{i+1} = y_i + m \dots\dots\dots (14)$$

Donde:

- m : es la pendiente calculada
- y_i : es el valor de la ordenada del pixel inicial
- x_i : es el valor de la abscisa del pixel inicial
- x_{i+1} : es el valor de la abscisa del siguiente pixel
- y_{i+1} : es el valor de la ordenada del siguiente pixel

Si la pendiente $|m| > 1$, es decir que $\Delta y > \Delta x$, se revierten las funciones de “x” y “y”, o sea, “y” tiene incrementos unitarios ($\Delta y = 1$ y $\Delta x = 1/m$ dado que $m = \Delta y / \Delta x$) y cada valor sucesivo de “x” se calcula con la siguiente ecuación:

$$y_{i+1} = y_k + m \dots\dots\dots (15)$$

$$x_{i+1} = x_k + 1/m \dots\dots\dots (16)$$

El subíndice $i+1$ toma valores enteros a partir de 1 y aumenta a razón de 1 hasta que el valor de “ y_{i+k} ” y el de “ x_{i+k} ” alcancen el valor de la coordenada final, mientras que el valor de “ y_i ” y “ x_i ”, corresponden al valor anterior calculado.

Sin embargo, es posible simplificar las ecuaciones para no estar eligiendo cual debe tener el incremento unitario. Para ello se selecciona el valor de la diferencia de mayor magnitud Δy ó Δx , a la que se le va a llamar “den”, obteniéndose las siguientes ecuaciones:

$$y_{i+1} = y_i + (\Delta y / den) = y_i + incy \dots\dots\dots (17)$$

$$x_{i+1} = x_i + (\Delta x / den) = x_i + incx \dots\dots\dots (18)$$

Ya que “ $incx$ ” e “ $incy$ ” pueden ser cualquier número real entre 0 y 1, los valores calculados de “ y_{i+1} ” y “ x_{i+1} ” deben redondearse al entero más cercano, si la fracción es < 0.5 , se redondea al valor entero inferior; y si es > 0.5 , al entero superior.

Entonces de esta manera, sustituyendo valores de las coordenadas entre los dos pixeles inicial y final: $P_i (7,5)$, $P_f (21,14)$, se tiene:

$$m_{12} = (y_2 - y_1) / (x_2 - x_1) = \Delta y / \Delta x \dots\dots\dots (19)$$

$$m_{12} = (5 - 3) / (16 - 3) = 2/13 = 0.15 \dots\dots\dots (20)$$

$$\Delta y = 2; \Delta x = 13; \text{ por lo tanto: } den = 13 \dots\dots\dots (21)$$

$$incy = 0.15 \dots\dots\dots (22)$$

$$incx = 1 \dots\dots\dots (23)$$

Sustituyendo los valores de los incrementos (22) (23) en las ecuaciones (17)(18)

$$y_{i+1} = 3 + 0.13 = 3.13 \dots\dots\dots (24)$$

$$x_{i+1} = 3 + 1 = 4 \dots\dots\dots (25)$$

Como es necesario redondear dado que la unidad más pequeña en una imagen es un pixel, no pueden existir fracciones de unidad, por lo que el valor obtenido se redondea al valor entero inmediato según el criterio de redondeo, siendo entonces las coordenadas del pixel $P_{i+1}(4,2)$, como se muestra en la Figura 2-8.

El cálculo del siguiente pixel continúa y sustituyendo los valores se tiene lo siguiente:

$$y_{i+2} = y_{i+1} + \text{inc}_y \dots\dots\dots (26)$$

$$x_{i+2} = x_{i+1} + \text{inc}_x \dots\dots\dots (27)$$

$$y_{i+2} = 3.13 + 0.13 = 3.26 \dots\dots\dots (28)$$

$$x_{i+2} = 4 + 1 = 5 \dots\dots\dots (29)$$

Y el pixel P_{i+2} tiene las coordenadas (5,3) (Figura 2-8) y el cálculo de las coordenadas continúa hasta llegar al pixel final Pf.

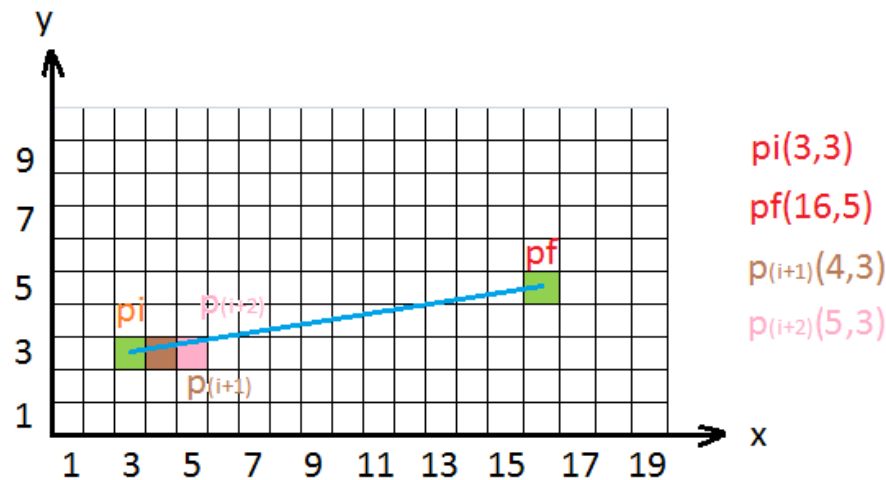


Figura 2-8. Ubicación del primer y segundo pixel calculado

Finalmente, la línea que se traza en una imagen binaria para unir dos puntos es como se muestra en la Figura 2-9.

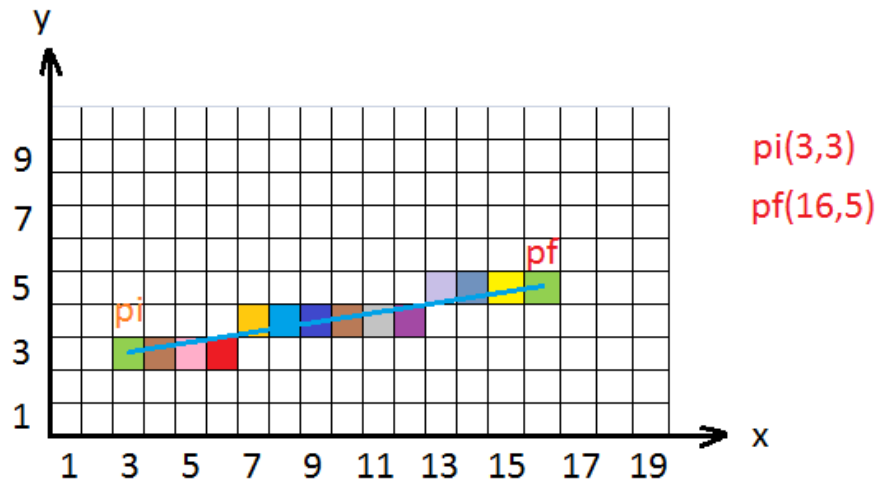


Figura 2-9. Línea formada con pixeles

Partiendo de la misma idea, se lleva el concepto de trazar líneas en dos dimensiones con pixeles para unir dos puntos, al de trazar líneas en el espacio con voxels para unir dos puntos en el espacio.

Supóngase que se han extraído los siguientes 3 vértices:

1. V1: P (5,4,7)
2. V2: Q (1,1,1)
3. V3: R (10,5,1)

La representación espacial de los voxels obtenidos es como se muestra en la Figura 2-10, donde se observan las aristas A1, A2 y A3, las cuales están formadas por los vértices V1 y V2, V2 y V3, V3 y V1 respectivamente.

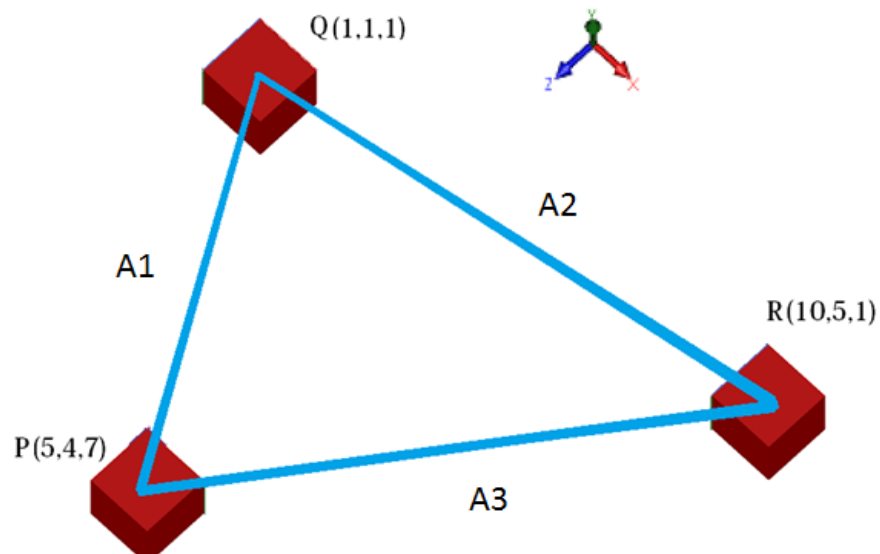


Figura 2-10. Ubicación espacial de los vértices extraídos

Para obtener la ubicación de los voxels que integran a cada una de ellas, se realizan cálculos de incrementos, los cuales se mencionan a continuación.

2.1.1.3 Cálculo de incrementos

Este procedimiento se realiza para conocer el incremento en cada coordenada, para obtener las coordenadas de los voxels que integran la arista. Para construir la arista A1 se procede a unir los vértices V1: P (5,4,7) y V2: Q (1,1,1), por lo que se realiza la diferencia de las coordenadas correspondientes y se elige el resultado de mayor magnitud, éste valor se le llama “den”.

$$\begin{aligned} dx &= |x_f - x_i| \\ dy &= |y_f - y_i| \\ dz &= |z_f - z_i| \\ \text{den} &\rightarrow \text{diferencia mayor} \end{aligned}$$

Sustituyendo los valores:

$$\begin{aligned} dx &= |1 - 5| = |-4| \\ dy &= |1 - 4| = |-3| \\ dz &= |1 - 7| = |-6| \\ \text{den} &\rightarrow dz = 6 \end{aligned}$$

Y para obtener el valor de los incrementos en x,y,z, se hace la división de las diferencias entre “den” como sigue:

$$\begin{aligned} \text{incx} &= (x_f - x_i) / \text{den} \\ \text{incy} &= (y_f - y_i) / \text{den} \\ \text{incz} &= (z_f - z_i) / \text{den} \end{aligned}$$

Sustituyendo valores se tiene:

$$\begin{aligned} \text{incx} &= (1 - 5) / 6 = -2/3 \\ \text{incy} &= (1 - 4) / 6 = -1/2 \\ \text{incz} &= (1 - 7) / 6 = -1 \end{aligned}$$

Esto quiere decir que el incremento en “z” es unitario y para “x” y “y” el incremento es fraccional.

Se procede con el cálculo de voxels entre dos vértices, para ello se toma el valor de la coordenada inicial que en este caso es V1: P (5,4,7). El total de voxels intermedios es igual al valor calculado de “den-1” dado que los incrementos son de 1 en 1. Y el total de puntos que conforman la arista es igual a “den+1”

Las coordenadas del voxel siguiente (P1) al voxel inicial (P0) se calculan como sigue:

Con el ejemplo de V1: P (5,4,7) y V2: Q (1,1,1):

$$\begin{aligned} P1(x) &= P0(x) + dx \\ P1(y) &= P0(y) + dy \\ P1(z) &= P0(z) + dz \end{aligned}$$

Sustituyendo los valores se tiene

$$P1(x) = 5 + (-0.66) = 4.34$$

$$P1(y) = 4 + (-0.50) = 3.5$$

$$P1(z) = 7 + (-1.00) = 6.0$$

Como los incrementos en 2 de las coordenadas es fraccional, y se está buscando la ubicación (x, y, z) de los voxels, no puede existir valor fracción, es decir (4.67, 3.5, 6), es necesario redondear al valor inmediato superior o inferior, si la fracción es mayor o igual a 0.5 o menor, respectivamente. Esto es en el supuesto de que el voxel tiene como medida 1 milímetro por lado.

$$P1(x) = 4.67 \rightarrow 4$$

$$P1(y) = 3.5 \rightarrow 4$$

$$P1(z) = 6.0 \rightarrow 6$$

Por lo que las coordenadas del voxel siguiente al vértice V1, es decir V1+1 ó P1 (4, 4,6), como se muestra en la Figura 2-11

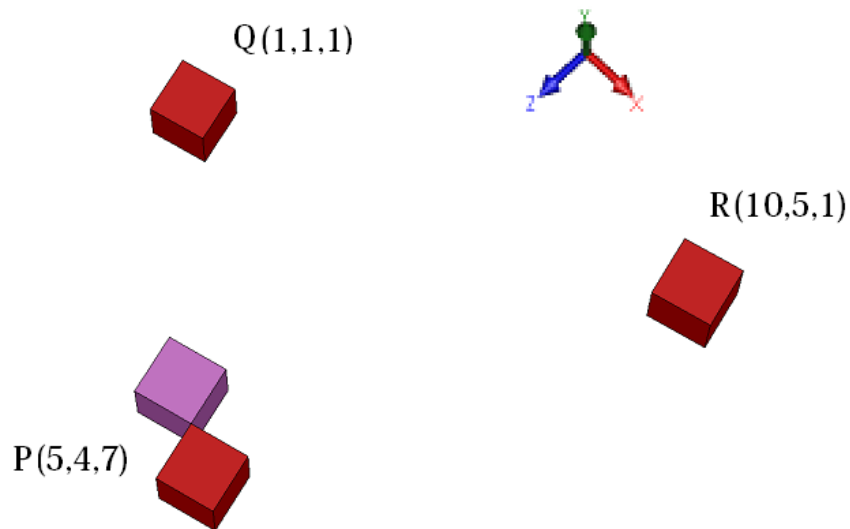


Figura 2-11. Ubicación espacial del voxel (V1+1)

Para el cálculo de las coordenadas de los siguientes voxels se toman en cuenta los valores reales obtenidos, no los valores redondeados como se muestra a continuación:

$$P2(x) = P1(x) + dx$$

$$P2(y) = P1(y) + dy$$

$$P2(z) = P1(z) + dz$$

Sustituyendo

$$P2(x) = 4.67 + (-0.33) = 4.34$$

$$P2(y) = 3.5 + (-0.50) = 3.0$$

$$P2(z) = 6.0 + (-1.00) = 5.0$$

Las coordenadas del voxel P2 son: (4, 3, 5). El procedimiento se repite sucesivamente, se calculan los valores reales de las coordenadas de los voxels siguientes, pero obteniendo un valor entero de acuerdo al criterio de redondeo, hasta terminar de unir los dos voxels. Los valores obtenidos se enlistan en la Tabla 2-12

Tabla 2-1. Coordenadas de los voxels de la arista A1

	x	y	z
V1	5	4	7
P1	5	4	6
P2	4	3	5
P3	4	3	4
P4	4	2	3
P5	3	2	2
V2	3	1	1

La representación gráfica de los voxels con los valores obtenidos y que forman la arista se muestra en Figura 2-12.

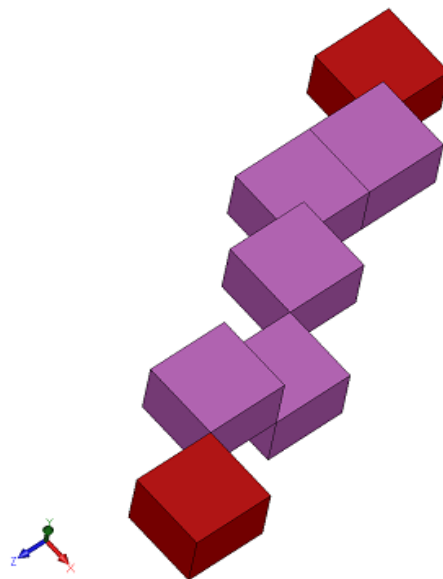


Figura 2-12. Construcción de voxels de una arista

Se realiza el mismo procedimiento para obtener las coordenadas de los voxels de las aristas A2 y A3, cuyos valores se muestran en la Tabla 2-2 y en la Tabla 2-3 respectivamente.

Tabla 2-2. Coordenadas de los voxels de la arista A2

	x	y	z
V2	1	1	1
P1	2	1	1
P2	3	2	1
P3	4	2	1
P4	5	3	1
P5	6	3	1
P6	7	4	1
P7	8	4	1
P8	9	5	1
V2	10	5	1

Tabla 2-3. Coordenadas de los voxels de la arista A3

	x	y	z
V3	10	5	1
P1	9	5	2
P2	8	5	3
P3	8	5	4
P4	7	4	5
P5	6	4	6
V1	5	4	7

Los valores de las coordenadas de los voxels obtenidos y enlistados en las tablas anteriores se representan de acuerdo con el color respectivo de cada tabla con su arista que se muestra en la Figura 2-13.

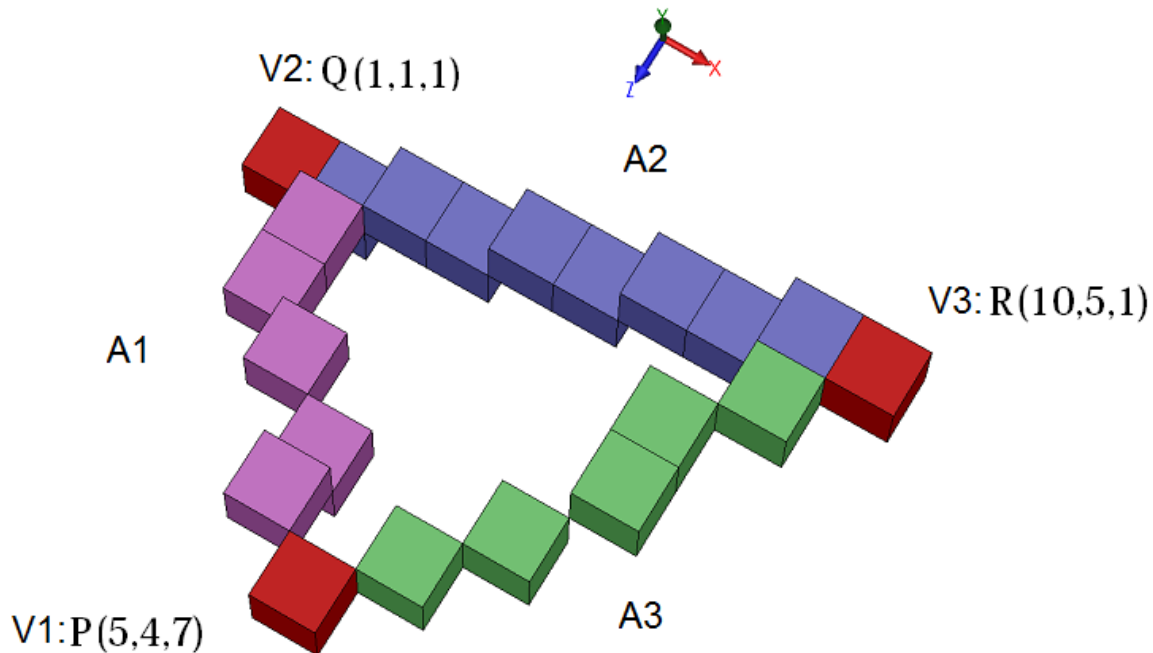


Figura 2-13. Construcción de las aristas del triángulo

2.1.1.4 Obtención de voxels del área del triángulo

Una vez conocidos los valores de los voxels que integran las aristas se procede al cálculo de obtener los voxels que integran el área, para ello se utiliza un método desarrollado, llamado método radial.

El método radial consiste en seleccionar la arista de mayor tamaño y el vértice opuesto a éste, posteriormente se une el vértice opuesto con cada uno de los voxels que integran la arista mayor.

Para seleccionar la arista mayor, se comparan los valores calculados en la variable “den” de cada arista, y se selecciona la de mayor magnitud, en consecuencia el vértice opuesto es aquel que no forma parte de la arista mayor. Para este caso, con el hecho de comparar la cantidad de voxels que integran cada arista en las tablas respectivas: 2-1, 2-2, 2-3, se observa que la arista A2 es la más grande, por lo que el vértice opuesto sería el vértice V1.

A continuación se une el vértice opuesto (VO) con cada uno de los voxels que integran la arista mayor (AM), como se muestra en la Figura 2-14, es decir, trazar una línea del VO al voxel “a” después con el “b” y así sucesivamente hasta llegar a la “h”.

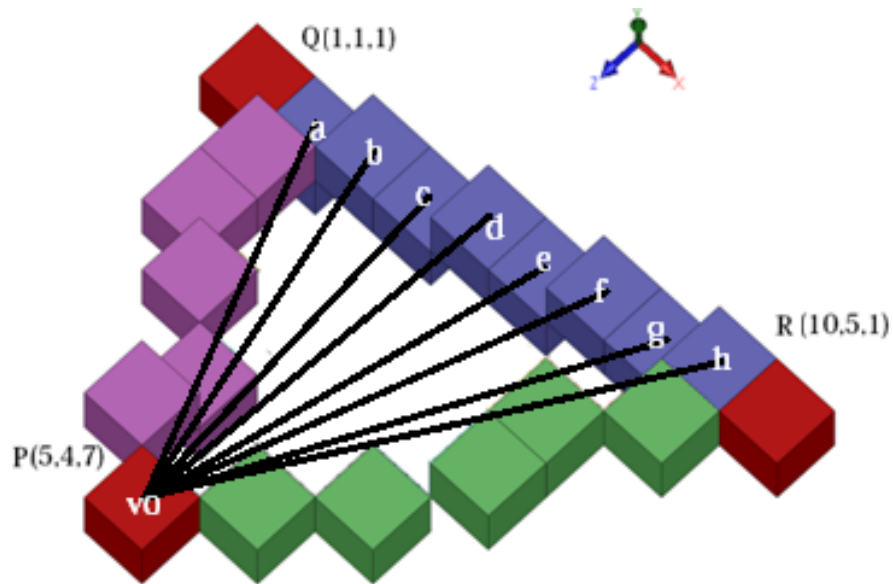


Figura 2-14. Método radial para obtención de voxels del área

Para unir el VO con los voxels que integran la AM, se retoman los procesos para la obtención de las aristas: calcular incrementos dx , dy , dz ; obtener la variable “den” y realizar el cálculo de los voxels intermedios y finalmente calcular los voxels que unen el VO al voxel “a” como se muestra en la Figura 2-15. En la Tabla 2-4, se muestran las coordenadas de los voxels de la línea que une el vértice opuesto y el voxel “a”.

Tabla 2-4. Coordenadas de la línea de VO al voxel “a”

	x	y	z
VO	5	4	7
1	5	4	6
2	4	3	5
3	4	3	4
4	3	2	3
5	3	2	2
a	2	1	1

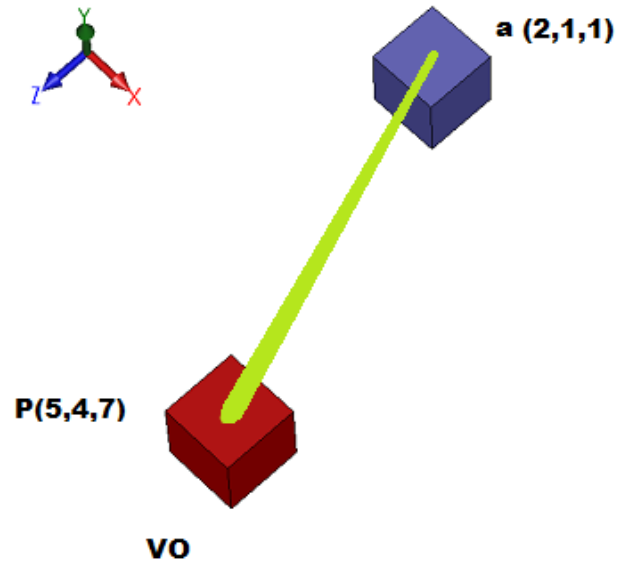


Figura 2-15. Imagen de los voxels a unirse por medio de una línea

La representación de los voxels calculados de la línea que une el VO y el voxel “a” se muestran a continuación en la Figura 2-16.

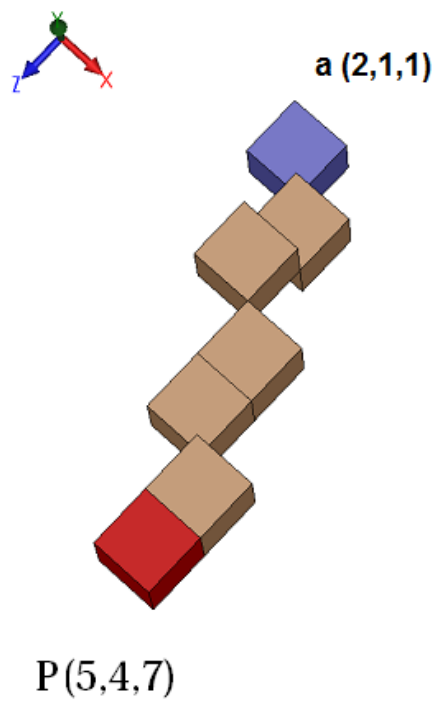


Figura 2-16. Imagen de la línea que une el vértice opuesto con el voxel "a"

Entonces con la obtención de los voxels de las aristas y de la primera línea del área la representación espacial se vería como se muestra en la Figura 2-17, donde se aprecia un voxel de color negro el cual indica que ha ocurrido una redundancia, es decir, que ese voxel se repite y se encuentra en la Tabla 2-1 y en la Tabla2-4.

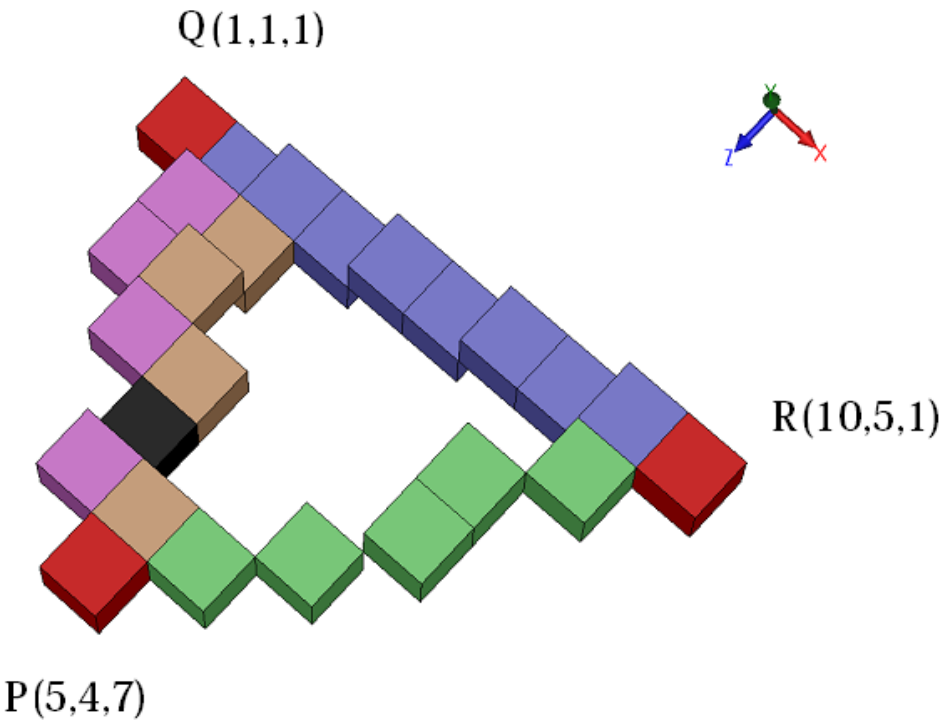


Figura 2-17. Muestra de los voxels calculados hasta el momento

El cálculo de los voxels de las líneas que integran el área del triángulo se realiza hasta terminar con todas las líneas. A continuación se muestran las tablas: 2-5, 2-6, 2-7, 2-8, 2-9, 2-10, 2-11, las coordenadas de los voxels que integran las líneas trazadas en la Figura 2-14.

Tabla 2-5. Cálculo de voxels entre el VO y "b"

	x	y	z
VO	5	4	7
1	5	4	6
2	4	3	5
3	4	3	4
4	4	3	3
5	3	2	2
b	3	2	1

Tabla 2-6. Cálculo de voxels entre el VO y "c"

	x	y	z
VO	5	4	7
1	5	4	6
2	5	3	5
3	5	3	4
4	4	3	3
5	4	2	2
c	4	2	1

Tabla 2-7. Cálculo de voxels entre el VO y "d"

	x	y	z
VO	5	4	7
1	5	4	6
2	5	4	5
3	5	4	4
4	5	3	3
5	5	3	2
d	5	3	1

Tabla 2-8. Cálculo de voxels entre el VO y "f"

	x	y	z
VO	5	4	7
1	5	4	6
2	6	4	5
3	6	4	4
4	6	4	3
5	7	4	2
f	7	4	1

Tabla 2-9. Cálculo de voxels entre el VO y "h"

	x	y	z
VO	5	4	7
1	6	4	6
2	6	4	5
3	7	4	4
4	8	3	3
5	8	3	2
h	9	3	1

Tabla 2-10. Cálculo de voxels entre el VO y "e"

	x	y	z
VO	5	4	7
1	5	4	6
2	5	4	5
3	6	4	4
4	6	3	3
5	6	3	2
e	6	3	1

Tabla 2-11. Cálculo de voxels entre el VO y "g"

	x	y	z
VO	5	4	7
1	6	4	6
2	6	4	5
3	7	4	4
4	7	4	3
5	8	4	2
g	8	4	1

En la Figura 2-18 se muestran los voxels que fueron calculados cuyas coordenadas se enlistaron en las tablas anteriormente mostradas. Los voxels tienen el mismo color de la tabla donde se encuentran sus coordenadas.

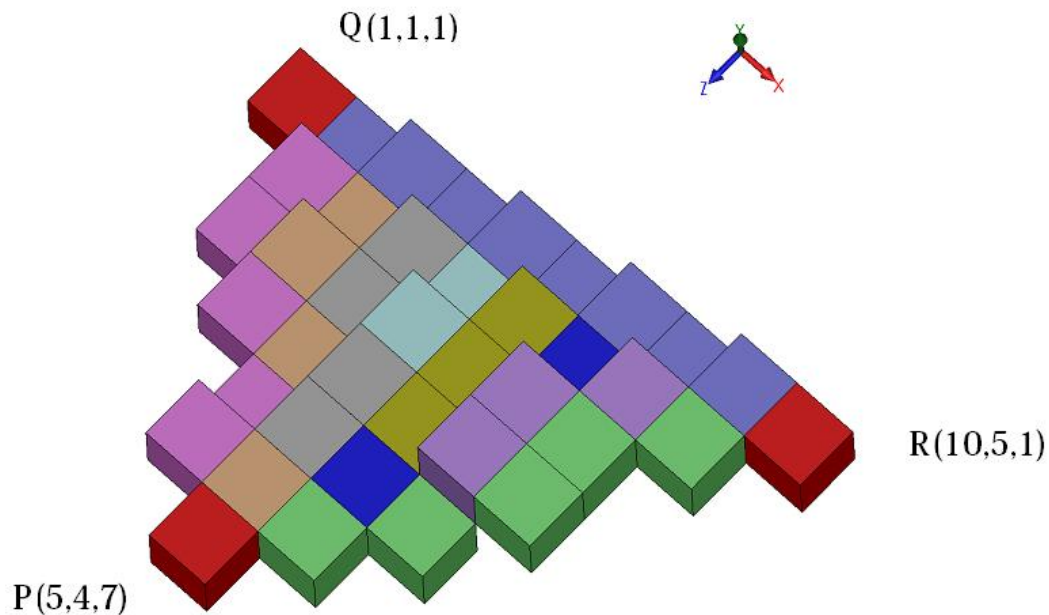


Figura 2-18. Ubicación de los voxels del área del triángulo

2.1.1.5 Redundancias y huecos

El cálculo de voxels que conforman tanto las aristas como los del área de los triángulos que representan la superficie total de la pieza, presenta dos situaciones no deseables para la conformación del contorno de las rebanadas que se generarán para el maquinado de la pieza que son las redundancias y los huecos, como se observa en la Figura 2-19.

La redundancia consiste en que dos o más voxels tengan las mismas coordenadas (x, y, z). Estas redundancias generan que la superficie conformada idealmente por un número determinado de voxels tenga un número excedente y que son totalmente innecesarios, puesto que se desean tener los voxels suficientes que cubran totalmente o en el peor de los casos la mayor parte de la superficie del triángulo, y para poder lograr esto se requiere eliminar esas redundancias. Además de que al calcular los voxels de las aristas, éstos pertenecen a uno o varios triángulos.

Para poder realizar la eliminación de las redundancias, se hace una comparación de las coordenadas de todos los voxels calculados y al existir voxels con las mismas coordenadas se eliminan hasta solo tener un voxel con esas coordenadas.

Los huecos se presentan cuando un espacio no es ocupado por un voxel calculado, que por cuestiones de redondeo no pasan por ese espacio y por lo tanto no se tiene registro del mismo. Esto ocasiona problemas pues al no estar completa la información del área al momento de obtener la rebanada del contorno, y si ésta pasa por donde exista algún hueco la trayectoria no será cerrada.

Sin embargo dicha falla se corrige en la imagen en 2D para llenar ese hueco con el pixel correspondiente y de esa manera obtener un contorno cerrado, esto se explica a detalle en la sección “Rellenado de huecos” en la pagina 77.

En la Figura 2-18, el conteo físico de los voxels que forman las aristas y del área del triángulo es de 39. Sin embargo, la cantidad de puntos calculados y enlistados en las tablas: 2-1 hasta 2-11, se obtuvieron un total de 61 voxels incluyendo aristas, esto quiere decir que existieron 22 redundancias.

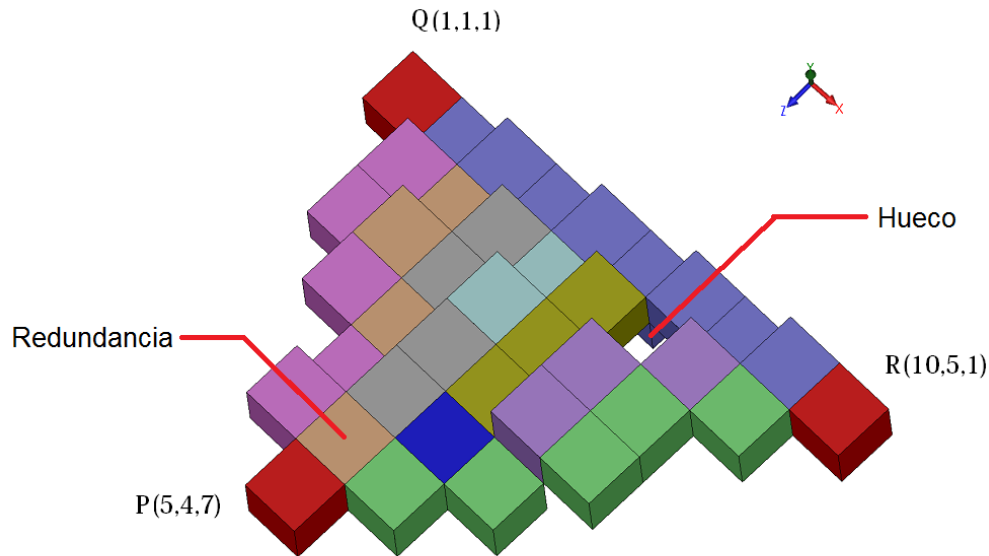


Figura 2-19. Duplicidad y huecos en el cálculo de obtención de voxels

2.1.1.6 Obtención de los voxels que integran la superficie total del sólido

Para obtener todos los voxels que integran la superficie de una pieza dibujada (Figura 2-20), se repiten los procesos anteriormente descritos, para la obtención de los voxels de un triángulo para todos los triángulos que la conforman (Figura 2-21).

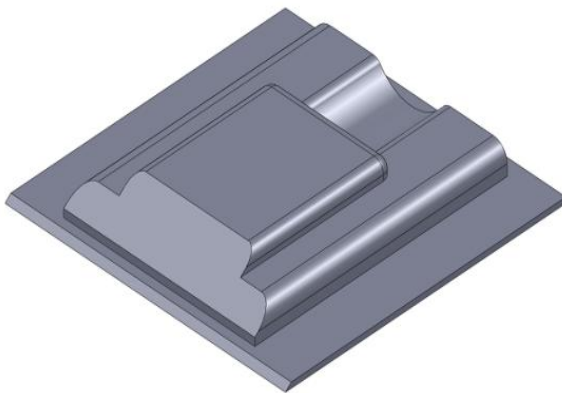


Figura 2-20. Pieza dibujada en CAD

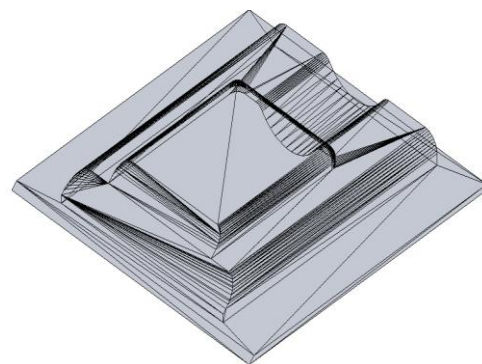


Figura 2-21. Pieza dibujada en CAD en formato STL

De la misma manera en que se obtuvieron tablas donde se alojaban los valores de las coordenadas de los voxels, al obtener todos los voxels que integran la superficie de la pieza, se obtiene una gran lista de coordenadas (x, y, z), sin embargo, solo para ejemplificar supóngase que en la Tabla 2-12 se enlistan las coordenadas de todos los voxels.

Tabla 2-12. Lista de coordenadas de voxels

X	Y	Z
2	3	4
1	2	3
2	1	2
1	2	3
1	1	4
3	2	2
2	3	1
1	2	2
2	3	4
1	3	3
2	1	1
2	3	4

2.1.1.7 Rebanado del sólido

Una vez conocidos las ubicaciones de los voxels que forman la superficie del sólido, lo que procede es seccionarlo en varias rebanadas para que de cada una se obtenga una imagen en dos dimensiones del contorno, es decir, que los voxels pasan a ser pixeles.

Dicha imagen será tratada con una serie de operaciones de imagen para mejorar las características del contorno, y así obtener un perfil, el cual será seguido y a su vez mandará la información correspondiente a la interfaz para el movimiento del mecanismo logrando que la herramienta describa dicha trayectoria y machine el modelo.

El rebanado de la pieza consiste en ordenar de menor a mayor los valores de la coordenada en “x”, y posteriormente en la coordenada en “y” para tener algo parecido como la Tabla 2-13. Siendo cada conjunto de coordenadas encerradas en el rectángulo una rebanada de las que se conforma el sólido.

Tabla 2-13. Ordenación de la coordenada X y Y

X	Y	Z
1	5	5
1	5	6
1	6	6
1	6	5
2	3	2
2	3	3
2	3	1
2	3	1
3	4	1
3	4	1
3	5	2
3	5	3

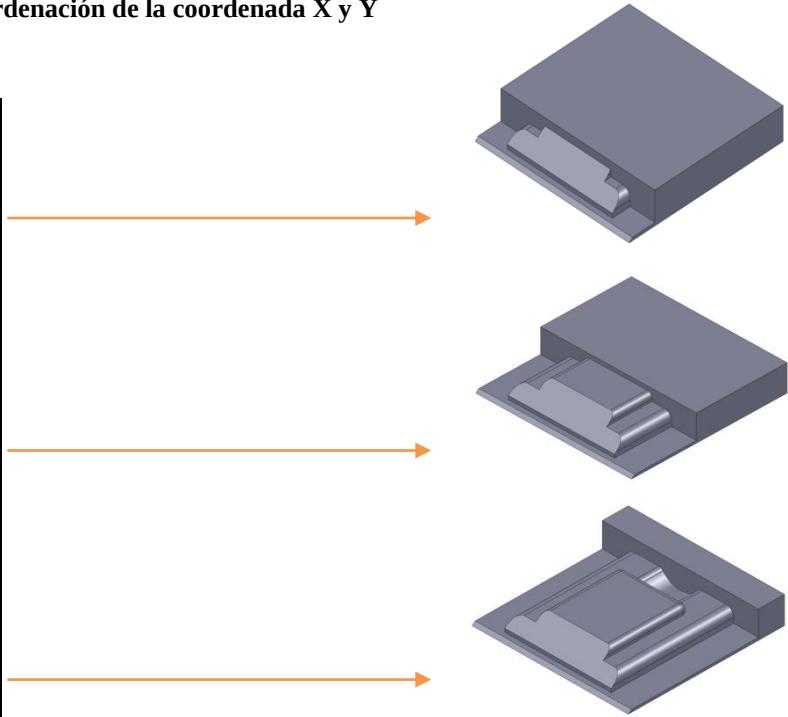


Figura 2-22. Imágenes de correspondencia de rebanada

2.1.2 Obtención de perfil

En esta sección se hablará de la necesidad de obtener un perfil a partir del contorno que se obtiene de la imagen.

El perfil es la trayectoria que debe seguir la herramienta, cuya característica principal es que debe ser una línea continua, debe empezar y terminar en el mismo renglón y el ancho debe ser de un pixel

2.1.2.1 Obtención del contorno

Para la obtención de la imagen de la rebanada se dibujan en color blanco los voxels con el mismo valor en “x” sobre una superficie negra y se obtiene en 2 dimensiones la ubicación de los voxels el contorno de la superficie del sólido pero en pixeles.

Como se está hablando de una imagen binaria se tiene dos valores: el “0” y el “1”, el primero representa los valores en color negro y el segundo en color blanco

Una posible imagen que se obtiene de la pieza rebanada es como la que se muestra en la Figura 2-23, donde se observa que existen pixeles en el contorno que no están de color blanco, esto debido a los huecos que se generaron en la pieza y las redundancias que pudieron haber quedado al representarse en la imagen binaria, simplemente no se aprecian ya que se sobre escribe en el mismo lugar, por lo que ese problema queda resuelto.

Además se observa que el contorno no es uniforme en cuanto al grosor, causado por la duplicidad en que resultó del cálculo de los voxels debido al redondeo, siendo el espesor adecuado de línea de 1 pixel, dado que cuando se realiza el seguimiento del perfil, puede ocasionar fallas ya que existe más de un camino por donde ir, es por ello que el contorno debe corregirse.

Para resolver los problemas causados por los huecos y la duplicidad, se realizan procedimientos para mejorar el contorno, el primero de ellos consiste en identificar lo que corresponde al sólido, es decir, identificar la parte interna del contorno, para ello se rellena en su totalidad el área que lo comprende, cambiando de color los pixeles delimitados por el contorno, con esto se elimina el problema de los huecos.

Posteriormente, se identifica el perfil, es decir, la orilla de contorno para eliminar el problema de la duplicidad, garantizando que el espesor del contorno sea de 1 pixel.

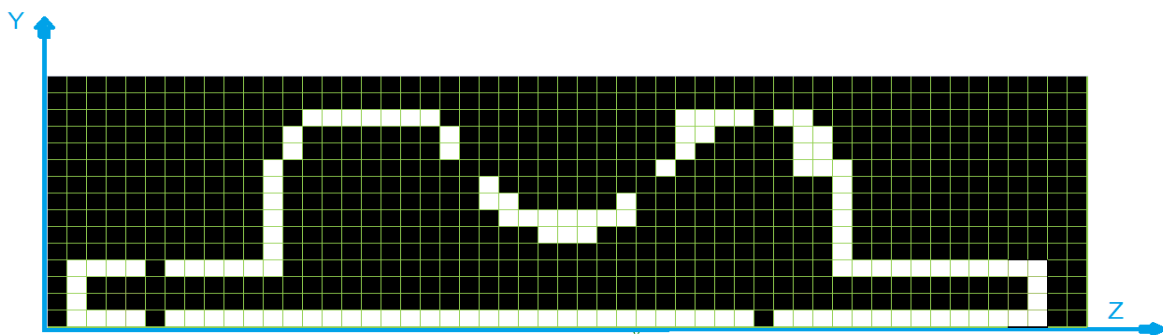


Figura 2-23. Imagen binaria con tamaño de pixel mayor

2.1.2.2 Rellenado del interior del contorno

De ante mano se sabe que el contorno es parte de la superficie de la figura y se puede determinar qué es interno y qué es externo, para ello se realiza el algoritmo de llenado de pixeles internos.

El algoritmo consiste en buscar en cada columna el primer pixel en blanco, empezando de arriba hacia abajo, es decir, buscar un pixel con un valor de 1 y a partir de él cambiar a blanco todos los pixeles que se encuentren debajo hasta llegar al límite, como se muestra en la Figura 2-24.

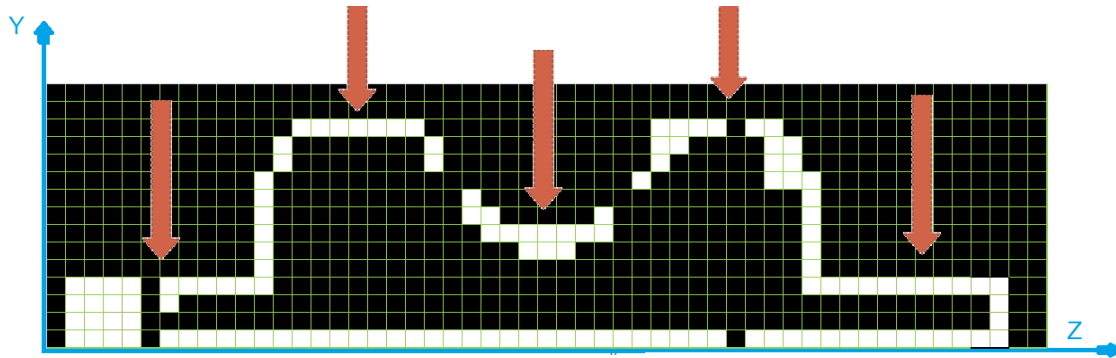


Figura 2-24. Procedimiento para el relleno del interior del contorno

Al final del proceso la imagen queda como se muestra en la Figura 2-25.

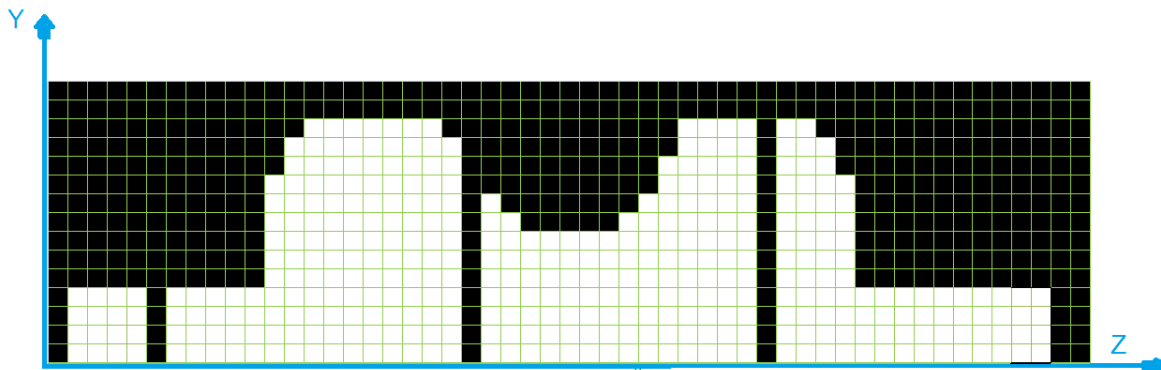


Figura 2-25. Píxeles internos rellenos con secuelas de los huecos

Se aprecia en la Figura 2-25, el daño que ocasionan los huecos en el contorno dado que esa ausencia se prolonga a lo largo de la imagen y para obtener totalmente identificado el interior de la pieza, se utiliza el relleno de huecos.

2.1.2.3 Rellenado de huecos

Este método consiste en llenar huecos que existan dentro del área y en el contorno para garantizar una trayectoria cerrada. Se realiza el proceso de cerradura con un elemento estructural de vecindad-2 horizontal como se muestra en la Figura 2-26 dado que los huecos son de 1 píxel.

EL EE recorre toda la imagen pixel por pixel, empezando de izquierda a derecha y de abajo hacia arriba. La condición para la cerradura de la imagen consiste en que si los valores de los pixeles de la imagen ubicados en ' $p[x+1][y]$ ' y ' $p[x+1][y]$ ' del EE son iguales a "1" el pixel de la imagen que se encuentra en ' $p[x][y]$ ' del EE, cambia su valor a "1", de lo contrario, simplemente el EE avanza al siguiente pixel y vuelve a realizar el procedimiento hasta terminar.



Figura 2-26. Elemento estructural para el llenado de huecos

La imagen después de este proceso se muestra a continuación en la Figura 2-27.

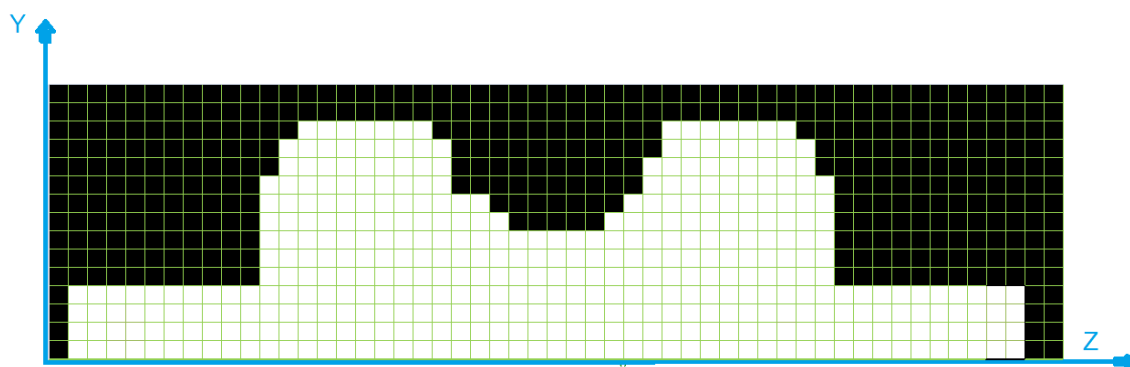


Figura 2-27. Imagen binaria después de rellenar huecos

2.1.2.4 Identificación de perfil

Una vez definido el área del sólido, es posible identificar el contorno. El contorno es una línea cerrada, en este caso es la cadena de píxeles conectados entre sí, por medio de arista o por vértice, y empieza y termina en el mismo punto (Figura 2-28).

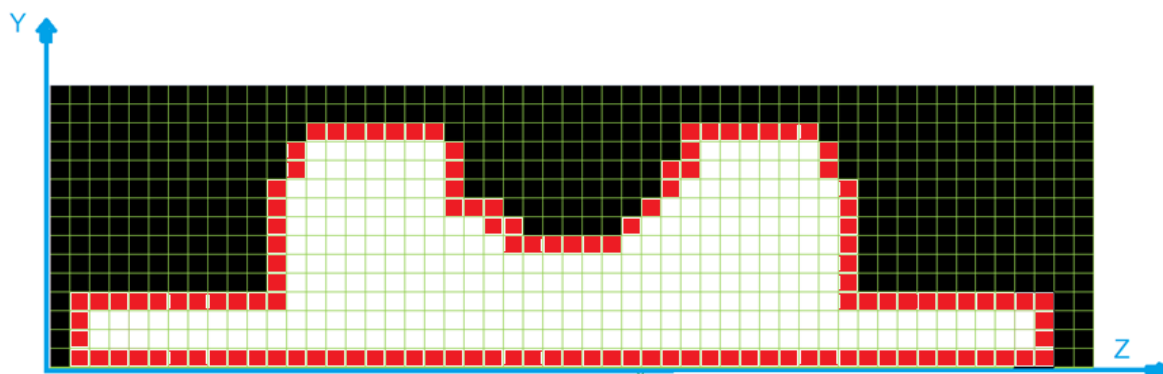


Figura 2-28. Contorno de la figura

El perfil es parte del contorno de la figura pero sin la base, es decir, sin la línea horizontal en el primer renglón, dado que es la trayectoria que debe seguir la herramienta (Figura 2-29).

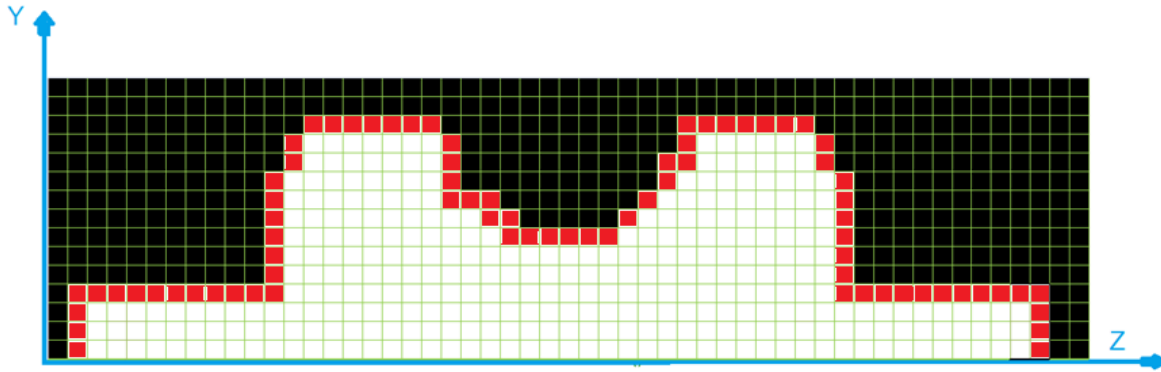


Figura 2-29. Perfil del contorno

Para marcar el perfil de la figura, se hace en dos partes:

- Barrido vertical
- Barrido horizontal

El primero consiste en hacer un barrido de arriba hacia abajo y de izquierda a derecha; en donde se pregunta por el valor de cada pixel y al encontrar aquel que tenga el valor de 1, que obviamente va a ser el pixel más externo su valor cambiará a 2 y la búsqueda se repite en la columna siguiente, así sucesivamente hasta terminar.

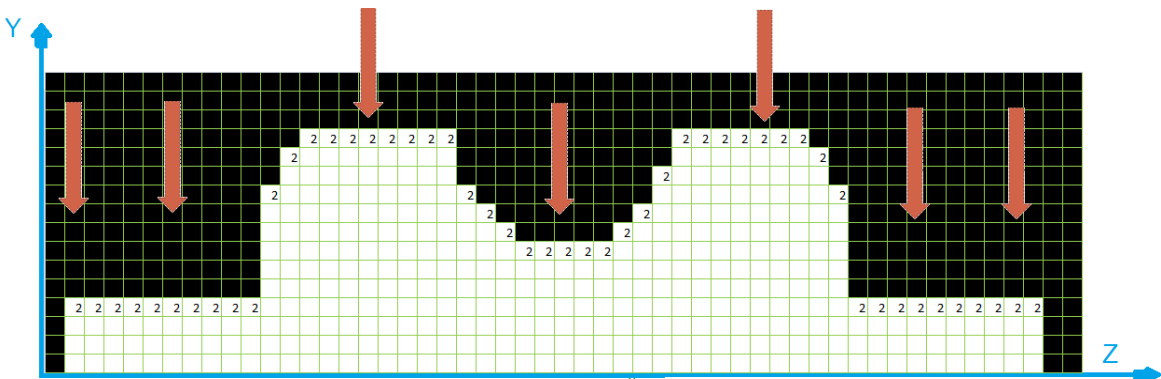


Figura 2-30. Proceso de marcar el contorno, con un barrido vertical hacia arriba

Con este método se identifican los pixeles exteriores de las líneas horizontales y de algunas curvas, sin embargo, aquellos que integran las líneas verticales no fueron identificadas, es por ello que se realiza la siguiente operación

La segunda parte consiste en hacer un barrido de izquierda a derecha y de arriba hacia abajo, en donde se pregunta por el valor del pixel actual ' $p[x][y]$ ' y del siguiente ' $p[x+1][y]$ ', si el valor del primero es 0 y del segundo es 1, el pixel ' $p[x+1][y]$ ' cambia su valor a 2.

Y si los valores del pixel actual ' $p[x][y]$ ' y del siguiente ' $p[x+1][y]$ ' son 1 y 0, respectivamente, el pixel ' $p[x][y]$ ', cambia su valor a 2, así sucesivamente hasta terminar con el renglón.

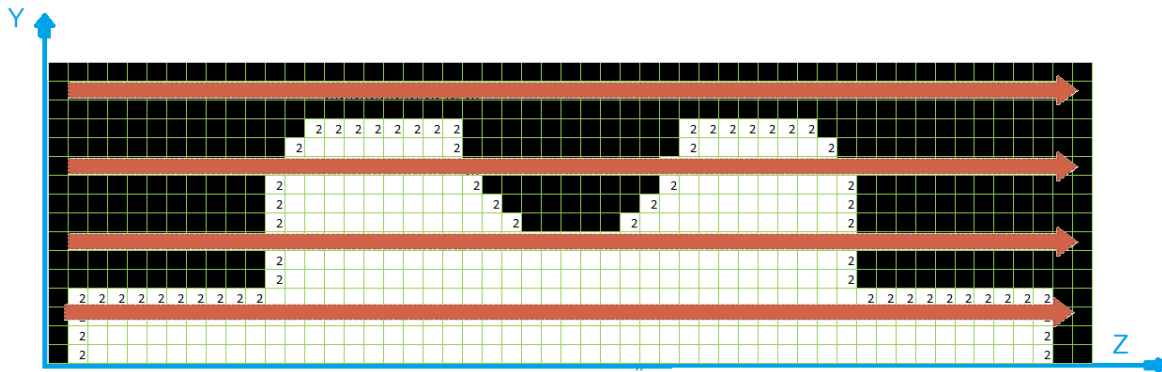


Figura 2-31. Proceso de marcar el contorno con un EE en barrido horizontal

Finalmente el contorno queda marcado de la siguiente manera en la Figura 2-32.

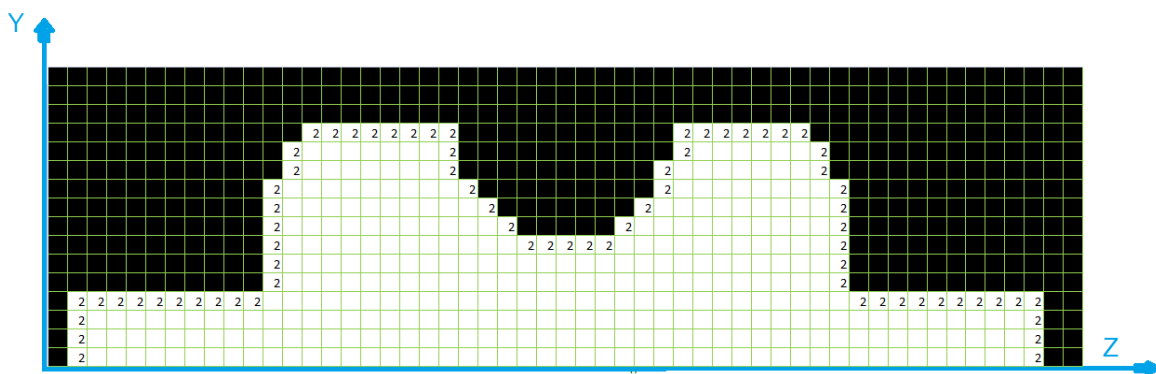


Figura 2-32. Proceso de marcado del contorno terminado

2.1.2.5 Eliminación de protuberancias de un pixel

Existen casos en la imagen donde existen protuberancias verticales con un ancho de un pixel, y es necesario eliminarlo o ensancharlo, con la finalidad de evitar problemas posteriores en el seguimiento del perfil (Figura 2-33).

Ya que como en la definición del contorno lo indica, es una cadena sucesiva de pixeles conectados por vértice o arista, es decir, un pixel siempre debe estar conectado con dos pixeles, pero si existe al menos otro pixel más conectado, el algoritmo para el seguimiento del perfil fallaría, debido a que existen 2 posibles caminos y si toma el de la protuberancia, se termina el proceso ya que no puede regresar por el mismo camino y la trayectoria que seguiría la herramienta sería la incorrecta.

Es por eso que si se ensancha la protuberancia, el algoritmo de búsqueda no presentaría ese problema porque tendría cómo continuar con el perfil hasta terminar.

Para ello se ocupa la operación morfológica de Cierre descrito en el Capítulo 1:, con un EE de 3x3, para dilatar y después erosionar la imagen en un pixel para que la imagen vuelva al tamaño original, con la diferencia de que las protuberancias cambian de espesor de 1 pixel a 2 como se muestra en la Figura 2-34.

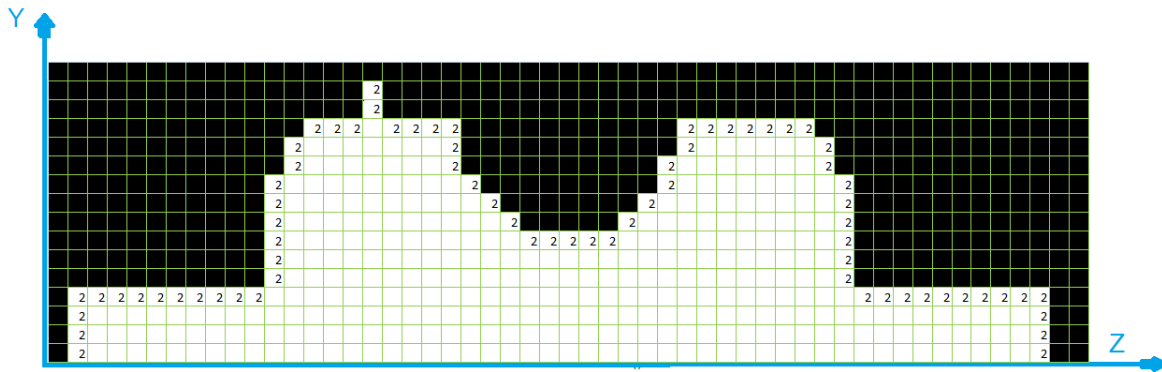


Figura 2-33. Imagen con protuberancia vertical

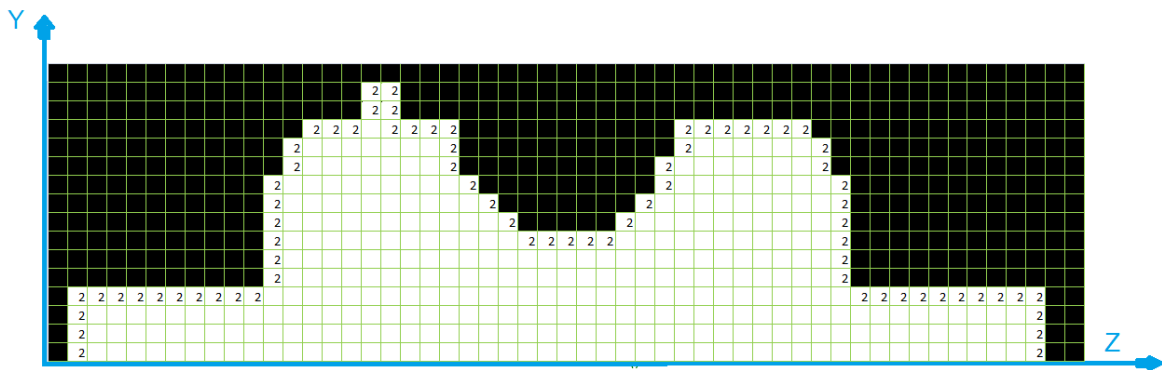


Figura 2-34. Imagen después del proceso de cierre y protuberancia dilatada

2.1.2.6 Compensación de la herramienta de corte

Existe un problema con la herramienta de corte ya que, aunque se le puede considerar como un punto de corte, maquinaria de más si dicho punto de corte sigue el perfil debido a sus dimensiones físicas.

El cortador es de forma cilíndrica y posee un punto de corte en el avance, si el corte es de izquierda a derecha, ya sea horizontal o diagonal ascendente, el punto de corte se ubica como muestra en el inciso a) de la Figura 2-35, sin embargo cuando el avance de la herramienta es de derecha a izquierda, horizontalmente o diagonal ascendente, el punto de corte cambia como en el inciso b) de la misma Figura 2-35.

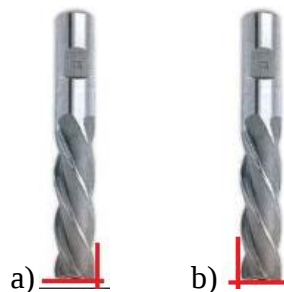


Figura 2-35. Cortador vertical con punto de corte: a) con avance hacia la derecha b) y la izquierda

Lo anterior quiere decir que para cualquiera de los dos casos, si el punto de corte (representado por el pixel de color rojo) de la herramienta sigue un perfil como se muestra en la Figura 2-36, la herramienta de corte maquinaría de más debido al diámetro de la misma como se muestra en la Figura 2-37, donde los pixeles en verde son los cortes cuando el avance de la herramienta es hacia la derecha y en azul cuando va hacia la izquierda, por lo que es necesario realizar un arreglo a la imagen para que no importando la dirección de corte, respete la medida nominal.

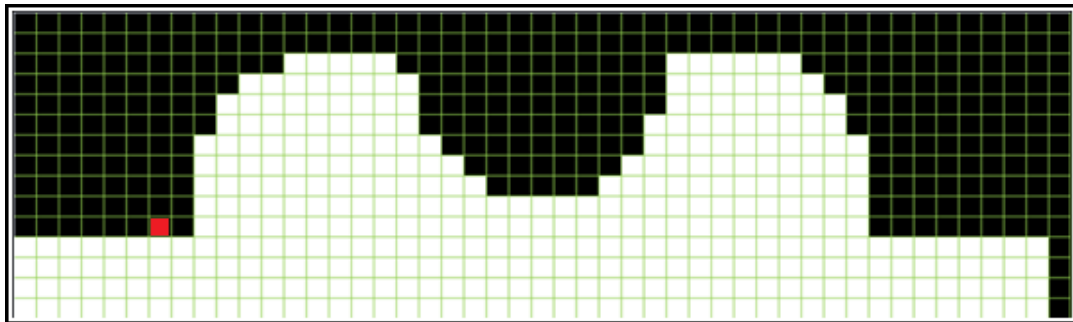


Figura 2-36. Imagen binaria con el punto de corte del cortador

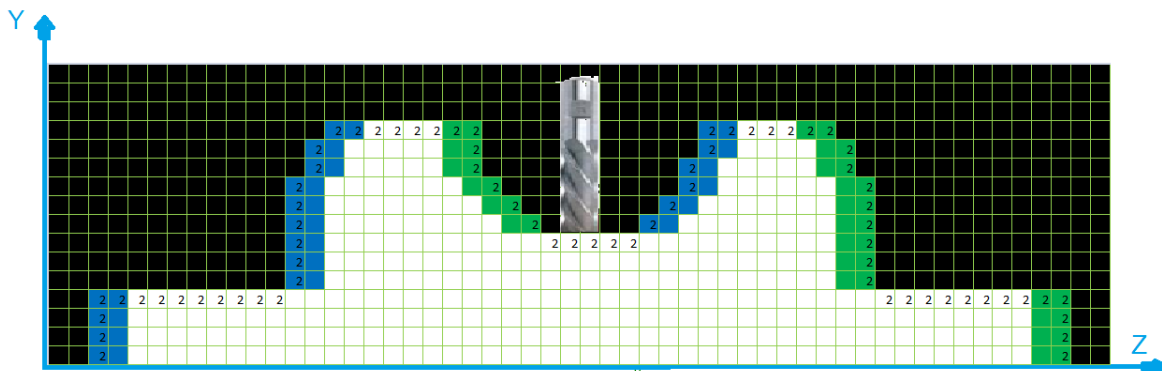


Figura 2-37. Cortede la herramienta sin compensación

Para esto se hace una dilatación hacia los lados en la figura, para que cuando la herramienta describa la trayectoria y machine de más, la pieza quede en su medida nominal. Se dilata con un EE de vecindad 2-horizontal, para hacerlo crecer en el ancho y no en la altura, y dicha dilatación va a estar ligada al diámetro radio de la herramienta, por ejemplo si el diámetro fuera de 3 mm. y en términos de pixeles equivaliera a 30, la dilatación crecería en 15 pixeles por lado, es decir, se repetiría 15 veces el proceso de dilatación en la imagen para obtener la dilatación necesaria.



Figura 2-38. EE para la dilatación de imagen

El EE recorre toda la imagen pixel por pixel, empezando de izquierda a derecha y de arriba hacia abajo para dilatar según la teoría. Para este caso, la condición para dilatar consiste en que si en cualquiera de los tres pixeles: $p[y][x-1]$, $p[y][x]$, $p[y][x+1]$, tienen un valor de 1, el pixel $p[y][x]$ cambia su valor a 1, de lo contrario avanza al siguiente pixel y vuelve a

realizar su procedimiento, hasta terminar. Finalmente se obtiene la imagen compensada y se muestra en la Figura 2-39.

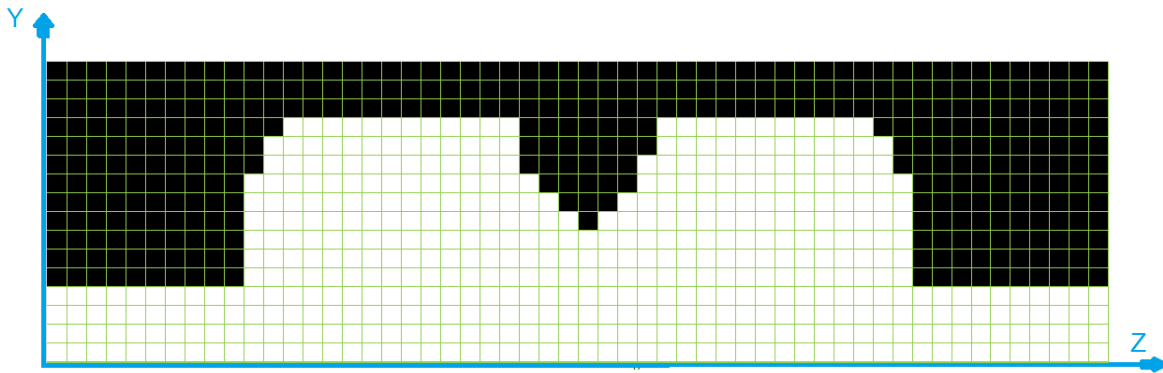


Figura 2-39. Imagen binaria con compensación de la herramienta

Así en la imagen anterior se le aplica el algoritmo para obtener su perfil, el cual se representa en color naranja en la Figura 2-40 y el punto de corte, quien sigue el perfil (con el algoritmo de seguimiento de perfil que se muestra en la siguiente sección), y que representa a la herramienta como un pixel, se ubica al centro de la misma, para que no importando hacia donde se desplace la herramienta, machine y elimine el material correspondiente a los pixeles de color verde y azul, para obtener la pieza deseada representada en color blanco.

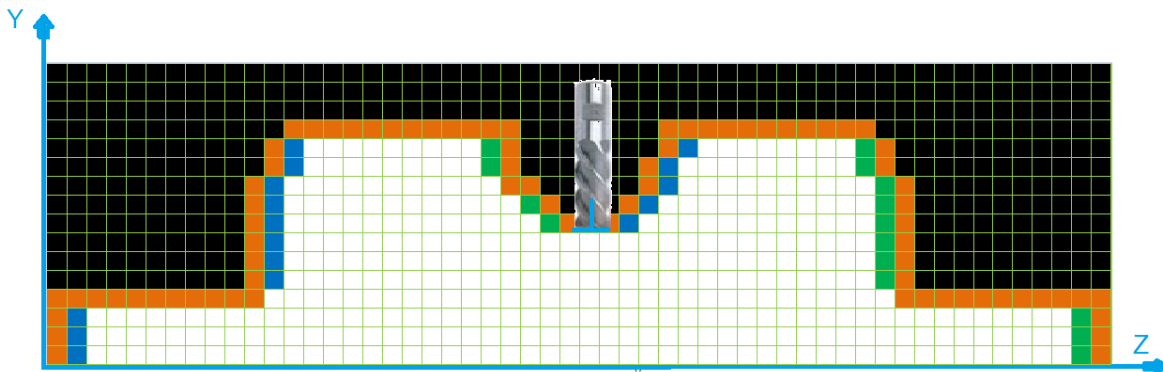


Figura 2-40. Compensación de la herramienta de corte

2.1.2.7 Seguimiento de perfil

Una vez teniendo el perfil compensado como se muestra en la Figura 2-41, es turno de seguir la trayectoria del perfil, esto se logra buscando los pixeles que tienen un valor de 2.

Primeramente se busca el primer pixel del perfil, empezando en el origen con un barrido de izquierda a derecha y de abajo hacia arriba hasta encontrarlo.

Después se busca el siguiente pixel en la vecindad-8, empezando por el pixel debajo 'p [y-1] [x]' y en sentido anti horario, como muestra el inciso a) de la

Figura 2-42, y cuando lo encuentra, el pixel 'p[y] [x]' cambia su valor a 3 y el buscador se mueve hacia el pixel encontrado y vuelve a hacer la búsqueda así sucesivamente hasta completar el perfil.

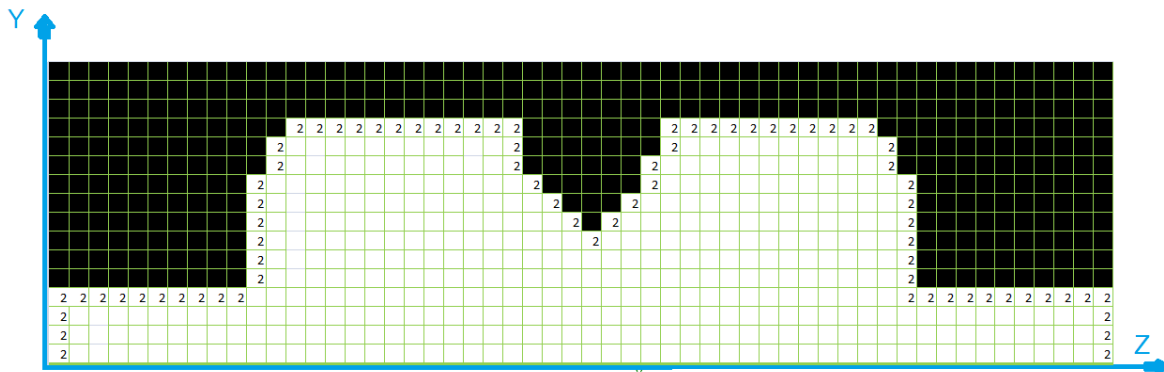


Figura 2-41. Contorno marcado con compensación de herramienta

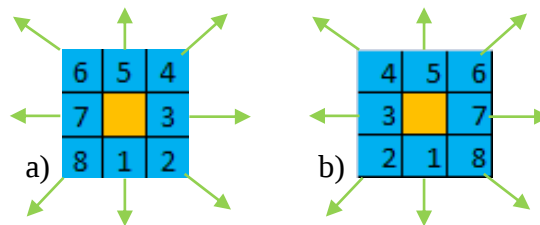


Figura 2-42. Dirección de búsqueda de perfil hacia la derecha (a) e izquierda (b) respectivamente

De esta manera, al marcar los píxeles ya recorridos con un valor de 3, no hay forma de que se regrese por el mismo camino (Figura 2-43).

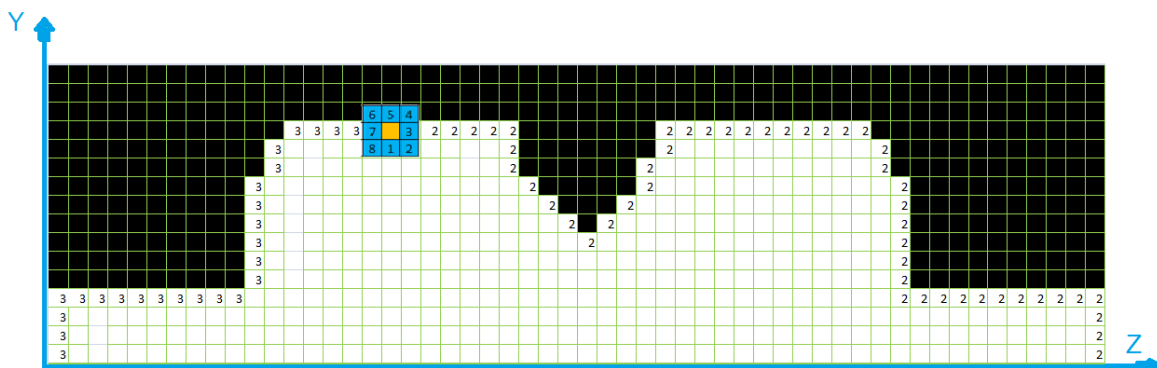


Figura 2-43. Recorrido del EE sobre el perfil hacia la derecha

El recorrido del perfil anteriormente descrito se hace de izquierda a derecha, sin embargo, cuando se busca el perfil de izquierda a derecha, dado que la estrategia de maquinado así lo requiere, el buscador cambia el sentido de búsqueda pero conservando el mismo pixel de inicio como se muestra en la

Figura 2-42 inciso b). En el seguimiento de perfil, se realiza el mismo procedimiento de marcar el camino que ya recorrió, como se muestra en la Figura 2-44.

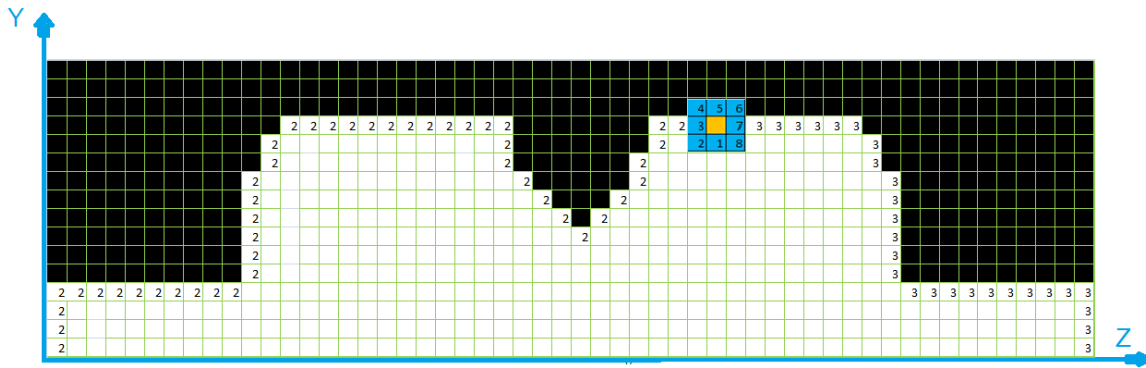


Figura 2-44. Recorrido del EE sobre del perfil hacia la izquierda

El seguimiento del perfil permite obtener los valores “y”, “z” que se envían al mecanismo. Aplicando una escala donde un pixel equivale a 0.02 mm. Al terminar de seguir el perfil de la imagen binaria, se pasa a la siguiente rebanada con el valor siguiente de la coordenada “x” donde también se manda la información al mecanismo, y la búsqueda de perfil hacia la derecha y hacia la izquierda se van alternando, es decir, cuando empieza la primera rebanada, la búsqueda del perfil se hace hacia la derecha y la rebanada siguiente hacia la izquierda, y así sucesivamente hasta terminar con todas las rebanadas del sólido.

2.2 Implementación del software

La programación es parte fundamental de este trabajo, ya que es el medio de implementación de los algoritmos y del control del mecanismo para fabricar la pieza sin la necesidad de la intervención del operador para corregir fallas o modificar algún programa.

En esta sección se explica cómo se realizó la implementación en la programación de lo que se desarrolló en la sección de diseño. Así mismo se muestra el mapa de cómo está organizado el software de las rutinas hechas, las cuales se explicarán una a una más adelante.

El CAM desarrollado se divide en tres grandes bloques:

1. Construcción de la superficie
2. Obtención del perfil
3. Movimiento de motores

El primero se refiere a los procedimientos que se realizan para obtener la ubicación espacial de los voxels que integran toda la superficie de la pieza dibujada en CAD.

El segundo explica la forma de seccionar el sólido obtenido en rebanadas y el procesamiento de cada una para obtener el perfil.

El tercero aborda sobre la interfaz o comunicación vía puerto paralelo con los sensores y controladores de cada motor.

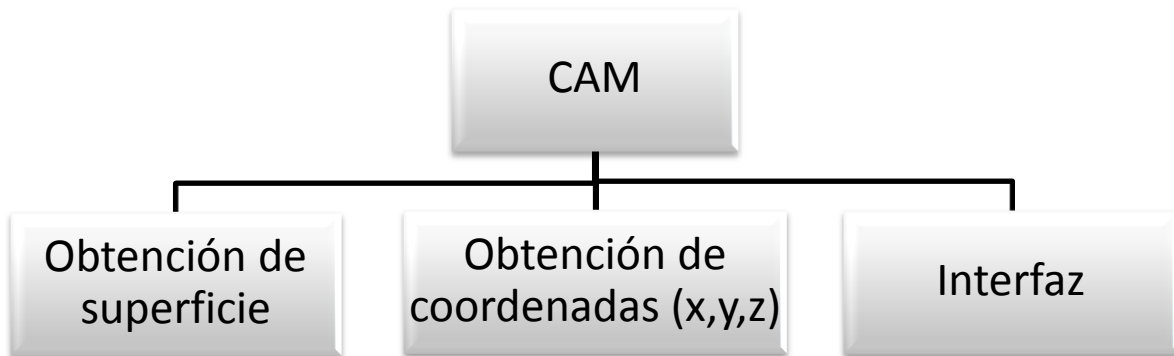


Diagrama 2-1. Esquema del CAM desarrollado

Entonces el proceso para maquinar una pieza dibujada en un programa CAD pasa por cuatro etapas que dependen progresivamente.

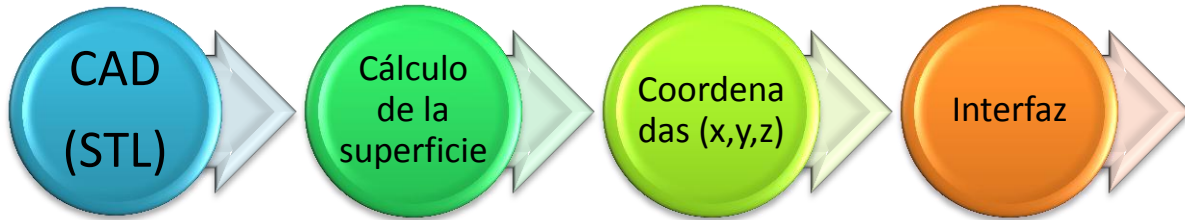


Diagrama 2-2. Proceso para la manufactura de una pieza dibujada en CAD

Dentro del proceso para la manufactura se empieza por el CAD, que es donde se realiza el dibujo de la pieza a fabricar, y se guarda en un formato STL para obtener un archivo en código ASCII. Después se realiza el cálculo de la superficie para después rebanar el sólido y obtener las coordenadas (x,y,z) para el movimiento de los motores

2.2.1 Construcción de superficie del sólido

La construcción de la superficie del sólido consiste en formar su superficie con una serie de voxels, de los cuales se obtendrán sus coordenadas (x,y,z) , con ello se conoce en todo momento la geometría de la pieza, como se muestra en la Figura 2-45.

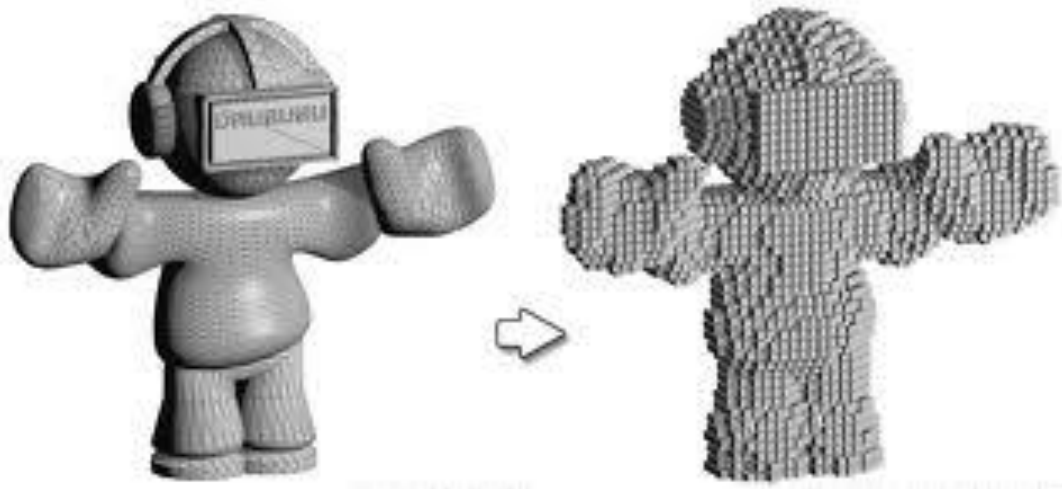


Figura 2-45. Construcción de la superficie con voxels

Para realizar la construcción virtual de la superficie del sólido dibujado en CAD, es primordial extraer la ubicación de los vértices del archivo en formato stl. Posteriormente estos datos son ocupados para realizar el cálculo de los voxels que integran el área del triángulo, para finalmente repetir dicho proceso para todos los triángulos, como muestra el Diagrama 2-3.

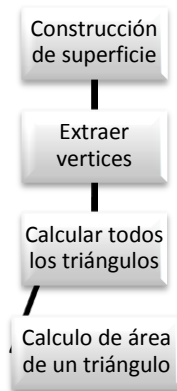


Diagrama 2-3. Proceso para la obtención de la superficie del sólido

2.2.1.1 Rutina de extracción de vértices

Esta sección explica cómo se extraen los valores (x,y,z) de los vértices V1, V2, V3 del triángulo del archivo STL, como se muestra en la Figura 2-46.



Figura 2-46. Proceso de extracción de vértices

En la implementación, el algoritmo de extracción de los valores de los vértices se efectúa primeramente abriendo el archivo stl en código ASCII y se empieza a recorrer hasta encontrar y capturar los valores (x, y, z) de cada vértice que forma un triángulo y así sucesivamente hasta terminar con el archivo, como muestra el Diagrama 2-4.

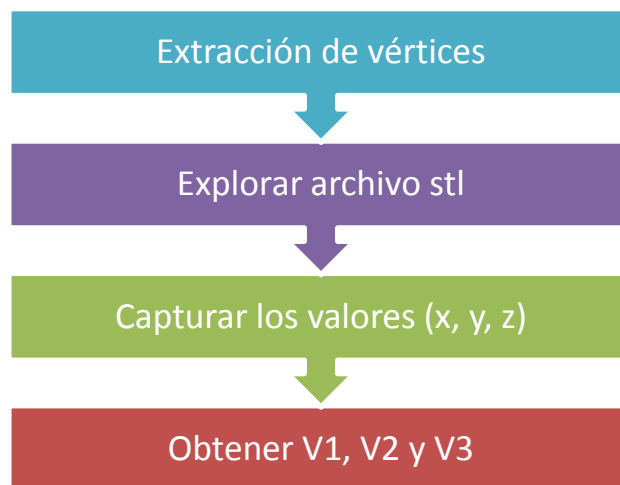


Diagrama 2-4. Proceso para obtención de vértices

Observando el formato en el que se presenta la información de las coordenadas de los vértices de cada triángulo, está dada en código ASCII (American Standard Code Information Interchange), que es un código de caracteres basados en el alfabeto latino, se tiene lo siguiente:

```
facet normal 1.000000e+000 0.000000e+000 0.000000e+000
outerloop
vertex 1.000000e+001 1.000000e+001 1.000000e+001
vertex 1.000000e+001 0.000000e+000 1.000000e+001
vertex 1.000000e+001 1.000000e+001 0.000000e+000
endloop
endfacet
```

En el primer renglón aparece la leyenda “facet normal”, los valores que se presentan se refieren a la ubicación vectorial normal al triángulo y dicha información no es útil para el cálculo de los voxels, por lo que se ignora esta información.

Posteriormente en el segundo renglón aparecen las palabras: “outerloop” que indica el inicio de las coordenadas de los vértices del triángulo.

En el tercer renglón aparece la palabra “vertex” y en seguida aparecen tres valores separados por un espacio, dichos valores corresponden a las tres coordenadas (x, y, z).

El primer vértice V1 tiene los siguientes valores:

1.000000e+001

Donde:

- 1.000000: es el valor de la base, compuesto por 8 caracteres
- e: representa la notación científica que equivale a 10^x
- +: representa el signo del exponente
- 001: es el valor del exponente, compuesto de tres caracteres

Esto quiere decir que el valor de la coordenada en “x” es: $1 \times 10^1 = 10$, las unidades están dadas en milímetros en este caso.

Para la extracción de los vértices, se observa que los valores empiezan después de la palabra “vertex” y dada la peculiaridad de que la “x” es la única letra inmediata anterior a los valores y que no existe otra “x” anterior a ella, por lo que primeramente se hace una búsqueda de dicho carácter.

Una vez localizado la “x” se avanza un carácter que es el que representa el espacio y después se captura el siguiente carácter que es la parte entera de la base y el siguiente carácter a capturar es el punto decimal, y posteriormente los siguientes 6 que corresponden a la parte fraccionaria.

Después se captura el siguiente carácter correspondiente a la letra “e” y se avanza otro carácter, ocupado por el signo “+”, que en todos los casos siempre es el mismo, es decir, que el exponente es positivo y finalmente se capturan los siguientes 3 valores que

representan el valor del exponente. Con este procedimiento se ha capturado el valor de la coordenada en “x”.

Dado que se han capturado caracteres, hay que realizar una transformación de dichos caracteres a números de tipo float, para poder tener números fraccionales.

Posteriormente se avanza un caracter, que es el espacio en blanco y se capturan los siguientes ocho caracteres, con las mismas características anteriormente descritas, y de esta manera se obtiene el valor de la coordenada en “y”, posteriormente se hace la conversión al float.

Finalmente se avanza otro caracter, y se capturan los siguientes ocho, y se convierte en números del tipo float, obteniéndose así el valor de la coordenada en z.

Con este proceso se han obtenido las tres coordenadas del primer vértice V1. El proceso descrito anteriormente se muestra a continuación en el Diagrama de flujo 2-1.

Posteriormente para capturar los valores de las coordenadas del vértice 2, se vuelve a buscar el caracter “x”, y se repite el proceso del Diagrama de flujo 2-1 y al finalizar, se hace una tercera búsqueda de la letra “x”, para capturar los valores del vértice 3.

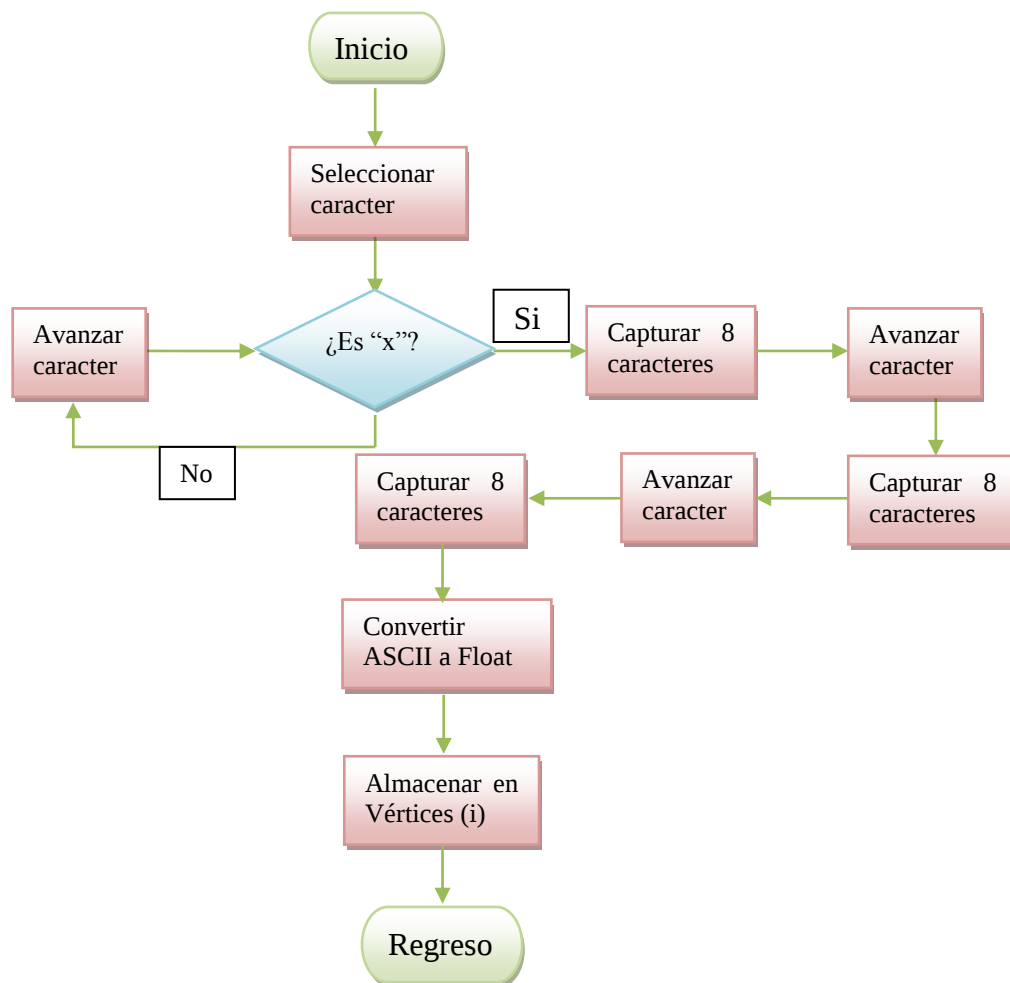


Diagrama de flujo 2-1. Algoritmo para obtención de coordenada de vértice

2.2.1.2 Obtención de las aristas

Una vez extraídos los vértices del triángulo, se procede a calcular los voxels que integran su superficie, por medio del método radial, pero para ello es necesario primeramente calcular los que forman las aristas como muestra la Figura 2-47.

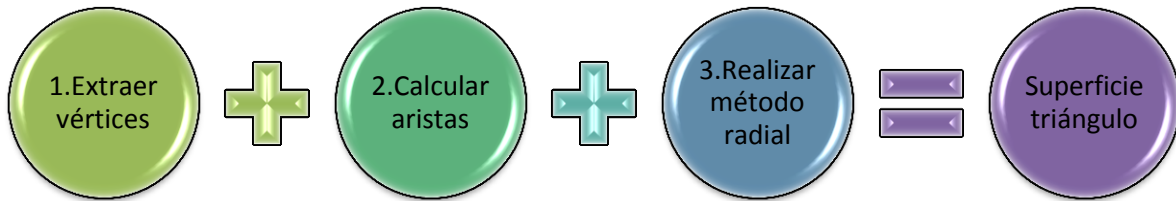


Figura 2-47. Proceso para obtención de voxels de la superficie de un triángulo

Lo que se conoce de las aristas son las coordenadas de los vértices, por lo que para calcular los voxels que la integran, se retoma la forma de unir dos puntos en el espacio para calcular una arista a la vez como muestra la Figura 2-48.

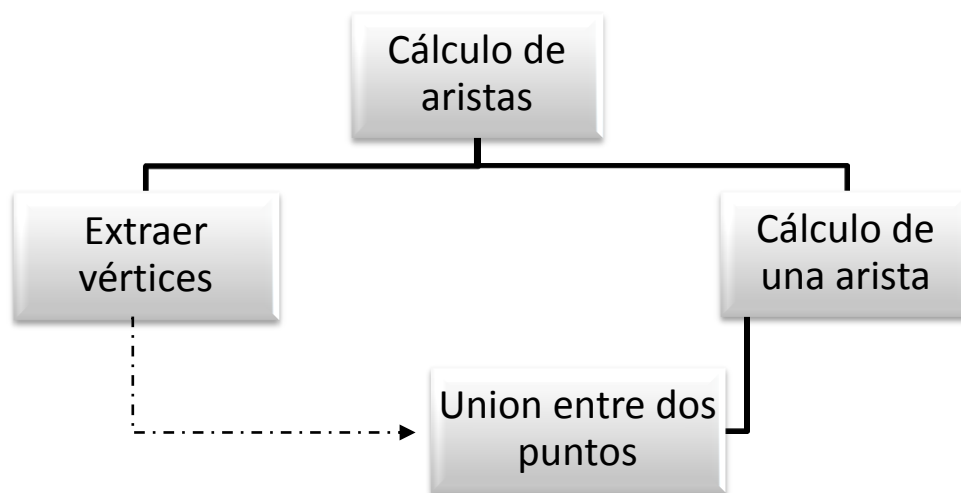


Figura 2-48. Proceso de la extracción de vértices

Supóngase que al igual que en la sección de diseño, se han obtenido los siguientes 3 vértices:

1. V1: P (5, 4, 7)
2. V2: Q (1, 1, 1)
3. V3: R (10, 5, 1)

Las aristas están formadas de los vértices como se muestra a continuación:

1. A1:V1 y V2
2. A2:V2 y V3
3. A3: V3 y V1

En la Figura 2-49, se muestra la ubicación espacial de los vértices y de las aristas de las que se obtendrán los voxels que las integran.

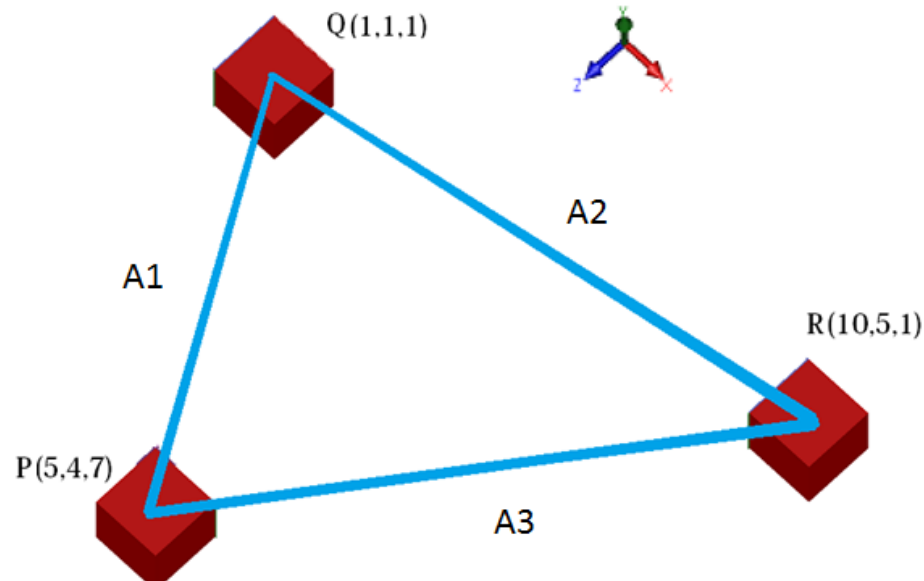


Figura 2-49. Ubicación espacial de los vértices del triángulo

2.2.1.2.1 Rutina de unión de dos puntos

Esta sección habla de la forma de obtener los voxels intermedios entre dos puntos que se ubican en el espacio, para ello se obtienen los incrementos para cada coordenada, para poder calcular las coordenadas del voxel siguiente, así se repite hasta tener los voxels con sus respectivas coordenadas (x,y,z) que integran esa línea como muestra la Figura 2-50.

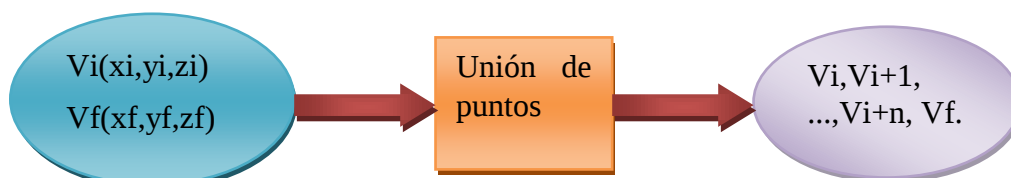


Figura 2-50. Proceso de cálculo de voxels de una línea

Para la obtención de los voxels que forman una línea, la rutina de obtención de voxels se apoya en la de obtener dos puntos, que los toma de los 3 vértices, después se procede a hacer el cálculo de incrementos, el cual consiste en calcular los incrementos en cada coordenada (x,y,z).

El cálculo del voxel siguiente consiste en obtener la ubicación espacial de los voxels, para ello se realiza haciendo una suma entre el punto inicial y el incremento, después el bloque de redondeo consiste en convertir el número fraccionario a un número, como se muestra en Diagrama 2-5.

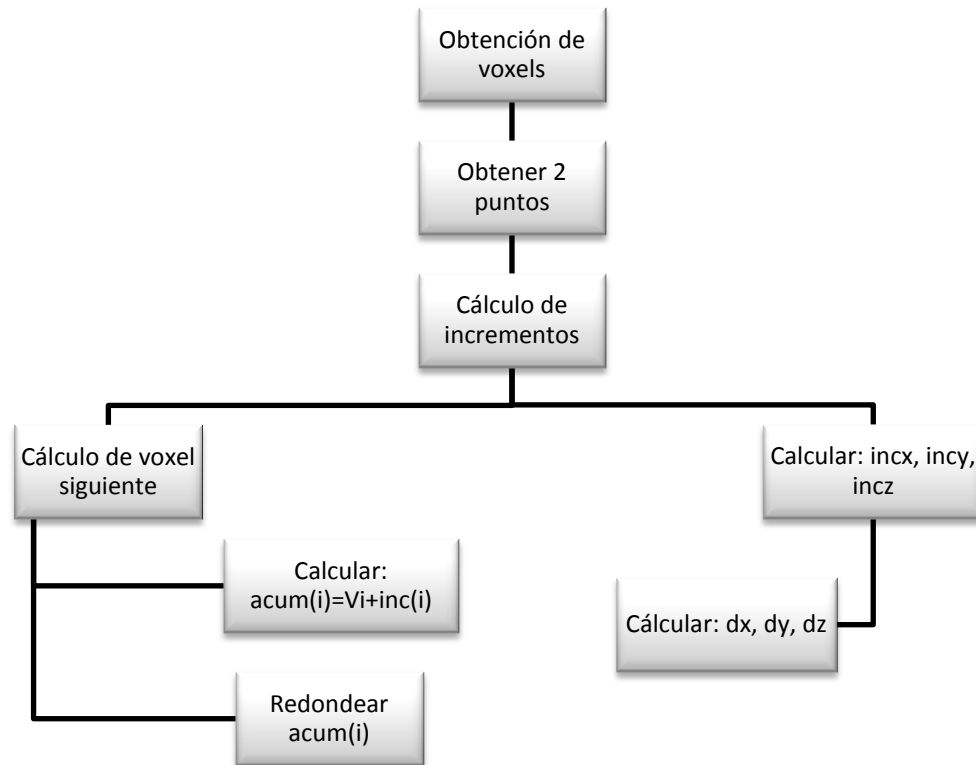


Diagrama 2-5. Proceso de rutina de obtención de voxels

Para calcular la arista A1, los vértices V1 y V2 pasan a ser Vi y Vf respectivamente, y con este ejemplo se realizan los cálculos siguientes.

Todos los valores calculados se almacenan en la memoria de la PC, utilizando lo que se conoce como memoria dinámica, donde se solicita memoria para ciertos cálculos, y se borra cuando ya no se necesita.

2.2.1.2.2 Cálculo de incrementos

Se procede a realizar los cálculos de los incrementos, para ello se realiza la diferencia de las coordenadas del punto final (Vf) y el inicial (Vi), obteniéndose el valor absoluto “dx”, “dy” y “dz”.

Después se selecciona la diferencia de mayor magnitud y se guarda en una variable llamada “den”, y el incremento en cada coordenada está dado al dividir las diferencias dx, dy, dz entre “den”, como se presenta a continuación en el Diagrama de flujo 2-2.

Al sustituir los valores de los vértices de las aristas y realizando los cálculos anteriormente mencionados, se tienen los incrementos:

$$\begin{aligned} \text{incx} &= -2/3 \\ \text{incy} &= -1/2 \\ \text{incz} &= -1 \end{aligned}$$

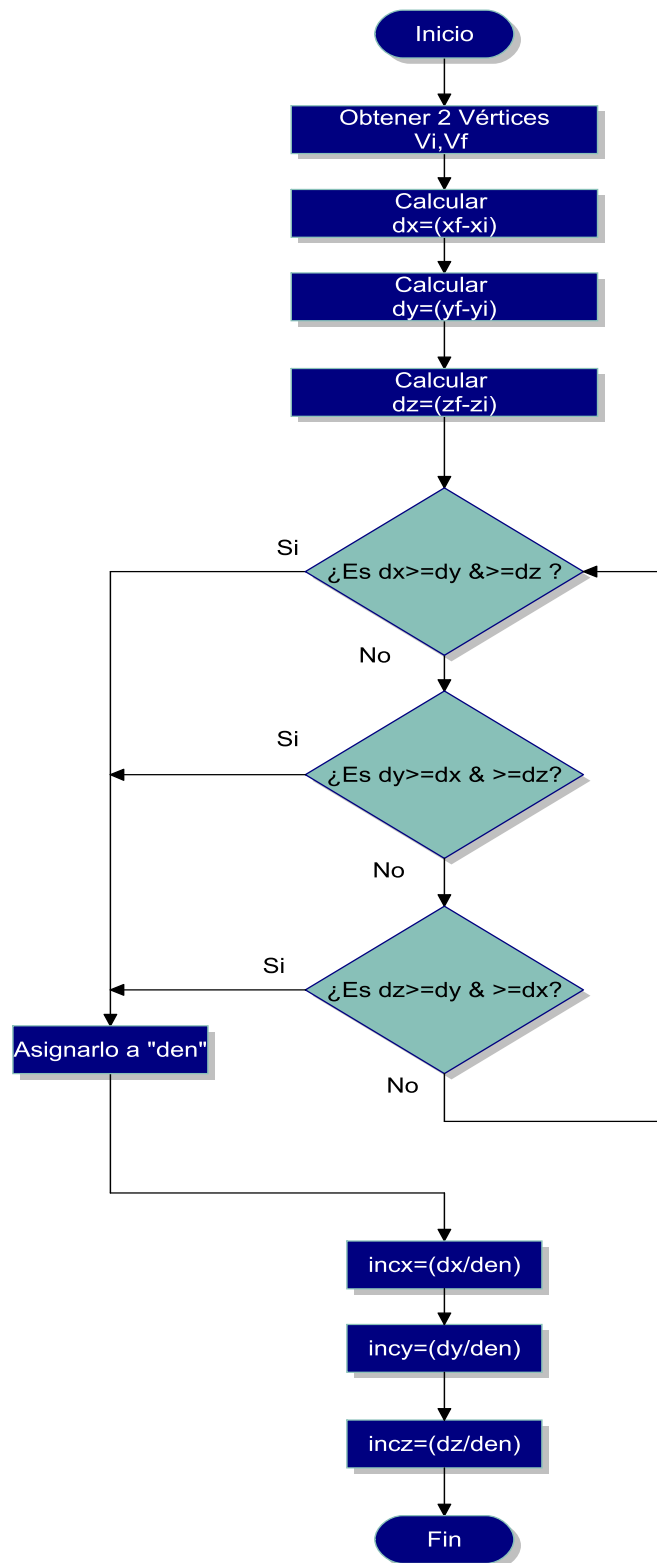


Diagrama de flujo 2-2. Algoritmo de cálculo de incrementos

2.2.1.2.3 Cálculo de los voxels

Se procede con el cálculo de voxels entre dos vértices de la arista A1, para ello se toma el valor de las coordenadas iniciales V1: P (5,4,7), que van hacia las finales V2: Q (1,1,1).

Las coordenadas iniciales son asignadas a las variables “acumx”, “acumy”, “acumz”, que significa acumulación, como se muestra a continuación

```
acumx=x1
acumy=y1
acumz=z1
```

Posteriormente se hace la operación de sumar los incrementos de las coordenadas, los cuales se presentan a continuación:

```
acumx=acumx+incx
acumy=acumy+incy
acumz=acumz+incz
```

Sustituyendo valores se tiene:

```
acumx=5+(- 2/3)= 4.67
acumy=4+(-1/2)=3.5
acumz=7+(-1)= 6.0
```

El valor de la coordenada del voxel calculado debería tener dichos valores, sin embargo como se está tomando en cuenta que el tamaño del voxel es de un milímetro, no pueden existir fracciones de voxel es por esta razón que se realiza el redondeo.

2.2.1.2.4 Redondeo

Para realizar el redondeo se tiene la condición de que si la parte decimal del valor es mayor o igual a 0.5, se redondea al número entero inmediato superior, y si es menor se redondea al número inmediato inferior.

Se utiliza una instrucción que toma en cuenta únicamente la parte entera del número, es por ello que se le suma al valor obtenido 0.5 para que cumpla esta condición.

```
*(ptosx)=floor(acumx+0.5)
*(ptosy)=floor(acumy+0.5)
*(ptosz)=floor(acumz+0.5)
```

Dónde:

floor: es la instrucción de tomar el valor entero

*(ptos_): es la localidad de memoria donde se almacena el valor numérico

Sustituyendo

```
*(ptosx)=floor(4.67+0.5)=5.17=5
*(ptosy)=floor(3.50+0.5)=4.00=4
*(ptosz)=floor(6.00+0.5)=6.50=6
```

Por lo que el voxel siguiente (V1+1) al vértice V1, tiene las coordenadas (5, 4, 6), y se representa en la siguiente Figura 2-51

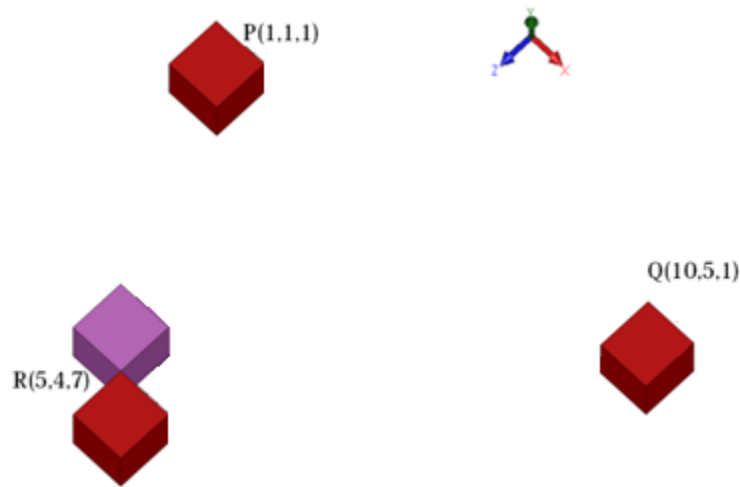


Figura 2-51. Ubicación espacial del voxel (V1+1)

Para obtener los valores del siguiente voxel (V1+2), se utilizan las mismas ecuaciones, nótese como el valor del voxel inicial cambia por el valor obtenido (V1+1).

$$\text{acumx} = 4.67 + (-2/3) = 4.67$$

$$\text{acumy} = 3.5 + (-1/2) = 3.0$$

$$\text{acumz} = 6.0 + (-1) = 5.0$$

Obteniéndose las coordenadas de (V1+2)= (5, 3, 5).

Para obtener los voxels siguientes, V1+3, V1+3,..., Vf-1, hay que repetir las ecuaciones anteriores un número “i” de veces, que corresponde al número de voxels que lo integran, es decir el valor “den”, por lo que en programación se tiene un ciclo:

```
for(i=1;i<=den;i++)
{
    acumx=acumx+incx;
    acumy=acumy+incy;
    acumz=acumz+incz;

    *(ptosx+i)=floor(acumx+0.5);
    *(ptosy+i)=floor(acumy+0.5);
    *(ptosz+i)=floor(acumz+0.5);
}
```

Dónde:

for: instrucción de ciclo

i: es la variable del número de iteraciones

Con este ciclo los valores han sido calculados las coordenadas de los voxels que integran la arista A1. Gráficamente los voxels se representarían de la siguiente manera en la Figura 2-52.

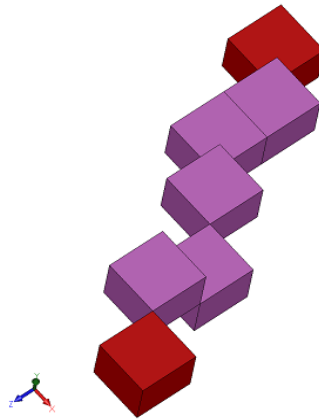


Figura 2-52. Construcción de voxels de una arista

El proceso del cálculo de las coordenadas de los voxels que integran una línea anteriormente explicado, es como se muestra en la Figura 2-53.

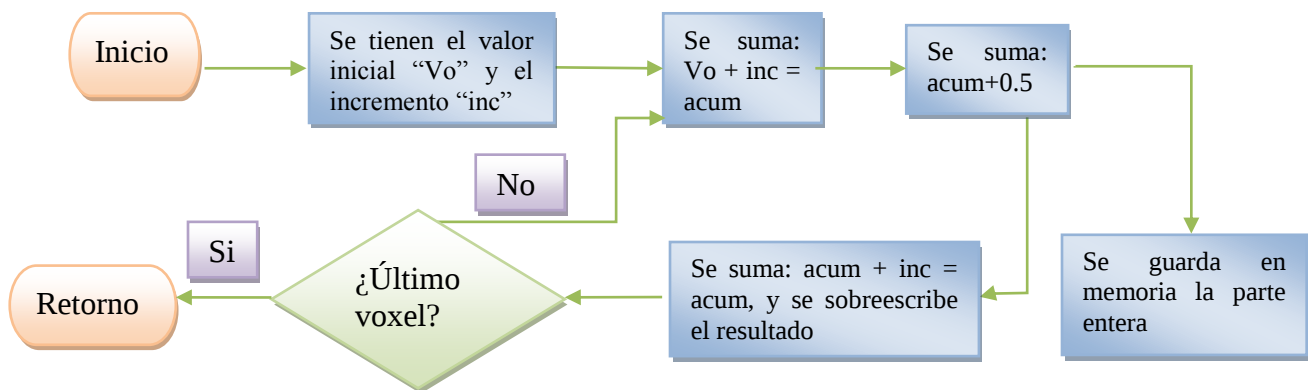


Figura 2-53. Algoritmo del cálculo de coordenadas de los voxels siguientes

Para obtener las aristas otras dos aristas A2 y A3, se repite el proceso anterior para así obtener la ubicación de los voxels que las integran, como se muestra en la Figura 2-54.

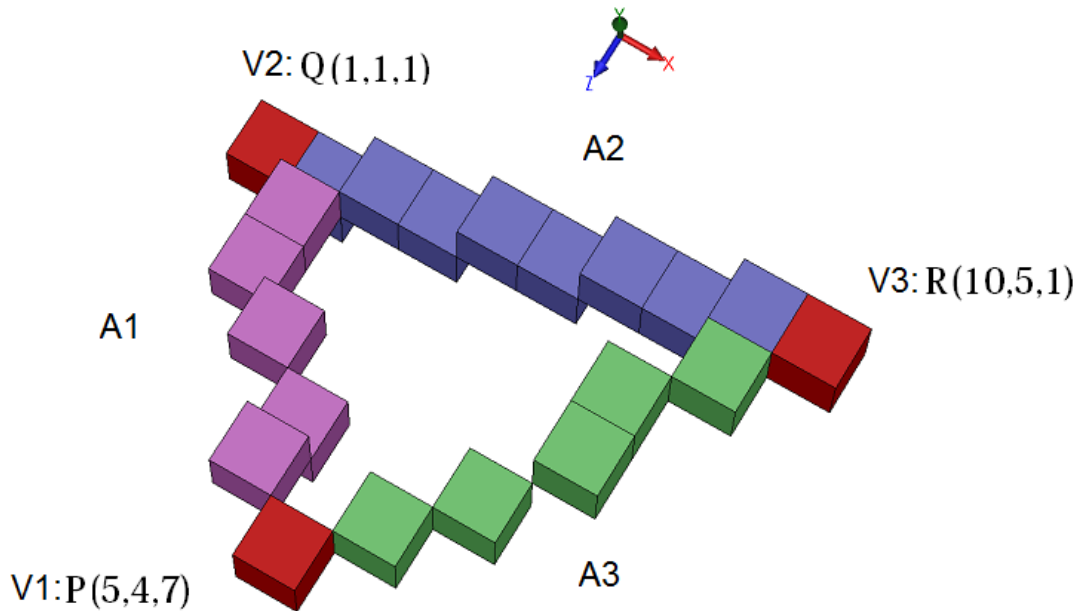


Figura 2-54. Construcción de las aristas del triángulo

2.2.1.3 Cálculo de voxels en la superficie del triángulo

Para calcular la superficie del triángulo se desarrolló el método llamado “radial”, el cual consiste en seleccionar la arista de mayor tamaño (AM) y el vértice opuesto (VO), para después unir el vértice opuesto con cada uno de los voxels que integran la arista mayor, utilizando la rutina de unión de puntos, explicado en el apartado 2.2.1.2.1, como se muestra en la Figura 2-55.

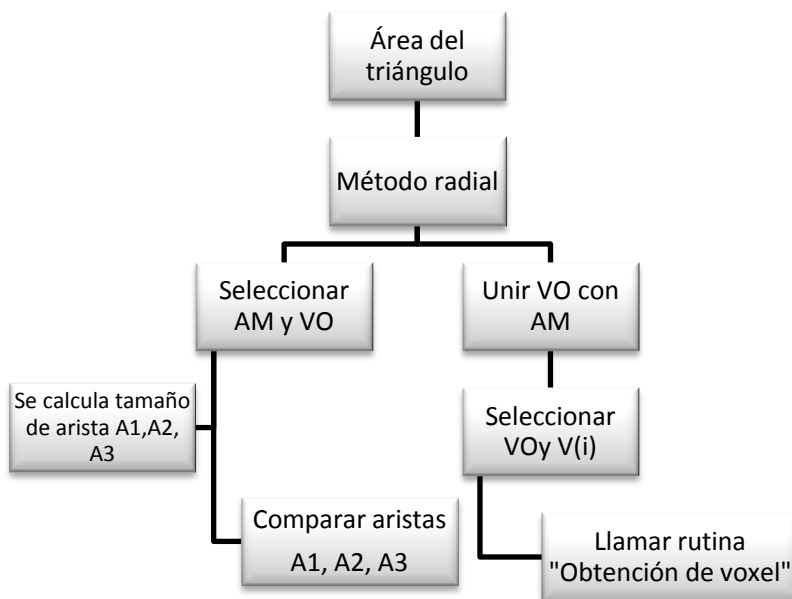


Figura 2-55. Proceso para obtener el área de un triángulo

2.2.1.3.1 Determinar la arista mayor y el vértice opuesto

Para determinar la arista mayor, se realizan las diferencias de los vértices que forman las aristas, posteriormente se comparan y se elige la de mayor magnitud, como se muestra en la Figura 2-56.

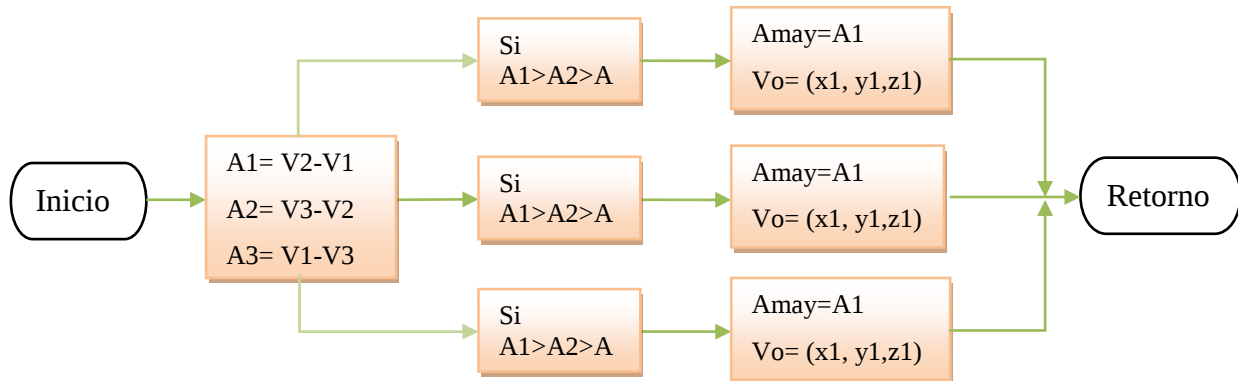


Figura 2-56. Proceso de determinar la arista mayor

Para este ejemplo A2 está constituido por V2 y V3 que se le asigna a la variable “Amay”, y el vértice opuesto a la arista es el vértice V1 y se le asigna a la variable “VO”.

Una vez identificados la arista mayor y el vértice opuesto lo que continúa es realizar el cálculo de los voxels que forman el área con el método radial, que consiste en unir el VO con cada uno de los voxels que integran la AM, como se muestra en la Figura 2-57. Para obtener los voxels, se toman los valores de las coordenadas del VO y las del voxel “Va” y se llama a la Rutina de unión de dos puntos, en la sección 2.2.1.2.1.

Al terminar con esa línea, se pasa a la siguiente constituida por VO y “Vb”, y se calculan los voxels. Dicho procedimiento se repite hasta terminar.

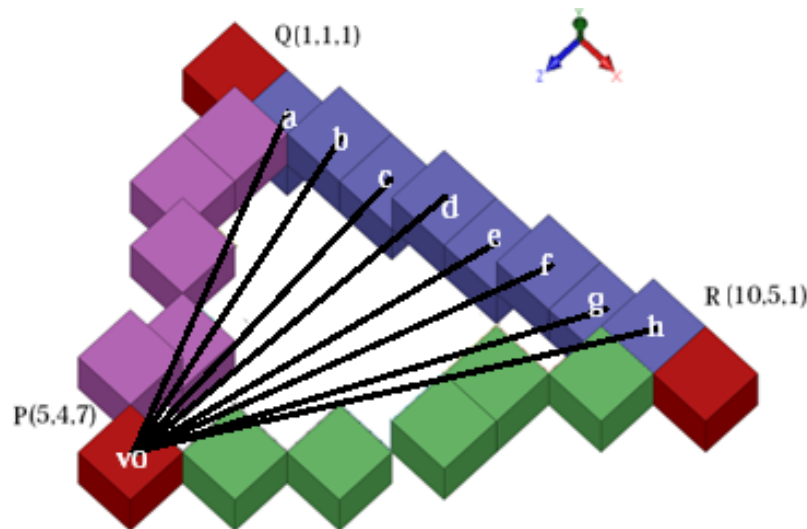


Figura 2-57. Método radial para obtención de voxels del área

De esta manera, se obtienen los voxels que integran la superficie del triángulo, como se muestra en la Figura 2-58.

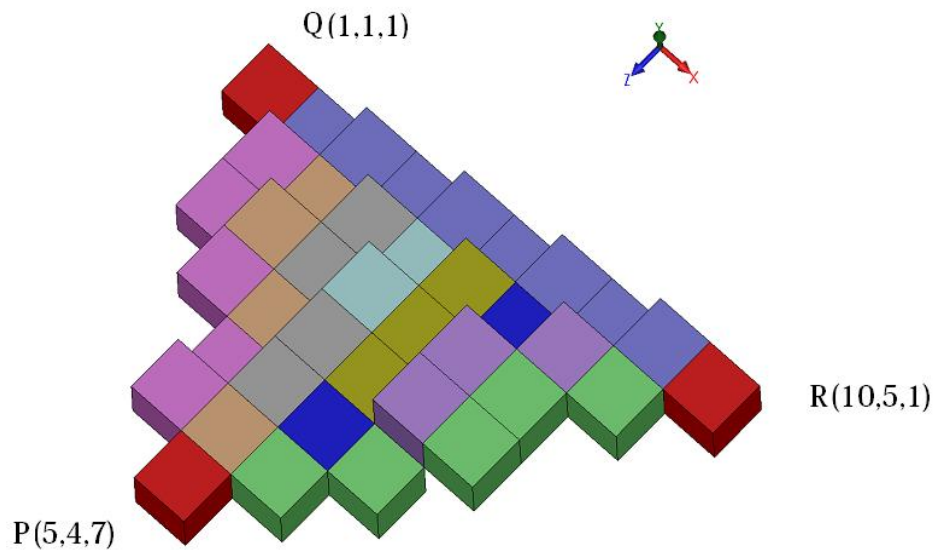


Figura 2-58. Ubicación de voxels del área del triángulo

2.2.1.4 Obtención de los voxels que integran la superficie total del sólido

Para obtener todos los voxels que integran la superficie de la pieza dibujada, se repiten las rutinas anteriormente descritas, para todos los triángulos que la conforman, obteniéndose una gran lista de coordenadas (x,y,z). Lográndose así la construcción virtual de la superficie del sólido dibujado en CAD, como se muestra en la .

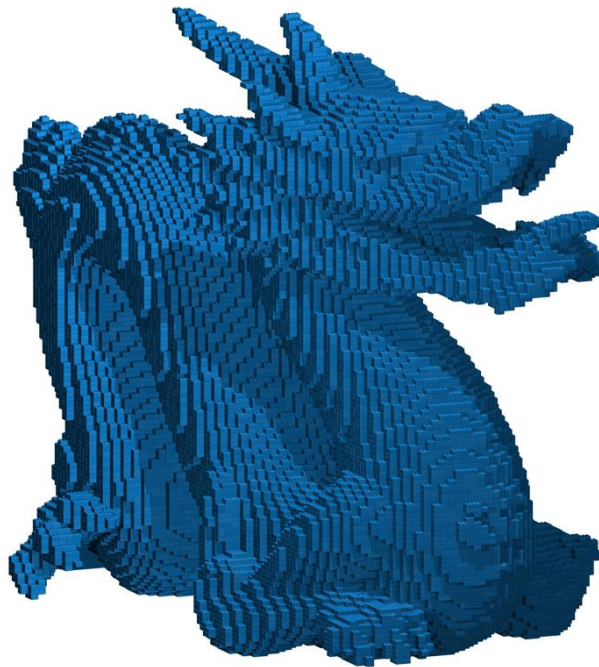


Figura 2-59. Construcción virtual de la superficie

2.2.2 Obtención de perfil

Una vez que se tiene la construcción virtual de la superficie, lo que continúa es rebanarlo, para obtener de cada una un perfil, el cual deberá seguir la herramienta de corte para maquinar la pieza como muestra la Figura 2-60, siendo las rebanadas los voxels que tiene el mismo color.



Figura 2-60. Ejemplo de rebanado de la superficie virtual

La rutina de obtención de perfil consiste en obtener una trayectoria continua que es la trayectoria que deberá seguir la herramienta de corte. Para ello, primeramente se rebana la pieza, para obtener un contorno, el cual puede presentar problemas como huecos y duplicidades, para ello se realizan varios procesos en la imagen para poder tener un contorno cerrado y poder obtener la trayectoria deseada, como se muestra en Figura 2-61.

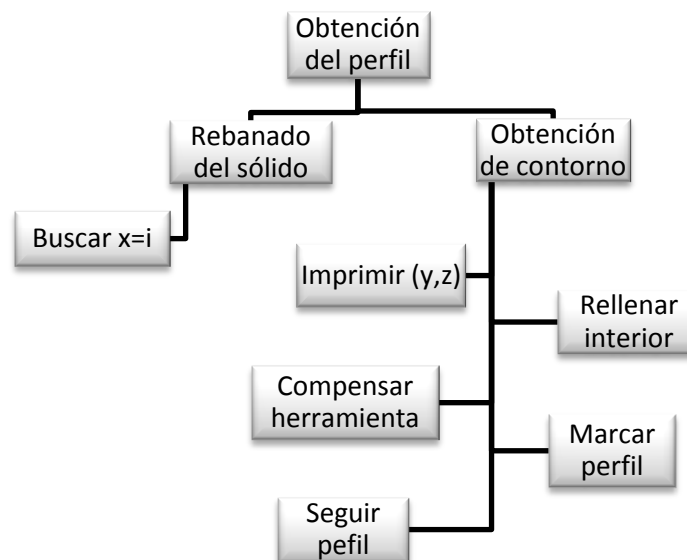


Figura 2-61. Proceso de la obtención del perfil

2.2.2.1 Obtención del contorno

Para la obtención del contorno de la rebanada, se realiza una búsqueda en la localidad de memoria donde se ubican todos los voxels calculados de la superficie del sólido, aquellos cuyo valor en la coordenada “x” sean iguales, y se imprimen en color blanco las coordenadas (y, z) de dichos voxels, sobre un lienzo de color negro. En la rebanada, los voxels del contorno pasan a ser pixeles.

En la implementación no fue necesario acomodar los valores de las coordenadas, dado que consumía mucho tiempo computacional, y es mucho más rápido buscar en la lista el valor solicitado en “x” e imprimir directamente.

Al imprimir los valores, se elimina el problema de las redundancias, puesto que simplemente se reimprimen en el mismo lugar, sin embargo, los huecos y las duplicidades se resuelven con operaciones morfológicas en la imagen.

Como se está hablando de una imagen binaria se tienen dos valores: el “0” y el “1”, el primero representa los valores en color negro y el segundo en color blanco, sin embargo, cuando se guarda una imagen en la computadora, se almacena en 1 byte (8 bits), por lo que se pueden ocupar 256 valores.

Una imagen que se obtiene de la pieza rebanada es como la que se muestra en la Figura 2-62, donde se observa que existen pixeles en el contorno donde hay huecos que se generaron en la pieza.

Además se aprecia que el contorno no es uniforme en cuanto al grosor debido a la duplicidad y dado que el espesor deseado es de 1 pixel, el contorno debe manipularse para obtener un perfil continuo y con las características descritas para poder ser seguido. Para ello definiremos primeramente el interior y exterior de la pieza.

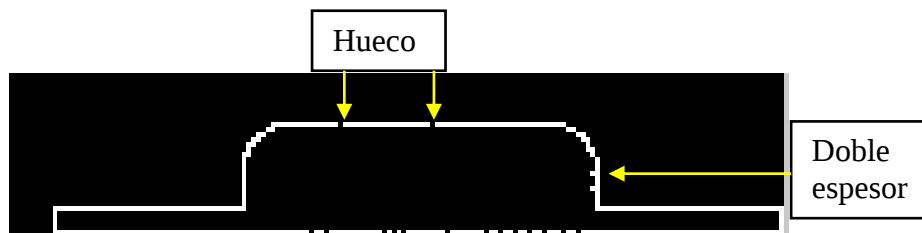


Figura 2-62. Imagen de contorno con huecos y doble espesor

2.2.2.2 Rellenado del interior del contorno

El relleno interior del contorno sirve para poder identificar lo que es parte de la pieza, para corregir las fallas en el contorno y obtener el perfil. Se realiza en dos etapas, primeramente se hace un barrido de arriba hacia abajo para identificar el primer punto en blanco, que corresponde al contorno, y después un barrido horizontal para eliminar las secuelas del hueco en el contorno



Figura 2-63. Diagrama de la rutina para el relleno del contorno

Se sabe que el contorno es parte de la superficie de la figura y se puede determinar qué es interno y qué es externo, para ello se realiza el algoritmo de llenado de pixeles internos, que consiste en buscar en cada columna el primer pixel en blanco de arriba hacia abajo, es decir, con un valor de 1 y a partir de él cambiar a blanco todos los pixeles que se encuentren debajo hasta llegar al límite, como se explicó en la sección 2.2.2.2. A continuación se muestra en el Diagrama de flujo 2-3 la rutina de relleno del interior de la figura.

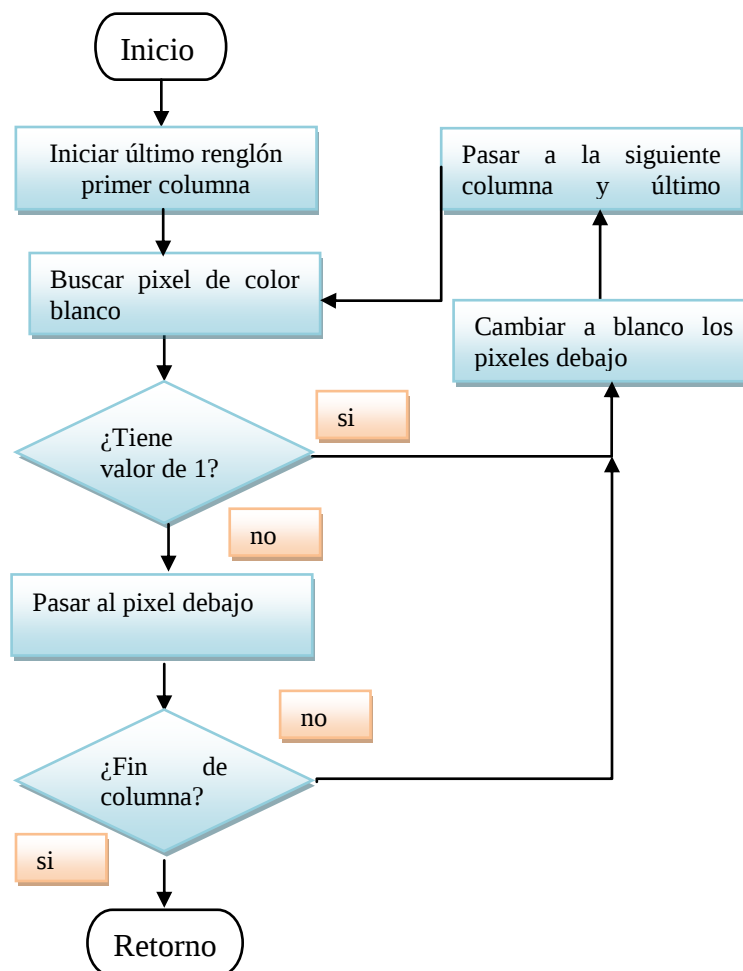


Diagrama de flujo 2-3. Algoritmo de relleno interno del contorno

Entonces la imagen queda como se muestra en la Figura 2-64.

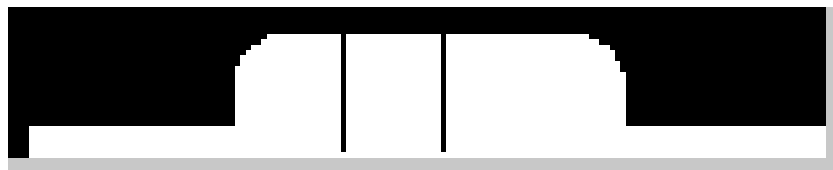


Figura 2-64. Imagen con el interior llenado y con secuelas de los huecos

Se aprecia el daño que ocasionan los huecos dado que se prolonga esa ausencia a lo largo y para ello necesitamos ocupar otro procedimiento para corregir la falla.

2.2.2.3 Rellenado de huecos

Este método consiste en llenar huecos que existan dentro del área y en el contorno para garantizar una trayectoria cerrada. Se realiza el proceso de dilatación con un elemento estructural de vecindad-2 horizontal como se muestra en la Figura 2-26, debido a que los huecos son de 1 pixel. La imagen después de este proceso se muestra a continuación en la Figura 2-65.



Figura 2-65. Imagen binaria después de rellenar huecos

A continuación se muestra el Diagrama de flujo 2-4 la operación de rellenado de huecos

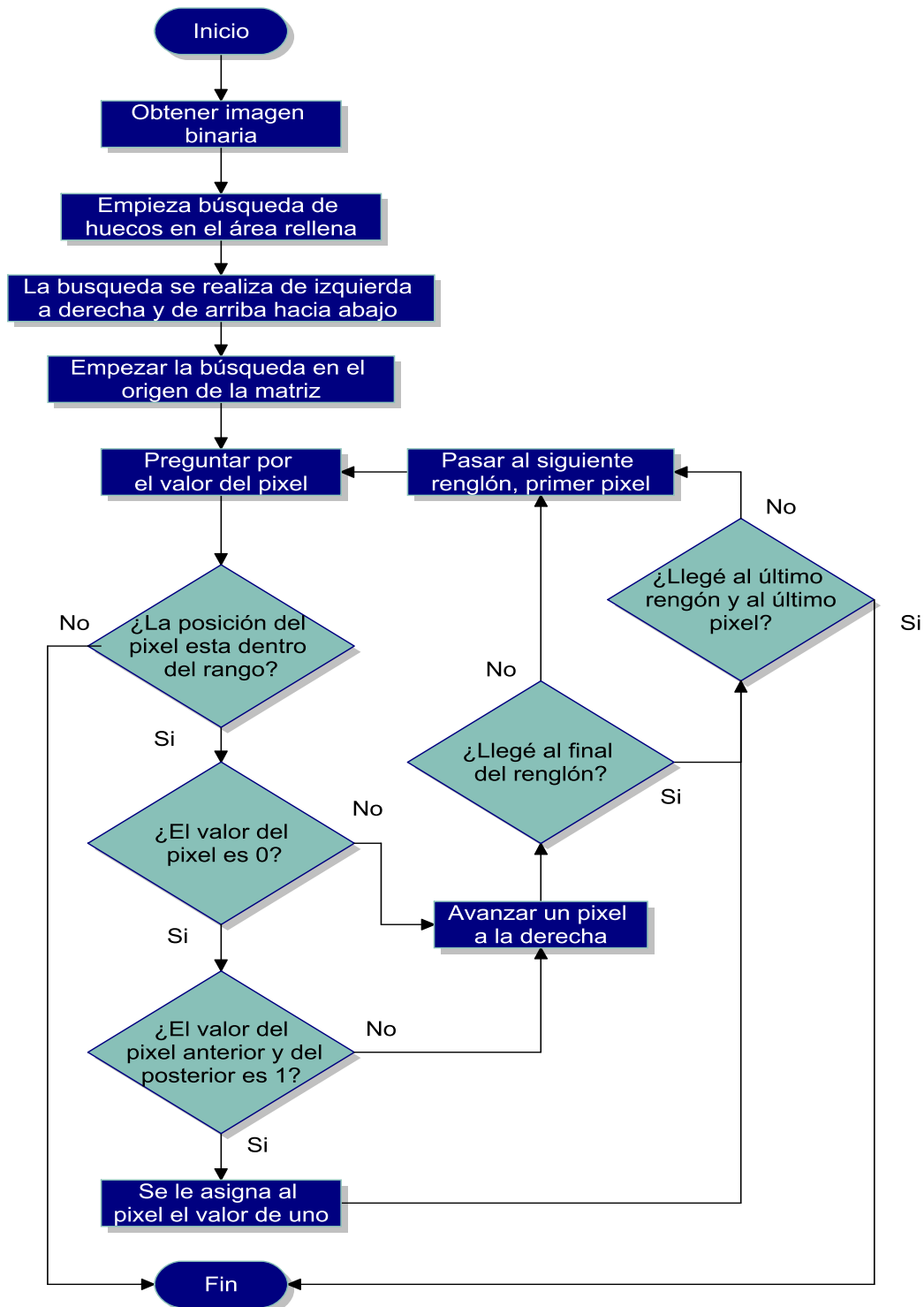


Diagrama de flujo 2-4. Algoritmo para el llenado de huecos

2.2.2.4 Eliminación de protuberancias de un pixel

Existen casos en la imagen donde existen protuberancias verticales con un ancho de un pixel y es necesario eliminarlo o ensancharlo con la finalidad de evitar problemas posteriores en el seguimiento del perfil, de esta manera mejora la apariencia de la figura (ver ejemplo en la sección 2.1.2.5, pág. 80).

Para ello se ocupa el proceso de cierre que consiste en dilatar y erosionar la figura con un EE de 3x3. Como muestran los algoritmos en el Diagrama de flujo 2-5 y Diagrama de flujo 2-6, donde el de dilatación el EE busca en su vecindad-8 si alguno tiene el valor de 1, el elemento director cambia su valor a 1.

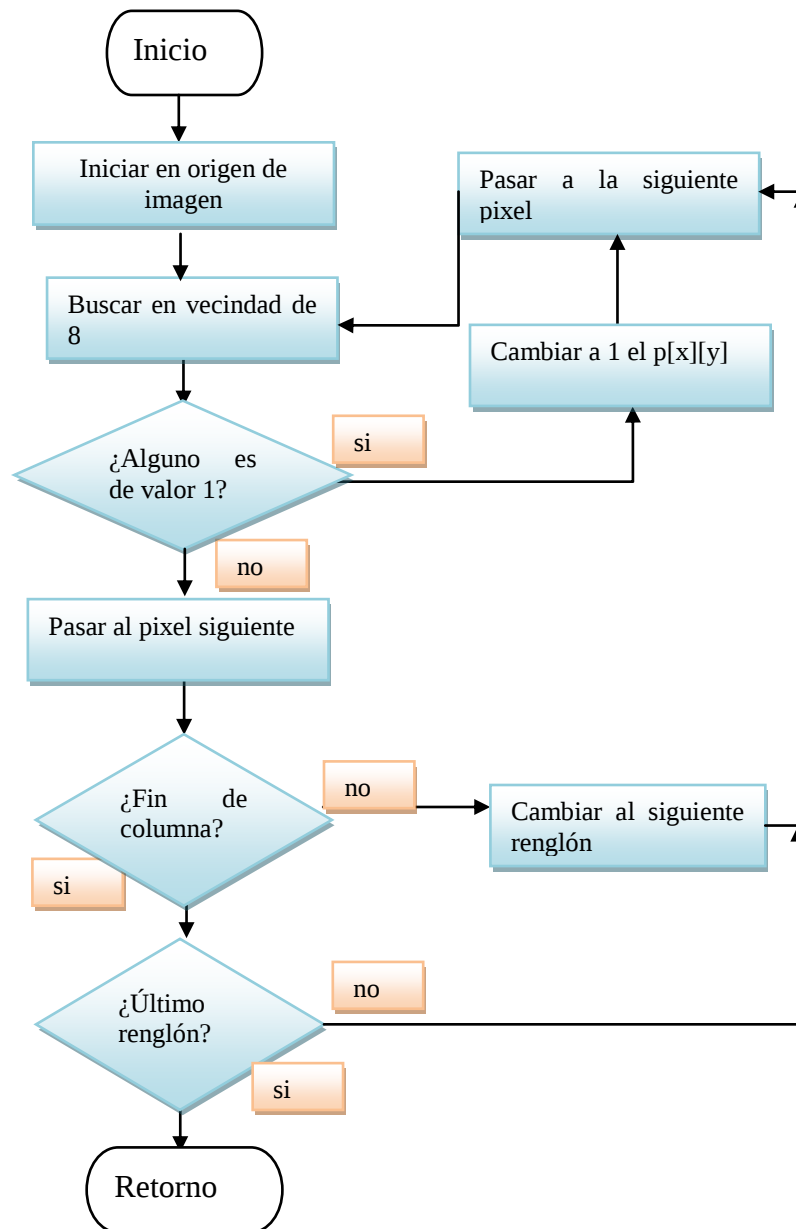


Diagrama de flujo 2-5. Algoritmo de dilatación

A continuación se muestra el diagrama de flujo de la rutina de erosión de la imagen, el cual consiste en que el EE busca en la vecindad-8 si todos tienen un valor de '1', el elemento director cambia su valor a '0'.

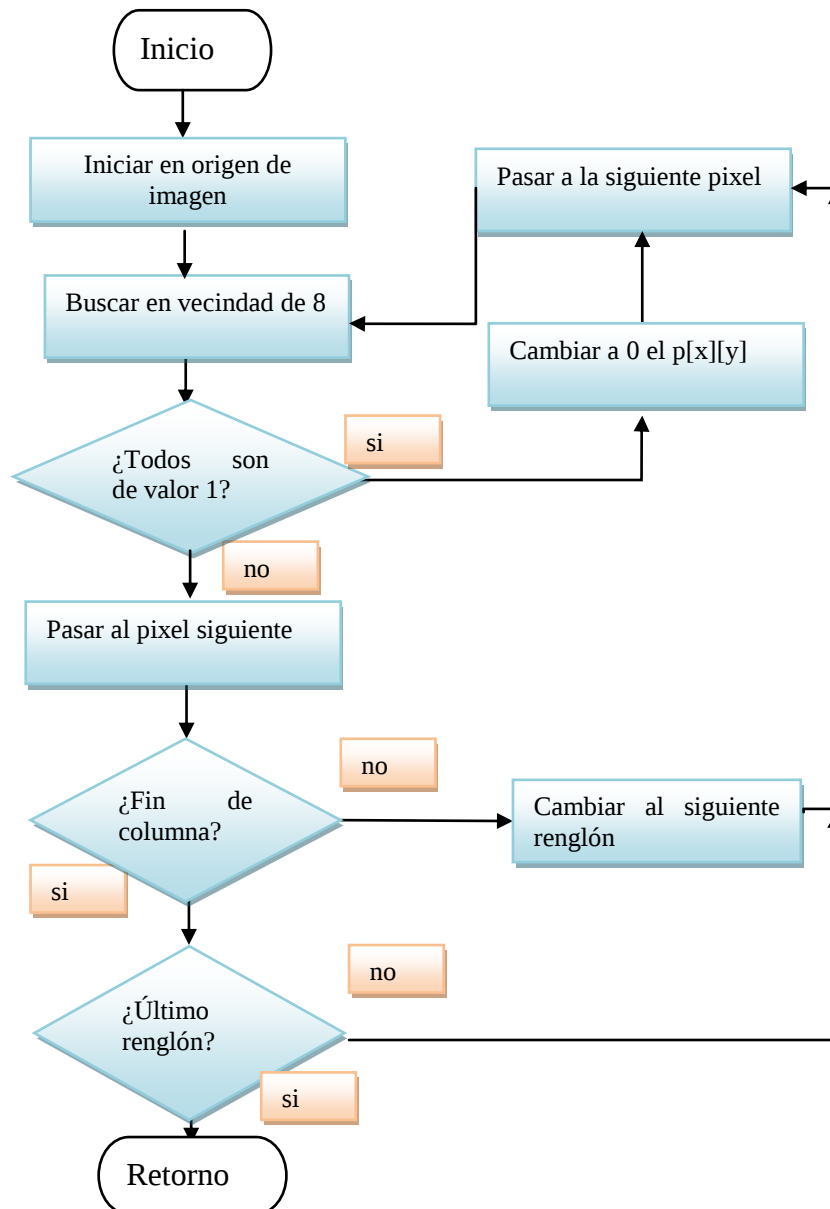


Diagrama de flujo 2-6. Algoritmo de erosión

2.2.2.5 Identificación del perfil

La identificación consiste en marcar todos los píxeles exteriores de la figura, exceptuando la base horizontal, con un valor distinto de los demás para señalar que se trata del perfil que debe ser seguido, y se realiza con dos procesos como se muestra en la Figura 2-66.



Figura 2-66. Proceso para la identificación del perfil

Una vez que se tiene el contorno relleno y sin huecos se procede a identificar el perfil a seguir, para ello se utilizan los siguientes procedimientos.

Para marcar el contorno de la figura, se hace en dos partes:

- Barrido vertical
- Barrido horizontal

El primero consiste en hacer un barrido de arriba hacia abajo y de izquierda a derecha; en donde se pregunta por el valor de cada píxel y al encontrar aquel que tenga el valor de 1, que obviamente va a ser el píxel más externo, su valor cambiará a 2, y la búsqueda se repite en la columna siguiente, así sucesivamente hasta terminar.

Con este método se identifican los píxeles exteriores de las líneas horizontales, y de algunas curvas, sin embargo, aquellos que integran las líneas verticales no fueron identificadas, es por ello que se realiza la siguiente operación

La segunda parte consiste en hacer un barrido de izquierda a derecha y de arriba hacia abajo, en donde se pregunta por el valor del píxel actual ' $p[x][y]$ ' y del siguiente ' $p[x+1][y]$ ', si el valor del primero es 0 y del segundo es 1, el píxel ' $p[x+1][y]$ ' cambia su valor a 2.

Y si los valores del píxel actual ' $p[x][y]$ ' y del siguiente ' $p[x+1][y]$ ' son 1 y 0, respectivamente, el píxel ' $p[x][y]$ ', cambia su valor a 2, así sucesivamente hasta terminar con el renglón.

El procedimiento de marcar contorno flujo se muestra a continuación en el Diagrama de flujo 2-7.

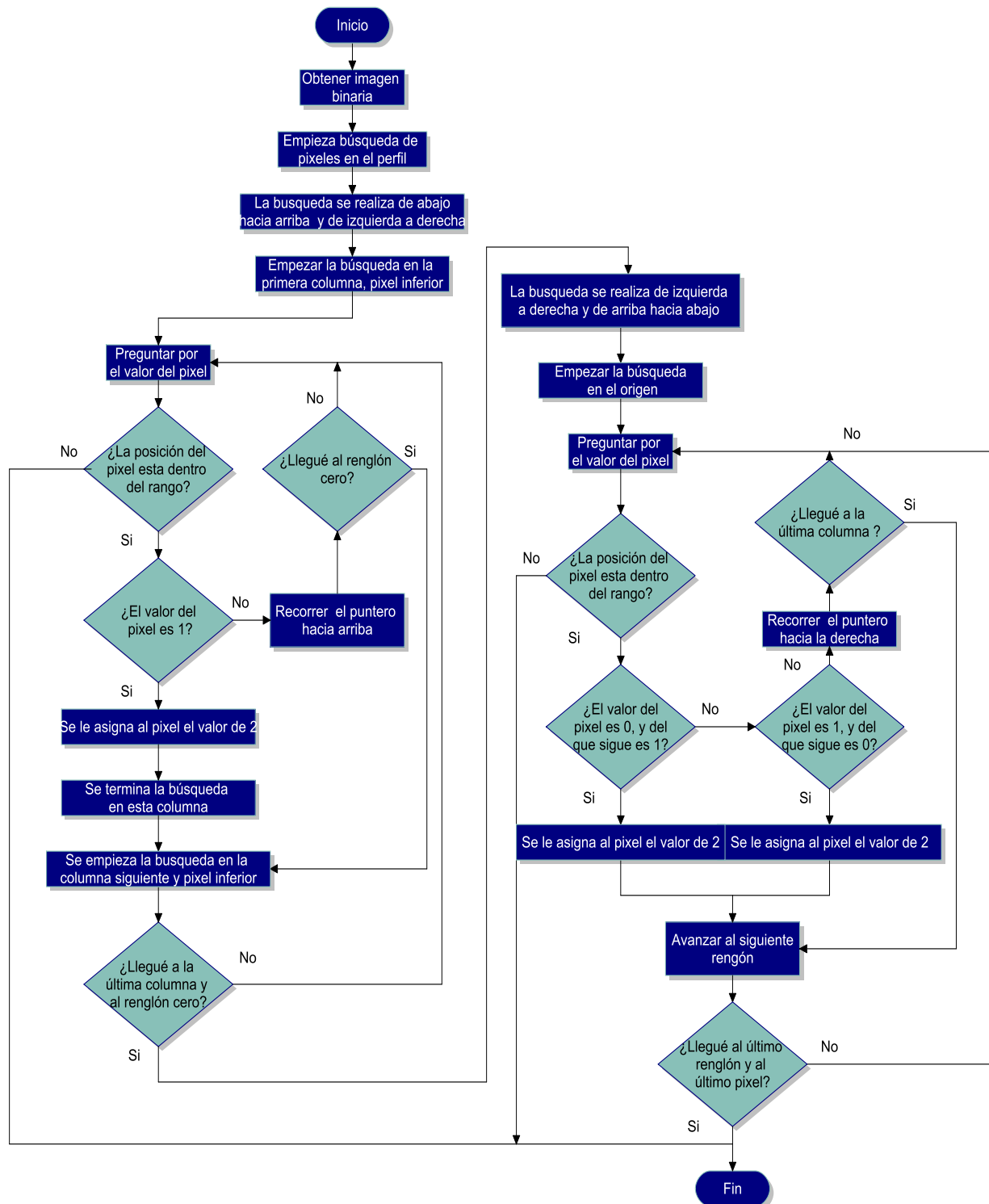


Diagrama de flujo 2-7. Algoritmo de marcar contorno

2.2.2.6 Compensación de la herramienta de corte

Como se explicó en la sección 2.2.2.6, es necesario hacer la compensación de la herramienta para que no importando hacia qué lado corte, izquierda o derecha, respete la geometría de la figura, para ello se realiza una dilatación hacia los lados con un EE de vecindad 2-horizontal, la dilatación incrementa en 1 pixel por lado, y se dilata las veces necesarias para que corresponda al radio de la herramienta.

El barrido del EE de vecindad 2-horizontal, se hace de izquierda a derecha y de arriba hacia abajo. En donde, si cualquiera de los tres pixeles: $p[y][x-1]$, $p[y][x]$, $p[y][x+1]$ tiene un valor de 1 el pixel $p[y][x]$, cambia su valor a 1. Todos los valores se almacenan en otra imagen binaria para no modificar la original.

Realizando la equivalencia física de longitud de un pixel dada la resolución de la máquina, un pixel mediría 0.02 milímetros si la herramienta midiera 1.5 mm de radio, el número de dilataciones sería de 75, para dilatarse 75 pixeles por lado. Una vez teniendo el contorno compensado, se obtiene el perfil de la rebanada como se muestra en la Figura 2-67.

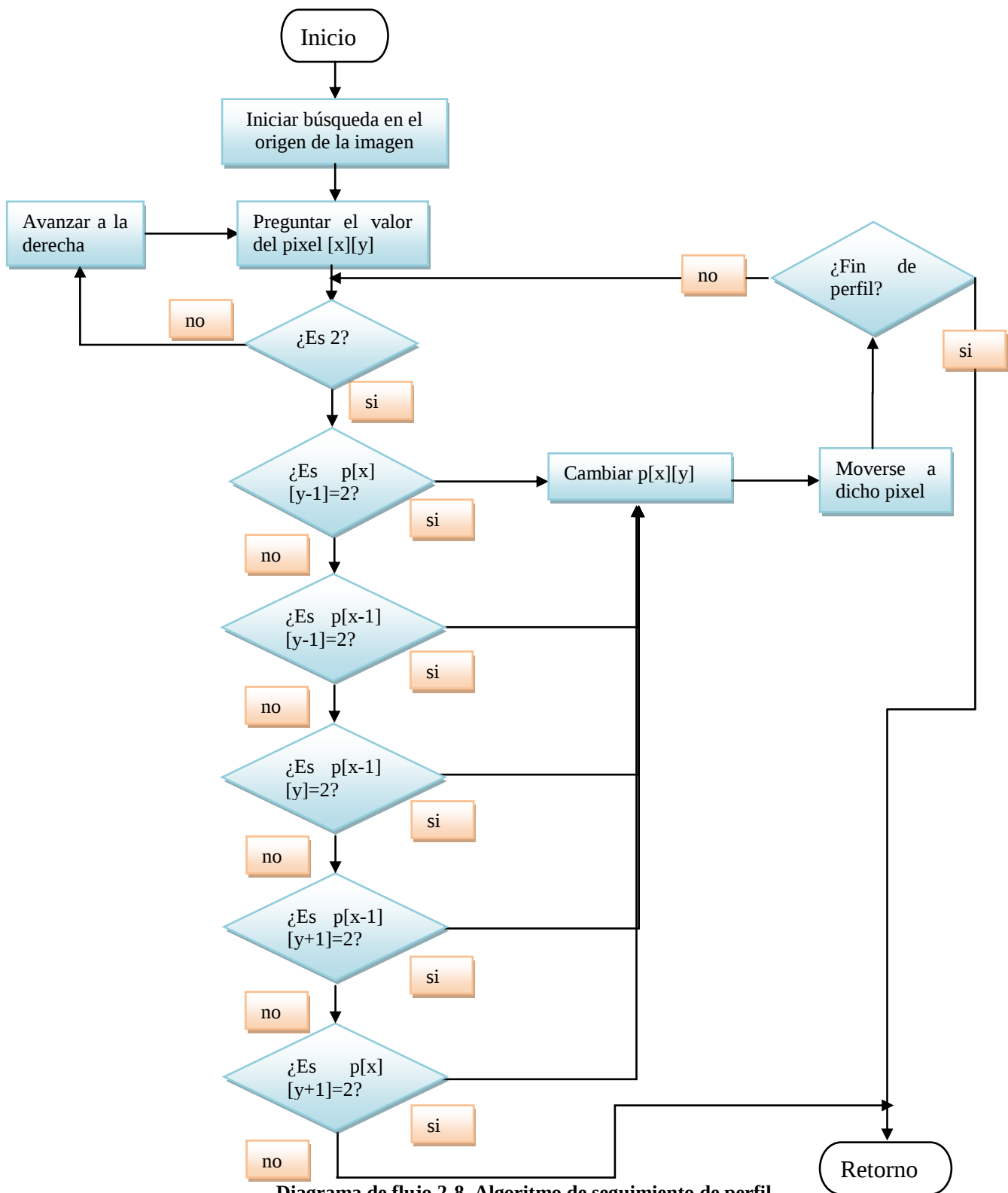


Figura 2-67. Imagen binaria con compensación de la herramienta

2.2.2.7 Seguimiento de perfil

Una vez teniendo la imagen dilatada por la compensación de la herramienta y habiendo marcado el contorno con un valor de 2 es turno de seguir la trayectoria del perfil, para ello se hace un barrido de izquierda a derecha con un EE de 3x3 hasta que encuentra el primer punto del perfil.

Posteriormente se empieza a buscar en la vecindad-8 el pixel con el valor de 2 en sentido anti horario empezando por el pixel debajo: ' $p[y+1][x]$ ', y cuando lo encuentra, el pixel actual ' $p[y][x]$ ' cambia su valor a 3 y el EE se mueve hacia el pixel con valor 2 y vuelve a hacer la búsqueda hasta completar el perfil. De esta manera, no hay forma de que se regrese por el camino ya recorrido, como se muestra en la Figura 2-68



2.2.3 Movimiento de los motores

En esta sección se aborda la manera en que se obtienen los movimientos de los motores a partir del perfil obtenido, cuya información se manda por el puerto paralelo.

2.2.3.1 Obtención de desplazamientos en los ejes Y y Z

Para obtener los desplazamientos de los ejes Y y Z, se realiza con ayuda del algoritmo de seguimiento de perfil, es decir, por cada desplazamiento que se realiza de un pixel a otro, se manda una un dato para que se ejecute el movimiento pertinente de los motores. Según la dirección de búsqueda de perfil que muestra la Figura 2-70, es el dato que envía para mover la herramienta como se muestra en la Tabla 2-14.

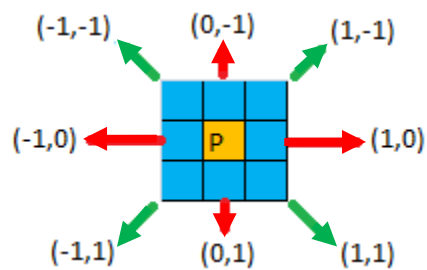


Figura 2-70. Dirección de búsqueda del perfil

Esto es que al tiempo en que se deslaza siguiendo el perfil, se manda la instrucción por el puerto paralelo para realizar los movimientos de los motores correspondientes, para maquinar el mismo perfil.

A continuación se muestra en la Tabla 2-14 el desplazamiento que realiza la herramienta de acuerdo al movimiento del EE cuando realiza el seguimiento del perfil.

Tabla 2-14. Desplazamiento del EE en el contorno y los movimientos de la herramienta

Desplazamiento del EE	Dirección del motor del eje Y	Dirección del motor del eje Z	Desplazamiento	
(0,1)	Arriba	-	Vertical	
(1,1)	Arriba	Derecha	Diagonal ascendente	
(1,0)	-	Derecha	Horizontal	
(1,-1)	Abajo	Derecha	Diagonal descendente	
(0,-1)	Abajo	-	Vertical	
(-1,-1)	Abajo	Izquierda	Diagonal descendente	
(-1,0)	-	Izquierda	Horizontal	
(-1,1)	Arriba	Izquierda	Diagonal ascendente	

La rutina de seguimiento del perfil manda el dato correspondiente para que la herramienta se mueva de tal forma que imite el movimiento y describa la misma trayectoria, como muestra la Figura 2-71

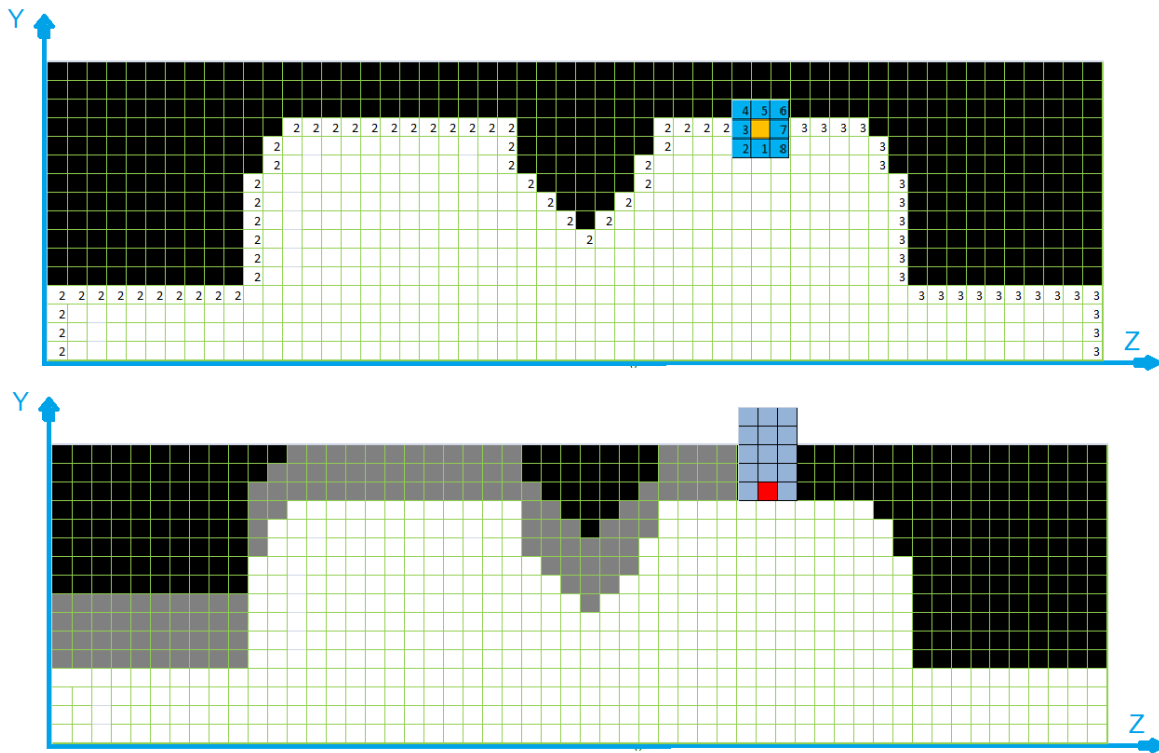


Figura 2-71. Desplazamiento simultáneo entre el EE sobre el perfil y de la herramienta de corte

Cuando el mecanismo termina de maquinarse el perfil y al avanzar a la siguiente rebanada, es decir, cuando los valores en “x” pasan al siguiente valor, el dato que se manda por el puerto paralelo es para que se mueva el motor del eje “x”, y vuelve a repetirse el proceso de maquinado.

En el siguiente capítulo se habla precisamente de los datos que manda el programa al puerto paralelo y de los dispositivos electrónicos necesarios para ello.

2.2.4 Interfaz gráfica

La interfaz gráfica de diseñador que se muestra a continuación ha sido utilizada para la prueba del desarrollo del software en cuanto a los algoritmos diseñados y explicados anteriormente. La interfaz gráfica para usuario, se muestra más adelante en la Figura 2-72.

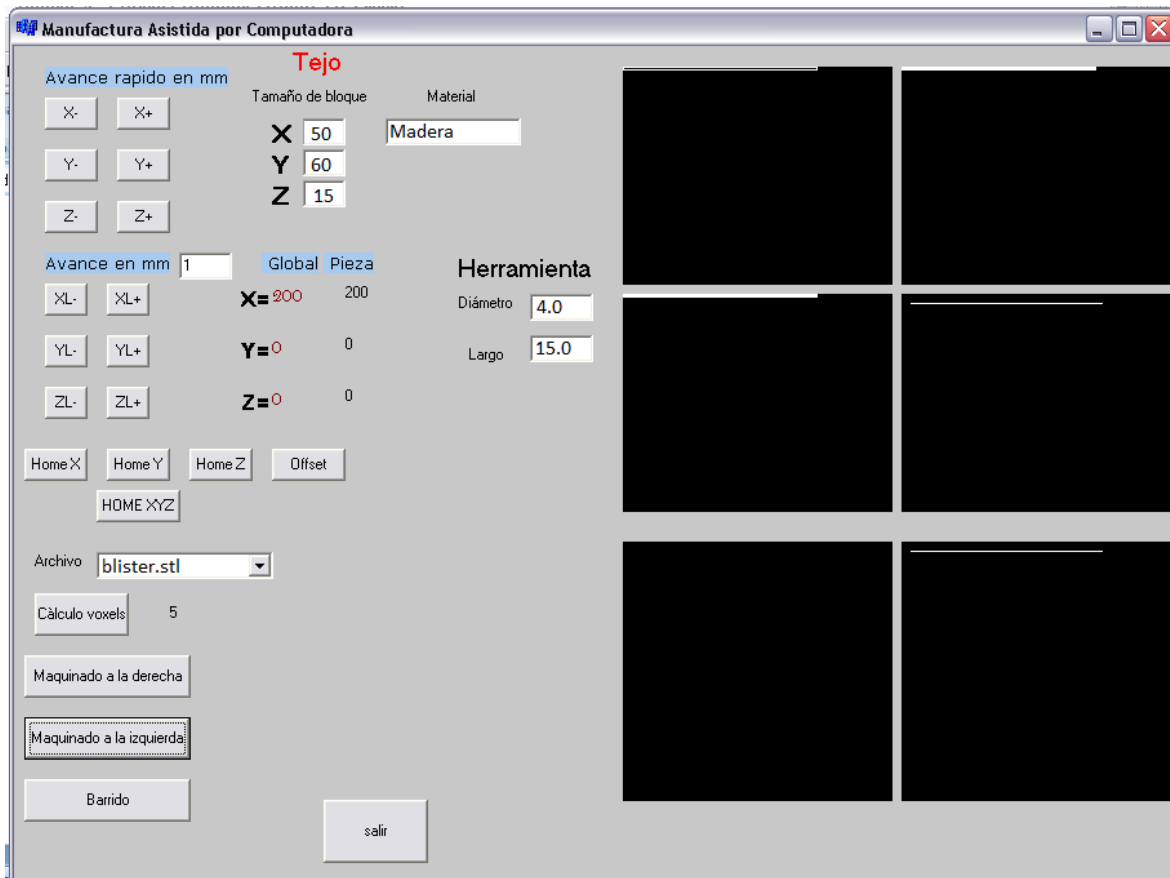


Figura 2-72. Interfaz de desarrollador

La interfaz gráfica ha sido desarrollada en lenguaje C en un ambiente de Windows, donde es posible apreciar 5 pantallas negras, una sección de botones distribuidos según las funciones que proporcionan y algunos cuadros con información.

En la interfaz, se tiene un apartado donde se busca el nombre del archivo a fabricar, con la restricción de incluir únicamente archivos con extensión stl.

De la misma manera, se incluyen las medidas del tejo y el tipo de materia, que por el momento es únicamente indicativo, no hay influencia para el proceso de maquinado.

La interfaz consta de una ventana que contiene botones de movimiento de los ejes X, Y, Z, una con desplazamientos determinados de 5 milímetros, llamado de avance rápido y la otra con desplazamientos fraccionales de hasta 0.02 milímetros, con la escritura en el cuadro de texto correspondiente, siendo el valor inicial de 1 milímetro, como se muestra a continuación en la Figura 2-73.

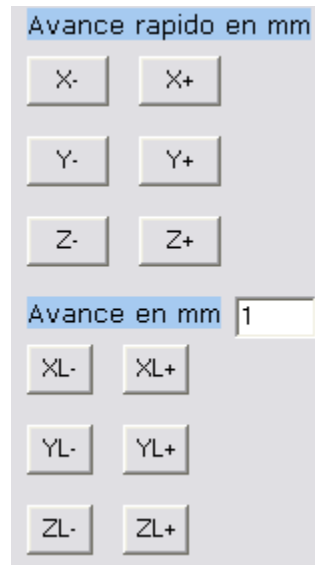


Figura 2-73. Botones de movimiento de ejes

Existen también 4 botones de inicio, uno para cada eje, (x,y,z), y el otro para los tres simultáneos como se muestra en la Figura 2-74. Estos botones realizan el movimiento del eje o ejes en un sentido, y después hacia el otro, hasta encontrar el sensor de inicio. Es en ese punto espacial es donde la máquina tiene una ubicación de (0,0,0) y es lo que se conoce como el origen de la máquina.

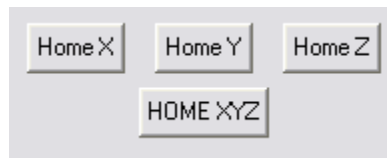


Figura 2-74. Botones de inicio de los ejes

También en la interfaz se muestra la posición de la herramienta en los 3 ejes, teniendo dos ubicaciones, una global y otro local, la primera se refiere a la ubicación con respecto al origen de la máquina, y el otro es cuando se encuentra en el vértice del bloque a maquinar, es decir, en el origen de la pieza (Figura 2-75).

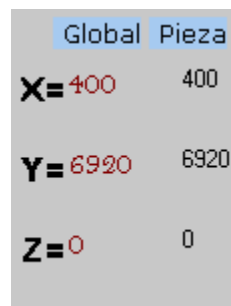


Figura 2-75. Displays de la ubicación en cada eje de la herramienta

Cuando la máquina se lleva a HOME, la posición global y local se inicializa en cero, y después de desplazarse, los valores se van actualizando de acuerdo a la ubicación en que estén, y para inicializar la ubicación local, cuando se está listo a maquinarse, se utiliza el botón de “offset”, ubicado debajo de los displays que muestran la ubicación como se muestra en la Figura 2-76.



Figura 2-76. Botón de inicio de la ubicación local de la herramienta

El inicio del cálculo de los voxels que integran la superficie de la pieza dibujada empieza al oprimir el botón “cálculo de voxels”. Aunque no se visualice nada ya se tienen las ubicaciones de los voxels (Figura 2-77).

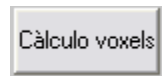


Figura 2-77. Inicio de cálculo de voxels de la superficie

Posteriormente para obtener el perfil deseado para su seguimiento por parte de la herramienta de corte, se da clic al botón “Maquinado a la derecha” y se obtienen 6 pantallas negras con imágenes binarias, las cuales aparecen giradas o de cabeza, dado que el origen de toda imagen es en la esquina superior izquierda (Figura 2-78).

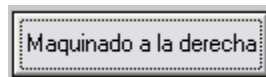


Figura 2-78. Maquinado hacia la derecha

La primera pantalla corresponde a la rebanada tomada directamente de la impresión de los voxels que tienen el mismo valor en la coordenada “x” (Figura 2-79).

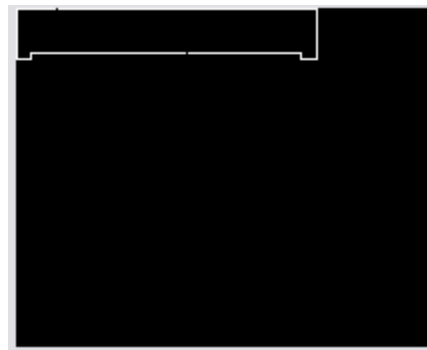


Figura 2-79. Rebanada obtenida de la impresión de voxels

La segunda corresponde a la imagen tratada con el algoritmo de rellenar el interior de la pieza (Figura 2-80).



Figura 2-80. Imagen del llenado interno de la pieza

La tercer imagen es corresponde cuando ha sido tratada con el algoritmo de llenado de huecos y con el proceso de la cerradura (Figura 2-81).



Figura 2-81. Imagen sin huecos internos

Y después se tiene la imagen del perfil con la compensación de la herramienta como se muestra en la Figura 2-82

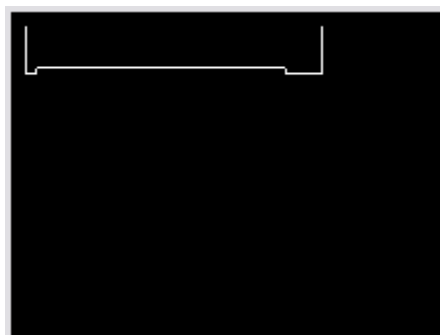


Figura 2-82. Perfil a seguir

La quinta y la sexta pantalla corresponden a la trayectoria que siguió la herramienta dependiendo hacia qué lado maquinó, si fue a la derecha aparecerá la trayectoria en la quinta pantalla y si fue a la izquierda aparecerá en la sexta, y que obviamente deben ser iguales a la cuarta imagen (Figura 2-83).



Figura 2-83. Imagen de la trayectoria descrita por la herramienta

Estas imágenes se obtuvieron al oprimir el botón “maquinar a la derecha” para realizar la prueba del maquinado de la rebanada 1 y dado que la herramienta se trasladó de izquierda a derecha, bien puede regresar por el mismo camino realizando la misma trayectoria para llegar al punto donde inició, pasar a la siguiente rebanada y avanzar en el eje “x” y continuar el maquinado de izquierda a derecha.

Sin embargo, para reducir el tiempo de maquinado y la energía consumida a la mitad, una vez terminada de maquinar la primera rebanada de izquierda a derecha en vez de regresar por el mismo camino, se avanzaría a la siguiente rebanada y se maquinaría en sentido contrario.

Para ello, una vez finaliza el maquinado, inmediatamente después se avanza en el eje “x”, como parte del proceso de maquinado. Después se oprime el botón “Maquinar a la izquierda”, y se vuelven a tener 5 pantallas con imágenes de la siguiente rebanada, como las que se explicaron anteriormente, posteriormente se sigue el perfil en sentido contrario y se maquina de derecha a izquierda. Es aquí donde se aplica el barrido como la estrategia de maquinado (Figura 2-84).

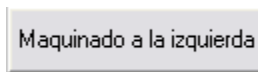


Figura 2-84. Botón de maquinado hacia la izquierda

Hasta ahorita se han maquinado 2 rebanadas, y lo que seguiría a continuación es oprimir el botón “Maquinado a la derecha” y después el “Maquinado a la izquierda”, hasta terminar con la pieza. Sin embargo, para hacer este procedimiento de manera automática desde el principio se oprime el botón “Barrido” después de haber oprimido el botón “Cálculo de voxels”, aunque de esta manera no se visualizará por mucho tiempo el resultado en la quinta y sexta imagen (Figura 2-85).

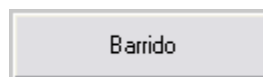


Figura 2-85. Botón de maquinado hacia la izquierda

En resumen la interfaz gráfica de diseñador se puede inicializar la herramienta de corte, realizar el movimiento manual de la herramienta de corte, distribuye de la siguiente manera, como se muestra en la Figura 2-86:

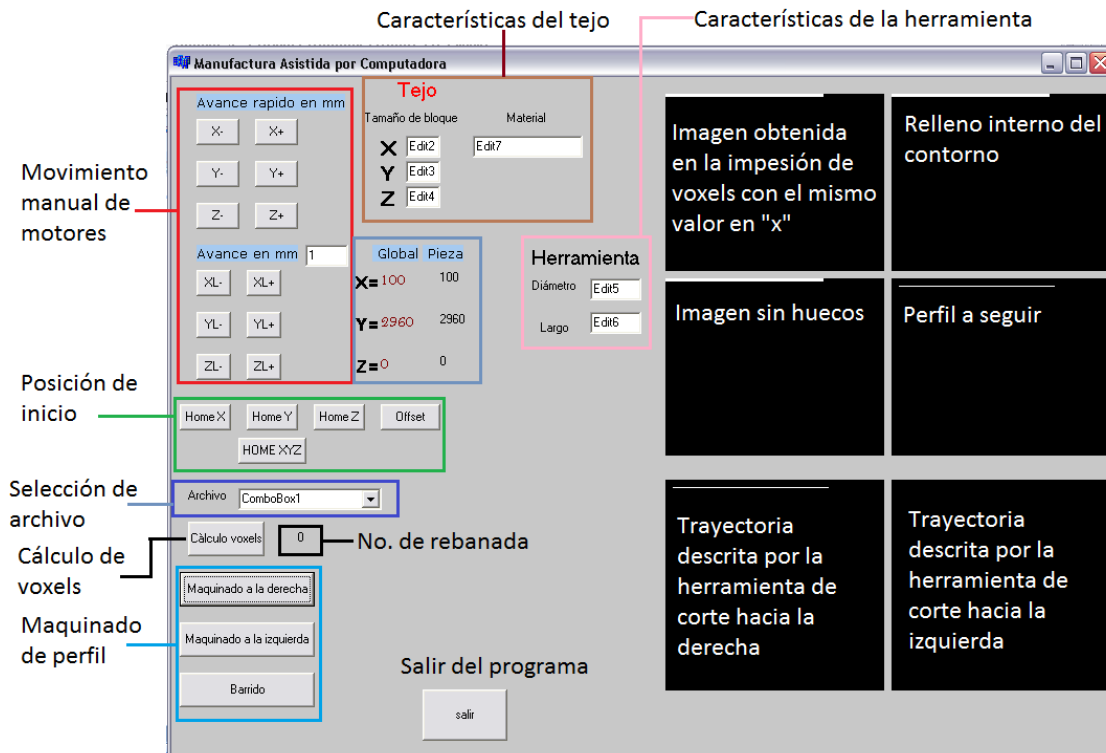


Figura 2-86. Identificación de las partes de la interfaz de diseñador

2.2.5 Interfaz de usuario

La interfaz del usuario, es decir, la parte visual del programa en donde el operador puede interactuar con la máquina se ha realizado en lenguaje C ++, uno de los lenguajes de programación más utilizados.

En esta pantalla el operador puede mover la herramienta a lo largo, ancho y alto del espacio de trabajo, además de poder mandarla a home y de aproximarse a la pieza a maquinar de forma sencilla.

En la interfaz el operador importa el archivo de la pieza que se desea manufacturar, posteriormente acerca la herramienta al bloque y después al oprimir el botón de inicio, la rutina de cálculo de voxels, de rebanado, de obtención de perfil trabajan en automático y la herramienta empieza a moverse y a maquinar los perfiles de las rebanadas obtenidas y así lo hace hasta terminar la pieza (Figura 2-87). A continuación se muestra el proceso que el operador realiza:

1. Dibuja y/o importa la pieza en formato stl.
2. Indica las dimensiones del tejo y de la herramienta
3. Ubica la herramienta en el punto inicial de la pieza
4. Da un clic al botón inicio
5. Se muestra el proceso del maquinado

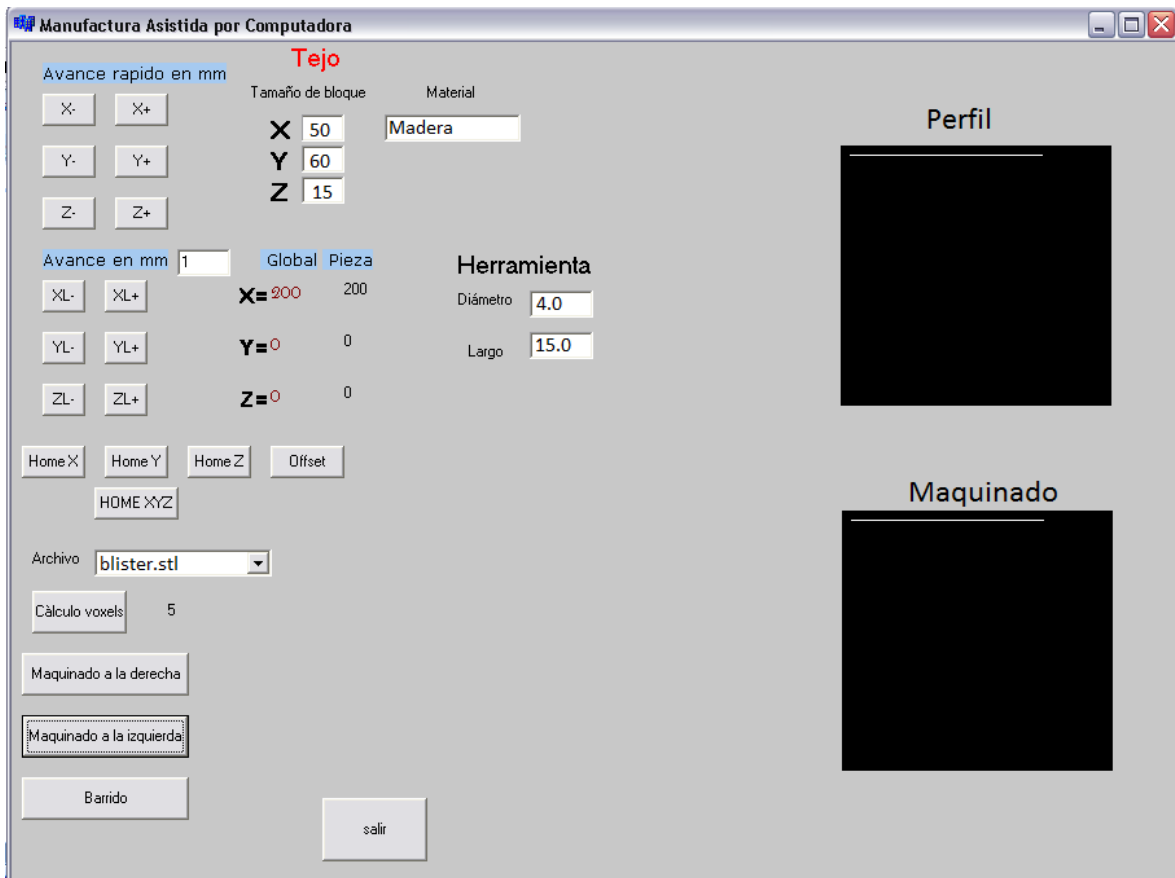


Figura 2-87. Pantalla de usuario

Capítulo 3:

Interfaz y

mecanismo

Introducción

En este capítulo se abordan los temas del mecanismo y de la interfaz desarrollada una vez habiendo explicado aspectos teóricos en el Capítulo 1:. El mecanismo construido sirvió para comprobar que los algoritmos desarrollados e implementados habían sido adecuados, ya que se pudieron realizar correcciones al software que no eran apreciables únicamente con las suposiciones, y que al ser implementados el panorama de mejoras incrementó y algunas se llevaron a cabo.

El mecanismo desarrollado es un robot de tipo cartesiano de tres grados de libertad, controlado por medio de la PC a través de la interfaz del puerto paralelo, por donde se mandan datos para el manejo de los controladores de los motores que realizan el movimiento de los husillos y poder desplazar la herramienta de corte.

3.1 Mecanismo

El prototipo que se ha construido es un mecanismo que cuenta con 3 ejes perpendiculares entre sí (X, Y, Z). En los ejes X y Y están dispuestas dos barras guías paralelas por donde se deslizan dos pares de rodamientos lineales, un par por cada guía, y en medio de las barras un husillo con tuerca embalada, el cual está conectado a un motor a pasos en un extremo Figura 3-1. La diferencia para el caso del eje Z es que sólo cuenta con un rodamiento lineal por cada barra.

El mecanismo del eje X soporta una estructura donde se encuentra el mecanismo del eje Y, y a su vez éste soporta el mecanismo del eje Z, donde se sujeta el motor que hace funcionar la herramienta de corte.

Se empieza a explicar cada uno de los mecanismos de cada eje empezando por aquel que sujeta a la herramienta de corte, o sea el eje Z.

Los mecanismos son semejantes en cuanto a los componentes de los que están integrados, por lo que será suficiente explicar de manera amplia uno de ellos.

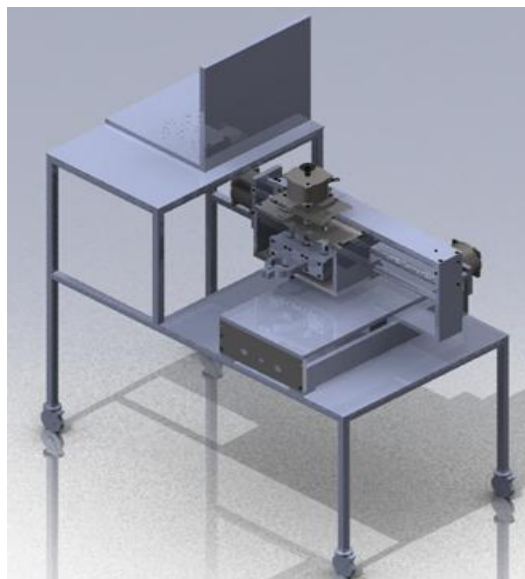


Figura 3-1. Diseño del prototipo en CAD

3.1.1 Mecanismo del eje Z

El eje Z es el mecanismo que se desplaza verticalmente y es donde se aloja la herramienta de corte por medio de un porta-herramienta que sujeta el mototool y quien finalmente sujeta el cortador.

El mecanismo se integra por dos barras guías de ½" de diámetro por 30 cm de largo, paralelas entre sí y por donde se deslizan dos rodamientos lineales, también cuenta con un husillo embalado con un paso de 8 mm. y 25 cm. de largo. Tanto la tuerca como los rodamientos lineales se fijan una placa rectangular de aluminio, la cual sujeta otra placa de forma redonda donde se sujeta al mototool, como muestran la Figura 3-2 y Figura 3-3.

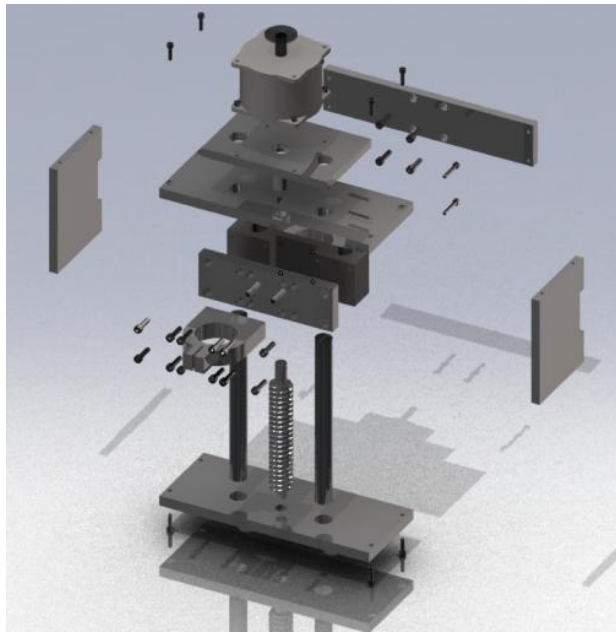


Figura 3-2. Vista explosionada del mecanismo del eje "z"

El husillo se conecta con el motor a pasos por medio de un cople, sujetos por dos pernos cónicos que atraviesan tanto el cople como el eje del motor y del husillo.

El desplazamiento lineal mínimo que se tiene al dar un medio paso del motor está dado por la siguiente relación:

Los motores ocupados son de 200 pasos, es decir, para realizar una revolución se manda ésta cantidad de pulsos, y por cada paso se avanza 1.8° sin embargo, al ocupar el medio paso se tiene un total de 400 pasos, por lo que el desplazamiento angular es de 0.9°.

Por lo tanto, la máxima resolución del movimiento lineal es de:

$$\text{Resolución} = \frac{\text{paso de cuerda}}{\text{pasos del motor}} = \frac{8 \text{ mm}}{400 \text{ pasos}} = 0.02 \frac{\text{mm}}{\text{paso}} = .00078 \frac{\text{pulgada}}{\text{paso}}$$

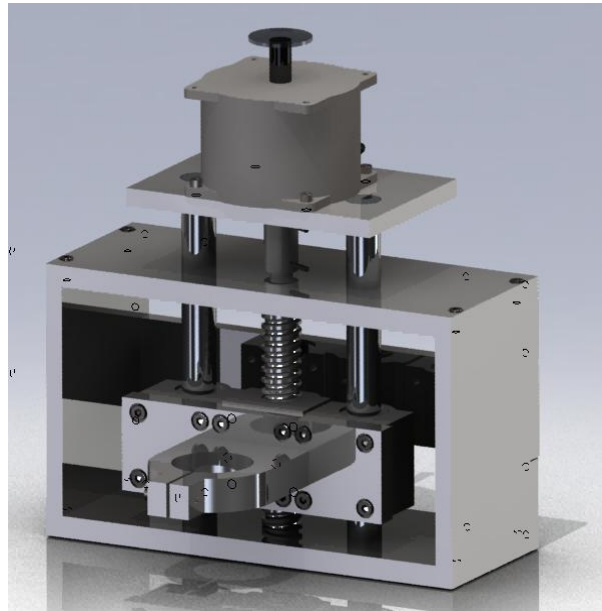


Figura 3-3. Imagen del diseño del mecanismo del eje Z

3.1.2 Mecanismo del eje Y

Es el mecanismo transversal que desplaza horizontalmente de izquierda a derecha y viceversa a la herramienta de corte, y al igual que en el Mecanismo del eje Z cuenta con dos barras guías paralelas por donde se deslizan los rodamientos lineales, también cuenta con un husillo embalado entre las barras que se conecta al motor a pasos en un extremo por medio de un cople y pernos cónicos.

El husillo y las barras están contenidas en un bastidor rectangular, formado de placas de aluminio de 3 pulgadas de ancho con un espesor de $\frac{1}{2}$ pulgada. La tuerca del husillo se sujeta de una placa de aluminio rectangular, quien a su vez sujeta el bastidor del mecanismo del Mecanismo del eje Z, mientras los rodamientos sujetan directamente a dicho bastidor. Dichas sujeciones son por medio de unos tornillos de cabeza allen (Figura 3-4 y Figura 3-5).

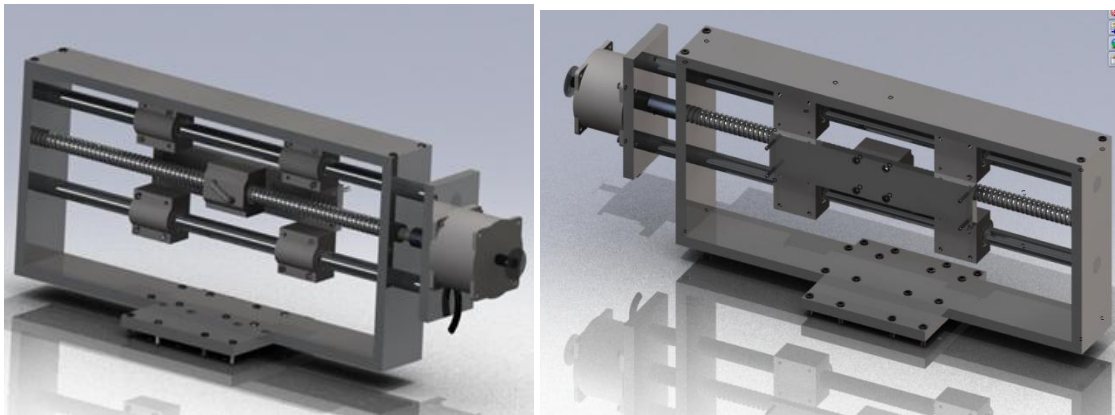


Figura 3-4. Diseño del mecanismo del eje Y

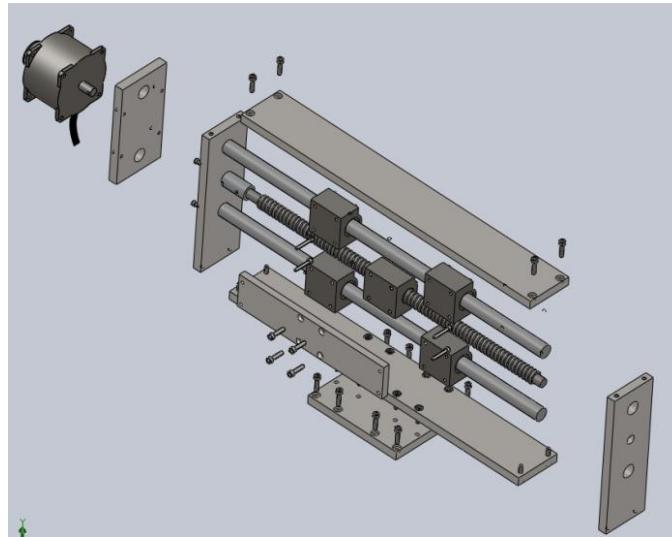


Figura 3-5. Vista explosionada del mecanismo del eje Y

3.1.3 Mecanismo del eje X

Es el mecanismo longitudinal que se mueve horizontalmente de atrás hacia adelante y viceversa, además es donde se monta la mesa para la pieza a maquinar, como se muestra en la Figura 3-6 y al igual que los mecanismos anteriormente mencionados, cuenta con dos barras guías y un husillo embalado, que se conecta al motor a pasos por medio de un cople.

La tuerca del husillo y los rodamientos lineales se sujetan a una placa rectangular, y ésta a su vez a la placa inferior del bastidor Mecanismo del eje Y, como se muestra en la Figura 3-7.

El husillo y las barras guías están dispuestas en un bastidor rectangular, formada por placas de aluminio de $\frac{1}{2}$ " de espesor (Figura 3-8).

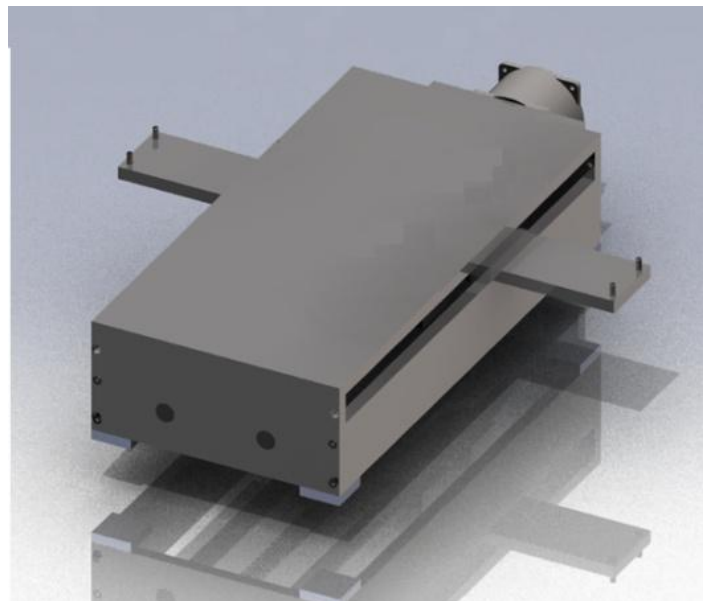


Figura 3-6. Mesa de trabajo donde se sujeta el bloque a maquinar

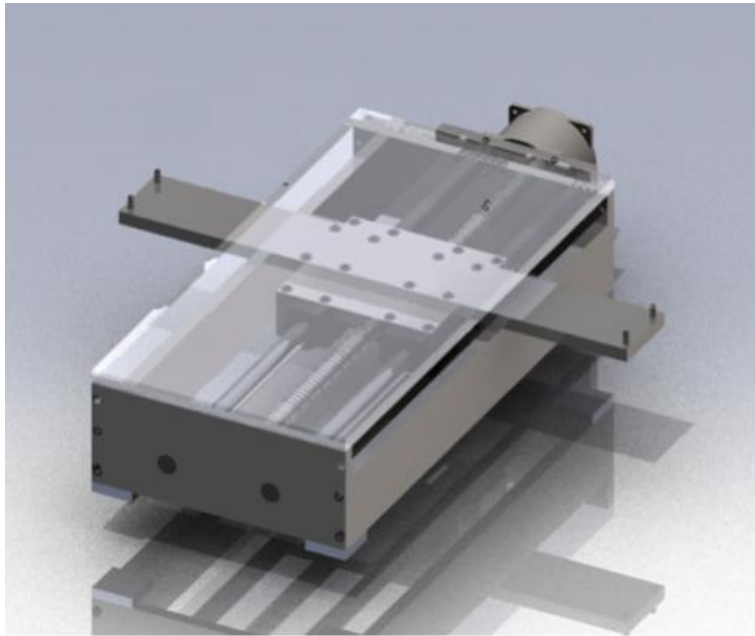


Figura 3-7. Vista del mecanismo del eje X

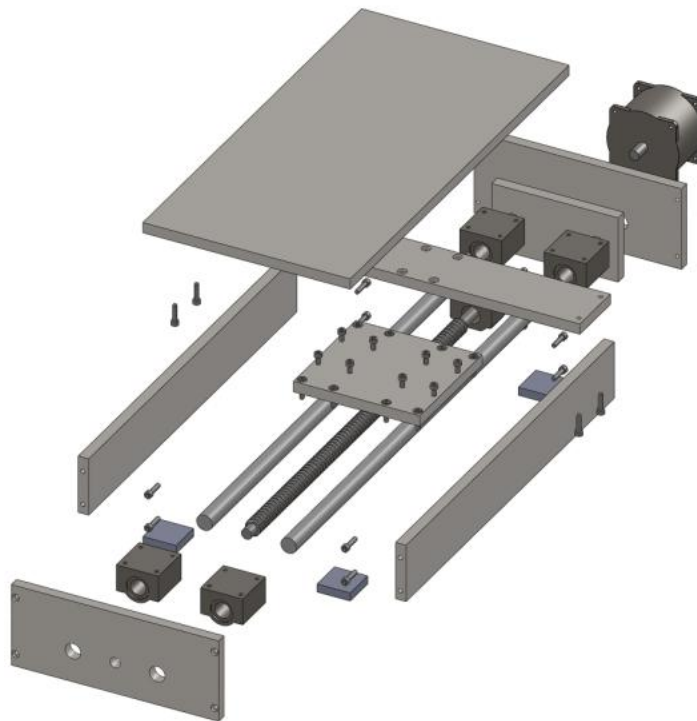


Figura 3-8. Vista explosionada del mecanismo del eje X

La sujeción de los motores a pasos con los mecanismos anteriormente descritos, se realiza por medio de una placa rectangular que se sujeta a las barras guías respectivas de cada eje, como se muestra en la Figura 3-9, y en la Figura 3-10 se muestra el despiece del mismo ensamble.

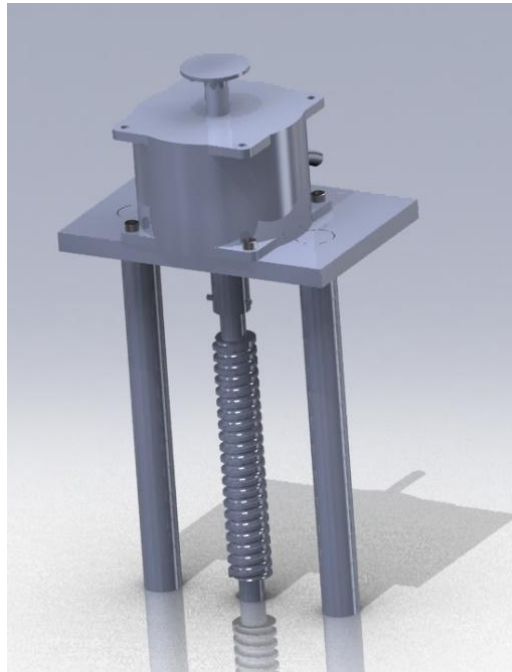


Figura 3-9. Sujeción de los motores al mecanismo



Figura 3-10. Vista explosionada

Se ha realizado una estructura de tubo cuadrado de 1" donde se monta el prototipo, dotada de pequeñas ruedas para poder desplazarse como se muestra en la Figura 3-11

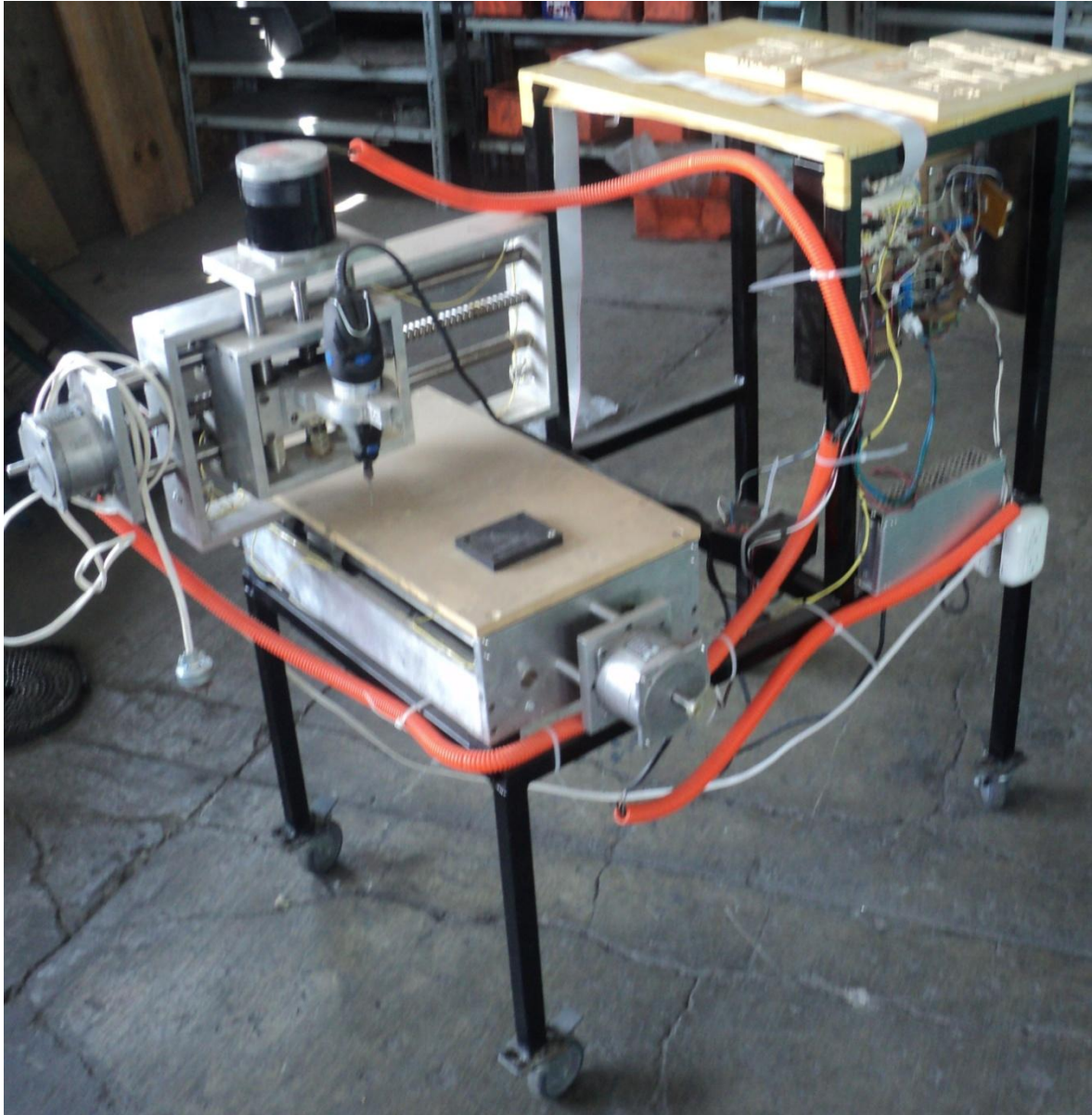


Figura 3-11. Prototipo terminado

En la siguiente imagen (Figura 3-12) se observa cómo está constituido el prototipo. Del lado superior derecho se encuentra la PC, donde se carga el programa que realiza las operaciones de construir la superficie del sólido dibujado en CAD y se aprecia la interfaz gráfica de diseñador.

En la parte baja de la PC se visualiza la interfaz PC-máquina que comprende el puerto paralelo y los drivers de los motores a pasos, cuyos cables de control viajan por las mangueras de color naranja a sus respectivos motores.

El mecanismo de tres ejes se encuentra del lado izquierdo de la PC, y se aprecia la forma en que está trabajando en la fabricación de una pieza, cuyo perfil obtenido se observa en el monitor.

La fuente conmutada de 12 Volts a 8 amperes para la alimentación de los motores se ubica debajo de la interfaz de comunicación, mientras que la fuente de 5 volts se encuentra de lado izquierdo, sobre el travesaño.

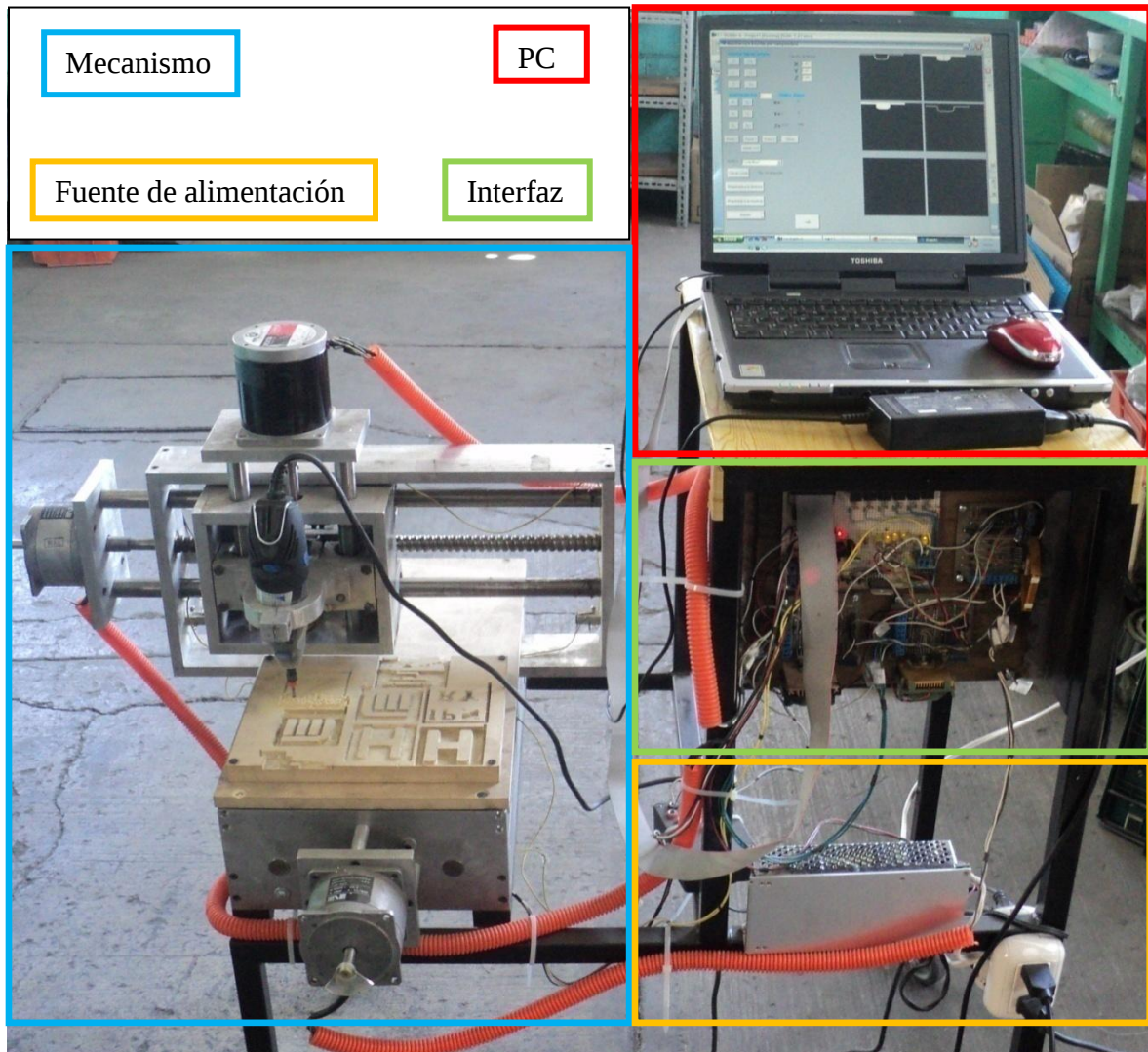


Figura 3-12. Fotografía del prototipo

3.1.4 Sensores

Los sensores son indispensables para mecanismos como el desarrollado para tener una retroalimentación respecto a la ubicación de la herramienta de corte, así como la prevención de colisiones al surgir algún tipo de falla en el envío de datos por el puerto paralelo. A continuación se explican los sensores utilizados en este trabajo.

3.1.4.1 Sensor de final de carrera

Para este caso, se han utilizado microswitch de rodillo los cuales se montan en una base de aluminio y son colocados en los extremos de cada husillo para ser accionados por el

mecanismo, al ejercer la presión necesaria a la palanca a una distancia de 10 milímetros antes de la colisión como muestra la Figura 3-13.

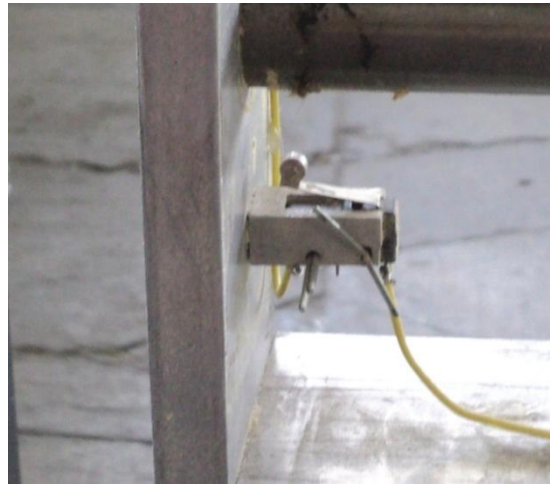


Figura 3-13. Fotografía de la ubicación de un sensor de final de carrera

La conexión de los 6 sensores de final de carrera es en serie, y conectados a la señal del “enable” del L297, para que cuando el rodillo de cualquier sensor sea accionado por el mecanismo, se abra el circuito y deshabilite el funcionamiento de la tarjetas de control de motores e inhiba el funcionamiento del motor, como muestra la Figura 3-14.

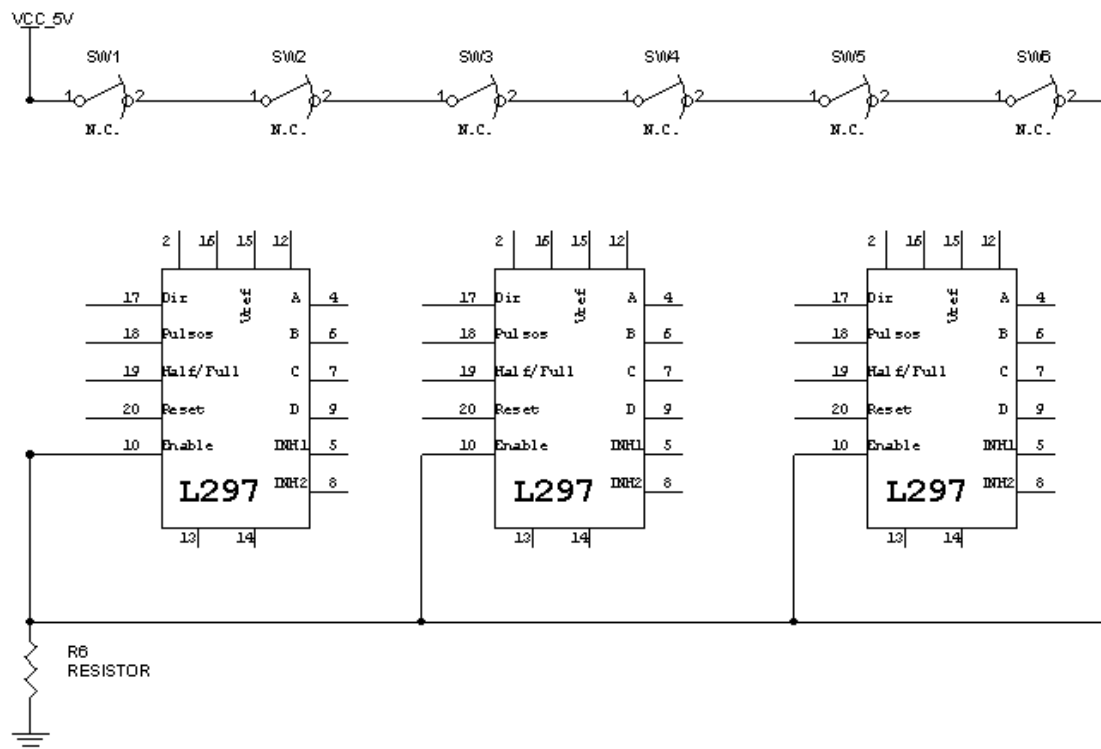


Figura 3-14. Circuito eléctrico de los sensores de finales de carrera

3.1.4.2 Sensores de inicio

Se han utilizado foto interruptores para detectar la llegada del mecanismo en un punto de referencia identificado como HOME.

En los elementos mecánicos que viajan sobre los ejes, se han colocado elementos físicos para activar los sensores que se encuentran a una distancia aproximada de 20 milímetros de los límites e indicar que el mecanismo ha llegado al inicio o al HOME de la máquina como se muestra en la Figura 3-15.

Las señales se conectan a la interfaz y a su vez a las entradas del puerto paralelo, para detectar el momento en que el dispositivo ha sido interrumpido y saber que ha llegado al punto de inicio de cada eje.

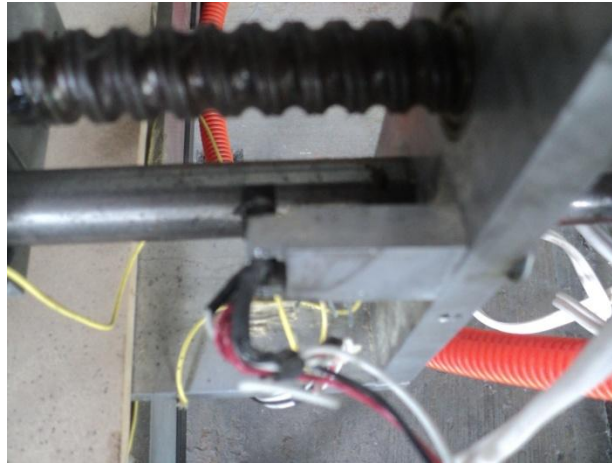


Figura 3-15. Fotografía de la ubicación de un sensor de inicio del eje Y

3.1.5 Fuente de alimentación

Para alimentar el sistema eléctrico de potencia, se ha utilizado una fuente conmutada de 12 volts de C.D. a 8.5 Amperes, debido a que cuando los motores están parados, es cuando mayor corriente consumen, 1 Amperes por cada uno, lo que hace un total de 3 Amperes (Figura 3-16). Para la parte de control se utiliza una fuente de 5 volts.

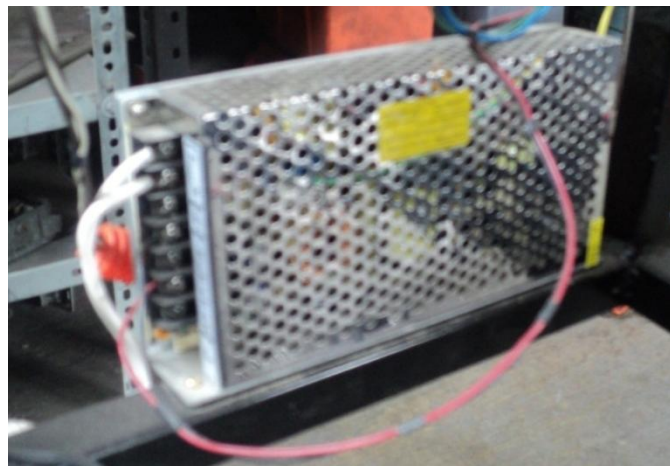


Figura 3-16. Fuente conmutada de 12 volts a 8.5 Amperes

3.1.6 Herramienta de corte

Se ha adaptado un motor rectificador pequeño al mecanismo y provee hasta 35,000 rpm como se muestra en la Figura 3-17. La herramienta de corte que se utiliza es una lima rotatoria, para el maquinado de piezas en MDF o madera, sin embargo, para poder maquinar metales, como el aluminio, que es uno de los metales más blandos o plásticos como nylamid o acrílico, es necesario usar un cortador vertical, y eso es una limitante ya que el zanco de las herramientas que ocupa el rectificador es de 1/8" (3.2 mm), y los cortadores de acero alta velocidad (HSS), tienen un zanco de 3/8".



Figura 3-17. Fotografía del mototool y la herramienta de corte montado en la máquina

3.2 Interfaz

La interfaz es un medio de comunicación entre dos elementos, para este caso es la comunicación entre la PC y los controladores de los motores. Dicha comunicación empieza desde el software de la PC donde se manda la información que se convierten en señales eléctricas que viajan a través del puerto paralelo y llegan a los controladores para el movimiento de los motores. Es por eso que se habla primeramente de lo que ocurre en la PC y después de la parte física.



Diagrama de bloques 3-1. Conexión entre los dispositivos para el control de motores

3.2.1 Software

En el programa se realizan las operaciones computacionales para obtener la información requerida para mandar las señales por el puerto paralelo. Las señales a mandar son las correspondientes al movimiento de los motores, por ello se muestra a continuación en la Tabla 3-1 la relación del bit con respecto a la acción que realiza.

Tabla 3-1. Acción de los niveles lógicos del puerto 378

Pin	Bit	Descripción	Acción	
			0	1
2	0	Paso Z		
3	1	Dir Z	Abajo	Arriba
4	2	Paso X		
5	3	Dir X	Atrás	Adelante
6	4	Paso Y		
7	5	Dir Y	Izquierda	Derecha

3.2.1.1 Movimientos para los ejes x, y, z

Como se mencionó anteriormente, no se puede manipular el dato a nivel de bit, es por eso que el dato de 1 byte se enmascara de acuerdo al movimiento de búsqueda en el algoritmo del seguimiento del perfil.

A continuación se muestra en la Figura 3-18 el proceso de la interacción del CAM con el mecanismo, como se observa el sistema se divide en dos bloques, el software y el hardware.

En el software se desarrolla el cálculo de los voxels que forman la superficie y obtiene las rebanadas del sólido, después y durante el seguimiento de cada perfil se manda simultáneamente los datos correspondientes a los controladores para mover los motores en el eje “y” y “z”.

Cuando ocurre el cambio de rebanada, se manda el dato para mover el eje “x”. En el hardware se encuentra el puerto paralelo y los drivers de los motores, en el esquema se muestran flechas en color negro representando el puerto paralelo y en los bloques en rojo a los drivers.

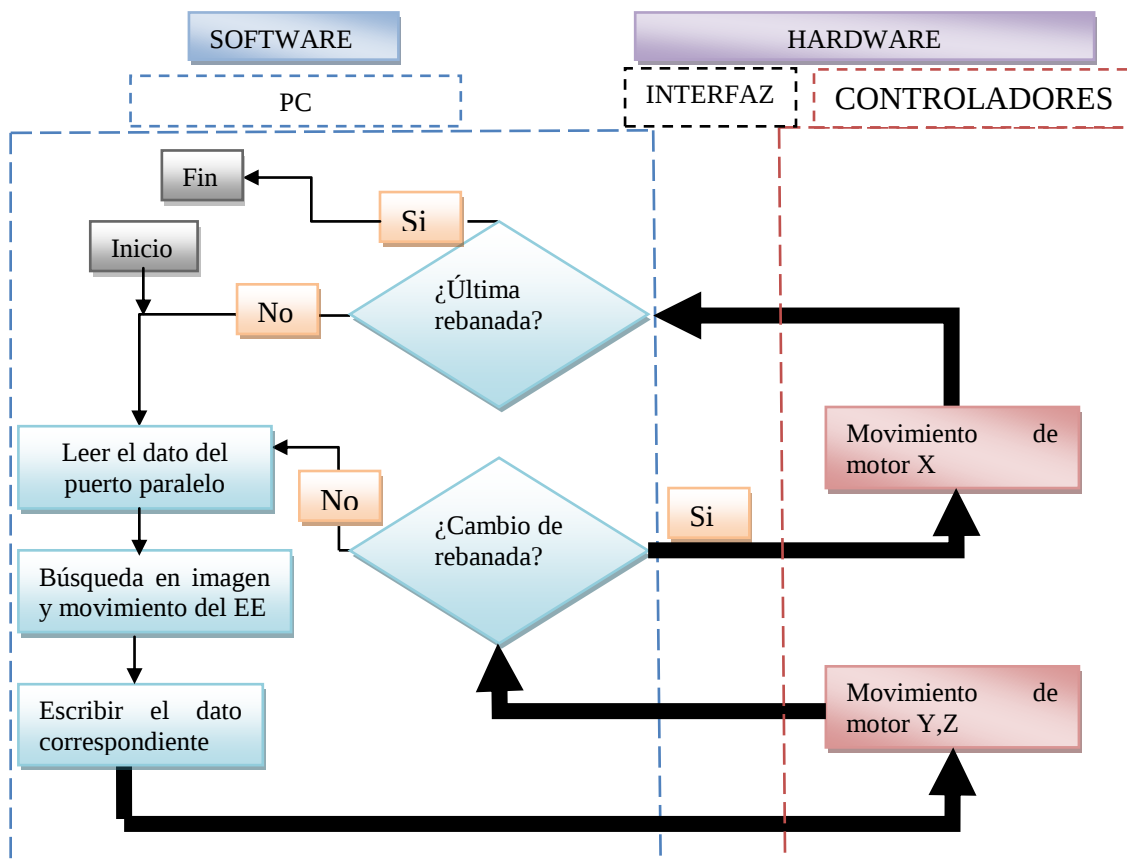


Figura 3-18. Diagrama de proceso de la interfaz

A continuación se muestra en la Tabla 3-2 las máscaras utilizadas correspondientes a los bits de salida de los motores.

Tabla 3-2. Máscaras de los bits de los motores

Pin	Bit	Descripción	Dato		Acción	
			0	1	0	1
2	0	Paso Z				
3	1	Dir Z	Dir & 0XFD	Dir 0x02	Sube	Baja
4	2	Paso X				
5	3	Dir X	Dir & F7	Dir 0x08	Avanza	Retrocede
6	4	Paso Y				
7	5	Dir Y	Dir & 0XDF	Dir 0X20	Derecha	Izquierda

Cuando se manda el dato para realizar el paso, se tiene que realizar una rutina de conteo, para darle tiempo al motor de moverse y no recibir los pulsos a una frecuencia alta, ya que puede comportarse de una forma distinta a la deseada.

Dado que se están transmitiendo datos por el puerto paralelo, la computadora está expuesta a recibir descargas de los circuitos que controlan los motores que dañarían gravemente a la misma y para evitar esto es importante tener un aislamiento entre la PC y los circuitos electrónicos de control y de potencia para ello se usan dispositivos electrónicos llamados acopladores ópticos.

3.2.2 Hardware

El puerto paralelo es la vía física en la que se mandan las señales y se han ocupado los 8 pines de salida (2 al 9) un par de ellos son para indicar la dirección y el pulso de un motor, si se tiene tres motores, se tienen tres pares de bits y los otros dos bits de salida, utilizados para el encendido y apagado de la fuente conmutada y del mototool.

La conexión física del puerto paralelo a los dispositivos de aislamiento hacia el dispositivo L297, para el control del motor a paso es como se muestra a continuación en la Figura 3-19:

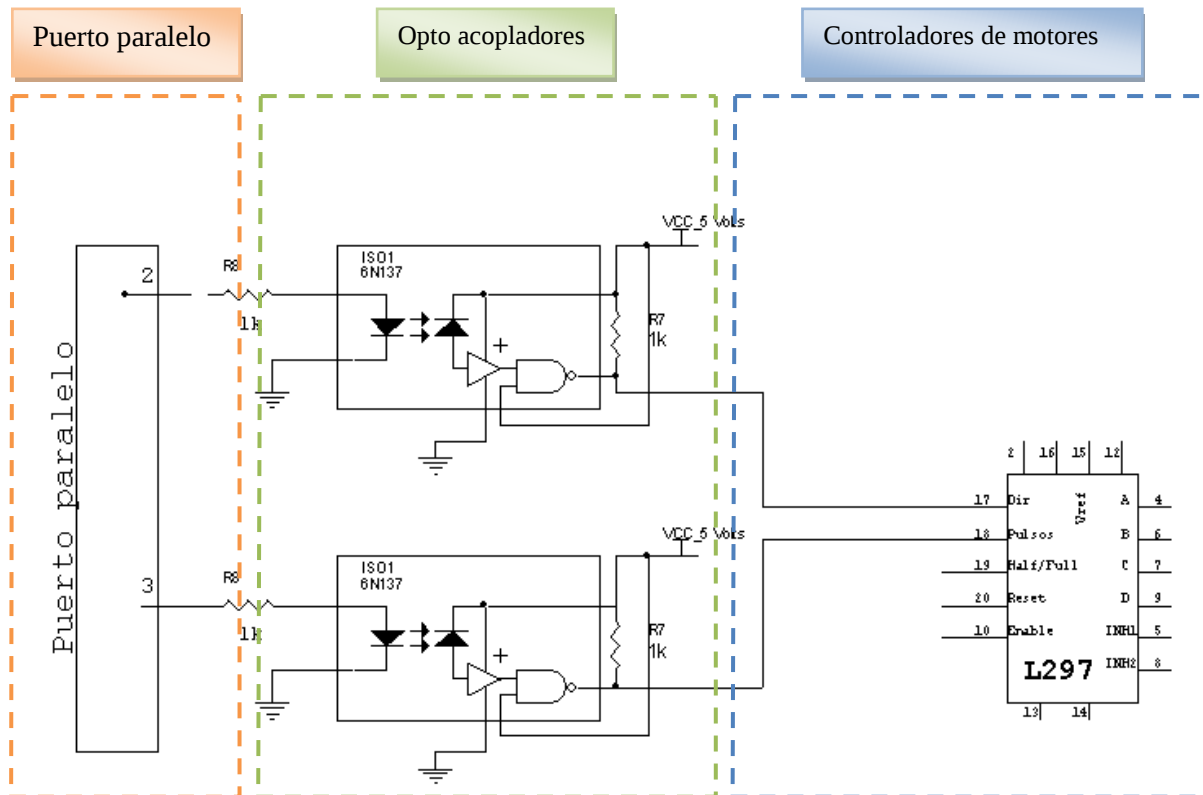


Figura 3-19. Conexión electrónica del puerto a los aisladores y a los drivers de los pines de dirección y paso del L297

También se han ocupado 3 pines de entrada del puerto paralelo correspondientes a los pines: 10, 12 y 13 para las señales de los sensores de home que mandan una señal en el momento en que se activen los sensores como muestra la Figura 3-20.

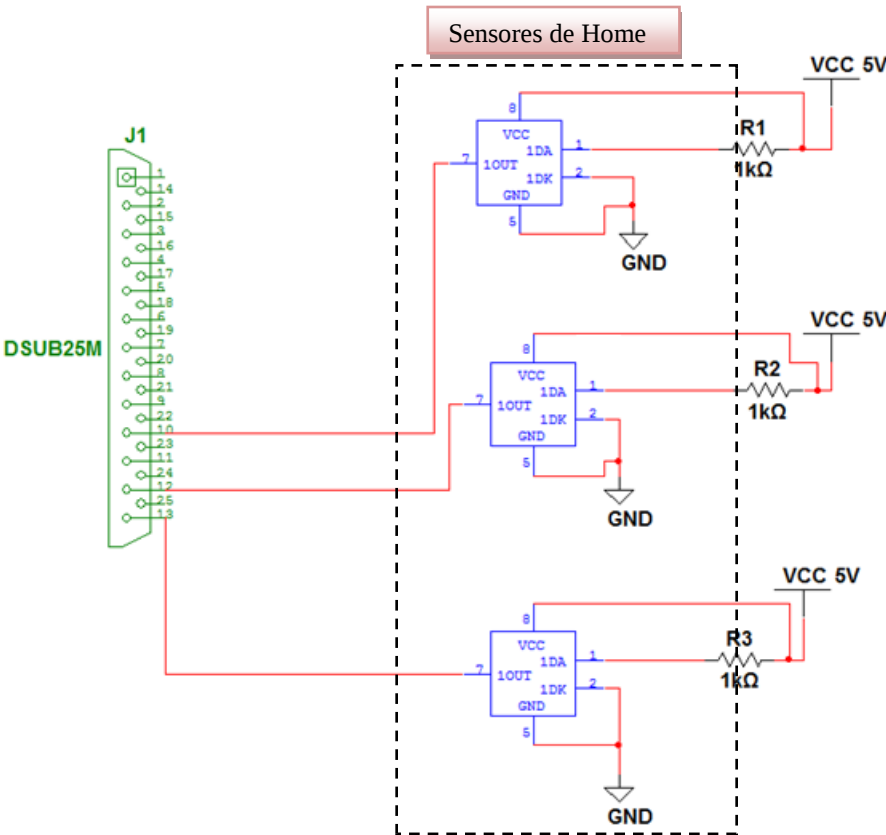


Figura 3-20. Conexión de las entradas al puerto de los sensores de inicio

La conexión entre los pines de entradas y salidas del puerto paralelo y las funciones de las señales se muestran en la Tabla 3-3 y la Tabla 3-4.

Tabla 3-3. Asignación de los pines y datos del puerto paralelo

PIN	9	8	7	6	5	4	3	2
BIT	7	6	5	4	3	2	1	0
Función	On/Off Fuente	On/Off Herramienta	Paso Y	Dir Y	Paso X	Dir X	Paso Z	Dir Z

Tabla 3-4. Asignación de pines de entrada de los sensores de home

PIN	13	12	10
BIT	4	5	6
Función	Home z	Home y	Home x

De esta manera se usa el puerto paralelo para mover los tres motores a pasos del mecanismo para manufacturar la pieza dibujada por el CAD interpretada por los algoritmos desarrollados.

3.2.2.1 Controlador de motor a pasos

El controlador del motor a pasos está compuesto del chip L297 y el L298. El primero es el que tiene comunicación directa con el puerto paralelo por medio de los opto acopladores 6N137, y es quien controla al puente H L298 quien realiza el movimiento del motor a pasos. El diseño electrónico para el control de los motores a pasos se muestra a continuación en la Figura 3-21:

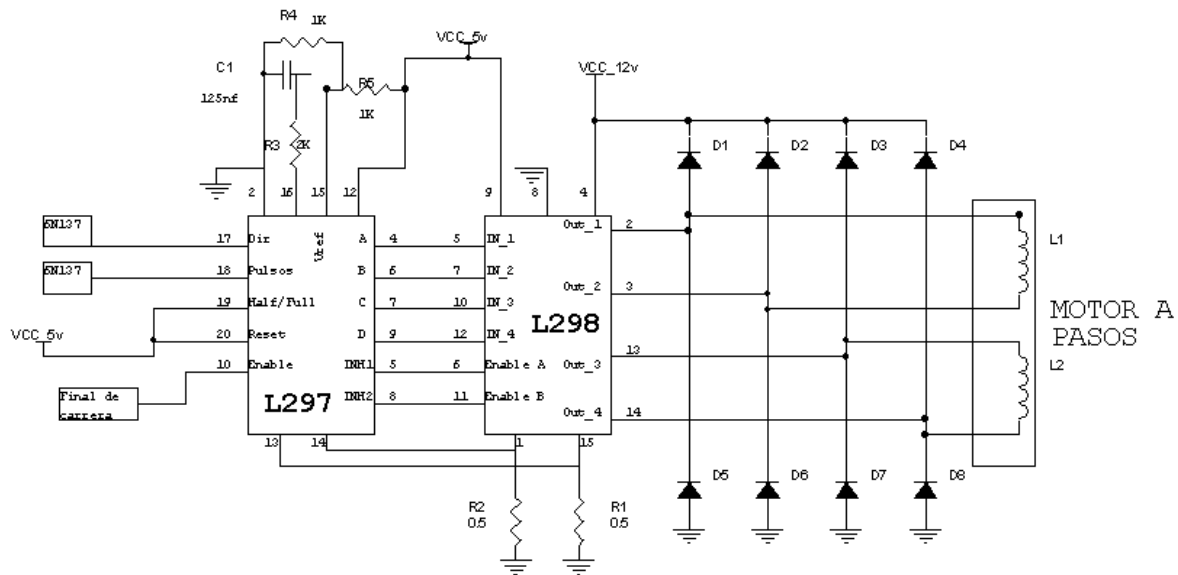


Figura 3-21. Circuito electrónico del driver del motor

Se muestra a continuación en la Figura 3-22 la fotografía del driver del motor a pasos

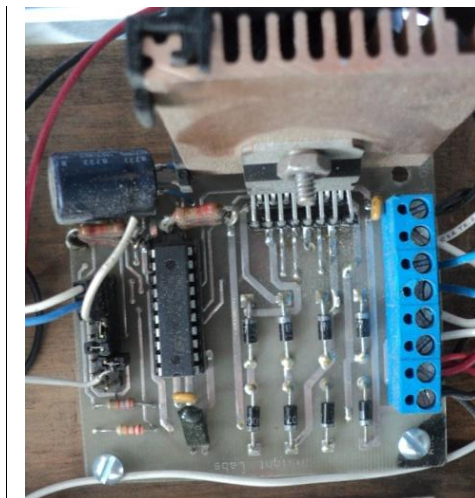


Figura 3-22. Fotografía del driver de un motor

3.2.2.2 Control de encendido y apagado de la fuente de alimentación

Este dispositivo es activado por un pin del puerto paralelo, que a su vez está conectado a un TRIAC externo para el control de encendido y apagado de las cargas.

Este dispositivo se ha ocupado para el accionamiento de la herramienta de corte, y también para la fuente conmutada de 12 volts a 8 amperes. A continuación se muestra el circuito realizado en la Figura 3-23

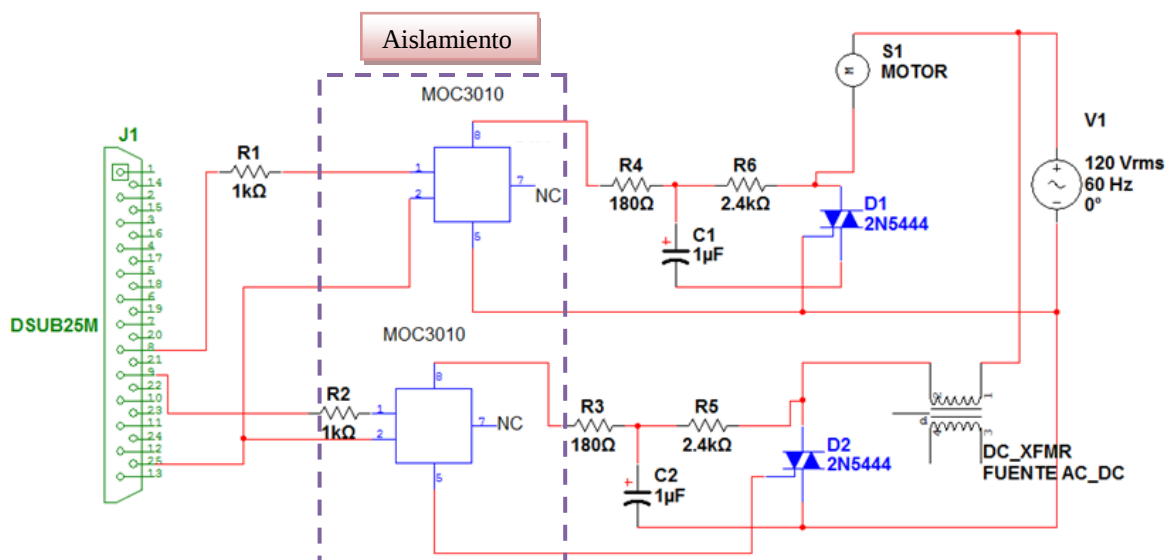


Figura 3-23. Conexión electrónica del encendido y apagado del mototool y la fuente de alimentación

Se muestra a continuación en la Figura 3-24 el esquemático de la conexión entre el puerto paralelo y los drivers que controlan los motores, así también con los sensores de inicio y con el control de encendido y apagado de la fuente y del mototool.

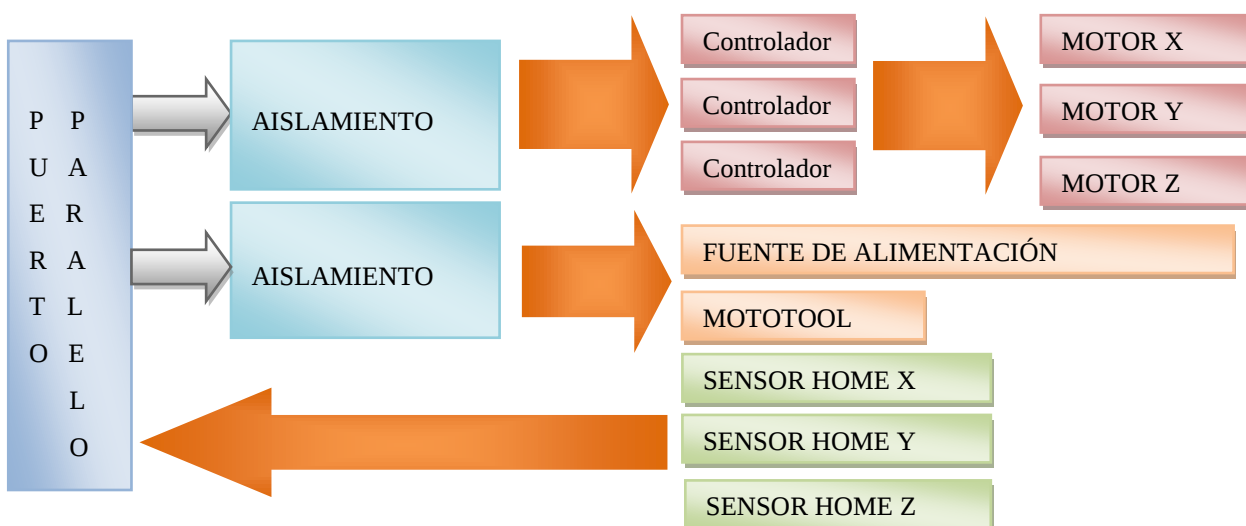


Figura 3-24. Diagrama de comunicación entre el puerto paralelo con los controladores, sensores de inicio y el control de encendido

A continuación en la Figura 3-25 se muestra la fotografía de la parte del aislamiento de la interfaz realizada

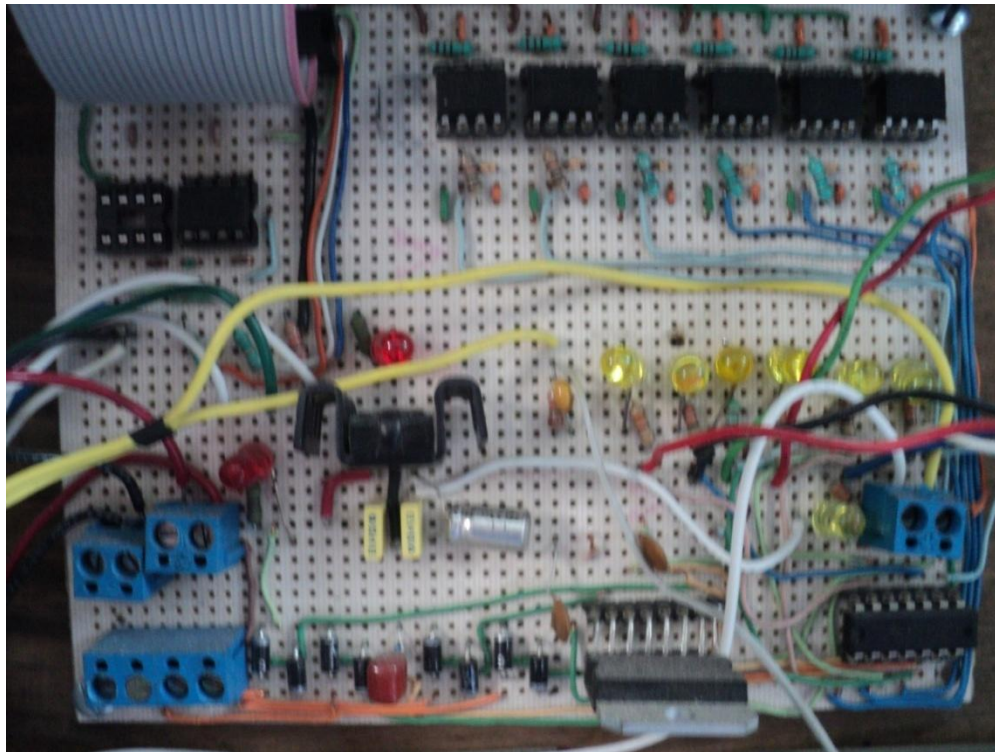


Figura 3-25. Circuito de interfaz entre PC y máquina

Finalmente los circuitos de control de motores a pasos, el de control de encendido y apagado de fuente y del motor de la herramienta, de las entradas de las señales de los sensores de inicio y de los sensores de final de carrera, se condensan en un circuito general.

La tarjeta de control y la de interfaz es como la que se muestra a continuación en la Figura 3-26.

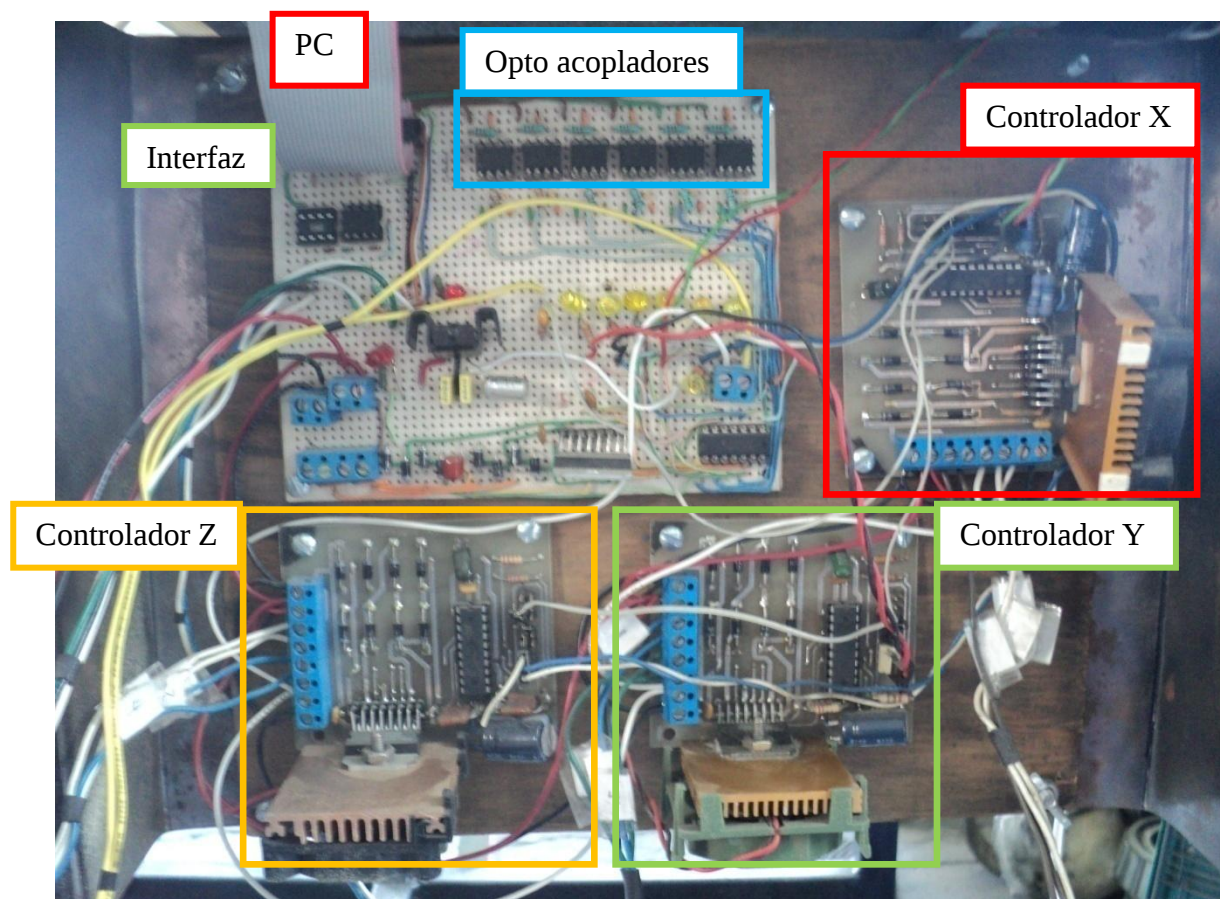


Figura 3-26. Fotografía de la tarjeta de interfaz y de controladores de los motores.

Capítulo 4:

Aplicación

Introducción

En este capítulo se describen ejemplos realizados en la máquina, que muestran la forma en que se lleva a cabo el proceso de manufactura sin la intervención del usuario u operario, simplemente con el uso del software desarrollado. Lo único que tiene que hacer el operario es mandar a inicio la herramienta y posteriormente ubicarla en el punto de inicio de maquinado en la pieza, e importar el archivo en STL para dar inicio en el botón correspondiente.

4.1 Molde

Como se explicó en el Capítulo 1:, en el apartado de Moldes se mencionó que la mayoría de los moldes están constituido en dos partes: la cavidad y el corazón. Un ejemplo de este tipo de molde son los ocupados para fabricar empaques de productos como lo son algunos artículos de higiene personal, juguetes, papelería, etc, como se muestra en la Figura 4-1.



Figura 4-1. Ejemplos de empaques con forma del producto

Uno de estos ejemplos es el empaque del hilo dental (Figura 4-2), y a continuación se muestra paso a paso cómo se desarrolla el proceso de manufactura, desde el dibujo hasta el producto final.



Figura 4-2. Empaques de hilo dental

El primer paso a seguir es dibujar las dos partes del molde que se quiere fabricar, como se muestra en la Figura 4-3, donde la pieza superior corresponde al corazón de la pieza y la pieza inferior corresponde a la cavidad. Aunque en algunos programas de CAD, el realizar los moldes es cuestión de unas operaciones a partir del producto final, es decir, que se dibujaría el blíster y se obtendría el molde.

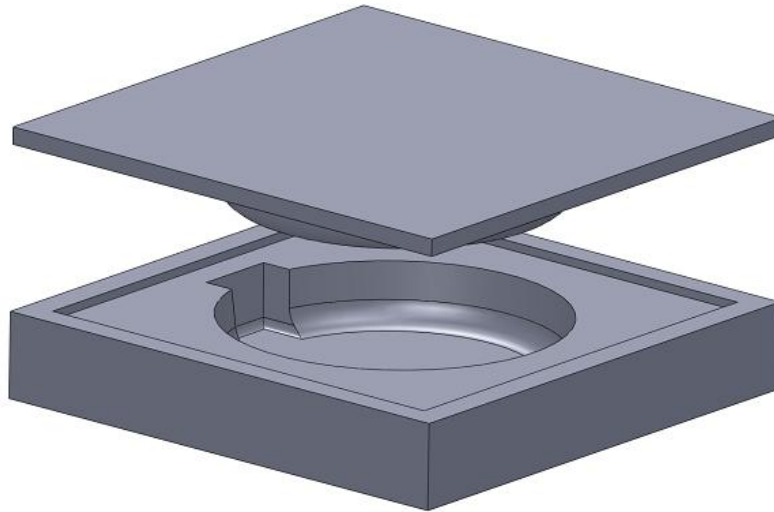


Figura 4-3. Molde para blíster de producto de hilo dental

Se muestra a continuación en la Figura 4-4 el ensamble del molde, con la cavidad en forma transparente para poder visualizar el corazón.

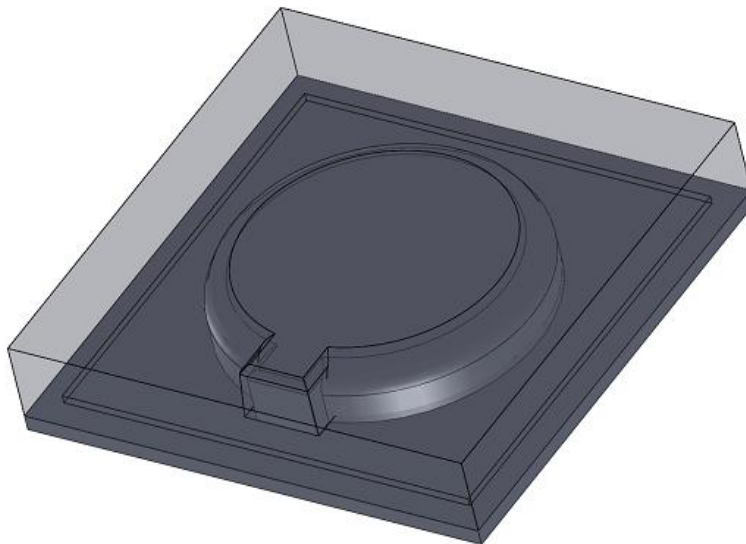


Figura 4-4. Ensamble con la cavidad en forma transparente

4.1.1 Cavidad

Primeramente se muestra la forma en que se manufactura la cavidad del molde. La cavidad del molde que se dibujó se muestra a continuación en la Figura 4-5

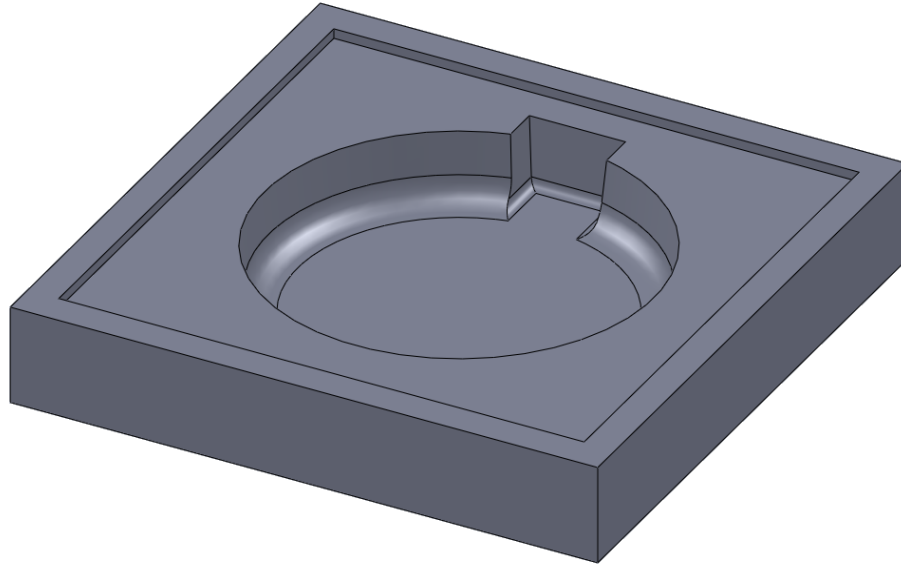


Figura 4-5. Cavidad del molde de empaque

4.1.1.1 Obtención de la pieza en formato STL

Una vez que se ha dibujado la pieza en un programa CAD se guarda en formato STL (Figura 4-6). Existe la posibilidad de seleccionar el tipo de código a guardarse en ASCII o binario (Figura 4-7) y como se mencionó anteriormente el tipo de código que se requiere, se selecciona el primero y de acuerdo a la calidad de la resolución se construye la superficie de la pieza con una serie de triángulos, como se muestra en la Figura 4-10, posteriormente aparece una ventana que anuncia el número de triángulos, el tamaño del archivo y el formato del archivo, es importante verificar que se haya guardado en formato ASCII.

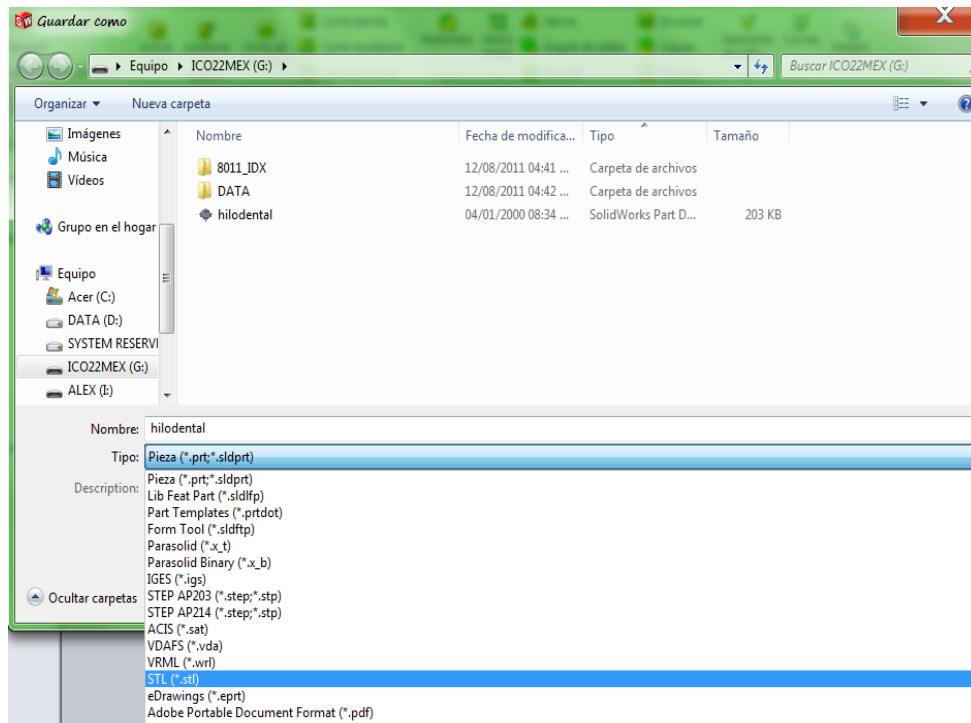


Figura 4-6. Selección del tipo de formato

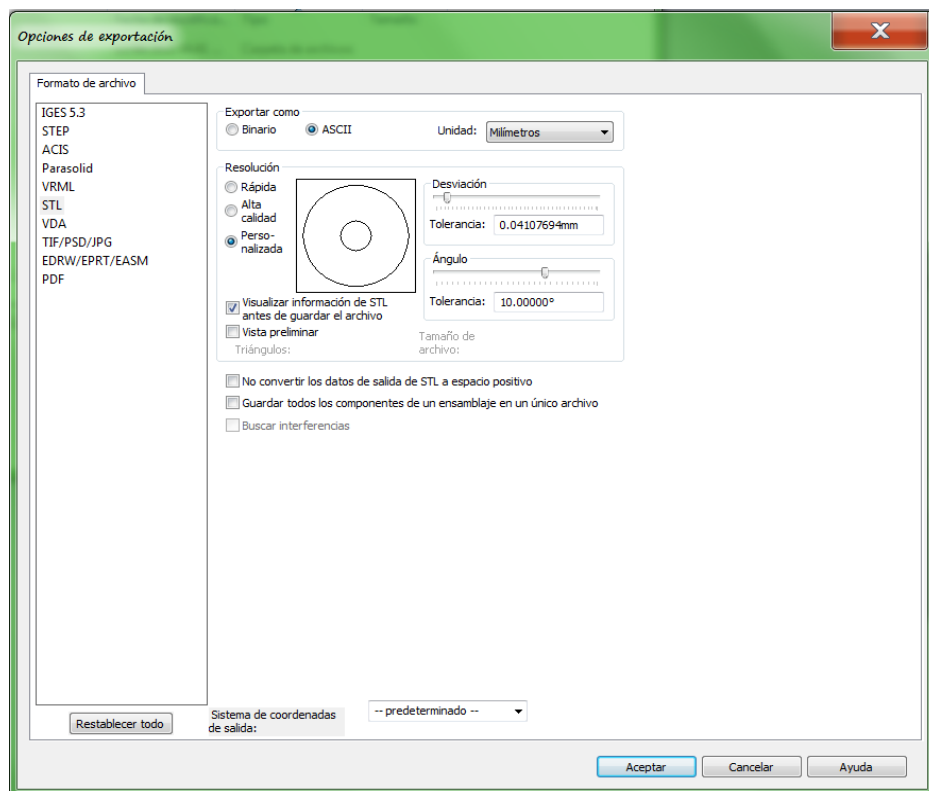


Figura 4-7. Ventana donde se selecciona el tipo de código, la resolución, unidades, etc.

Se muestran a continuación las ventanas que aparecen cuando se está guardando la pieza, en una se muestra cuando se ha guardado en formato ASCII (Figura 4-8) y la segunda cuando se ha guardado en binario (Figura 4-9), nótese la diferencia en el tamaño del archivo entre una y otra.

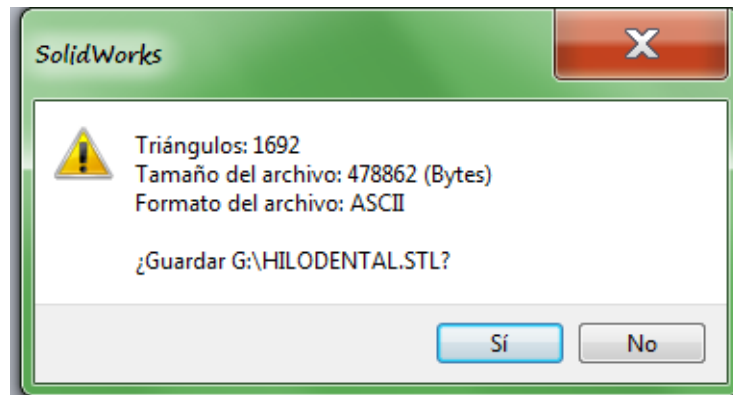


Figura 4-8. Ventana mostrada al guardar la pieza en código ASCII

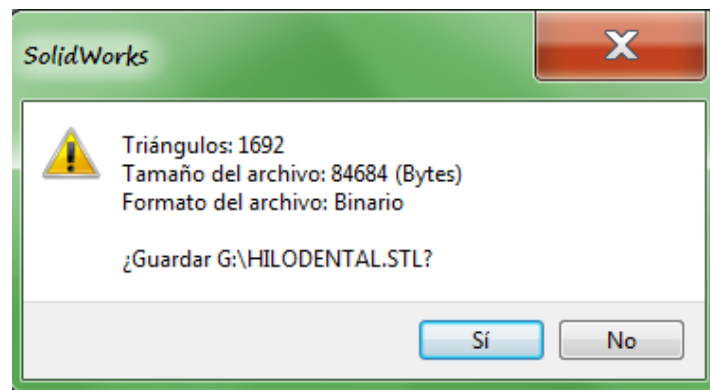


Figura 4-9. Ventana mostrada al guardar la pieza en código binario

Para el dibujo de la cavidad del molde, el total de triángulos a procesar es de 1692, esto quiere decir que de cada uno se calculan los voxels que forman la superficie.

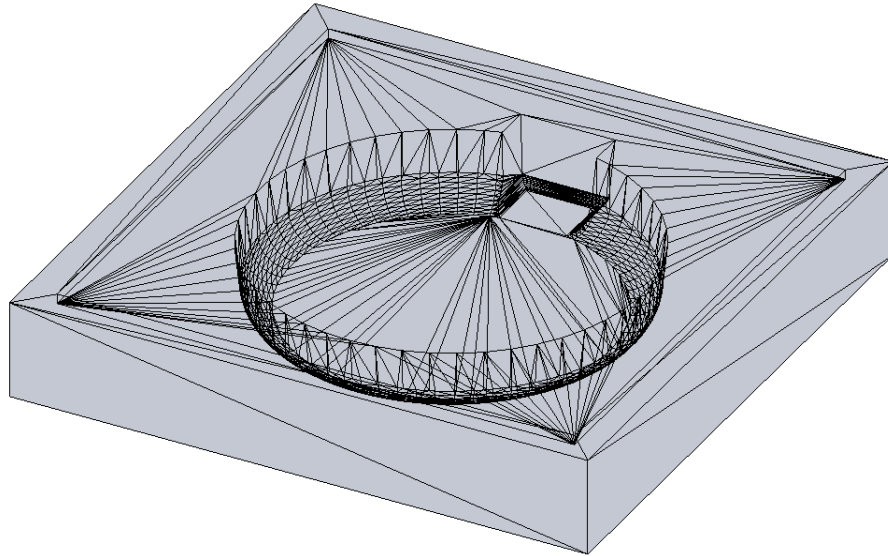


Figura 4-10. Formato stl de la pieza dibujada

4.1.1.2 Inicio del programa

Una vez que se tiene la pieza dibujada y guardada en formato STL, se procede a correr el software desarrollado y se obtiene la ventana de trabajo y lo que sigue es importar el archivo de la pieza que se desea maquinar.

Una vez realizado esto, se escriben las características de la herramienta a utilizar (Figura 4-11), posteriormente se desplaza a la herramienta a la posición de HOME. Posteriormente se acerca la herramienta al punto inicial donde empezará a maquinar y se oprime el botón de “Cálculo de voxel” y finalmente el botón de “Barrido”.

Herramienta	
Diámetro	4.0
Largo	15.0

Figura 4-11. Sección de anotación de las características de la herramienta de corte

4.1.1.3 Imágenes obtenidas del CAM

Se muestra a continuación las pantallas obtenidas por el software desarrollado para cada una de las rebanadas distribuidas en cuatro pantallas como se muestra en la Figura 4-12, donde en cada una se realiza cierta operación para la obtención del perfil a seguir.

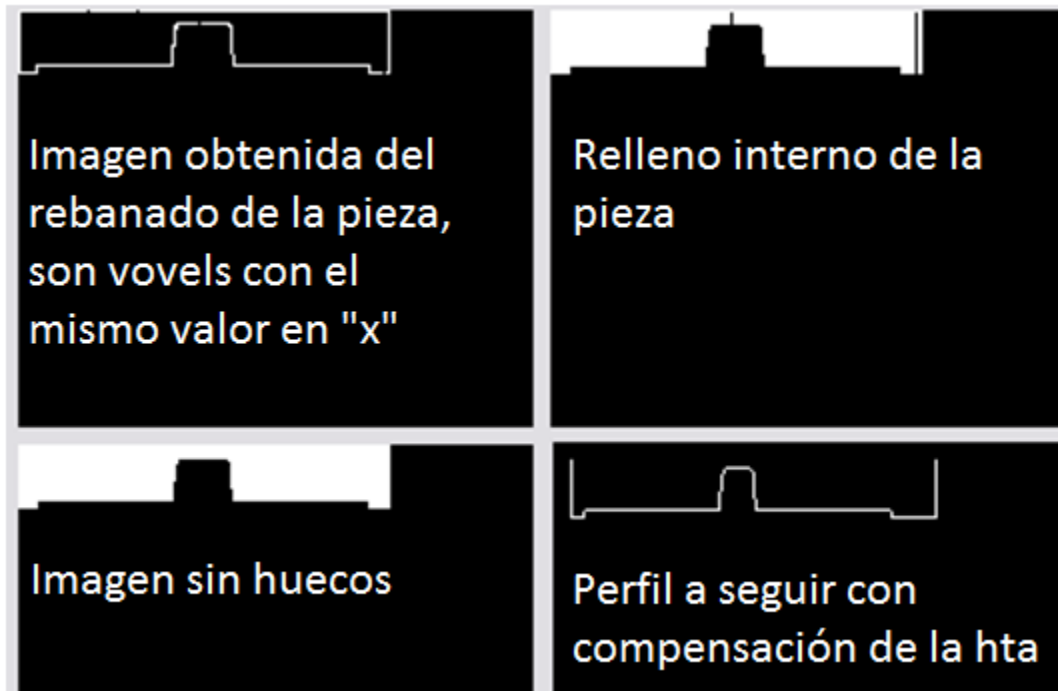


Figura 4-12. Distribución de las pantallas de la interfaz de desarrollador

En la rebanada número 1 se obtuvieron las imágenes que se muestran en la Figura 4-13

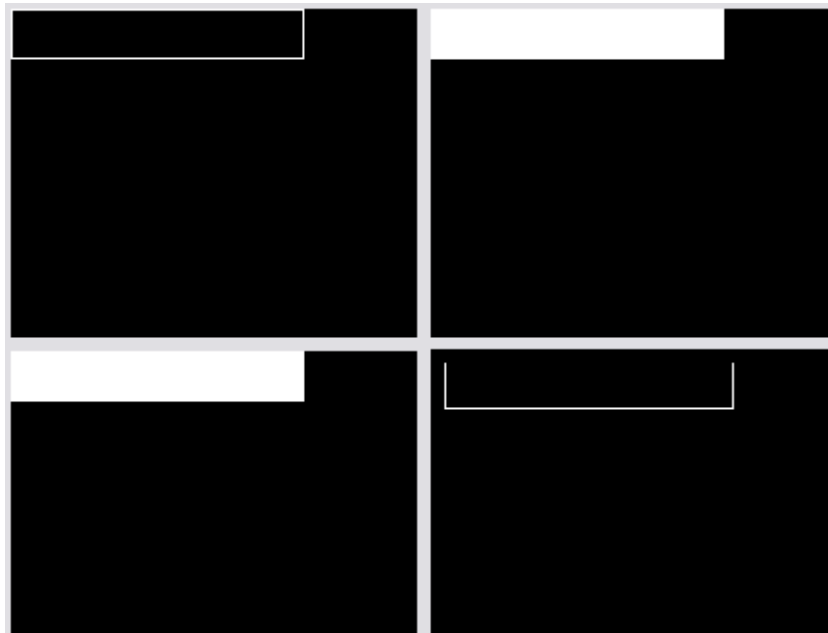


Figura 4-13. Pantallas tomadas de la rebanada 1

Como se observa, la imagen obtenida de la impresión de los voxels no presenta ningún hueco, ni duplicidad, por lo que en los algoritmos posteriores para mejorar la imagen, al menos en este caso sólo reafirma el perfil a seguir. El mecanismo de 3 grados de libertad,

describe la trayectoria de la imagen del perfil a seguir, y se muestra a continuación en la Figura 4-14 la fotografía del maquinando un bloque de madera.



Figura 4-14. Maquinado de rebanada

Como la máquina está empezando a manufacturar la pieza, el proceso de maquinado es el de desbaste, por lo que el avance de corte en el eje x es de 5 rebanadas. En la rebanada número 25 se obtuvieron las imágenes que se muestran en la Figura 4-15.

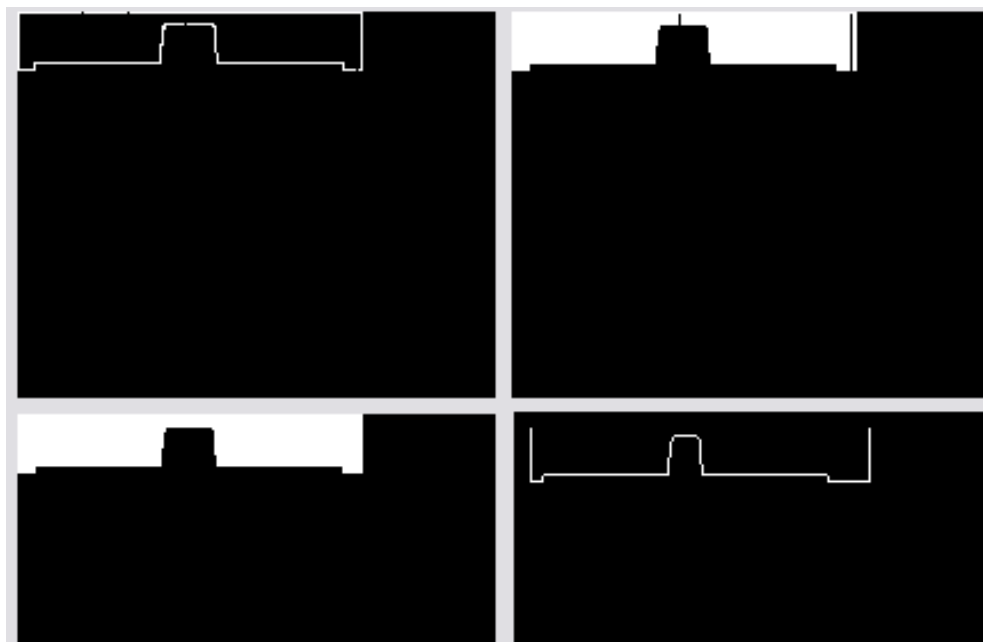


Figura 4-15. Pantallas tomadas de la rebanada 25

En las pantallas de la Figura 4-15 se aprecia en la imagen superior izquierda algunos huecos en el contorno de la pieza, y se ve reflejada en la imagen superior derecha el efecto de los huecos, que son rellenados ya en la imagen inferior izquierda y en la última se aprecia la dilatación conforme al diámetro de la herramienta, y es el perfil que sigue la herramienta.

En la rebanada número 50 se obtuvieron las imágenes mostradas en la Figura 4-16, donde se aprecia más la forma de la cavidad del molde.

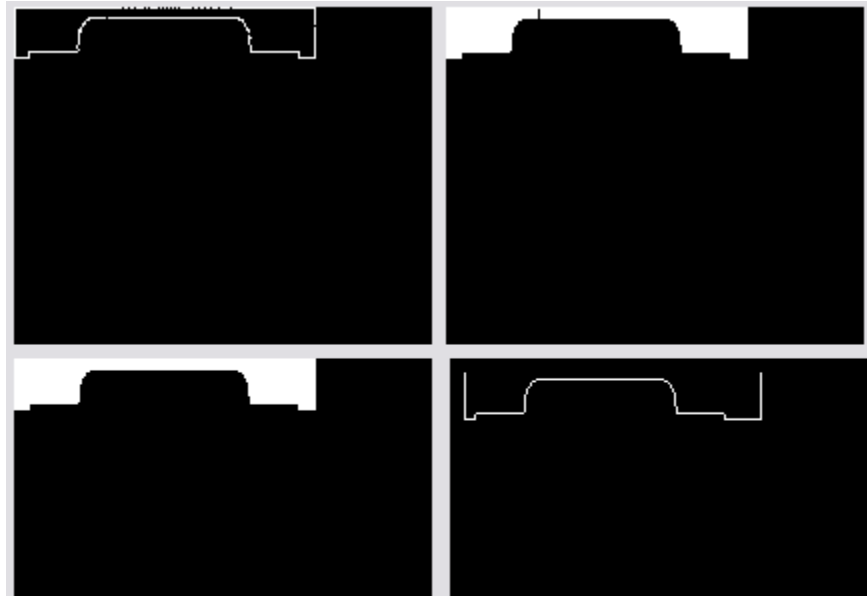


Figura 4-16. Pantallas tomadas de la rebanada 50

Se observa en las imágenes anteriores las fallas en el contorno, así también como las correcciones de las mismas para obtener el perfil compensado.

A continuación se muestran las imágenes de las rebanada número 75 en la Figura 4-17 y la número 120 en Figura 4-18, donde en la imagen correspondiente a la rebanada 75 se observa que no tiene problemas de huecos ni duplicidad, al igual que en las primeras rebanadas, sin embargo son tratadas para obtener el perfil compensado acorde al diámetro de la herramienta.

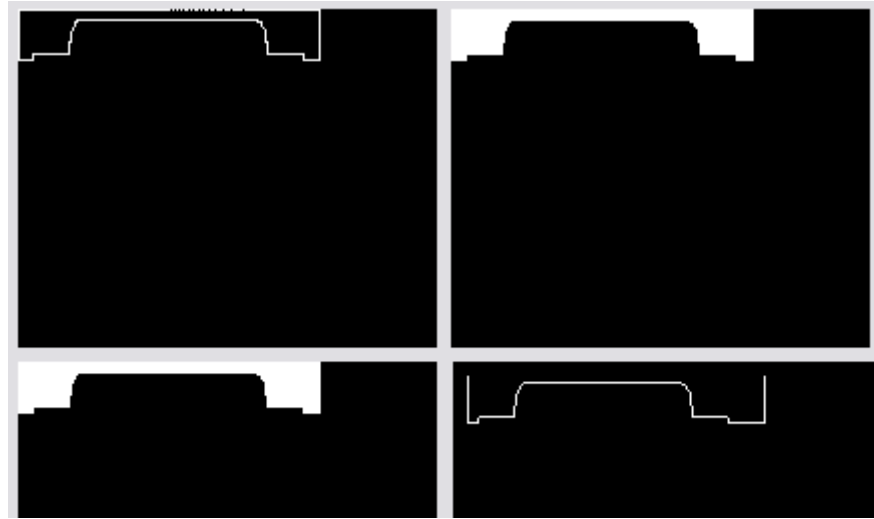


Figura 4-17. Pantallas tomadas de la rebanada 75

En la rebanada número 120 se obtuvo las siguientes imágenes mostradas en la Figura 4-18, donde es posible apreciar la cantidad de duplicidades en lo que corresponde al inicio de la cavidad y de dos huecos en el contorno, pero que finalmente con los algoritmos se resuelven dichos problemas.



Figura 4-18. Pantallas tomadas de la rebanada 120

El proceso de obtener las imágenes y desbastar continúa hasta terminar con todas las rebanadas, posteriormente se aumenta la profundidad en el corte y se vuelven a maquinar hasta terminar. Finalmente con el proceso del acabado se realiza con una profundidad de corte pequeño y con un avance de rebanada de 1, para que la pieza quede terminada.

A continuación se muestra una fotografía en la Figura 4-19 de la herramienta de corte con una profundidad de 0.1 milímetros y con un avance de 1 rebanada en el proceso de acabado.



Figura 4-19. Fotografía del maquinado de la rebanada en acabado

Finalmente lo que se hace es extraer la pieza del bloque, cortando el pequeño bloque como se muestra en la Figura 4-20.



Figura 4-20. Fotografía de la pieza terminada

4.1.2 Corazón de molde

Una vez terminado de fabricar la cavidad del molde, toca el turno a la fabricación corazón, cuyas dimensiones son inferiores a la cavidad para que el espacio entre ellos sea el espesor del material en el blíster, el dibujo del corazón se muestra a continuación en la Figura 4-21.

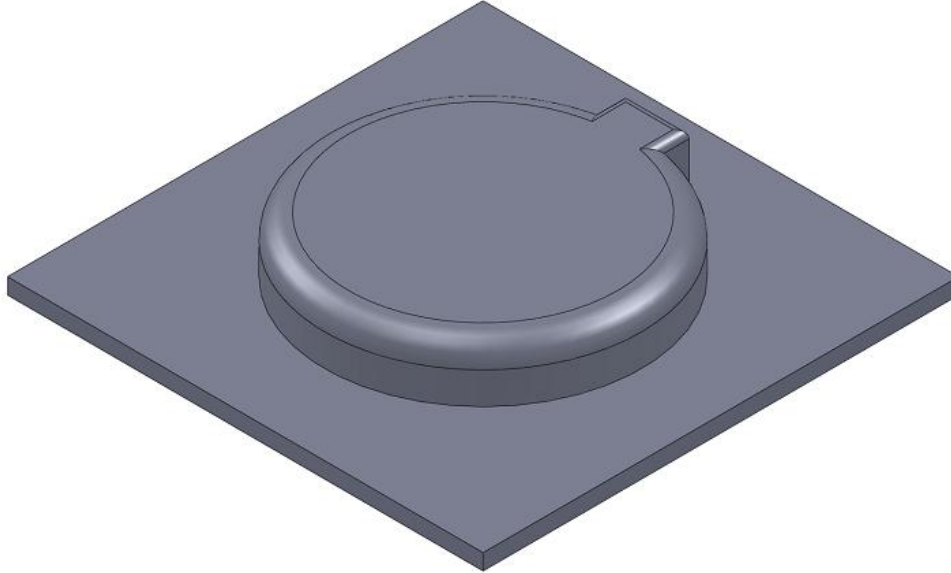


Figura 4-21. Corazón del molde

4.1.2.1 Pieza en formato stl

Se repite el procedimiento de guardar la pieza con un nombre y en formato stl, y al hacerlo, aparecerá una ventana indicando las características del archivo, como se muestra en la Figura 4-22.

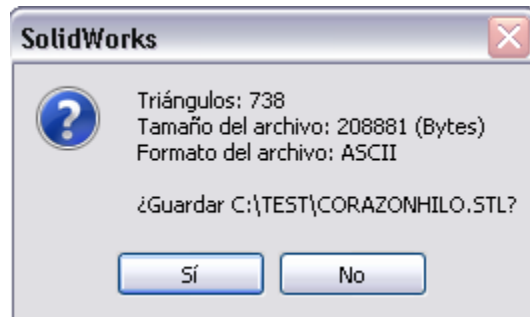


Figura 4-22. Ventada de características del formato stl

Por lo que la superficie de la pieza se forma con una serie de 738 triángulos, como se muestra en la Figura 4-23

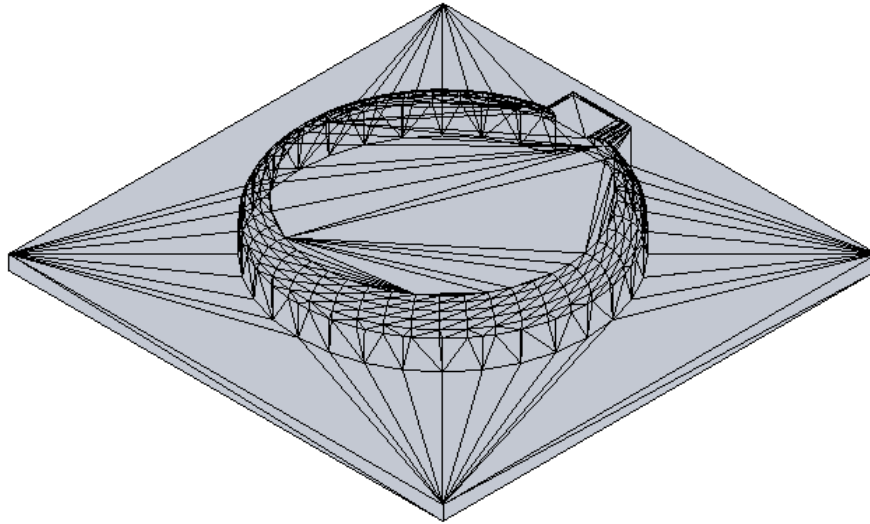


Figura 4-23. Superficie de la pieza en formato stl

4.1.2.2 Imágenes obtenidas del CAM

A continuación se muestran las pantallas capturadas. Generalmente, las primeras rebanadas cumplen con las características del perfil deseado, y como se observa en la Figura 4-24, no existen fallas, salvo el hueco interior, pero que no afecta al perfil.



Figura 4-24. Imagen de la rebanada 1 del corazón del molde

A continuación en la Figura 4-25 se muestra una fotografía del maquinado de las primeras rebanadas.

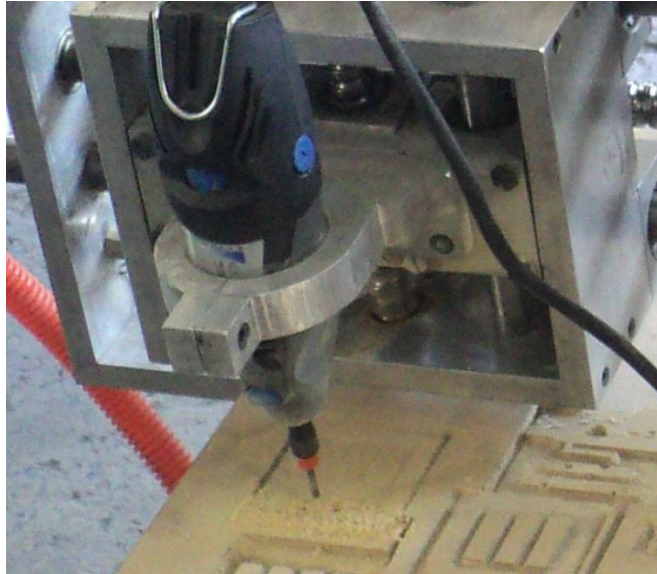


Figura 4-25. Fotografía de maquinado

Se aprecia, como el aserrín se acumula en la pieza, por lo que es necesario contar con una aspiradora para quitar las virutas y poder visualizar la pieza maquinándose.

Se muestra a continuación las imágenes de la rebanada 25 en la Figura 4-26, donde se aprecia que la trayectoria corresponde a nariz del corazón del molde, y se alcanza apreciar que no tiene muchas irregularidades ya que sólo cuenta con un hueco en el contorno.

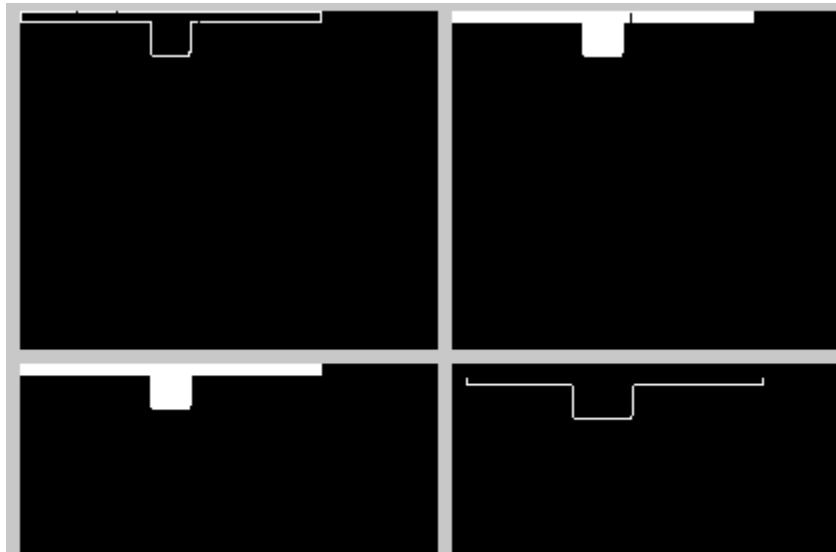


Figura 4-26. Imagen capturada de la rebanada 25

En la imagen de la Figura 4-27, se muestra la PC donde se aprecian las 4 imágenes que muestran el proceso de obtención de perfil, y del lado izquierdo el mecanismo, observándose que durante el proceso de maquinado no existe la intervención del usuario.

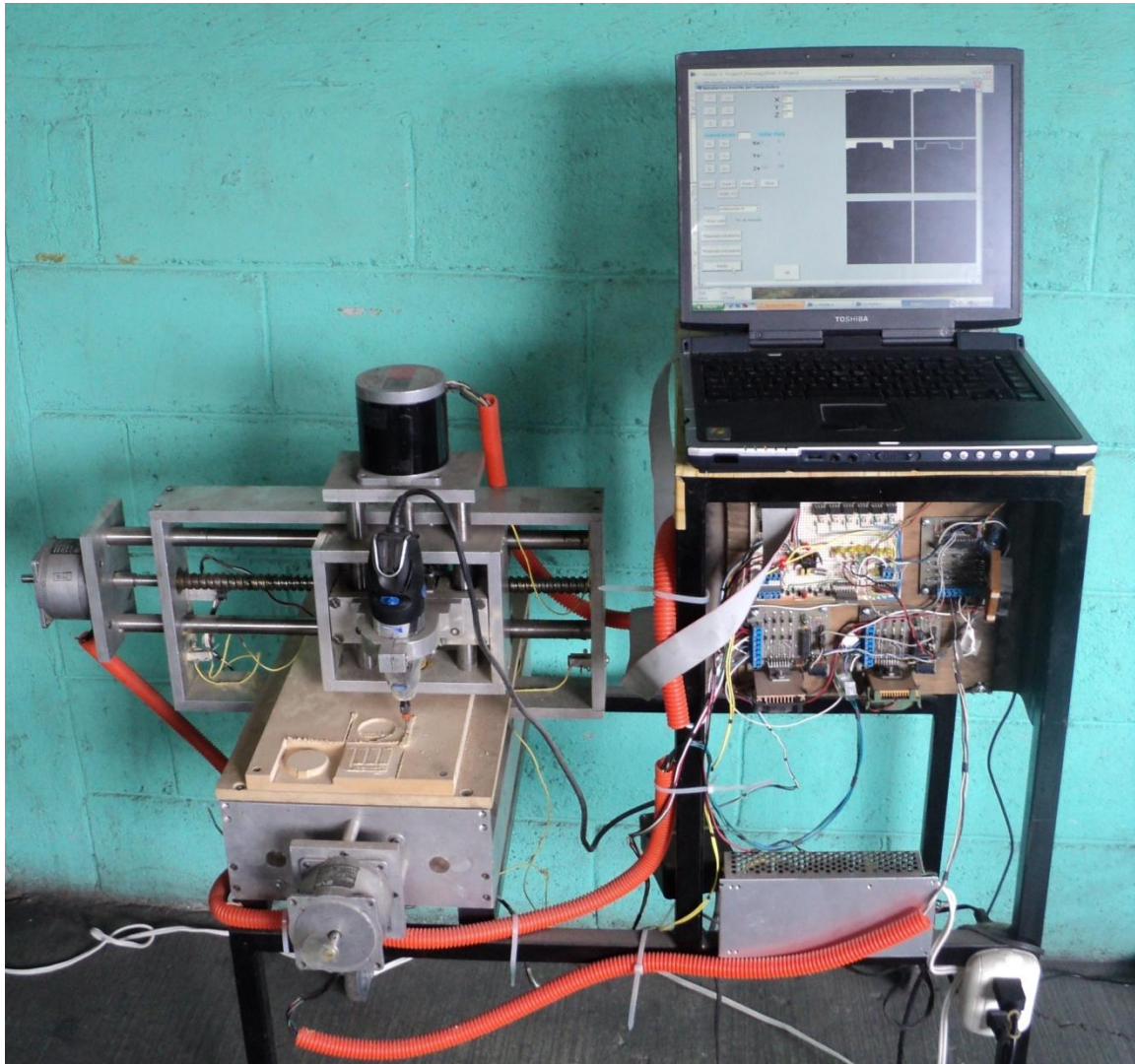


Figura 4-27. Fotografía del mecanismo maquinando siguiendo el perfil en la computadora

El proceso de maquinado se realiza primeramente con el desbaste, con un avance de 5 rebanadas y para el acabado con avance de 1, cuando se aproxima al producto final, bien utilizando el mismo cortador o de preferencia uno de menor diámetro.

En la imagen de la Figura 4-28 se muestra el maquinado de la pieza en el proceso de desbaste, en donde se alcanzan a ver las huellas de la herramienta de corte.

En la Figura 4-29, se muestra el proceso de acabado de la pieza, donde se observa que la calidad de la misma mejora



Figura 4-28. Fotografía del maquinado de las últimas rebanadas en el desbaste

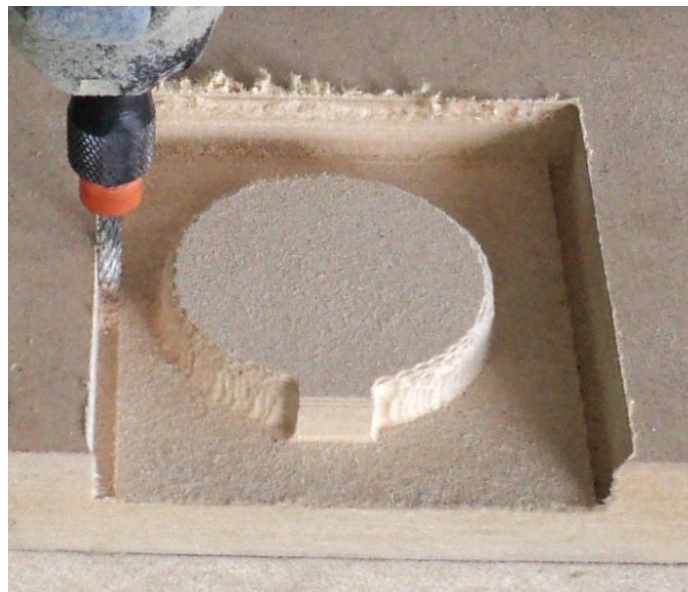


Figura 4-29. Fotografía del maquinado en la etapa de acabado

Finalmente las piezas tanto la cavidad como el corazón, en forma virtual y las reales fabricadas en el prototipo son como las que se muestran en la Figura 4-30y en la Figura 4-31

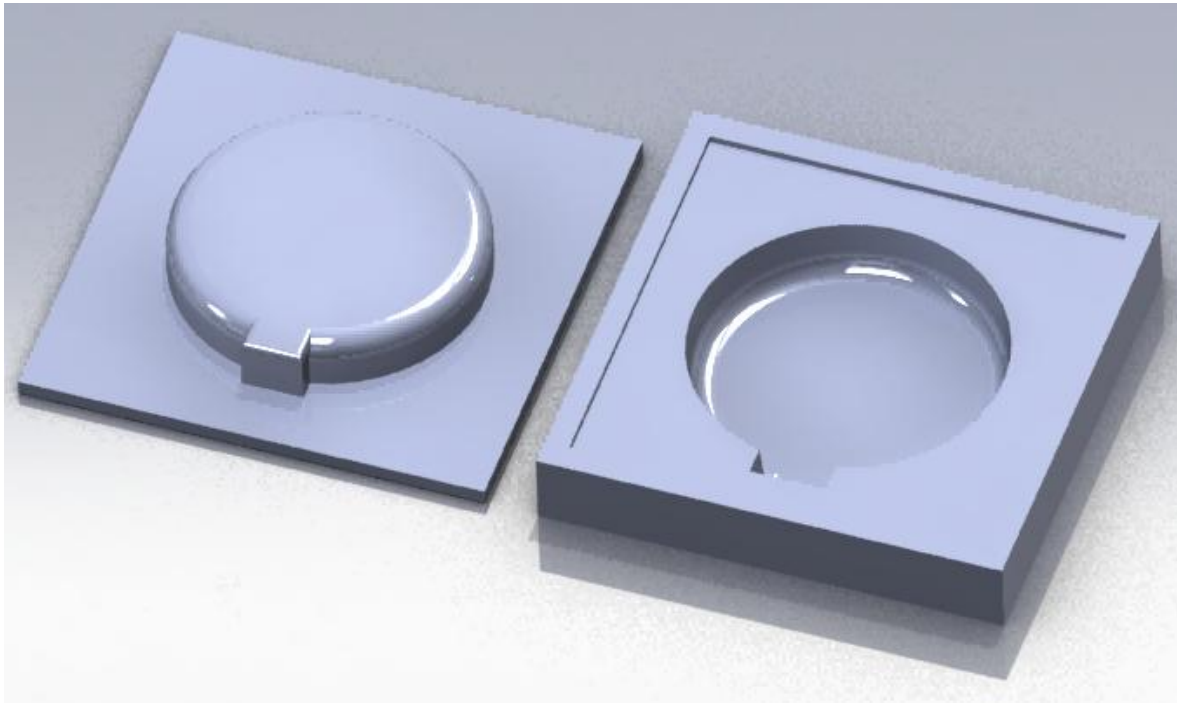


Figura 4-30. Renderizado de las piezas en CAD



Figura 4-31. Fotografía del molde maquinado

Ambas piezas tienen una rugosidad media, sin embargo si la herramienta de corte es mucho más delgada que la utilizada, la calidad hubiese sido mucho mayor, y durante el proceso no existió ninguna intervención del usuario, salvo para mover la herramienta en el lugar para iniciar el maquinado.

4.2 Relieve

Se presenta un caso en el que se muestra la forma de la pieza con relieve, donde se tiene un plano inclinado, curvas de distintos radios, suponiendo que es el modelo para un molde de arena para fundición (Figura 4-32). Como en los ejemplos anteriores, se guarda en archivo STL (Figura 4-33) en código ASCII y a continuación el CAM se encarga del cálculo de los voxels y de la obtención de la rebanada y del perfil a seguir.

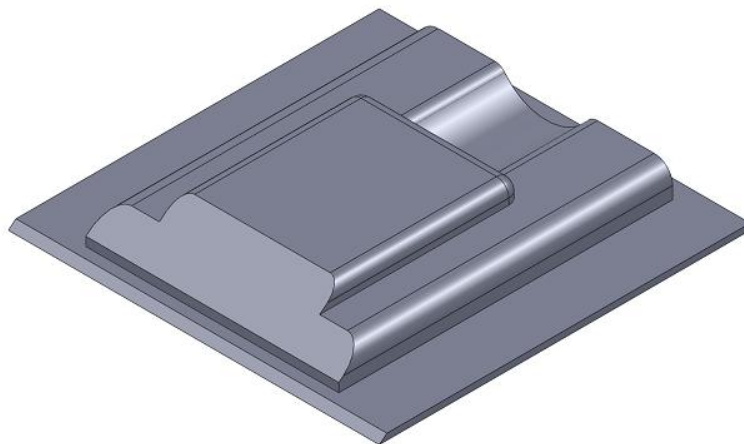


Figura 4-32. Dibujo del modelo a fabricar

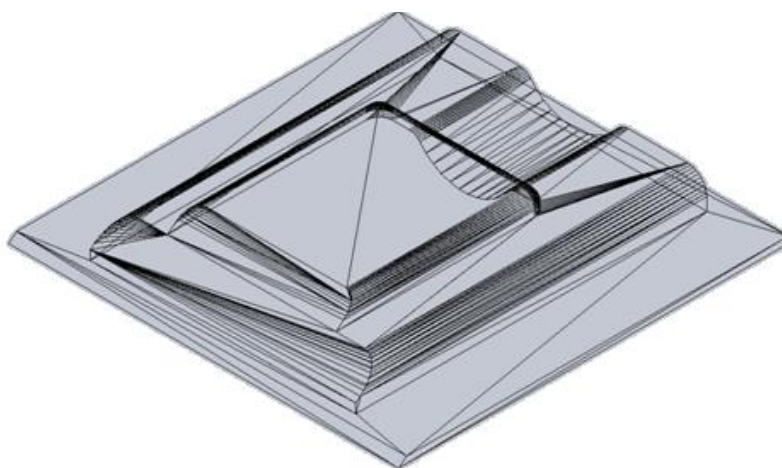


Figura 4-33. Superficie formada por triángulos en el formato stl

Sólo para mostrar un ejemplo, de la rebanada número 75 se obtuvo las siguientes imágenes que se observan en la Figura 4-34, donde se observa que a pesar de estar en una zona donde existe un plano horizontal, existe el problema de los huecos y se presentan 5.

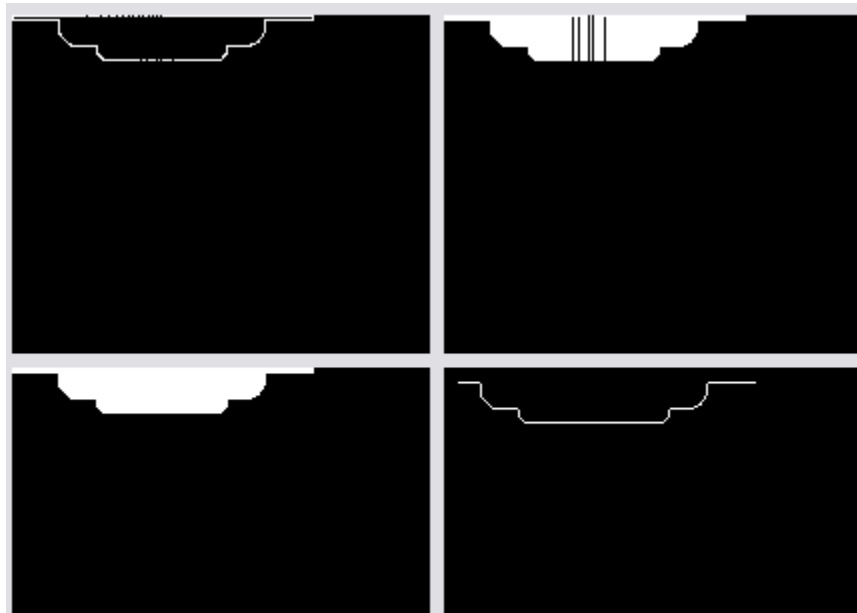


Figura 4-34. Pantallas tomadas de la rebanada número 75

Se maquinan las rebanadas en desbaste y con acabado y finalmente se tiene la pieza terminada, como se observa en la Figura 4-35 la curva maquinada queda bien incluso la curva de las orillas, sin embargo existió una falla en la curva encerrado en la elipse, debido al ancho de la herramienta, ya que si hubiera sido más delgada, hubiera tenido un mejor acabado.

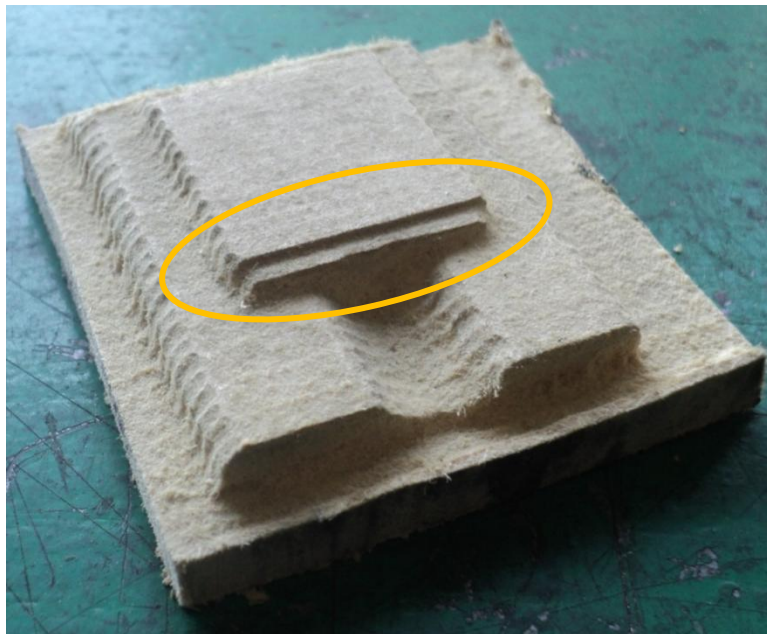


Figura 4-35. Fotografía isométrica del modelo terminado

De la misma manera ocurrió con el plano inclinado que se muestra en Figura 4-36, en donde se aprecia escalonado, que sin embargo puede resolverse con una herramienta aún más delgada.



Figura 4-36. Fotografía del modelo terminado

4.3 Ejemplos

Se muestran también las fotografías de otros ejemplos realizados en la máquina, como por ejemplo, en la Figura 4-37 se puede apreciar que se han maquinado las letras IPN, así también se nota la huella de la herramienta de corte.



Figura 4-37. Fotografía del maquinado de las letras IPN

En la imagen siguiente de la Figura 4-38 se muestra la diferencia entre el desbaste y el acabado. En la parte superior, el avance fue de 1 rebanada con una herramienta de corte de 1 mm., mientras que en la parte inferior el avance fue de 5 rebanadas con una herramienta de corte de 3 mm.



Figura 4-38. Fotografía del logo de "Hecho en México"

Como es de esperarse y como se observa en la imagen de la Figura 4-38, no es posible realizar detalles más pequeños que el mismo diámetro de la herramienta, por lo que la idea de remplazar las herramientas de corte por una tecnología que permita el maquinado en proporciones mucho más pequeñas es posible realizarlo.

Conclusiones y trabajo a futuro

Conclusiones

Después de las pruebas obtenidas del CAM y comprobadas en el mecanismo, las conclusiones que se obtuvieron fueron los siguientes:

Por la selección del barrido como estrategia de maquinado, se pudo obtener un sistema de manufactura asistida por computadora exento de errores. Esto fue gracias a que el sistema genera de manera virtual todos los puntos que forman la superficie de la pieza, conociendo así la geometría de la misma, y al seccionar la superficie en rebanadas, se obtienen los perfiles de cada una de los cuales se desprenden los movimientos de los motores para describir las trayectorias.

Debido a que los movimientos de los motores se generan del seguimiento del perfil, y al ser éste parte del contorno pero sin la base, y al cumplir con las características de línea con el ancho de un pixel y continuo, no se tiene un programa al que se le tenga que revisar y depurar, por lo que se elimina el tiempo destinado a dicho trabajo, reduciendo tiempo y costo en producción.

Gracias a que con este sistema se puede pasar del dibujo en CAD directamente al maquinado, es decir, que no existe trabajo previo al maquinado después del dibujo hecho en CAD, es costeable la fabricación de una sola pieza, y con mayor razón, el de una producción en serie.

Los algoritmos desarrollados e implementados fueron de costo computacional bajo, ya que el cálculo de los puntos de la superficie es rápido, así también la obtención del perfil de cada rebanada.

Dada la calidad del acabado superficial de la pieza depende en gran medida de la herramienta, resulta deseable que la herramienta de corte sea lo más delgada posible, por lo que el láser sería una herramienta de corte ideal para este sistema, ya que el ancho de corte de un láser puede ser de unas cuantas micras.

Sugerencias para trabajo futuro

Fabricar un mecanismo con dimensiones mayores para poder manufacturar piezas de mayor tamaño, así como aumentar el número de grados de libertad para no limitar el tipo de piezas a fabricar, dado que el mecanismo utilizado es de 3 grados de libertad, no permite fabricar piezas de mayor complejidad. Al menos con un grado de libertad en el eje z, que gire la herramienta de corte en el plano YZ, para que de ésta manera, la herramienta siempre corte de forma perpendicular a la superficie de maquinado y así aumentar la calidad en el maquinado.

Así también, al aumentar el tamaño del mecanismo, se utilizaría un motor más grande para el giro de la herramienta y poder adaptar una boquilla con diferentes tamaños en el diámetro y utilizar cortadores verticales que se encuentran fácilmente en el mercado, además de que esto haría factible la manufactura de piezas en materiales diferentes al aluminio, como lo es el acero al carbón u otras aleaciones que requieren de cortadores especiales.

Otra de las sugerencias para el trabajo a futuro es utilizar las coordenadas del vector de posición normal a la superficie del triángulo que entrega el archivo STL para que en

conjunto con la aplicaciones del láser y con la pieza virtual desarrollar impresoras 3D y poder fabricar piezas, moldes o modelos tan complejos, incluso con cavidades como los ejemplos que se muestran más adelante [44].

En la imagen de la se muestra una válvula neumática funcional, impresa con todos los elementos en un mismo proceso por lo que puede realizar el movimiento como si fuera la real. Además se ha impreso a color, de ésta manera es fácil identificar las partes de las que estaría hecha el ensamble de la válvula real [45][46][47].

La impresión del modelo puede ser en tamaño real para observarlo, revisarlo, verificar que esté bien en cuanto al funcionamiento y visualizar posibles fallas en el diseño, rugosidades, etc.

Otra sugerencia para trabajo a futuro es utilizar el láser como herramienta de corte, dado que el CAM propuesto soporta perfectamente con esta idea, ya que al sustituir al cortador convencional, aporta una ventaja mayor porque con el láser es factible tener un diámetro de unas cuantas micras para poder realizar maquinados tan finos y con ello poder marcar o grabar dentro del mismo molde.

Una de las aplicaciones del láser en las que está teniendo mucho auge es en el grabado de moldes para obtener relieves que difícilmente se obtendrían con una herramienta de corte usual como el cortador vertical, debido a que a pesar de poder encontrar algunos tan delgados de unas cuantas decimas de milímetro y de tener buena resistencia mecánica, son susceptibles de quebrarse ya sea por fatiga, mal uso de la herramienta, etc.

Referencias

- [1] International Organization for Standardization – ISO 6983/1– Numerical control of machines – Program format and definition of address words – Part 1: Data format for positioning, line and contouring control systems. First edition, (1982).
- [2] Garrido, J., Hardwick, M. (2006). “A traceability information model for CNC manufacturing”. *Computer-Aided Design*, 38, 540-551.
- [3] Albert M. 2002, “STEP NC - The End of G Codes?” *Modern Machine Shop*, 70-80.
- [4] F. C. Teruel 2007, “Control numérico y programación” Ed. AlfaOmega, 25-50.
- [5] X. Guo, Y. Liu, K. Yamazaki, K. Kashihara, M. Fujishima, (2008), “A study of a universal NC program processor for a CNC”, *International Journal Advanced Manufacturing Technology*.
- [6] Garrido, J., Hardwick, M. (2006). “A traceability information model for CNC manufacturing”. *Computer-Aided Design*, 38, 540-551.
- [7] Chen, L., Song, Z. Feng, L. (2003), “Internet enabled real-time collaborative assembly modeling via an e-Assembly system: status and promise”, *Computer-Aided Design*, 35-47.
- [8] Hoppe H. (1994), *Surface Reconstruction from Unorganized Points*, PhD Thesis, University of Washington USA, 30-67.
- [9] Boubekeur T. (2004), *Reconstruction de surface à l'aide de surfaces de subdivision*, Research Master MM, UQTR, Canada.
- [10] Choi, B. K. (1991), *Surface Modeling for CAD-CAM*, Elsevier.
- [11] Xu, X.W., He, Q. (2004). *Striving for a total integration of CAD, CAPP, CAM and CNC. Robotics and Computer-Integrated Manufacturing*.
- [12] de Jager, P. J. (1996), “Using Slanted and Ruled Layers for Rapid Prototyping,” *Proceedings of the 5th European conference on Rapid Prototyping and Manufacturing*, Finland, 15-30.
- [13] Choi, B. K. (1991), *Surface Modeling for CAD-CAM*, Elsevier.
- [14] G. Allen Brady and John W. Halloran, (1997), “Stereolithography of Ceramic Suspensions”, *Rapid Prototyping Journal*, 61-65.
- [15] Karunakaran, K. P., Shivmurthy Dibbi, P. Vivekananda Shanmuganathan, Srinivasarao Kakaraparti and D. Sathyanarayana Raju (April 2002), “Efficient Stock Cutting in Laminated Manufacturing”, *Computer-Aided Design*, 34(4), 281-298.
- [16] Jamieson, R. and Hacker, Herbert (1995), “Direct slicing of CAD model for rapid prototyping,” *Rapid Prototyping Journal*, 1(2), 4-12.

-
- [17] Im, Y. and Walczyk, D. F. (2002), "Development of a Computer-Aided Manufacturing System for Profiled edge lamination Tooling," Transactions of the ASME, Journal of Manufacturing Science and Engineering, 754-761.
- [18] M.L. Griffith and J. W. Halloran, (1997) "Scattering of Ultraviolet Radiation in Turbid Ceramic Suspensions", Journal of Applied Physics, 25, 38-46.
- [19] E. D. De Garmo, J.T.Black, R.A. Rosher. "Materiales y procesos de fabricación" Edit. Ed. Reventé S.A.
- [20] Hans Gastrow y Peter-Hunger, "Moldes de inyección por plástico", Dipl-ing Plastic Communication, S.L. 4ta edic, Grupo Emitec S.A.45-69,
- [21] Menges, G. et. al. (1986), "How to Make Injection Molds", Hanser Publisher, Munich.
- [22] Kruth, J.P. and Willems, R. (1994), "Intelligent support system for the design of injection moulds", Journal of Engineering Design, 4(5), 339-351.
- [23] V. M. Faires, (1987), "Design of machine elements", Ed. UTEHA, 4th Edition, 201-234.
- [24] R. Jouvinall (1997) "Fundamentos de diseño para ingeniería mecánica" Ed. Limusa, 1ra ed., 311-316.
- [25] S. Cetinkunt (2007) "Mechatronics" Ed. Jhon Wiley & Sons, Inc. 87-89.
- [26] S. Cetinkunt (2007) "Mechatronics" Ed. Jhon Wiley & Sons, Inc. 461-478.
- [27] P. Acarnley, "*Stepping Motors: A Guide to Theory and Practice*", 4th ed. Stevenage, U.K.: Inst. Elect. Eng., 2002.
- [28] D.G. Alciatore, M.B. Hiestand (2008) "Introducción a la mecatrónica y los sistemas de medición" Ed. Mc Graw Hill, 3rd ed., 413-421.
- [29] T. Kanjo and A. Sugawara, *Stepping Motors and Their Microprocessor Controls*. Oxford, U.K.: Clarendon, 1994.
- [30] P. Acarnley, *Stepping Motors: A Guide to Theory and Practice*, 4th ed. Stevenage, U.K.: Inst. Elect. Eng., 2002.
- [31] Baluta, G, "Microstepping Mode for Stepper Motor Control", IEEE International Symposium on Signals Circuits and Systems, 2, pp.1-4, July 2007.
- [32] F. Bianconi "Bridging the gap between CAD and CAE using STL files". Department of Industrial Engineering, University of Perugia, Perugia, Italy.
- [33] Y.H. Chen and C.T. Ng (1997), Integrated Reverse. Engineering and Rapid Prototyping. Computers & Industrial Engineering, 33(3-4), 481-484.
- [34] Y.H. Chen and Y.Z. Wang (1999), Genetic algorithms for re-triangulation in the context of reverse engineering Computer-Aided Design, 31(4), 261-271.
- [35] P. Soille, *Morphological Image Analysis - Principles and Applications*, Springer Verlag, 1999.
-

-
- [36] Hyun Geun Yu: Morphological Image Segmentation For Co-Aligned Multiple Images Using Watersheds Transformation, 2004.
 - [37] S.Beucher, Interpolation of sets, of partitions and of functions, In: H.Heijmans and J.Roerdink Mathematical morphology and its application to image and signal processing, Kluwer, 1998.
 - [38] J.Serra, Image Analysis and Mathematical Morphology vol.1, Academic Press, 1982.
 - [39] Xiaojun Qi: Advanced (Mid-Level) DIP - Morphological Image Processing.
 - [40] J.Serra, Image Analysis and Mathematical Morphology vol.2, Academic Press, 1988.
 - [41] Dharmaraj, G.: Algorithms for Automatic Vectorization of Scanned Maps, 2005.
 - [42] "Volume Graphics" IEEE Computer, Vol. 26, No. 7 July 1993, pp. 51-64.
 - [43] A. Weitzenfeld, "Ingeniería de software orientada a objetos con UML, java e internet", ediciones paraninfo, s.a., 2005.
 - [44] de Jager, P. J., Broek, J. J. and Vergeest, J. S. M. (1996), "Rapid prototyping: extending the layer concept," Proceedings of the 2nd International Conference on Rapid Product Development, Stuttgart, 325-334.
 - [45] Hope, R. L., Jacobs, P. A. and Roth, R. N. (1997), "Adaptive Slicing with sloping layer surfaces," *Rapid Prototyping Journal*, 3(3), 89-98.
 - [46] Hope, R. L., Jacobs, P. A. and Roth, R. N. (1997), "Rapid Prototyping with sloping surfaces," *Rapid Prototyping Journal*, 3(1), 12-19.
 - [47] Hope, R. L., Rick, A. T. and Roth, R. N. (1996), "Layer Building with Sloping Edges for Rapid Prototyping of large objects," Proceedings of the 5th European conference on Rapid Prototyping and Manufacturing, Finland, June