



universidad  
de león



# Escuela de Ingenierías Industrial, Informática y Aeroespacial

## GRADO EN INGENIERÍA MECÁNICA

Trabajo de Fin de Grado

PROGRAMACIÓN EN CÓDIGOS ISO PARA CONTROL  
NUMÉRICO FAGOR 8050

PROGRAMMING IN ISO CODES FOR NUMERICAL CONTROL  
FAGOR 8050

Autor: Kaloyan Yordanov Yordanov  
Tutor: Susana Martínez Pellitero

(Febrero, 2021)

**UNIVERSIDAD DE LEÓN**  
**Escuela de Ingenierías Industrial, Informática y**  
**Aeroespacial**

**GRADO EN INGENIERÍA MECÁNICA**  
**Trabajo de Fin de Grado**

**ALUMNO:** Kaloyan Yordanov Yordanov

**TUTOR:** Susana Martínez Pellitero

**TÍTULO:** Programación en Códigos ISO para Control Numérico FAGOR 8050

**TITLE:** Programming in ISO Codes for Numerical Control FAGOR 8050

**CONVOCATORIA:** Febrero, 2021

**RESUMEN:**

Este proyecto irá dirigido a la utilización del software Adobe Animate para desarrollar una herramienta educativa. Ésta se centrará en el Control Numérico en los Procesos de Mecanizado, y su función, será hacer transmitir los conocimientos de una manera didáctica, con ejemplos prácticos y una teoría mucho más enfocada al alumno principiante. En la introducción se hablará de la necesidad de desarrollar este tipo de herramientas y el cambio en la docencia que se está produciendo actualmente.

Posteriormente se desarrollará la metodología y el software utilizado, explicando conceptos elementales del Control Numérico como pueden ser las Interpolaciones Lineales o las Compensaciones, y los programas utilizados para desarrollar todo el trabajo.

Seguidamente se hará un análisis de programa Adobe Animate, describiendo su entorno de trabajo, las herramientas de las que consta y cómo poder elaborar una plantilla de presentación.

Finalmente se estudiará la aplicación desarrollada, las partes de las que consta, el árbol de actividades, y la teoría y ejemplos expuestos en cada una de las plantillas.

Por último, se añadirán las conclusiones personales con respecto al trabajo realizado como las conclusiones personales.

**ABSTRACT:**

This project will consist in developing an educational tool using software Adobe Animate.

The main focus of this tool, is to be useful to future students to get started with Numerical in Machining Processes, their functioning and introducing the lectures at beginner level with more didactical features. In the introduction we will talk about the necessity of developing these kinds of tools and the current changes the teaching is suffering.

Next, it will be addressed the methodology and the software used, explaining basic concepts of Numerical Control like linear movements, compensations, and the rest of programs used as aid to develop this project.

Following a deep analysis of Adobe Animate will be done, describing the work environment, the working tools and how to elaborate a basic presentation slide.

To finalize, the developed tool will be under study: the parts in which consists, the focus tree, theory and the examples showed.

At last, personal conclusions about the work and the teaching tool will be added, the same way as the personal conclusion about the idea behind the whole project.

<b>Palabras clave:</b> Control Numérico, herramienta educativa, Adobe Animate.	
<b>Firma del alumno:</b>	<b>VºBº Tutor/es:</b>

# I. Resumen

Este proyecto está orientado al desarrollo de una aplicación multimedia, como herramienta educativa en el contexto de la programación con Control Numérico para procesos de mecanizado. Para ello, se utiliza el software Adobe Animate. El objetivo de esta herramienta es el de transmitir los conocimientos a los alumnos de una manera didáctica, con ejemplos prácticos y una teoría enfocada al alumno principiante. En la introducción se hablará de la necesidad de desarrollar este tipo de herramientas y el cambio en la docencia que se está produciendo actualmente.

Posteriormente se desarrollará la metodología y el software utilizado, explicando conceptos elementales del Control Numérico como pueden ser las Interpolaciones Lineales o las Compensaciones, y los programas utilizados para desarrollar todo el trabajo.

Seguidamente se hará un análisis de programa Adobe Animate, describiendo su entorno de trabajo, las herramientas de las que consta y cómo poder elaborar una plantilla de presentación.

Finalmente se estudiará la aplicación desarrollada, las partes de las que consta, el árbol de actividades, y la teoría y ejemplos expuestos en cada una de las plantillas.

Por último, se añadirán las conclusiones personales con respecto al trabajo realizado como las conclusiones personales.

## II. Abstract

This project will consist in developing an educational tool using software Adobe Animate. The main focus of this tool, is to be useful to future students to get started with Numerical Control in Machining Processes, their functioning and introducing the lectures at beginner level with more didactical features. In the introduction we will talk about the necessity of developing these kinds of tools and the current changes the teaching is suffering.

Next, it will be addressed the methodology and the software used, explaining basic concepts of Numerical Control like linear movements, compensations, and the rest of programs used as aid to develop this project.

Following a deep analysis of Adobe Animate will be done, describing the work environment, the working tools and how to elaborate a basic presentation slide.

To finalize, the developed tool will be under study: the parts in which consists, the focus tree, theory and the examples showed.

At last, personal conclusions about the work and the teaching tool will be added, the same way as the personal conclusion about the idea behind the whole project.

# III. Índice de contenidos

I.	Resumen .....	4
II.	Abstract .....	5
III.	Índice de contenidos .....	6
IV.	Índice de figuras .....	8
V.	Índice de cuadros y tablas .....	10
VI.	Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos .....	11
1.	Introducción .....	12
1.1.	¿Qué es el Control Numérico Computarizado (CNC)?.....	14
1.2	Objetivo .....	18
2.	Funciones del Control Numérico .....	20
2.1	Funciones preparatorias.....	21
2.1.1	Decalajes de origen .....	23
2.1.2	Compensación del radio de la herramienta .....	26
2.2	Ciclos fijos .....	27
2.3	Funciones auxiliares .....	30
2.4	Funciones Complementarias .....	31
3	Software Adobe Animate .....	34
3.1	Configuración de partida .....	35
3.2	Espacio de trabajo .....	36
3.3	Funcionalidad de Adobe Animate .....	38
3.3.1	Línea de tiempo .....	38
3.3.2	Herramientas.....	41
3.3.3	Símbolos y botones .....	42
3.3.4	Código JavaScript.....	45
3.3.5	Compilación .....	48
3.3.6	Publicación.....	49
4	Descripción de la aplicación desarrollada .....	53
4.1	Menú Principal e índice .....	53
4.2	Diapositiva Introducción.....	55
4.2.1	Estructura de un bloque .....	55
4.2.2	Diapositiva de Funciones preparatorias (G) .....	57
4.2.3	Diapositiva de Funciones auxiliares (M) .....	58

4.3	Diapositiva de Interpolaciones .....	59
4.3.1	Posicionamiento rápido (G00).....	60
4.3.2	Diapositiva de Interpolaciones Lineales en régimen de trabajo (G01) .....	61
4.3.3	Diapositiva de Interpolaciones Circulares (G02, G03).....	63
4.4	Diapositiva de Decalajes.....	66
4.5	Diapositiva de Compensaciones.....	69
4.6	Diapositiva de Ciclos fijos .....	71
4.7	Ejemplos prácticos.....	74
4.7.1	Ejemplo práctico 1.....	74
4.7.2	Ejemplo práctico 2.....	82
4.7.3	Ejemplo práctico 3.....	84
4.8	Autoevaluación.....	87
4.8.1	Ejercicio 1.....	88
4.8.2	Ejercicio 2.....	90
4.8.3	Ejercicio 3.....	91
4.8.4	Ejercicio 4.....	92
4.8.5	Ejercicio 5.....	93
4.8.6	Ejercicio 6.....	94
4.8.7	Ejercicio 7.....	95
5	Conclusiones.....	97
6	Referencias .....	99

## IV. Índice de figuras

Figura 1-1. Diagrama de bloques de una máquina CNC [6] .....	16
Figura 1-2. Elaborado con Adobe Animate.....	17
Figura 2-1. Manual FAGOR 8050M [7]. .....	21
Figura 2-2. Desplazamiento lineal “A” a “B” .....	23
Figura 2-3. Traslado de origen de coordenadas.....	25
Figura 2-4. Representación gráfica de G41 y G42 .....	26
Figura 2-5. FAGOR 8050 [7] .....	27
Figura 2-6. FAGOR 8050 [7] .....	29
Figura 3-1. Configuración inicial .....	35
Figura 3-2. Interfaz de usuario de Adobe Animate .....	36
Figura 3-3. Línea de tiempo.....	38
Figura 3-4. Tipos de fotograma [9] .....	39
Figura 3-5. Añadir capa y eliminar capa .....	39
Figura 3-6. Ocultar y bloquear capas.....	40
Figura 3-7. Ejemplo de Capa de Máscara y Capa de Guía .....	40
Figura 3-8. Herramientas.....	41
Figura 3-9. Ampliación de la Figura 3-8.....	41
Figura 3-10. Creación de un símbolo.....	43
Figura 3-11. Nombre de instancia .....	43
Figura 3-12. Configuración de un botón.....	44
Figura 3-13. Fragmentos de código .....	45
Figura 3-14. Menús desplegables de Fragmentos de código .....	46
Figura 3-15. Ventana Acciones .....	46
Figura 3-16. Fragmento de código “Navegación en la línea de tiempo”.....	47
Figura 3-17. Fragmento de código “Controladores de eventos” .....	48
Figura 3-18. Ejemplo mensaje en el panel de Salida .....	49
Figura 3-19. Menú Exportar .....	50
Figura 3-20. Configuración de publicación .....	51
Figura 3-21. Compartir y publicar.....	52
Figura 3-22. Publicar HTML Canvas .....	52
Figura 4-1. Índice .....	54
Figura 4-2. Ejemplo de menú desplegable del tema Introducción .....	55
Figura 4-3. Estructura de un bloque.....	56
Figura 4-4. Pestaña adicional Número de bloque .....	57
Figura 4-5. Funciones Preparatorias (G).....	58
Figura 4-6. Funciones Auxiliares (M) .....	59
Figura 4-7. Posicionamiento rápido (G00).....	60
Figura 4-8. Interpolaciones lineales (G01).....	62
Figura 4-9. Diagrama Intepolaciones Lineales.....	63
Figura 4-10. Interpolaciones Circulares (G02, G03).....	64
Figura 4-11. Pestaña auxiliar Interpolaciones Circulares .....	64
Figura 4-12. Parámetros I,J.....	65
Figura 4-13. Diagrama de Interpolaciones Circulares .....	66
Figura 4-14. Cotas según X, Y, Z.....	67
Figura 4-15. Decalaje .....	67



Figura 4-16. Cero pieza .....	68
Figura 4-17. Pestaña adicional Decalaje.....	69
Figura 4-18. Compensación del radio de la herramienta .....	70
Figura 4-19. Pestaña auxiliar compensaciones de herramienta .....	71
Figura 4-20. Ciclos Fijos .....	72
Figura 4-21. Pestaña auxiliar Ciclo G83 .....	73
Figura 4-22. Ejemplo práctico 1.....	75
Figura 4-23. Ejemplo práctico 1-Planeado .....	76
Figura 4-24. Ejemplo práctico 1-Vista eje Z.....	77
Figura 4-25. Ejemplo práctico1-Contorneado .....	79
Figura 4-26. Ejemplo práctico 1-Cajera central .....	80
Figura 4-27. Ejemplo Práctico 1-Taladrado .....	81
Figura 4-28. Ejemplo práctico 1-Taladrado mediante Ciclo Fijo.....	82
Figura 4-29. Ejemplo práctico 2.....	83
Figura 4-30. Parámetros I y J .....	84
Figura 4-31. Ejemplo práctico 3.....	84
Figura 4-32. Ejemplo práctico 3-Planeado .....	85
Figura 4-33. Cálculo de las coordenadas A y B.....	86
Figura 4-34. Ejemplo práctico 3-Contorneado .....	87
Figura 4-35. Ejercicio 1 .....	88
Figura 4-36. Ejercicio 1-Opción correcta .....	89
Figura 4-37. Ejercicio 1-Opción incorrecta .....	89
Figura 4-38. Ejercicio 2 .....	90
Figura 4-39. Ejercicio 3 .....	91
Figura 4-40. Ejercicio 4 .....	92
Figura 4-41. Ejercicio 4-Opción correcta .....	93
Figura 4-42. Ejercicio 5 .....	94
Figura 4-43. Ejercicio 6 .....	94
Figura 4-44. Ejercicio 7 .....	96

# V. Índice de cuadros y tablas

Tabla 1. Ejemplo en Cotas Absolutas e Incrementales.....	23
Tabla 2. Códigos ISO .....	33
Tabla 3. Parámetros I,J .....	83
Tabla 4. Coordenadas Ejemplo Práctico 3 .....	86
Tabla 5. Ejercicio 6-Coordenadas .....	96
Tabla 6. Ejercicio 7-Parámetros I y J .....	96

## VI. Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos

<b>Término</b>	<b>Significado</b>
CNC	Computer Numerical Control (Control Numérico Computarizado)
CAD	Computer-Aided Design (Diseño asistido por computadora)
CAM	Computer-Aided Manufacturing (Fabricación asistida por computadora)
ISO	International Organization for Standardization (Organización Internacional de Estandarización)
TIC	Tecnologías de la Información y la Comunicación

# 1. Introducción

El objetivo de este Trabajo de Fin de Grado, será el de afianzar los conocimientos sobre programación en control numérico conseguidos durante el Grado, y plasmarlos en el desarrollo de una herramienta interactiva que pueda ayudar a futuros estudiantes de asignaturas relacionadas con la Tecnología Mecánica. Para ello se utilizará la herramienta Adobe Animate. La finalidad última es que los estudiantes puedan aprender los conceptos de Control Numérico de una manera más didáctica y visual, así como aplicar lo aprendido en sencillos ejercicios de autoevaluación.

Esta herramienta educativa, pretende también, dar respuesta a los constantes cambios que se producen en la docencia tradicional, donde el uso de herramientas digitales se está imponiendo con cada vez más fuerza en el ámbito educativo.

La irrupción de las nuevas tecnologías ha provocado una rápida transformación en el perfil de los docentes y los estudiantes, así como la relación tradicional que une estos dos grupos. Así pues, desde principios de este siglo, el papel que el profesor o maestro desempeñaba en el aula, era el de una figura autoritaria que imponía estrictas normas que el alumno debía obedecer y cumplir en todo momento. El profesor tradicional [1], era el encargado de impartir clases magistrales en los que la lección o temática es impartida mientras que los alumnos escuchaban atentamente desde sus pupitres. Este método de enseñanza tradicional, relegaba al alumno a un segundo plano, en el que su deber era escuchar las enseñanzas del maestro y tomar nota de todo lo contado durante la clase. La escasa participación del alumno durante estas clases magistrales, podía provocar un freno o limitar el desarrollo de sus capacidades y talentos, ya que lo enseñado o leído en un libro o una pizarra no es garantía de que el alumno haya comprendido a fondo la materia y el objetivo de la clase.

Como se está demostrando, cada alumno aprende a ritmos y de maneras muy diferentes. En este sentido, la enseñanza tradicional podría crear discriminaciones diversas dentro de una misma clase.

Es por esto, que se están implantando nuevos sistemas educativos más centrados en el alumno. Estas nuevas metodologías, por supuesto, han cambiado el rol tradicional profesor-alumno. Ahora, el alumno es una parte activa dentro del proceso de aprendizaje

donde se han añadido activos como los seminarios, trabajos en grupo, resolución conjunta de problemas, prácticas en laboratorio, visitas de campo, etc.

El resultado de estas innovaciones educativas, es que el alumno se siente mucho más integrado y protagonista en su aprendizaje, participando activamente dentro del aula, mejorando su comprensión y creatividad a la vez que sus habilidades o pensamiento crítico.

Es en este apartado donde posiblemente se estén apreciando los mayores cambios en el mundo docente, los libros de texto tradicionales están dando paso a plataformas digitales donde el alumno tiene infinitas posibilidades de interactuar con el profesor y con la lección que se está impartiendo.

La figura del profesor como único facilitador del conocimiento y el alumno como elemento pasivo, tiende a desaparecer. El alumno obtiene una mayor autonomía e independencia adquiriendo un mayor control y más capacidad de decisión en el proceso de aprendizaje. El profesor se convierte, por tanto, en un punto de apoyo y de guía para el alumno debiendo monitorizar de manera constante el proceso de aprendizaje [2].

Se puede llegar a la conclusión, que la sociedad está demandando cada vez más sistemas educativos más flexibles y accesibles, menos costosos y que se puedan implementar a lo largo de su vida profesional. Es en este ámbito donde la docencia debe plantearse un cambio de estrategia en la forma de distribuir los materiales de aprendizaje, en los sistemas de comunicación y cambios en las estrategias didácticas de los profesores.

Actualmente, el personal académico, está debatiendo cómo desarrollar estos nuevos cambios y cómo implantarlos en el mundo docente actual de una forma eficiente. Las universidades en particular son agentes muy sensibles a estos cambios. La comunidad docente universitaria, se encuentra con constantes retos de mantener actualizada y al día la oferta educativa, ofreciendo nuevas asignaturas que son demandadas, actualizando o cambiando los softwares que pueden ser utilizados en prácticas de laboratorio o de campo.

Las universidades deben innovar de manera permanente. La combinación de una docencia que utiliza las nuevas tecnologías educativas junto al contacto cercano y conocimiento de los estudiantes es la forma de plantear un proyecto educativo de futuro.

Es necesaria una nueva forma de enseñar, utilizando las nuevas metodologías de enseñanza que incluyen tecnología y aprendizaje online, junto al apoyo y acompañamiento académico a los estudiantes que poseen brechas de conocimiento al ingreso a la universidad [3].

Igualmente, para fomentar la participación de los estudiantes en la investigación, es necesario una mirada innovadora y de futuro para poder adaptarse a un mundo en permanente cambio. El avance en esta labor incluye incorporar cursos de exploración en investigación, invertir en nuevos proyectos de investigación, potenciar la formación doctoral, crear nuevas redes internacionales, y la participación de estudiantes en centros de innovación con fuerte vinculación al sector productivo, entre otros [3].

En la actualidad, es necesario fomentar nuevos estímulos de formación y soporte en innovación docente, para transformar las clases actuales en un espacio de aprendizaje activo, con nuevas herramientas de gestión e integración de tecnología de mayor calidad, siempre desde la consideración de la diversidad de los estudiantes universitarios. Por otra parte, incorporar experiencia en investigación en la formación universitaria, supone impulsar la curiosidad de los alumnos junto a la pasión por la exploración. Esto implica la búsqueda permanente de nuevas preguntas, que es la base de una formación universitaria innovadora y de futuro [3].

## 1.1. ¿Qué es el Control Numérico Computarizado (CNC)?

El Control Numérico o Control Numérico Computarizado (CNC) es un sistema de automatización de máquina herramienta que utiliza código normalizado. El desarrollo de esta tecnología empezó en los años 40 cuando se diseñó la primera computadora electrónica. Con el sucesivo avance de la tecnología, mediante el uso de relés, transistores y circuitos integrados, los controladores o computadoras se hicieron más baratos y accesibles, y por lo tanto su integración en el mundo de máquinas y herramientas fue mucho más sencilla [4]. En 1952, el Instituto Tecnológico de Massachusetts (MIT), logró desarrollar la primera máquina controlada por código CNC. Esto fue posible gracias al mando de un sistema operativo informático que dirigía cada movimiento y acción de la máquina [5].

Hasta ese momento, las máquinas herramientas utilizaban la tecnología de válvulas de vacío combinada con tarjetas perforadas, que se utilizaban para la carga de datos. Ya en los años 60, las válvulas de vacío eran reemplazadas por transistores, hasta que la introducción de las computadoras en la década del 70 sentó las bases definitivas de lo que hoy conocemos como tecnología del control numérico computarizado (CNC) [6].

La constante evolución y desarrollo de las máquinas herramienta ha logrado posicionar el sector de la industria del metal como uno de los más modernos y eficientes. Actualmente se le considera como uno de los de mayor relevancia, al ofrecer bajos costes de producción y altos niveles de precisión. En contraparte, realizar los procesos de forma manual conlleva mayor estrés en los trabajadores y altos costes de producción, a parte de una significativa pérdida de precisión y por lo tanto de calidad.

Otra de las ventajas estratégicas que ofrece el mecanizado CNC, es la de lograr una mayor complejidad en las piezas y diseños que se pueden fabricar con respecto a las máquinas tradicionales. Esto ha producido un notable desarrollo y salto tecnológico en diversas industrias como la aeronáutica, naval o automoción.

Así mismo, se ha reducido drásticamente el desperdicio de material al aumentar notablemente la precisión; el riesgo de sufrir accidentes laborales también ha disminuido, ya que las máquinas cuentan con mecanismos de seguridad como puertas que no permiten ser abiertas durante el mecanizado o diversos mecanismos de seguridad integrados en el propio software de la computadora, que indican errores de código y posibles fallos que se pueden producir durante el mecanizado.

En cuanto a las características y tipos de máquinas CNC, existen dos tipos: horizontales y verticales. La diferencia se encuentra en la posición del motor del husillo. Existen diferentes configuraciones y modelos que pueden trabajar con 4, 5 o 6 ejes. Algunas de las máquinas herramientas controladas por Código Numérico más comunes son: fresadora, torno, corte laser, electroerosión, etc.

El controlador CNC trabaja conjuntamente con una serie de motores y diversos componentes para trasladar los ejes de la máquina, y así poder ejecutar el código programado.

Los elementos principales que componen una máquina CNC son los siguientes (Figura 1-1):

- Dispositivo de entrada
- Unidad de control
- Máquina herramienta
- Sistema de accionamiento
- Monitor

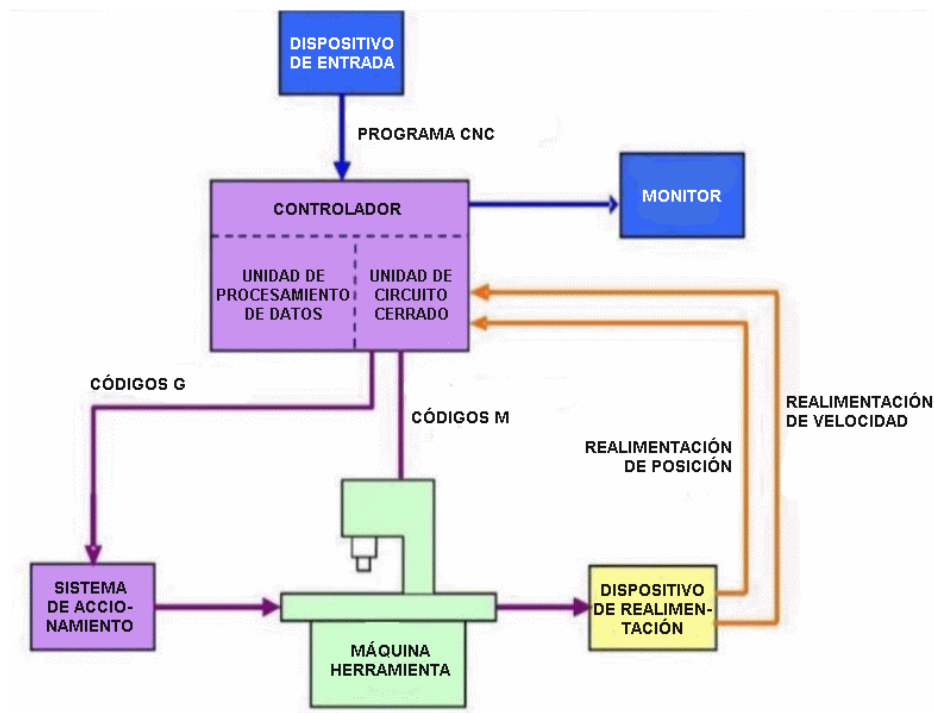


Figura 1-1. Diagrama de bloques de una máquina CNC [6]

De manera general, la unidad de control recibe instrucciones de la computadora en forma de códigos G y M, y a través del propio software, la máquina convierte estos códigos en señales eléctricas, que harán funcionar las partes móviles de la máquina herramienta permitiéndonos así realizar el trabajo que hemos programado.

Los códigos G y M, son un lenguaje de bajo nivel. Está estandarizado internacionalmente a través de la norma ISO 6983. Un programa típico que una máquina CNC podría interpretar, consiste en una serie de bloques encabezadas por la letra N. Esta letra representa un bloque de código, cuya estructura por lo general se dispone de la siguiente manera (Figura 1-2):



**N T G X Y Z F S M**

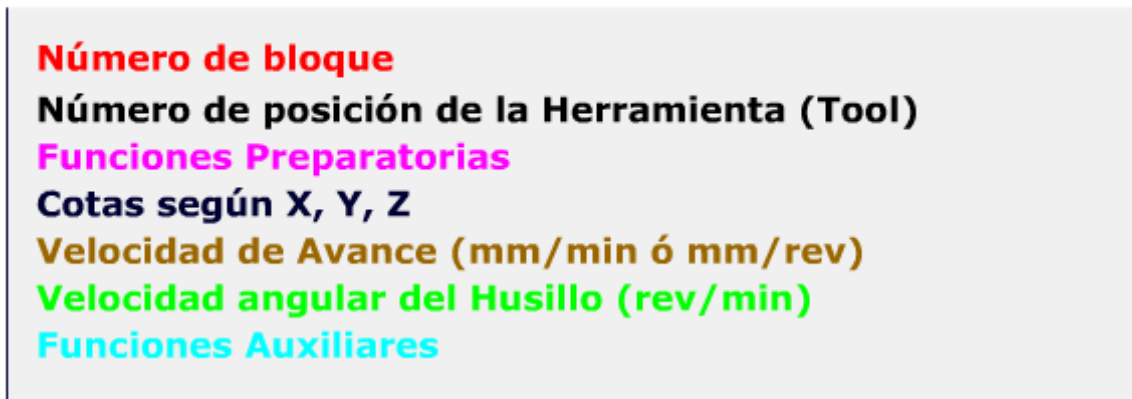


Figura 1-2. Elaborado con Adobe Animate

La programación se puede realizar de manera manual o con la ayuda de software específico CAD/CAM, que permiten pasar de la etapa de diseño del producto a la etapa de fabricación, sin necesidad de realizar mediciones manualmente o emplear tiempo en describir la trayectoria de la herramienta para fabricar la pieza en cuestión. Además, se deben de introducir todas las características de nuestra máquina, a la vez que las herramientas con las que queremos trabajar, esto incluye: diámetros y dimensiones de las herramientas, material sobre el que vamos a trabajar, velocidades de corte o revolución deseadas, etc.

A la hora de fabricar piezas de complejas dimensiones y formas, los ingenieros y operarios se valen de programas CAM para extraer el código CNC, ya que de la manera tradicional sería muy complejo y costoso desarrollar a mano todo el código necesario para fabricar correctamente nuestra pieza.

Otra de las ventajas que presenta el CAD/CAM, es la posibilidad de realizar una simulación previa al mecanizado. Esto permite comprobar si el código que ha generado el software es el adecuado y cumple con todas las órdenes necesarias, y también permite hacer una comprobación de seguridad con el fin de detectar: posibles colisiones de la herramienta con soportes, paredes, u otras partes de las máquinas.

## 1.2 Objetivo

El objetivo de este trabajo es el de, proponer una respuesta a una necesidad cada vez más urgente en el ámbito educativo universitario, involucrar al alumno en el proceso educativo mediante el uso de las Tecnologías de la Información y Comunicación (TICs). En este caso, mediante el diseño de una herramienta educativa cuyo objetivo es describir de manera clara, concisa y didáctica, los principios de programación de un Control Numérico (CNC), mediante códigos ISO. Esta herramienta formará parte de las actividades que se propondrán en las clases de la asignatura de Tecnología Mecánica, del grado de Ingeniería Mecánica que se imparte en la Escuela de Ingenierías Industrial, Informática y Aeroespacial de la Universidad de León. El alumno matriculado en esta asignatura podrá tener acceso a la herramienta en cualquier momento del semestre y desde cualquier lugar con acceso a Internet.

El primer paso, es identificar qué carencias presenta el método educativo tradicional. Impartir lecciones magistrales en el aula, donde el maestro enseña la lección y el alumno únicamente se limita a escuchar y tomar notas, es un método que se está quedando anticuado, y está siendo revisado.

Las plataformas educativas digitales presentan una oportunidad excelente de avanzar hacia métodos educativos más inclusivos y didácticos, donde el alumno no necesariamente debe estar en el aula para poder recibir la lección.

Internet se está mostrando como una ayuda indispensable, tanto para el alumnado como para el profesorado. Impartir lecciones, compartir apuntes o realizar exámenes a través de plataformas digitales se ha convertido una práctica común y muy extendida.

Una vez que hemos definido cuál es el objetivo principal de nuestro proyecto, nos debemos centrar en evaluar qué exigencias puede tener el alumnado a cerca de esta herramienta, es decir, qué contenidos se van a impartir. La enseñanza del Control Numérico, puede ser una tarea complicada si no se tienen al alcance herramientas gráficas como pueden ser fotos, vídeos, ilustraciones o diagramas. Es por ello, que una gran parte de los esfuerzos de este trabajo, se han centrado en presentar de la manera más gráfica posible todos los conocimientos que se pretenden explicar en este campo. Se han elaborado diagramas interactivos con los que el alumno puede establecer una primera toma de contacto con la programación de Control Numérico.

Por último, debemos centrarnos en desarrollar un temario coherente y adecuado para nuestra herramienta educativa, por ello, se han consultado las lecciones de la asignatura Tecnología Mecánica, de donde se ha podido extraer la información sobre qué aprendizajes se deben incluir, a la vez que, el nivel de dificultad debe tener nuestra plataforma.

Para el desarrollo de esta herramienta didáctica se han utilizado diferentes softwares:

- Adobe Animate: es el programa que hemos utilizado como base para desarrollar toda nuestra herramienta educativa. Debido al desconocimiento inicial sobre el funcionamiento de este programa, uno de los objetivos principales de este trabajo, ha sido aprender a trabajar con soltura el software Adobe Animate, a la vez que expandir mi conocimiento sobre el lenguaje de programación JavaScript, que es el lenguaje de programación que utiliza dicho software.  
Para dar mis primeros pasos con Adobe Animate, he seguido los tutoriales que ofrece el propio programa, a la vez que video tutoriales gratuitos que se pueden encontrar en plataformas como YouTube.
- Draw.io: es una herramienta gratuita online, con recursos para elaborar los diagramas que utilizaremos en nuestra herramienta. Algunos de los diagramas elaborados con esta herramienta son Figura 2-2 O Figura 2-3.
- SolidWorks: es un potente software CAD para modelado y diseño 2D y 3D. Con este software, se han diseñado las piezas que se van a utilizar como modelo en los ejercicios de Ejemplos prácticos. Adicionalmente, SolidWorks ofrece la posibilidad de trabajar con el software CAM, el cual ha sido utilizado para ejecutar los diferentes mecanizados de nuestras piezas. Posteriormente, utilizaremos estas imágenes para mostrar el mecanizado de la pieza en nuestra herramienta educativa, podemos observar esto en la Figura 4-32.
- ScreenToGif: es una herramienta gratuita que permite tomar imágenes de vídeos o movimientos que ocurren en la pantalla del ordenador, y convertirlas en fotogramas. En nuestro caso, se ha utilizado para capturar el movimiento que describe la herramienta en el programa SolidWorks, para posteriormente poder utilizar los fotogramas generados. A continuación, mediante los fotogramas generados, en Adobe Animate podremos crear una interpolación de movimiento, con el fin de poder reproducir fielmente el movimiento que realiza la herramienta.

## 2. Funciones del Control Numérico

Como ya se ha explicado anteriormente, el Control Numérico es un lenguaje estandarizado según ISO 6983, y un programa CNC contiene todas las instrucciones necesarias para el mecanizado de la pieza, así como una unidad de control o controlador que cumple la función clave de interpretar, y accionar los diversos mecanismos que componen la máquina para ejecutar correctamente todo el código e instrucciones que se han indicado. Adicionalmente, la unidad de control tiene otros propósitos de vital importancia: modificar o avisar al operario en caso de que se detecten errores de programación en el código, verificar mediante una simulación que la exactitud del programa CNC, o servir de apoyo al operario para indicar entradas importantes como pueden ser los valores de longitud de las herramientas.

La estructura típica de un bloque, como se señala en la Figura 1-1, es la siguiente:

Cabecera de bloque + bloque del programa + final del bloque

El código G describe las funciones preparatorias, es decir, los movimientos de la herramienta o la geometría y las condiciones de trabajo (por ejemplo, movimientos rápidos, interpolaciones lineales o circulares). Siempre que se programa una función preparatoria G, la debemos programar al comienzo del cuerpo del bloque, además deben estar acompañadas de dos cifras, siguiendo el modelo Gxx.

Mientras, el código M describe las funciones auxiliares necesarias para el mecanizado de la pieza, pero que no corresponden a los movimientos de la herramienta (por ejemplo, arranque y detención del husillo, cambio de herramienta, refrigerante, detención del programa). De igual manera que las funciones preparatorias (G), la función auxiliar M se debe programar seguida de dos cifras: Mxx.

A su vez, existen funciones complementarias, identificadas con otras letras y definidas por el programador para cada función específica. Por ejemplo, F define la velocidad de avance, S la velocidad del husillo, T la herramienta seleccionada, D es el número de corrector de la herramienta, X, Y y Z representan las cotas (dichas cotas se pueden

programar de forma absoluta o relativa, es decir, respecto al cero pieza o con respecto a la última cota), I, J y K la localización del centro de un arco, etc [7].

Toda máquina CNC se caracteriza por tener dos o más ejes; es decir, dos o más direcciones programables. Cada eje representa posibles direcciones y movimientos que la máquina puede ejecutar, de manera que cuantos más ejes tenga una máquina, más compleja será. Un eje no sólo se puede referir a ejecutar movimientos lineales, también se pueden efectuar movimientos giratorios, como los que por ejemplo utiliza el husillo de una máquina fresadora. Comúnmente, los ejes que representan movimientos lineales son X, Y y Z, mientras que los ejes que representan movimientos circulares se les asignan las letras A, B y C (Figura 2-1).

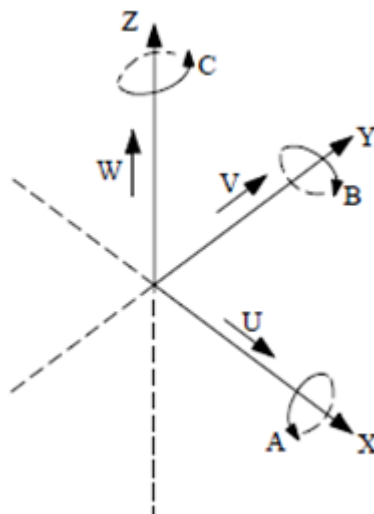


Figura 2-1. Manual FAGOR 8050M [7].

## 2.1 Funciones preparatorias

La finalidad principal de este capítulo, es describir concisamente cómo trabaja cada función y las características principales que tiene. A la hora de programar, es importante comprender que hay tres tipos diferentes de funciones:

- Las funciones modales, que son las funciones que seguirán activas hasta que anulen o se programe otra función G. Esta característica nos puede ayudar a simplificar mucho nuestro lenguaje, a la vez que podemos ahorrar tiempo no haciendo falta escribir el comando en todas las líneas.

- Las funciones secuenciales, que solamente estarán activas mientras se está ejecutando el bloque en el que están. Esto es debido a que las funciones de interpolación circular son muy específicas, y se entiende que en vacío nunca vamos a realizar un movimiento circular para desplazar la herramienta desde un punto A hasta uno B, ya que lo más sencillo es desplazarse en línea recta. Es por eso que las interpolaciones circulares son siempre utilizadas en régimen de trabajo y no en desplazamiento en vacío.
- Las funciones preferenciales, que como bien lo indica el nombre, sirven para ajustar la programación a nuestras propias preferencias personales. Así, cada operario podrá elegir la forma de programar que más sencilla resulte, o elegir uno u otro tipo de programación en función de las exigencias del cliente, la máquina o el boceto sobre el que se está trabajando.

Por ejemplo, el movimiento de una máquina controlada por Control Numérico, se puede programar o controlar de dos formas, mediante funciones preparatorias de tipo modal:

- Valores Absolutos (G90): la función G90 sirve para indicarle al control que queremos definir las cotas de forma absoluta, es decir, la distancia que separa ese punto del cero pieza, independientemente de la posición donde esté la herramienta. De esta manera, haremos el programa pensando en el punto al que queremos desplazar la herramienta con respecto al cero pieza, pudiendo usar valores positivos o negativos [12]. Esta función es modal, lo que significa que está activa por defecto; si no se programa otra función para reemplazarla, la máquina asumirá todas las cotas en valores absolutos.
- Valores Incrementales (G91): indicamos al control que las cotas están definidas de manera incremental, esto quiere decir que se desplace de un punto a otro tomando como referencia el punto en el que estamos actualmente. Así pues, en este caso debemos calcular cuántas unidades queremos desplazarnos en los ejes X, Y, Z para alcanzar el punto deseado, independientemente de donde esté situado el cero pieza. Por eso, al programar en cotas incrementales debemos indicar las unidades, además también el sentido del desplazamiento positivo o negativo según proceda.

Sirva la Figura 2-2 como ejemplo para entender las funciones G90/G91:

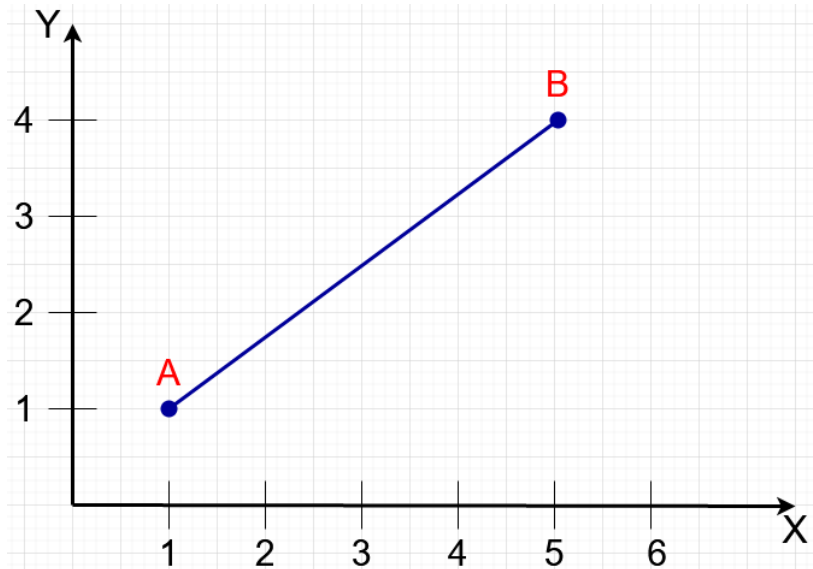


Figura 2-2. Desplazamiento lineal "A" a "B"

De esta manera, si queremos efectuar un movimiento desde el punto "A" hasta el "B", los códigos quedarían de la siguiente manera (Tabla 1):

Tabla 1. Ejemplo en Cotas Absolutas e Incrementales

Cotas Absolutas G90	Cotas Incrementales G91
G90 G01 X5 Y4	G91 G01 X4 Y3

En los apartados siguientes se describirán algunas de las funciones preparatorias más utilizadas y que se verán reflejadas en la herramienta desarrollada en este TFG.

### 2.1.1 Decalajes de origen

Para construir un programa es necesario poder definir los puntos adecuados y sus cotas. Para poder indicar las cotas adecuadas de cada punto, tenemos que referirlas o medirlas desde los ejes (X, Y, Z) de coordenadas cuyo origen (X0, Y0, Z0) podremos situarlo a nuestro antojo o necesidad [7].

En el momento en el que arrancamos cualquier máquina CNC, lo primero que debemos centrarnos es en establecer un punto concreto, un punto de referencia. Para ello, los fabricantes añaden sensores con los que calibrar nuestras herramientas y establecer los puntos de referencia y origen de pieza que necesitamos para empezar nuestro

mecanizado. Ese es el "cero máquina", es decir, es el origen de coordenadas, ("X0", "Y0", "Z0"), de la máquina, lo que significa que ya tiene un punto de referencia, que normalmente será alguna de las "esquinas" a donde pueda llegar. A partir de aquí el desplazamiento hasta cualquier cota que esté en su rango de desplazamiento es perfectamente posible.

Esta información, pese a que es relevante, a la hora de programar nuestro mecanizado y empezar el trabajo, significa muy poco, puesto que, si colocamos un tocho de material en las abrazaderas, no podemos tener nuestro sistema de coordenadas con referencia al cero máquina, puesto que tendríamos que confeccionar un programa "a medida" para cada pieza que quisiéramos mecanizar.

Para poder iniciar la sesión, hace falta una referencia para que a partir de ahí cualquier cota tenga sentido. Es por eso que el origen de máquina es inamovible y ya viene fijado por el fabricante (Figura 4-15).

Para efectuar un traslado de origen de coordenadas, disponemos de las funciones G54 al G59 que se encargan de desplazar el origen desde el cero máquina hasta el nuevo origen elegido (cero pieza), disponiendo así de seis desplazamientos distintos.

Para situar este cero pieza se utilizan las funciones "G53", "G54", "G55", "G56", "G57", "G58" y "G59", indicando seguido las cotas en los ejes X, Y, Z, siempre referidas al cero máquina (Figura 2-3). Teniendo en cuenta que "G53" se debe utilizar únicamente si queremos programar a partir del cero máquina. No es modal, y puede ser añadida a cualquier bloque que contenga funciones de control de trayectoria.

En las máquinas, se puede introducir directamente desde la consola de mandos, también desde el programa o utilizando la propia herramienta como "sensor" marcando nosotros dónde queremos situar el origen pieza. Además, se pueden programar todas las funciones de traslado de origen, (G53 a G59), que quedarán guardadas en la "tabla de decalaje", y luego durante el programa basta con tan solo nombrarlas para que carguen los valores almacenados.



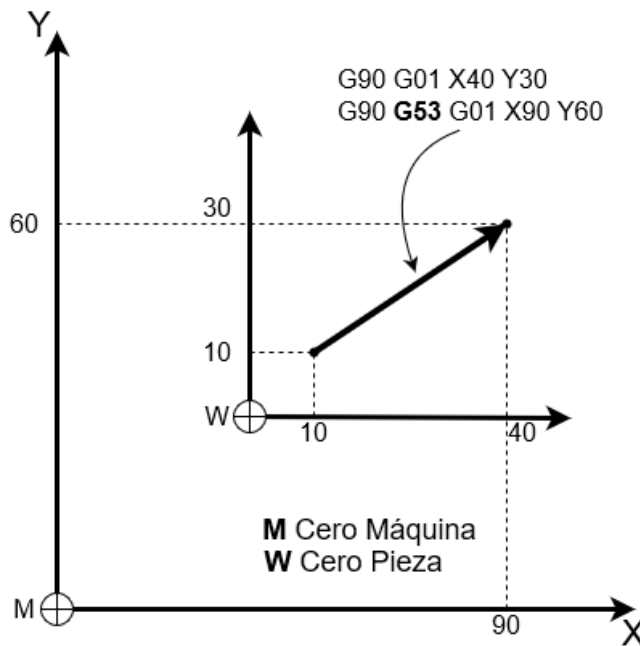


Figura 2-3. Traslado de origen de coordenadas

El resto de funciones de preselección de cotas y traslados de origen (G54 a G59), realizan una labor similar a G53, con la diferencia que las nuevas cotas introducidas son la distancia existente entre el cero máquina y el nuevo cero pieza. Por lo tanto, a la hora de programar, estaríamos utilizando como origen de coordenadas el nuevo cero pieza que hayamos seleccionado. Para facilitar el trabajo de programación, se podrán tener guardados diferentes “cero pieza”. Estos traslados de origen, o decalajes de origen, no se pueden programar fuera del entorno de la máquina al igual que G53, es decir, necesitamos calibrar los nuevos orígenes in situ en la máquina, aproximando la punta o centro de la herramienta hasta tocar físicamente la pieza en el punto exacto en el que queremos establecer un nuevo decalaje de origen.

Adicionalmente, existe la función “G92”, que permite hacer la preselección del nuevo cero pieza que queremos establecer y así modificar sobre la marcha el origen y establecerlo donde nos sea más cómodo, es decir, el control asume las cotas de los ejes programados a continuación de G92, como nuevos valores para X Y y Z. Este código no es tan utilizado como los anteriores, pero puede resultar de gran utilidad a la hora de mecanizar piezas con geometrías complejas.

## 2.1.2 Compensación del radio de la herramienta

A la hora de trabajar correctamente, debemos tener en cuenta el radio de la herramienta para poder obtener las dimensiones deseadas de la pieza. Para hacer más sencilla la programación y no tener que calcular las coordenadas sumando o restando el radio, podemos utilizar las compensaciones del radio de la herramienta, G41 (compensación a izquierdas) y G42 (compensación a derechas). Las funciones G41 y G42 son modales e incompatibles entre sí, y son anuladas mediante G40. Tanto G41 como G42 como G40 se deben activar en un bloque de interpolación lineal (G00 o G01), nunca circular. El CNC calcula automáticamente la trayectoria que debe seguir la herramienta, a partir del contorno de la pieza y del valor del radio de la herramienta almacenado en la tabla de correctores [7].

- **G41:** Compensación de radio de herramienta a izquierda. Cuando en la dirección del avance de la herramienta, ésta se coloca a la izquierda del perfil real, se debe usar la función G41 para activar la compensación a izquierdas. Esto significa que con G41 el material siempre se va a quedar a la derecha de la herramienta siguiendo el sentido de mecanizado. En la Figura 2-4, podemos ver gráficamente las diferencias entre G41 y G42, esta figura ha sido elaborada con Adobe Animate.

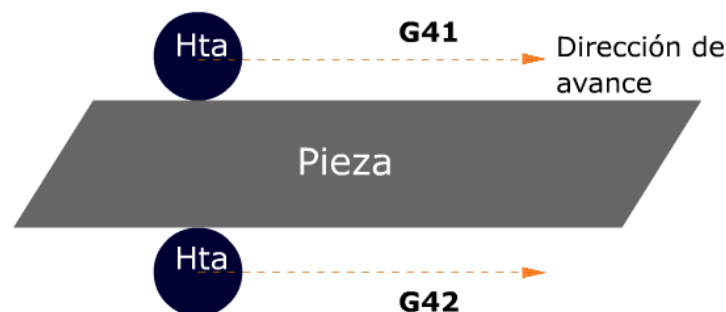


Figura 2-4. Representación gráfica de G41 y G42

- **G42:** Compensación de radio de herramienta a derechas. Cuando en la dirección del avance de la herramienta, ésta se coloca a la derecha del perfil real de la pieza a contornear, se debe usar la función G42 para activar la compensación a derechas. Es decir, contrariamente a lo que ocurre con G41, el material a mecanizar quedará siempre a la izquierda de la herramienta siguiendo el sentido de mecanizado.

- **G40:** Anulación de la compensación de radio. Como se ha comentado, esta siempre se debe programar al final de cada bloque cuando queremos apagar este ciclo de compensado o también, se puede programar al principio de cada bloque, con la única intención de apagar y anular cualquier ciclo de un bloque anterior y que no queremos que esté activo durante el bloque actual.

La Figura 2-5 muestra cómo se establece el inicio de compensación de radio en la primera trayectoria lineal, y en la esquina, empieza la compensación total del radio de la herramienta.



Figura 2-5. FAGOR 8050 [7]

## 2.2 Ciclos fijos

Además, los fabricantes, definen ciclos fijos propios para cada tecnología y que no tienen por qué estar recogidos en la norma. Estos ciclos definen operaciones completas de mecanizado, como por ejemplo, los ciclos de taladrado o de cajeado. Estos ciclos se definen mediante la función G, y no se podrán definir en un bloque que contenga G02 o G03. Una vez programado un ciclo fijo, se mantiene activo hasta que no se anule.

Se va a tomar como referencia para este TFG el CNC de Fagor, por lo que en este apartado vamos a particularizar alguno de los ciclos más utilizados con este CNC.

Existen tres cotas que siempre debemos tener en cuenta en el eje longitudinal:

- **Plano de partida:** posición de la herramienta respecto al cero máquina cuando se activa el ciclo.
- **Plano de referencia:** cota de aproximación a la pieza en coordenadas absolutas o incrementales, siempre referido al plano de partida.
- **Profundidad de mecanizado:** se refiere al plano de referencia y se puede programar en cotas absolutas o incrementales. En este caso también debemos

seleccionar el retroceso del eje longitudinal después del mecanizado, G98 (retroceso hasta el plano de partida) y G99 (retroceso al plano de referencia).

En nuestro caso, vamos a describir los ciclos G81 (Ciclo fijo de taladrado), G83 (Ciclo de taladrado profundo) y G87 (Ciclo de cajado rectangular):

- **G81:** Este es el ciclo más sencillo, realizamos un taladrado en el punto indicado hasta alcanzar la cota final. La estructura del bloque es:

G81 G98/99 X Y Z I K

Donde:

Z: Cota del plano de referencia

I: Profundidad del taladrado

K: Tiempo de espera en centésimas de segundo, tras el taladrado y antes del retroceso. Si no se programa se tomará K0.

- **G83:** Este ciclo cumple la misma función que G81, con la diferencia de que la herramienta retrocede hasta el plano de referencia tras cada paso de taladrado. Se realizan sucesivos pasos hasta conseguir la cota final. Es un ciclo que cuida más el desgaste de la herramienta, ya que el refrigerante puede penetrar mejor en el agujero. La estructura del bloque es:

G83 G98/99 X Y Z I J

Donde:

I: Paso de cada taladrado

J: Número de pasos que realiza el taladrado

- **G87:** Este ciclo permite realizar una cajetera hasta alcanzar el punto programado (Figura 2-6). Adicionalmente podemos programar un ciclo de acabado en las paredes de la cajera. La estructura del bloque es:

G87 G98/99 X Y Z I J K B C D H L

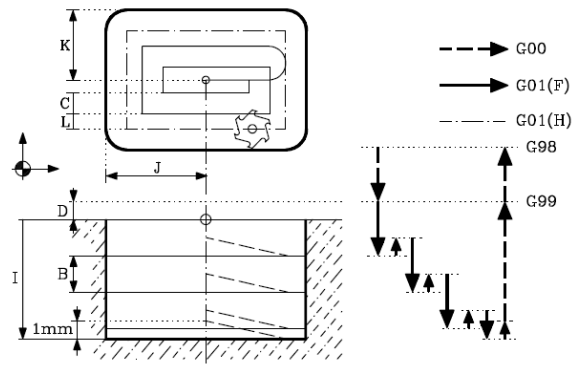


Figura 2-6. FAGOR 8050 [7]

Donde:

- J: Distancia del centro al borde de la cajera según abscisas.
- K: Distancia del centro al borde de la cajera según ordenadas.
- B: Paso de profundizado según el eje longitudinal.
- C: Paso de fresado según el plano principal.
- D: Distancia entre el plano de referencia y la superficie de la pieza.
- H: Velocidad de avance en la pasada de acabado.
- L: Valor de la pasada de acabado.

## 2.3 Funciones auxiliares

Una máquina herramienta tendría una utilidad muy limitada si sólo contara con un control de movimiento. Es aquí donde las funciones auxiliares (M) son muy útiles e imprescindibles para aprovechar al máximo las posibilidades que nos ofrecen estas máquinas herramienta.

Según los diseños más habituales de controladores CNC, en un mismo bloque (N), se pueden programar un máximo de 7 funciones auxiliares. El control ejecutará estas funciones en el orden en el que hayan sido programadas.

Entre las funciones más utilizadas y que siempre estarán incluidas en un programa, está la orientación de giro del cabezal, que puede ser a derechas, es decir, sentido horario (M03), o puede ser en sentido antihorario (M04). El refrigerante que posee la máquina, es adecuado y está pensado para aumentar la vida útil de la herramienta, se utiliza mediante las funciones M08 y M09, que sirven para encender y apagar el chorro respectivamente. Adicionalmente, el refrigerante sirve para mejorar las condiciones de trabajo de la herramienta, ya que se reduce el calentamiento del material, lo que provoca unos acabados superficiales y tolerancias mucho más precisas.

- M00: esta función detiene la ejecución de programa de programa. Cuando se detiene el programa, permanece invariable todas las funciones modales existentes. Para reanudar toda la ejecución del programa, debemos pulsar el botón de marcha de la máquina.
- M02: cuando el control lee esta función, da por finalizada la ejecución del programa y realiza una función de reset del control, es decir, coloca al control en las condiciones iniciales de encendido. Al mismo tiempo, se realiza una parada del giro de cabezal.
- M03/M04: la función M03, significa que el cabezal comenzará a girar en sentido horario, según la velocidad S programada. La opción contraria es la función M04, que realiza un giro en sentido anti horario.
- M05: esta función detiene el giro del cabezal, y como se indica en la lección, esta función es personalizable, se recomienda ejecutarla al final del bloque en el que

está programada. Podremos volver a girar el cabezal con las funciones M03 y M04 ya explicadas.

- M06: esta función es adecuado utilizarla cuando se dispone de un almacén de herramientas, el M06 ejecuta la subrutina de cambio de herramienta; esta función auxiliar está programada por el fabricante y entre otras cosas supone la parada de programa, paro del giro del cabezal (M05), posicionamiento de los ejes en la posición de cambio de herramienta, orientación del cabezal y accionamiento de los mecanismos de cambio neumáticos.
- M08/M09: con la función M08 podremos activar el refrigerante, esto es importante porque el refrigerante es necesario enfriar la herramienta y la pieza del calor producido por el corte y ayudar a la evacuación de viruta. Por el contrario, con la función M09 apagamos el refrigerante.
- M30: el código M30 indica el fin del programa, y se programa al final de cada bloque. La funcionalidad es idéntica al código M02, con la diferencia de que el nuevo bloque activo pasa a ser el primero del programa (N10). Es recomendable utilizar el código M30 cuando queremos pasar de un ciclo de trabajo a otro, de esta manera reseteamos la memoria interna del control y nos disponemos a iniciar un bloque completamente nuevo.

Una vez que se hayan ejecutado todos los bloques programados, debemos ejecutar el final del programa mediante la función M30, que provoca que el bloque vuelva al comienzo.

## 2.4 Funciones Complementarias

Con anterioridad, se han mencionado las funciones complementarias, son imprescindibles para que el programa CNC esté completo y pueda recibir todas las órdenes con precisión. Las funciones complementarias son las que aprovechan todo el potencial que puede ofrecer una máquina herramienta moderna. Una de las funciones más importantes es la velocidad de avance, se representa con la letra F y según nuestras necesidades o preferencias se puede programar en mm/minuto o mm/revolución mediante las funciones preparatorias G94 o G95 respectivamente.

La velocidad de giro del cabezal, se representa con la letra S y sus unidades siempre serán revoluciones por minuto (rpm). Programar adecuadamente la velocidad de giro del cabezal depende del material que vamos a mecanizar, del tipo de herramienta a utilizar durante el mecanizado (diámetro de la herramienta, tipo de fresa o broca, etc).

Son también funciones complementarias del control CNC los códigos T y D. La letra T representa el número de herramienta, y la letra D representa el corrector de herramienta. La identificación de la herramienta se realiza mediante dos cifras y puede tener un valor comprendido entre 1 y 99. Este valor se emplea para identificar la herramienta en el programa y en el almacén de herramientas [13]. Cuando el control ejecuta esta función, el CNC asume como herramienta activa la T que hemos programado; adicionalmente es necesario programar M06 para que se ejecute el cambio de herramienta.

Al igual que ocurre con la función T, el corrector de la herramienta (D) también se identifica mediante 2 cifras comprendidas entre 1 y 99. La tarea del corrector es mantener un informe sobre diferentes parámetros de la herramienta, como son el radio (R), longitud (L), desgaste del radio (I) y desgaste en la longitud (K). Los correctores I y K son valores que corrigen las distancias en X, Y y Z de los correctores R y L respectivamente. Se suelen usar para compensar los desgastes que sufre la herramienta durante el mecanizado.

Como resumen, los códigos más utilizados son los que se muestran en la Tabla 2:



Tabla 2. Códigos ISO

<b>Códigos G</b>	<b>Códigos M</b>
<p>G00: movimiento rápido                      G01: interpolación lineal (régimen de trabajo)                      G02: interpolación circular a derechas (sentido horario)                      G03: interpolación circular a izquierdas (sentido antihorario)                      G17: selección de plano principal X-Y                      G18: selección de plano principal Z-X                      G19: selección de plano principal Y-Z                      G40: cancelar compensación                      G41: compensación a izquierdas                      G42: compensación a derechas                      G53: programación respecto al cero máquina                      G70: programación en pulgadas                      G71: programación en milímetros                      G80: anulación de ciclo fijo                      G81: ciclo fijo de taladrado                      G82: ciclo fijo de taladrado con temporización                      G83: ciclo fijo de taladrado profundo con paso constante                      G84: ciclo fijo de roscado                      G90: programación absoluta                      G91: programación incremental                      G94: avance en mm/minuto                      G95: avance en mm/revolución</p>	<p>M00: parada de programa                      M01: parada opcional                      M02: reinicio de programa                      M03: giro de cabezal a derechas (horario)                      M04: giro de cabezal a izquierdas (antihorario)                      M05: parar el cabezal                      M08: encender refrigerante                      M09: apagar refrigerante                      M10: abrir mordazas                      M11: cerrar mordazas                      M30: fin de programa</p>

### 3 Software Adobe Animate

El software elegido para la elaboración de la herramienta educativa es Adobe Animate. Esta aplicación, junto con Adobe Photoshop o Adobe Illustrator es de las más populares que desarrolla la empresa Adobe Systems. Permite crear, manipular y modificar gráficos vectoriales mediante el uso de ActionScript, que es el lenguaje propio que utiliza Flash Player [8], pero este lenguaje, actualmente está en desuso.

El programa adquirió una gran popularidad para la elaboración de animaciones Flash, ya que facilitaba la creación de animaciones para plataformas como YouTube.

Este programa, es el creado por Adobe para sustituir al ya difunto Adobe Flash, cuya plataforma Flash Player ha dejado de ser utilizada a nivel mundial. Para nuestra herramienta, utilizaremos el lenguaje JAVA SCRIPT, que es el que actualmente utilizan las plataformas HTML5.

El contenido creado a través de Animate, se puede publicar y tener la apariencia de una página web y ser visto a través de un navegador, o puede ser reproducido independientemente por un reproductor Flash. También es posible crear contenido para ser publicado en móviles o tabletas.

En diciembre de 2015, Adobe anuncia que el cambiará el nombre del programa al actual Adobe Animate, antes conocido como Macromedia Flash. La finalidad de este cambio fue desvincular el programa de Flash Player para dar paso a las nuevas plataformas HTML5, que estaban creciendo en popularidad.

Se ha elegido esta aplicación porque es una de las más populares que ofrece Adobe, y actualmente ofrece nuevas opciones que mejoran la interacción y experiencia del usuario, facilitando la programación con fragmentos de código ya escritos.

La aplicación Adobe Animate trabaja con fotogramas. Estos fotogramas se colocan sobre una línea de tiempo que es la encargada de gobernar el funcionamiento del programa. A su vez, el programa también trabaja con capas, siendo Adobe famoso por implantar este sistema en otros programas tan populares como Photoshop.

## 3.1 Configuración de partida

Para empezar a trabajar con Animate, lo primero que debemos hacer es seleccionar el tipo de documento con el que vamos a trabajar. Adobe nos ofrece diferentes opciones en función de a qué estará enfocado nuestro proyecto. Como podemos ver en la Figura 3-1, podremos trabajar con escenarios específicos y enfocados a la creación de juegos, anuncios o páginas web.

En nuestro caso seleccionaremos la opción de educación, ya que nuestro propósito es crear una plataforma educativa. Para el tamaño de nuestra aplicación seleccionaremos alta de 960 x 640 píxeles. En base a pruebas empíricas, se ha llegado a la conclusión que este es el tamaño que más se adecua a nuestra aplicación, a la vez es el que mejor se adapta a una pantalla de un ordenador portátil y de sobremesa.

Como ya se ha comentado con anterioridad, las aplicaciones con la tecnología Flash Player no están en uso, por lo que trabajaremos con HTML5 Canvas, cuyo lenguaje de programación de Java.

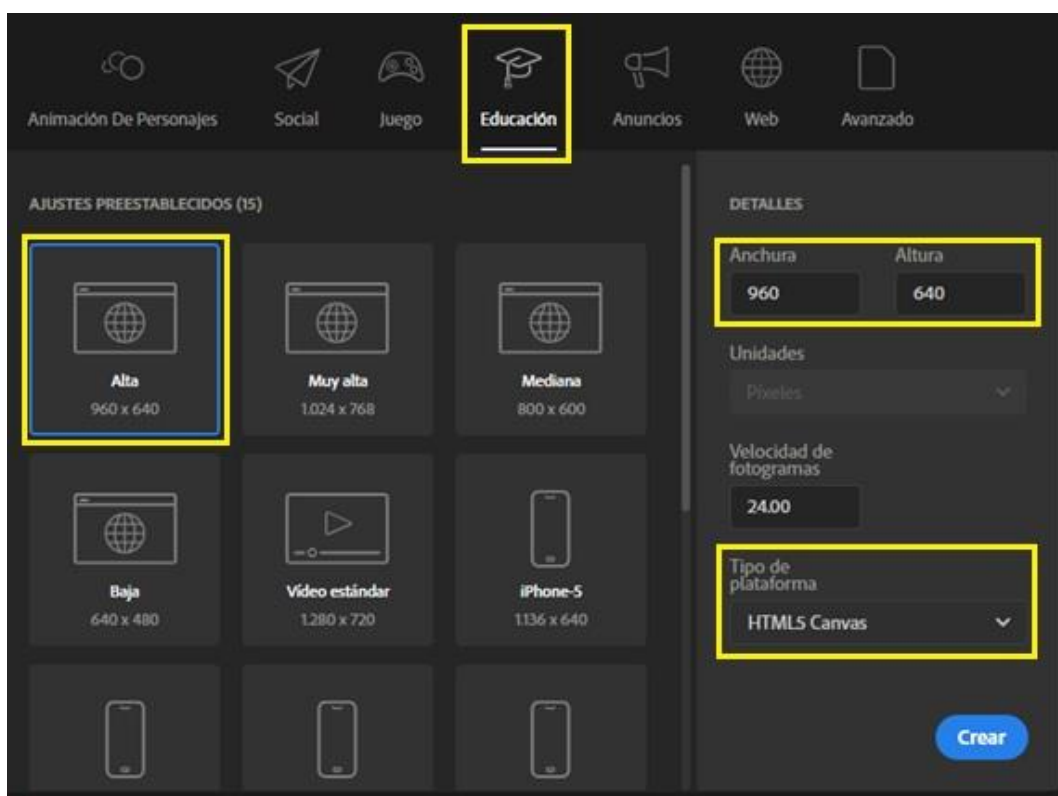


Figura 3-1. Configuración inicial

## 3.2 Espacio de trabajo

Inmediatamente después de pulsar el botón “Crear” en nuestra configuración inicial, nos aparecerá el escenario de trabajo o interfaz de usuario, que podemos ver en la Figura 3-2. El escenario principal está rodeado de diferentes paneles, son los que nos proporcionan todas las herramientas que necesitaremos para elaborar nuestra plataforma educativa de Control Numérico.

Estos paneles son intercambiables, y los podemos ajustar y colocar a nuestro antojo según nos sea más cómodo para trabajar. Desde el botón “Ventana” en la parte central superior, podremos elegir qué herramientas mostrar u ocultar, en función de nuestras necesidades.

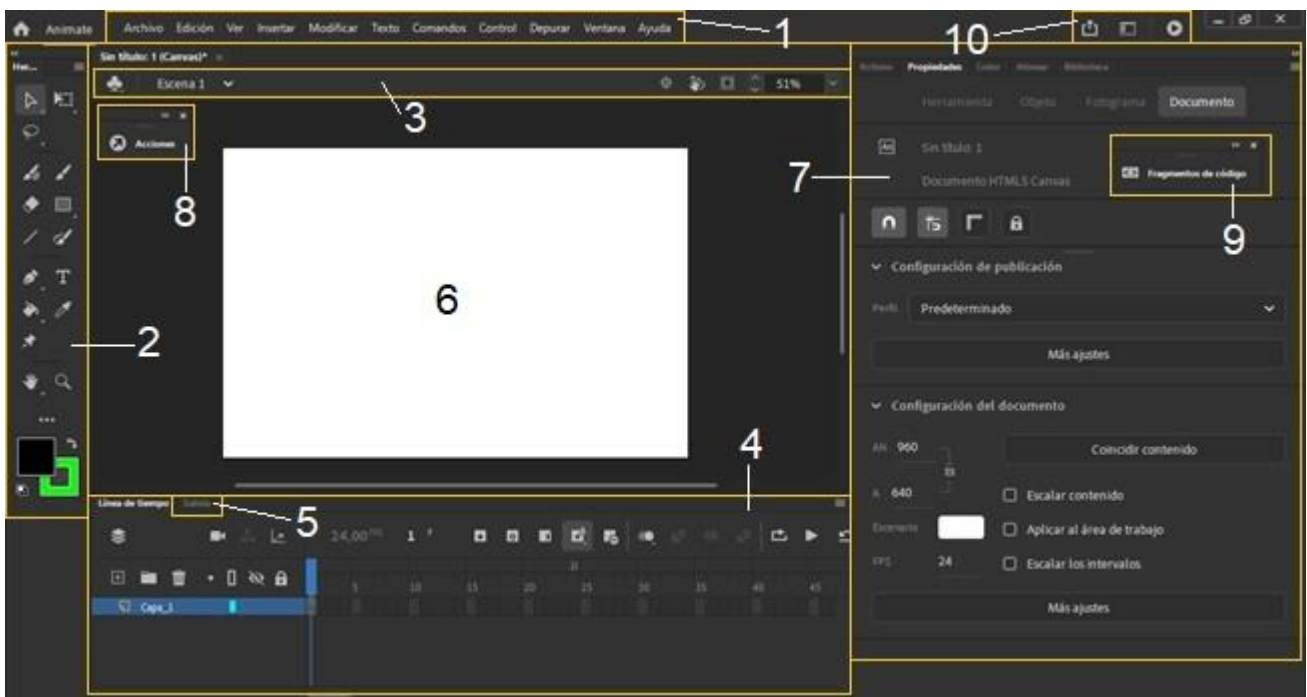


Figura 3-2. Interfaz de usuario de Adobe Animate

- 1- *Barra de tareas*: es la barra de menú en la que aparecen todas las utilidades que nos puede ofrecer el programa y que por lo tanto gobierna todo el programa.

- 2- *Barra de herramientas*: es la que contiene las herramientas más básicas y las que más frecuentemente vamos a utilizar. Podemos destacar la herramienta de selección y modificación, la herramienta pincel o la herramienta texto.
- 3- *Barra de edición*: nos indica en todo momento sobre qué objeto estamos trabajando, a la vez que nos ofrece otras posibilidades como centrar el escenario o el zoom.
- 4- *Línea de tiempo*: es la que gobierna el funcionamiento de todo nuestro proyecto. Sobre ella podemos crear capas y fotogramas, a la vez que interpolaciones de movimiento. Nos muestra en todo momento qué fotograma y capa tenemos seleccionada y sobre la que estamos trabajando.
- 5- *Salida*: durante la ejecución de pruebas, nos indica si tenemos algún error de código que puede interferir en el resultado final.
- 6- *Escenario de trabajo principal*: es sobre el que vamos a trabajar y crear nuestra herramienta educativa.
- 7- *Panel de propiedades, alinear, color y biblioteca*: este panel nos presenta algunas de las opciones más importantes que vamos a necesitar durante el trabajo. Desde la pestaña propiedades, podemos elegir las diferentes opciones que nos ofrecen las diferentes herramientas (p.ej.: pincel, texto, herramienta línea, ...). En cuanto al resto de pestañas, la que más vamos a utilizar es la pestaña biblioteca, ya que es donde se nos almacenarán todos los ítems y objetos que hayamos creado a en nuestro trabajo.
- 8- *Acciones*: es donde podremos editar nuestro código de texto en JAVA, desde este panel, básicamente podremos programar todas las acciones que ejecutará nuestra plataforma.
- 9- *Fragmentos de código*: es la herramienta desarrollada por Adobe para ayudarnos a seleccionar las acciones que necesitamos para nuestra plataforma, es decir, es el código para interacciones en un documento HTML5 Canvas.

10- *Botones de compartir y publicar y probar película*: son accesos rápidos que podemos utilizar a la hora de publicar todo nuestro trabajo, o mediante el botón probar película, ejecutar temporalmente la aplicación en un navegador web para comprobar el resultado antes de publicarlo.

## 3.3 Funcionalidad de Adobe Animate

A continuación, se hará una detallada explicación de todas las herramientas y componentes que conforman la aplicación Adobe Animate y que han sido utilizados a lo largo de todo el trabajo.

### 3.3.1 Línea de tiempo

La línea de tiempo es la que organiza y controla el contenido de un documento a través del tiempo en capas y fotogramas. La característica más significativa de Adobe Animate es que divide el tiempo en fotogramas. Los componentes principales de la línea de tiempo son las capas, los fotogramas y el cabezal de reproducción (Figura 3-3).



Figura 3-3. Línea de tiempo

Las capas son varias bandas de película apiladas unas sobre otras, cada una de las cuales contiene una imagen diferente que aparece en el escenario [9]. Las primeras capas y las últimas, se diferencian en que últimas se encuentran en el fondo del escenario.

Los fotogramas son la base de la animación, dictando cada fracción de tiempo y movimiento. La longitud total de la película, será determinada por el número total de fotogramas y la velocidad a la que se reproducen. En la línea de tiempo, para organizar y controlar el contenido, se trabaja con los fotogramas. Dichos fotogramas, se colocan en

la línea de tiempo, con el propósito de que el contenido aparezca en la pantalla según el orden que se desea.

Para empezar a crear contenido, lo primero que necesitamos hacer es crear un fotograma clave (Figura 3-4). Es un fotograma en el que una nueva instancia del símbolo aparece en la línea de tiempo. En los fotogramas clave también podemos incluir código de programación para controlar distintas acciones del documento. Al usar el fotograma clave, puede establecer una posición, añadir puntos de anclaje, acciones y comentarios, y mucho más. Esto nos será especialmente útil a la hora de asignar un nombre a cada fotograma clave desde el panel de propiedades, con el propósito de facilitarnos la programación posterior en JavaScript.

Los fotogramas en gris claro que siguen a un fotograma clave contienen el mismo contenido, sin cambios. Estos fotogramas tienen una línea negra vertical y un rectángulo vacío en el último fotograma del grupo.

También se puede añadir un fotograma clave vacío en la línea de tiempo como marcador de posición de los símbolos que se pretendan añadir más adelante, o bien para dejar el fotograma vacío de forma premeditada [9].

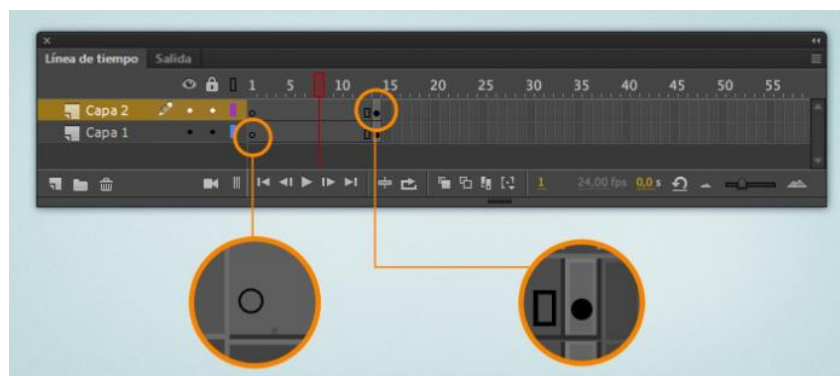


Figura 3-4. Tipos de fotograma [9]

En la línea de tiempo, tenemos otras opciones que son básicas para poder interactuar con Adobe Animate y elaborar el documento (Figura 3-5).



Figura 3-5. Añadir capa y eliminar capa

El botón “más” es el utilizado a la hora de crear una capa. Esta nueva capa se crea por defecto encima de la última capa creada. En cambio, el símbolo de la papelera representa el botón “eliminar” capa.

Cada capa creada, puede ser ocultada en caso de que queramos que no se nos muestre su contenido en el escenario, o también puede ser bloqueada para evitar que por error editemos o modifiquemos algo sin intención (Figura 3-6).



Figura 3-6. Ocultar y bloquear capas

Estos botones son de gran utilidad a la hora de trabajar con el programa. Por ejemplo, a la hora de elaborar las trayectorias que sigue la herramienta en las diapositivas de compensación del radio, necesitaremos bloquear y ocultar algunas capas para que los ítems ya creados en ellas no interfieran en nuestra labor. Bloquear y ocultar una capa no significa que se quede oculta en el producto final, es una acción que únicamente se cumplirá en el editor de trabajo mientras estamos trabajando en él, y siempre y cuando mantengamos nosotros la capa bloqueada u oculta.

Durante la elaboración de nuestra plataforma educativa, utilizaremos distintas funcionalidades de capa, como son la Capa de Máscara y la Capa de Guía (Figura 3-7). Las capas de máscara incluyen objetos utilizados como máscaras para ocultar partes seleccionadas de las capas que se encuentran debajo. Únicamente se encuentra visible la parte de la capa de máscara que no cubre la máscara.

En cambio, las capas guía, sirven para alinear los objetos de otras capas con los objetos creados en esas capas de guía. Cualquier capa puede ser una capa de guía. Las capas de guía muestran un icono de guía a la izquierda del nombre de la capa [9].

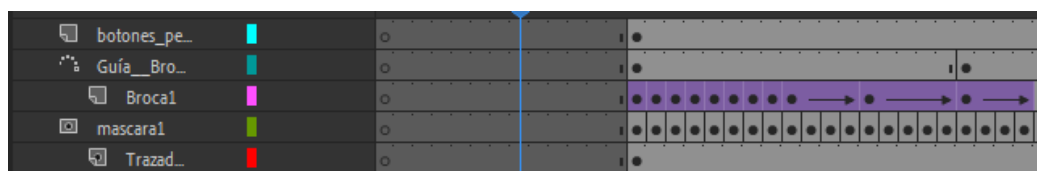


Figura 3-7. Ejemplo de Capa de Máscara y Capa de Guía



### 3.3.2 Herramientas

Las herramientas de trabajo es el apoyo que nos ofrece Adobe Animate para trabajar y editar nuestro documento. Se presentan como un apoyo indispensable que vamos a necesitar a lo largo todo el proyecto (Figura 3-8).

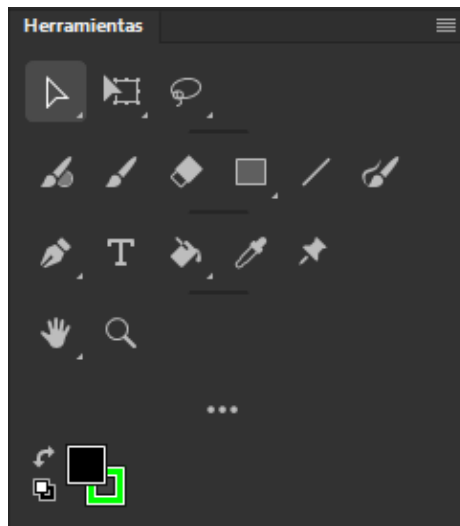


Figura 3-8. Herramientas

En la primera fila están situadas las herramientas de Selección, Transformación libre y lazo. Con la herramienta Selección, podremos seleccionar y resaltar cualquier objeto que está situado en nuestro escenario, con el fin de moverlo o seleccionarlo para otra edición. Debemos tener en cuenta que, si una capa se encuentra bloqueada desde la línea de tiempo, no podremos seleccionar o modificar los objeto que se hayan en esa capa. Con la herramienta de Transformación Libre, podremos modificar las dimensiones de los objetos seleccionados o girarlos según nuestras necesidades.



Figura 3-9. Ampliación de la Figura 3-8

Como podemos ver en la Figura 3-9, la segunda fila del panel de herramientas está compuesto por la Pincel, Borrador, Herramienta Rectángulo y la Herramienta Línea. La

Herramienta Pincel, la utilizaremos a la hora de marcar las trayectorias que describe la herramienta en las diferentes diapositivas de Compensaciones de Herramienta.

Con las herramientas Rectángulo y Línea, crearemos diferentes formas y objetos que nos serán muy útiles y necesarios para crear una plataforma educativa lo más didáctica posible. Un ejemplo del uso de estas dos herramientas, lo podemos ver en la Figura 3-10.

En la tercera línea, la herramienta más utilizada será la Herramienta Texto, que será indispensable para nosotros para crear todo el contenido de nuestra plataforma educativa.

Finalmente, al fondo de nuestro panel de Herramientas, podremos ver la paleta de colores de relleno y colores de trazo. Esta paleta la utilizaremos siempre que queramos cambiar el color del texto, rellenar las figuras creadas, o dar color a bordes de rectángulos y líneas. Esta misma opción está presente en el panel de Propiedades/ Herramienta.

### 3.3.3 Símbolos y botones

Los símbolos en Adobe Animate pueden ser un botón, un gráfico o un clip de película. A su vez, un símbolo puede ser una figura u objeto importados de otras aplicaciones. Mediante la función *Convertir en símbolo* (cuyo acceso directo en el teclado es F8) es como vamos a crear todos los botones que vamos a necesitar en nuestra aplicación. Todos los símbolos creados se van a almacenar automáticamente en la biblioteca de Animate, siendo así accesibles siempre los necesitemos, como puede ser a la hora de exportar o publicar nuestro trabajo, ya que sin esta biblioteca nuestro trabajo quedará “vacío” de contenido y no se mostrará tal y como lo vemos al ejecutarlo desde el editor de Animate.

En la Figura 3-10 podemos ver cómo es el proceso de creación de de un símbolo:

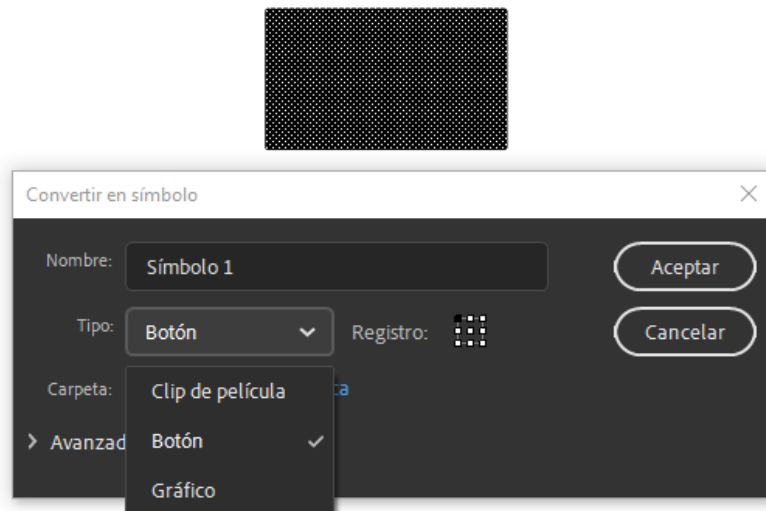


Figura 3-10. Creación de un símbolo

En este caso, el símbolo creado sería un botón, al cual le debemos dar un nombre de instancia. Este nombre de instancia, es el que vamos a utilizar para definir nuestro botón. Es importante que cada botón tenga asignado un único y exclusivo nombre de instancia, ya que a la hora de programar podríamos llamar por error otro símbolo, y posteriormente obtener errores o acciones indeseadas en la ejecución final como pueden ser saltos de diapositiva o reproducciones de película.

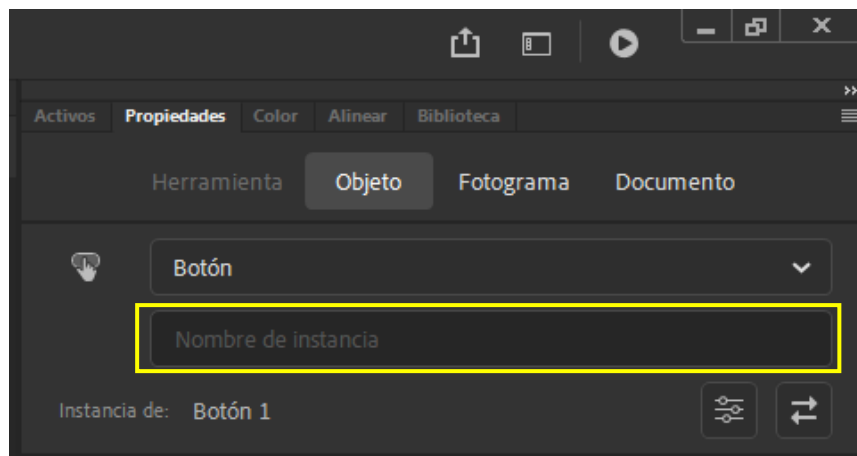


Figura 3-11. Nombre de instancia

Encontraremos el campo para introducir el Nombre de instancia en la parte superior derecha del interfaz de usuario de Animate, como podemos ver en la Figura 3-11

En nuestro programa, los botones serán utilizados para desplazarse hacia adelante o hacia atrás en las diapositivas, y para interactuar con distintos elementos como pueden ser botones para reproducir un clip de película.

Todos los botones que hayamos creado se pueden personalizar y dotar de una interactividad extra, disponemos de cuatro estados que son: Reposo, Sobre, Presionado y Zona activa.

- Reposo: es el estado natural del botón, aspecto a simple vista.
- Sobre: es posible dotar de “vida” al botón, y que cuando situamos el cursor encima de éste realice una acción.
- Presionado: la acción de puede realizar el botón cuando el usuario lo clic.
- Zona activa: es la zona de actuación del botón cuando se coloca el ratón encima, es decir, la zona en la que se activa “Sobre”. Podemos darle la forma o color que deseemos, ya que esta zona no será visible en nuestra zona de trabajo ni tampoco sobre el producto acabado. Este fotograma es muy útil a la hora de crear un botón a partir de texto y delimitar su zona de trabajo.

Al igual que todo el programa, estos cuatro estados de los botones funcionan mediante fotogramas (Figura 3-12). Para acceder a estos fotogramas y modificar los estados de un botón, debemos hacer doble clic sobre nuestro objeto (botón), y nos aparecerá el siguiente menú en la línea de tiempo:

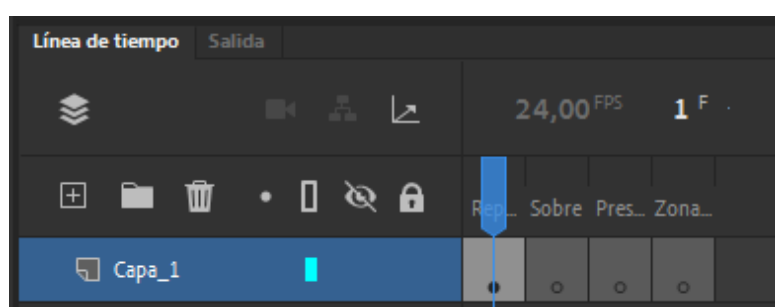


Figura 3-12. Configuración de un botón

### 3.3.4 Código JavaScript

Como ya se ha mencionado anteriormente, originalmente el programa Adobe Animate utilizaba el lenguaje de programación ActionScript3.0, que Adobe desarrolló específicamente para Animate. Su sintaxis se basaba en una mezcla de Java y C++. Este lenguaje era el utilizado para la elaboración de aplicaciones Flash y se compilaba en un archivo “.swf”. Una vez que Flash ha caído definitivamente en desuso, Animate ha decidido implantar el lenguaje JavaScript, intentado transformar el programa en una herramienta de animación profesional y publicando el producto final en la plataforma HTML5.

Para facilitarnos la programación, Animate dispone de Fragmentos de código, con el fin de seleccionar la acción que queremos y acortar el tiempo de programación, así como suministrar un código base sobre el que podemos trabajar directamente, únicamente añadiendo los elementos específicos (nombres de instancia) para hacer el programa final perfectamente funcional y sin errores de compilación.



Figura 3-13. Fragmentos de código

Como podemos ver en la Figura 3-13, Animate nos proporciona un amplio abanico de posibles acciones, navegación en la línea de tiempo o controladores de eventos con los

que vamos a dotar de interactividad nuestro trabajo. Estos tres apartados serán los más recorridos para realizar nuestro trabajo (Figura 3-14).



Figura 3-14. Menús desplegables de Fragmentos de código

La implantación de un código es obligatoria si queremos dotar de interactividad a nuestros botones, gráficos o clips de película, y que así puedan realizar las acciones que nosotros queremos. Para ello, en el momento de implementar nuestra primera acción a través de código, Animate creará en la sección Capas una capa llamada Actions.

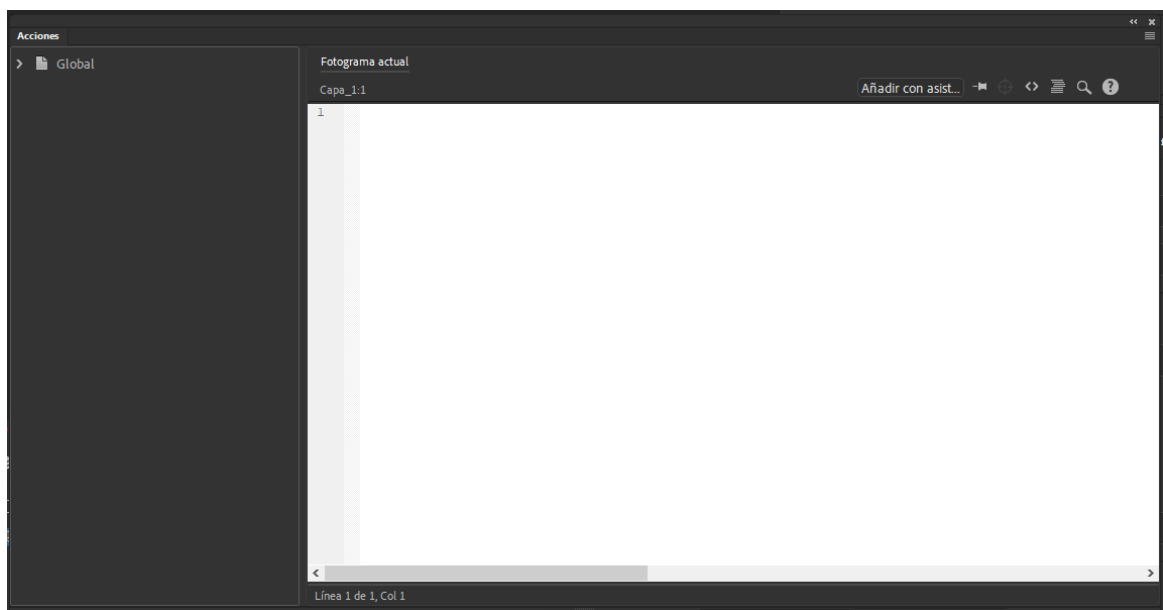


Figura 3-15. Ventana Acciones

En la ventana de Acciones que nos muestra la Figura 3-15 es en la que vamos a ver reflejado el código suministrado por Animate a modo de ayuda y donde lo podremos editar o escribir todo el código adicional que necesitamos para dotar de interactividad a nuestra aplicación.

Como también se ha mencionado en el punto anterior, es muy importante que todo el código escrito en la ventana Acciones no se repita sintácticamente y que cada fotograma tenga un código único e irrepetible, para así evitar errores de compilación y reproducción.

El menú Navegación en la línea de tiempo (Figura 3-15) será muy recurrido para dotar de utilidad todos los botones que componen nuestra aplicación. Con ellos podremos avanzar, retroceder o acceder al menú principal de la aplicación. Es decir, avanzar y retroceder entre fotogramas. Al implementar este código, siempre debemos utilizar la función “Hacer clic para ir al fotograma y detener”, ya que queremos no queremos que el programa interprete nuestra diapositiva como un vídeo y reproduzca todos los fotogramas hasta el final.

```
8
9  /* Hacer clic para ir al fotograma y detener
10 Al hacer clic en la instancia del símbolo especificado, la cabeza lectora se mueve hasta el fotogr
11 Se puede utilizar en la línea de tiempo principal o en líneas de tiempo de clips de película.
12
13 Instrucciones:
14 1. Reemplace el número 5 del siguiente código por el número de fotograma hasta el que quiere que s
15 2. Los números de fotograma en EaselJS empiezan con 0 en vez de 1
16 */
17
18
19 this.next_btn_1.addEventListener("click", fl_ClickToGoToAndStopAtFrame.bind(this));
20
21 function fl_ClickToGoToAndStopAtFrame ()
22 {
23     this.gotoAndStop("Diapositiva2");
24 }
25
26
```

Figura 3-16. Fragmento de código “Navegación en la línea de tiempo”

En este caso, (Figura 3-16) el código dotará de interactividad al primer botón de avanzar diapositiva, llamado “next\_btn\_1” mediante la función “ClickToGoToAndStopAtFrame()”. A continuación, debemos asignar a qué fotograma queremos que nos dirija el botón: podemos establecer directamente el número de fotograma, o para que sea más comprensible y sencillo a la hora de trabajar, Animate nos permite asignar un nombre a cada fotograma, con lo cual y siempre entre comillas, podemos escribir esa asignación y el código inmediatamente entenderá a donde nos debe dirigir.

Por otro lado, con el menú “Controladores de eventos” (Figura 3-17), podemos dar interactividad y controlar las acciones dentro de cada fotograma. La acción “Evento MouseOver” conseguirá que, al pasar el ratón por encima de la instancia del símbolo

especificado, se ejecuta una función a la que puede añadir su código personalizado. Esto es, como por ejemplo conseguir que un menú adicional a cada botón se despliegue con la simple acción de colocar el ratón sobre el botón.

```
21  
22     var frequency = 3;  
23     stage.enableMouseOver(frequency);  
24     this.interpolaciones_btn.addEventListener("mouseover", fl_MouseOverHandler.bind(this));  
25  
26     function fl_MouseOverHandler()  
27     {  
28         this.gotoAndStop("fuera_interpolaciones");  
29     }  
30  
31
```

Figura 3-17. Fragmento de código “Controladores de eventos”

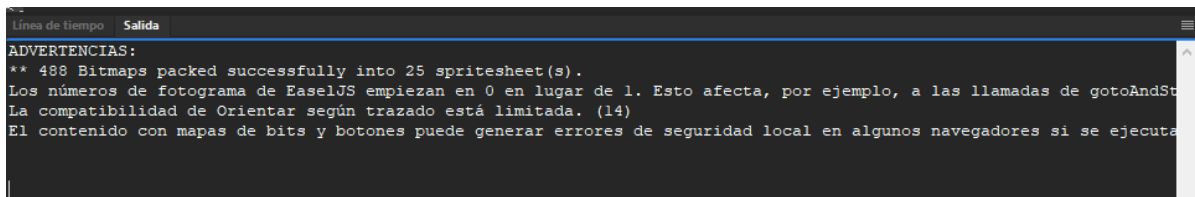
Vemos que nuestra función “MouseOverHandler” será la encargada de dotar de interactividad al botón “interpolaciones\_btn”. Para ello primero debemos habilitar la función, escribiremos “stage.enableMouseOver(frequency)”, en donde frequency es una variable que debemos tener en cuenta a la hora de programar y es la cantidad de veces que debería habilitarse el evento, este número siempre deberá ser mayor de 0.

### 3.3.5 Compilación

El proceso de compilación es en el cual se traducen las instrucciones escritas en un lenguaje de programación de alto nivel (en este caso JavaScript) a un lenguaje más sencillo de bajo nivel o lenguaje máquina [10]. A la hora de ejecutar nuestra aplicación para comprobarla, se abre otra ventana emergente del buscador Google Chrome, que por defecto es el que utiliza Animate para reproducir todas las aplicaciones creadas. Adicionalmente, Animate siempre realiza una compilación para comprobar que el código está escrito correctamente y sin errores.

En caso de haber algún error en el código, Animate nos mostrará esos errores en el panel de salida (Figura 3-18), así como advertencias y posibles puntos débiles en los que es posible que el futuro se produzcan fallos de compilación.





```
Línea de tiempo Salida
ADVERTENCIAS:
** 488 Bitmaps packed successfully into 25 spritesheet(s).
Los números de fotograma de EaselJS empiezan en 0 en lugar de 1. Esto afecta, por ejemplo, a las llamadas de gotoAndSt
La compatibilidad de Orientar según trazado está limitada. (14)
El contenido con mapas de bits y botones puede generar errores de seguridad local en algunos navegadores si se ejecuta
```

Figura 3-18. Ejemplo mensaje en el panel de Salida

Un posible indicativo de que nuestro código contiene errores, es que, al ejecutar nuestra aplicación en Chrome para comprobarla, es que esta aparezca totalmente en blanco sin mostrar nada del trabajo realizado. Una de las posibles maneras de resolver esto es comprobar todas las líneas de código una por una hasta encontrar el error, pero esta es una tarea tediosa que requiere de bastante tiempo y poder de observación para encontrar el error y ni mucho menos garantiza encontrar el error.

Las dos formas más adecuadas para encontrar el error es clicar con el botón sobre el mismo espacio en blanco de nuestro buscador, y seleccionar “Inspeccionar”. Entonces se nos abre una ventana en la que directamente nos dice cuál es nuestro error. La otra forma es escribiendo en nuestro código la sentencia “trace()”, donde entre paréntesis debemos colocar todo el código que queremos inspeccionar y que sospechamos que puede haber un error de sintaxis. El resultado final se mostrará en el panel de salida y nos indicará dónde está el error.

### 3.3.6 Publicación

Adobe Animate nos permite varias formas de exportar o publicar nuestro trabajo en función de nuestras necesidades finales. La función Exportar (Figura 3-19) en el menú Archivo no es la más adecuada para publicar y compartir una aplicación como la nuestra, ya que no aprovechamos todas las opciones que nos ofrece el programa, pero sí que permite mostrar pequeñas partes como puede ser un vídeo, una animación o un fotograma.

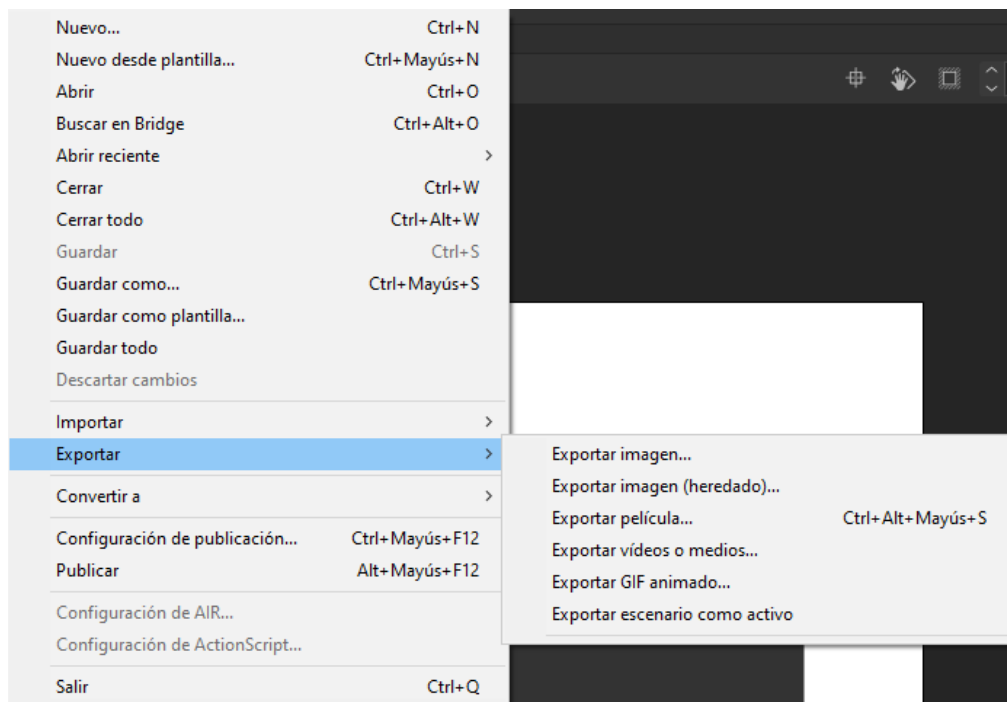


Figura 3-19. Menú Exportar

Por lo tanto, la solución que mejor se adecúa a nuestras necesidades es la opción Publicar. Con ella podremos compartir y hacer accesible nuestra aplicación de Control Numérico. Podemos encontrar esta función más abajo de la función Exportar, en el mismo menú Archivo.

El primer paso es configurar todas las variables que intervienen en nuestra aplicación, para ello acudimos a la función Configuración de publicación. Publicaremos en el formato JavaScript/HTML, porque queremos que la aplicación sea accesible desde cualquier ordenador y navegador. El formato HTML nos permite reproducir y visualizar correctamente todas las funcionalidades de nuestra aplicación, como pueden ser figuras animadas.

Como podemos ver en la Figura 3-20, Animate va a generar automáticamente 3 carpetas que van a contener las imágenes, los sonidos (en caso de que los haya), y los recursos de JavaScript. Animate creará estas carpetas en el mismo directorio en el que hasta ahora hemos estado guardando todo nuestro progreso.

Adicionalmente, cuando publicamos cualquier trabajo en Adobe Animate, a parte del archivo HTML, se nos genera un archivo .fla que contiene todas las etiquetas necesarias para reproducir correctamente nuestra aplicación. Estos dos archivos deben ir siempre juntos para asegurar el correcto funcionamiento del producto final.

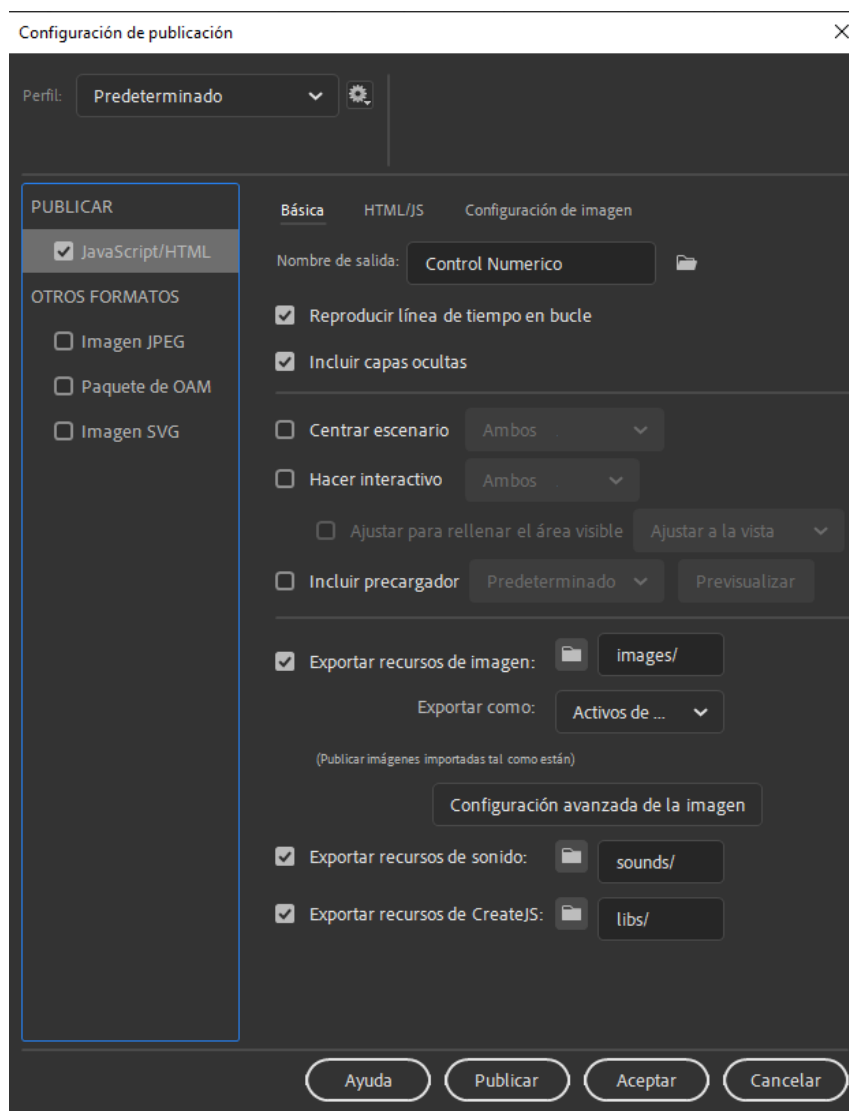


Figura 3-20. Configuración de publicación

Una vez establecidos todos estos requisitos previos, podemos proceder a publicar definitivamente todo nuestro trabajo. Para ello tenemos dos maneras, podemos acceder a través del menú *Archivo > Publicar*, o a través de los botones rápidos de acceso directo situado en la esquina superior derecha del programa (Figura 3-21).

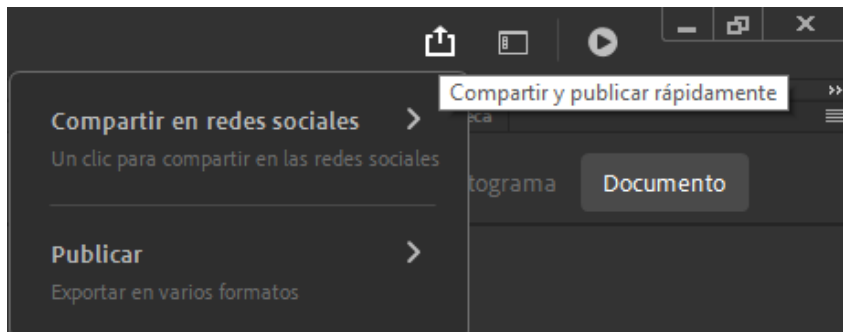


Figura 3-21. Compartir y publicar

Elegimos la opción Publicar y seguidamente nos dan a elegir el formato en el que deseamos publicar el trabajo. Como se ha explicado anteriormente, nosotros elegiremos la opción HTML Canvas (.html) (Figura 3-22).

Una vez hecho esto, podremos ver que en nuestro directorio se ha creado un archivo HTML que es nuestro acceso directo a la aplicación que se abre en el navegador.

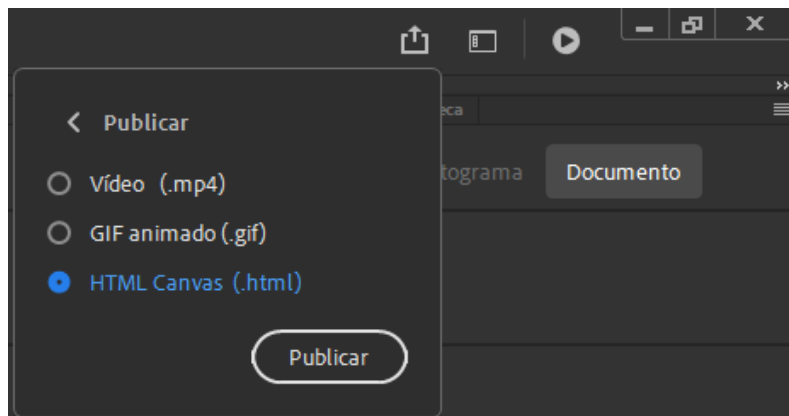


Figura 3-22. Publicar HTML Canvas

# 4 Descripción de la aplicación desarrollada

En este capítulo se va a explicar la aplicación creada, así como todos los pasos seguidos y la metodología, intentado transmitir todos los conocimientos adquiridos de alumno a alumno. La interactividad del programa es clave en el aprendizaje final, ya que el alumno puede comprender de una manera más visual la lección.

Antes de comenzar a elaborar una herramienta educativa, se recomienda hacer una búsqueda y consultar en internet las diferentes propuestas de plataformas educativas que podemos encontrar, de esta manera nos podemos hacer una idea global de cuál es nuestro objetivo final, comparando todas plataformas para poder encontrar una idea de compromiso que sea lo suficientemente didáctica e interactiva para poder transmitir todos los conocimientos que deseamos. Sin duda este es un paso muy importante, ya que nuestro boceto o idea inicial sobre la aplicación puede verse fuertemente modificada (y mejorada) con el simple hecho de buscar y estudiar distintos modelos de plataformas ya existentes.

## 4.1 Menú Principal e índice

Inicialmente, nuestra aplicación tendrá un menú de bienvenida que muestra que el alumno está accediendo a una aplicación diseñada para impartir conocimientos sobre Control Numérico y programación en códigos ISO, seguidamente y navegando a través de los botones habilitados, nos da paso al índice principal, donde el alumno podrá elegir cómo seguir su aprendizaje. En este caso, nuestra aplicación va a estar centrada en explicar cómo se programa una operación de fresado y cuáles son los distintos componentes y variables que intervienen durante todo el proceso.

Como podemos ver en la Figura 4-1, los temas que se van a desarrollar en profundidad serán:

- Introducción: en este primer punto se van a presentar las funciones básicas de las que consta un mecanizado CNC, (funciones preparatorias G y funciones auxiliares M) a la vez que enseñar cuál es la estructura que debe seguir cualquier código si queremos que nuestro CNC interprete y ejecute bien la programación.

- Interpolaciones: se van a desarrollar las interpolaciones lineales (G01), las interpolaciones circulares (G02) y el movimiento o posicionamiento rápido (G00)
- Decalaje: se va a explicar la distancia existente entre el cero máquina y el cero pieza, siendo este uno de los puntos fundamentales, ya que, si no se programa adecuadamente nuestro origen de pieza, toda la programación siguiente carecerá de valor ya que nuestro mecanizado no será acorde a los planos.
- Compensaciones: son códigos cuya función principal es facilitar la programación al operador, evitando tener que añadir sumas y restas del radio de la herramienta a los cálculos.
- Ciclos fijos: son muy útiles para programar taladrados o realizar el “vaciado” de una superficie, es decir, cajeado.
- Ejemplos prácticos: en este punto se van a proponer distintas piezas, sobre ellas se van a plantear unos enunciados, en función de los cuales se va a enseñar cómo se debería programar el mecanizado real de las piezas dadas.
- Autoevaluación: por último, se han redactado diferentes ejercicios de auto evaluación para que el alumno ponga a prueba los conocimientos adquiridos durante toda la lección.



Figura 4-1. Índice

Adicionalmente, y para hacer más accesibles todas las opciones secundarias, se han habilitado accesos directos en forma de botón que aparecen cuando el ratón se sitúa sobre alguno de los temas principales. En la Figura 4-2, podemos ver un ejemplo del menú adicional que aparece cuando colocamos el ratón sobre el Introducción.



Figura 4-2. Ejemplo de menú desplegable del tema Introducción

## 4.2 Diapositiva Introducción

Como en cualquier asignatura o curso, primeramente, debemos realizar una lección de introducción, para que el alumno se vaya familiarizando y asumiendo todos los conceptos, estructuras, abreviaciones y significados con los que vamos a trabajar a lo largo de todo el curso.

Las nociones básicas que debe aprender cualquier alumno que se inicie en la programación CNC, es comprender cómo escribir y cómo funciona la estructura de un bloque básico de código CNC, comprender todas las funciones preparatorias (G) así como entender qué trabajo realiza cada función, y, por último, están las funciones auxiliares (M), sin las cuales no podríamos realizar labores básicas como cambios de herramienta, sentidos de giro del cabezal o poner condiciones al movimiento.

### 4.2.1 Estructura de un bloque

Para comprender y saber interpretar el significado de cada línea de código, como primera tarea debemos entender cómo está estructurado. Todos los programas escritos están compuestos por líneas a las que llamamos bloques. Cada bloque contiene la información necesaria para realizar una operación, bien sea un movimiento de los ejes de la máquina, o una operación auxiliar.

El control CNC está programado para leer o ejecutar de manera secuencial los bloques de un programa, de modo que procesa la información de un programa más rápido de lo

que la máquina ejecuta las órdenes. En el modo ejecución, el control lee el programa bloque a bloque según se han escrito, interpreta el código escrito, chequea los errores y si no encuentra error ejecuta el movimiento u la orden en la máquina [14].

En la Figura 4-3 podemos ver la diapositiva creada para explicar la estructura general en nuestra aplicación. Cada letra mayúscula, está colocada de tal manera que representa el orden habitual con el que debería programar cada línea de código, si no seguimos este orden establecido, el control CNC no entenderá el código y nos mostrará el mensaje de error en la pantalla.



Figura 4-3. Estructura de un bloque

Adicionalmente, para completar y acabar de transmitir la información necesaria para comprender plenamente cómo funciona un bloque, el alumno podrá pulsar sobre cada letra, donde se han habilitado 7 pestañas adicionales, con el fin de incrementar la información sobre cada función. También, sobre cada pestaña se han añadido ejemplos sencillos a modo de primera toma de contacto, para que el alumno se pueda ir familiarizando con todas las estructuras, códigos y funciones, y así se hacer una idea sobre las nociones que se van a explicar a lo largo de toda la lección.

Los bloques contienen informaciones tan importantes como:

- Selección de herramienta para cada operación
- Forma de desplazamiento de la herramienta
- Activar funciones de máquina con funciones complementarias
- Funciones de control de avances de los ejes y de velocidades del cabezal



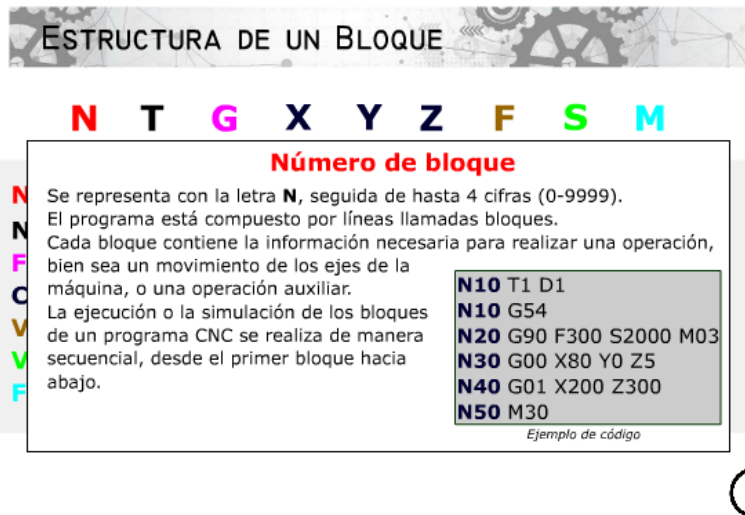


Figura 4-4. Pestaña adicional Número de bloque

En nuestra herramienta educativa, todas las lecciones y nociones que se van a explicar están basadas en el lenguaje de programación Fagor 8050. Existen otros lenguajes de programación ISO muy populares y extendidos, como es el FANUC. Este lenguaje de programación presente muchas similitudes y coincidencias con el Fagor 8050, sin embargo, hay una diferencia notable que debemos destacar: el orden N G X Z F S T D M es obligatorio en el lenguaje Fagor 8050, aunque en ambos el número de bloque debe ir al inicio. Este número de bloque (N) no es necesario programarlo, es decir, los bloques pueden ir sin numerar, y cuando se numeran tampoco es necesario que vayan en orden creciente. Los números de bloque van desde N0 hasta N9999, como se puede observar en la Figura 4-4.

#### 4.2.2 Diapositiva de Funciones preparatorias (G)

A continuación, en el bloque introductorio debemos hablar sobre las Funciones Preparatorias (G). Las funciones preparatorias, informan al control numérico de aspectos relacionados con el movimiento que desea realizar. En un mismo bloque pueden existir varias funciones preparatorias, siendo indiferente el orden en que se escriban [13]. Este capítulo introductorio es de vital importancia, ya que las funciones preparatorias son las más importantes en la programación CNC. Es fundamental que el alumno pueda entender el papel que desempeñan estas funciones que controlan el modo en que la máquina va a realizar un trazado, o el modo en que va a desplazarse sobre la superficie de la pieza que está trabajando.

Los comandos G se escriben siempre seguidos de 2 cifras, van de 00 a 99, y cada uno tiene una función determinada. Como ya se describió anteriormente, los tres tipos de funciones preparatorias se explican en una de las diapositivas de la herramienta desarrollada, tal como se ve en la Figura 4-5.

**FUNCIONES PREPARATORIAS (G)**

Son funciones que siempre presentan la siguiente estructura:

- La letra G seguida de dos cifras (GXX).
- Se deben programar siempre al comienzo del bloque.

Funciones Modales	Funciones Secuenciales	Funciones preferenciales
Permanecen activas hasta que no se anulen ó se programe otra función G. Por ejemplo: G00: avance y posicionamiento rápido G01: interpolación lineal en régimen de trabajo	Sólo están activas mientras se está ejecutando el bloque en el que están. Por ejemplo: G02/G03: en régimen de trabajo, interpolación lineal a derechas o izquierdas	Estas funciones estarán activas por defecto al encender el CN. Por ejemplo: G70/G71: programación en pugas/mm G90/G91: programación en coordenadas absolutas/relativas

Figura 4-5. Funciones Preparatorias (G)

Dado que este es el capítulo introductorio, únicamente se pretende familiarizar al alumno con funciones tan importantes como son G01 o G02. Más adelante, se dedica un capítulo entero a desarrollar y explicar en profundidad las interpolaciones lineales (G00, G01) y circulares (G02, G03).

### 4.2.3 Diapositiva de Funciones auxiliares (M)

Las funciones auxiliares son las utilizadas para definir el funcionamiento de la máquina, tales como el sentido de giro y parada de la pieza o herramienta, activación y desactivación del refrigerante, cambios de herramienta, orientación de la parada del cabezal, cierre y apertura de puertas, final del programa, etc.

De manera general, las funciones auxiliares se pueden clasificar en tres grupos: las relacionadas con el programa (M00, M30), con la herramienta (M03, M04) y con el funcionamiento de la máquina en general (M08, M09). Además, se pueden considerar las funciones auxiliares M usadas por el propio fabricante de la máquina CNC, como es la apertura de puertas o abrir y cerrar las garras. Estas funciones en particular son modales, ya que vienen programadas por el fabricante y no se pueden modificar y personalizar.

## FUNCIONES AUXILIARES (M)

Estas funciones sirven para ejecutar funciones y acciones que complementan el régimen de trabajo, como son las paradas o los giros de cabezal.

Se permiten programar hasta 7 funciones auxiliares y el CNC las ejecuta correlativamente en el orden que han sido programadas.

Las funciones más utilizadas son las siguientes:

**M00**: parada de programa, para reanudarlo se debe pulsar el botón marcha.

**M02**: fin de programa con vuelta al primer bloque.

**M03/M04**: giro del cabezal a derechas o izquierdas, es decir, sentido horario/sentido antihorario.

**M05**: parada de cabezal, se puede personalizar para ejecutarla al final del bloque

**M06**: cambio automático de herramienta.

**M08/M09**: activar/desactivar el refrigerante en modo ducha.

**M30**: fin de programa.



Figura 4-6. Funciones Auxiliares (M)

Como se destaca en la Figura 4-6, en un bloque se pueden programar hasta 7 funciones M, siempre que no sean incompatibles entre sí. Una característica muy importante a tener en cuenta de las funciones auxiliares es, que se ejecutan sucesivamente en el orden que han sido programadas.

En esta diapositiva se han mostrado 9 de las funciones más básicas y más utilizadas que el alumno debe comprender y saber utilizar adecuadamente en cada momento.

## 4.3 Diapositiva de Interpolaciones

En el mundo de la programación ISO -en este caso FAGOR 8050- podríamos definir el término interpolación como el cálculo o programación de contornos y trayectorias. En todo dibujo o pieza, solo existen básicamente dos formas posibles de trayectoria a seguir: son la trayectoria en línea recta y línea curva. Con la combinación de estas dos formas básicas, se puede diseñar cualquier dibujo o pieza por complejo que sea. La línea recta, es más fácil de comprender, ya que no es más que es la trayectoria más corta que existe entre esos dos puntos dados. La línea curva en realidad siempre es un trozo de circunferencia a la que llamamos arco, y sabemos que para definirla, necesitaremos el punto final y el punto inicial, además de al menos otro punto, que será el centro de la circunferencia en el que está contenido.

Muchas instrucciones G, pueden tener distintos funcionamientos en otras máquinas dependiendo de los parámetros globales que tenga activados el control numérico. Por

eso, lo verdaderamente importante es afianzar bien los conceptos, así como el funcionamiento básico de los procesos.

### 4.3.1 Posicionamiento rápido (G00)

Con esta función indicamos a la máquina que se desplace linealmente a la máxima velocidad hasta alcanzar el punto indicado. En FAGOR 8050, la herramienta se desplaza de tal manera que se consigue que la velocidad del avance en la dirección del movimiento sea máxima. Mientras se ejecuta G00 el avance programado F no le afecta.

Como podemos observar en la definición la Figura 4-7, “El principal propósito de esta función es ejecutar desplazamientos rápidos de la herramienta sin estar en régimen de trabajo”

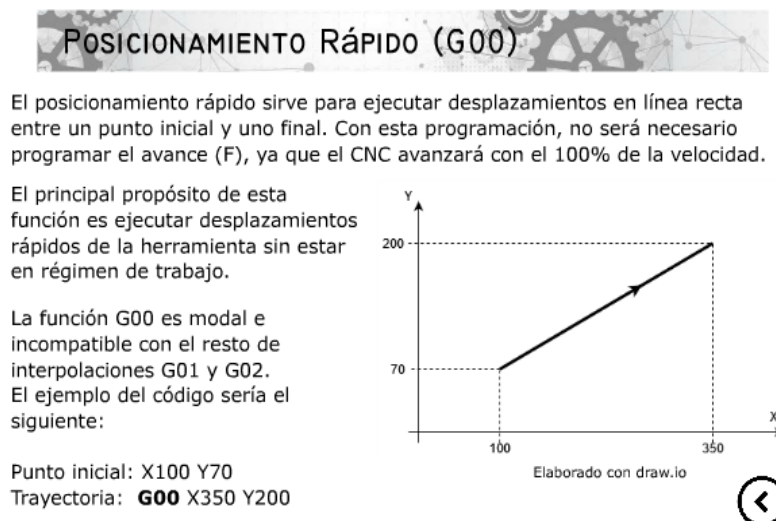


Figura 4-7. Posicionamiento rápido (G00)

Independientemente del número de ejes que se mueven, la trayectoria resultante es siempre una línea recta entre el punto inicial y el punto final. Es por eso que debemos precisar lo máximo posible nuestras coordenadas, a la vez que, hay que tener mucho cuidado cuando en un mismo bloque se ordena que se desplace en varios ejes a la vez, (G00 X70 Y50 Z10), pues el control hará sus cálculos para llegar al punto indicado en los tres ejes a la vez, pudiendo ocasionar situaciones de riesgo como puede ser golpear la herramienta con tocho o las abrazaderas.

Durante el avance rápido, el interruptor de corrección de avance 0-120% está activado. Por lo tanto, si el corrector de avance está a 0 no hay movimiento [7].

G00 es modal e incompatible con otras interpolaciones G01, G02, G03. G00 es la interpolación de movimiento lineal asumida por defecto por el control, cuando se enciende, después de ejecutar M02 o M30, o tras pulsar el interruptor de emergencia o de resetear el control.

#### 4.3.2 Diapositiva de Interpolaciones Lineales en régimen de trabajo (G01)

La interpolación lineal G01 es de los códigos que más vamos a utilizar en nuestra herramienta educativa, esto es debido a que en el mundo profesional comprender y saber programar correctamente este código es básico para realizar el mecanizado de cualquier pieza. Es por eso que se hará especial énfasis en intentar explicar lo más precisamente posible la función de este código, cómo trabaja y en qué se diferencia de los demás.

Esta instrucción, sirve para programar una trayectoria en línea recta en régimen de trabajo, mecanizando a la velocidad de avance programado (F). El origen de la línea recta se encuentra en el punto en el que tenemos la herramienta en este momento, es decir en las cotas actuales. El punto final de esa recta será el que programamos en esta instrucción, en cotas absolutas (G90), o en cotas incrementales (G91).

Adicionalmente, debemos explicar los comandos G90 y G91, como se ha mencionado, estos sirven para programar nuestras coordenadas en cotas absolutas y cotas incrementales respectivamente. Es por ello que en esta lección se muestra cómo programar la trayectoria de la imagen (Figura 4-8) en coordenadas absolutas y coordenadas incrementales, con el objetivo de que el alumno pueda entender y saber señalar las diferencias entre los dos tipos de programación.

Al igual que ocurre con la función de desplazamiento rápido (G00) -y como más adelante se explicará con las interpolaciones circulares-, los desplazamientos programados después de G01 se ejecutan en línea recta y según el avance F programado, así podremos ahorrar tiempo de programación al no necesitar escribir constantemente el código. Es por ello que es una función modal.

## INTERPOLACIONES LINEALES (G01)

Esta función se ejecuta en régimen de trabajo: se deben introducir las coordenadas X e Y para definir la trayectoria, así como el avance F para definir la velocidad de avance de la herramienta.

Adicionalmente (entre G01 y las coordenadas X/Y), se puede incluir la función G90/G91 para trabajar con programación absoluta o incremental.

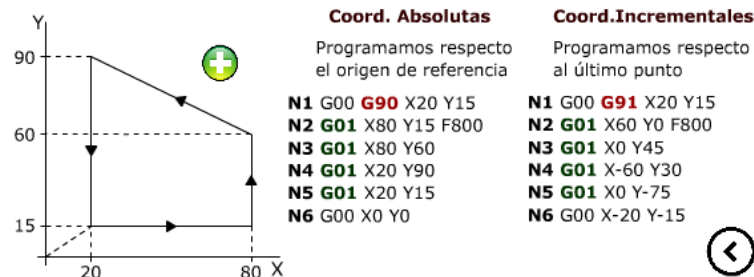


Figura 4-8. Interpolaciones lineales (G01)

Un avance G01 puede ser un movimiento de un solo eje o una combinación de los ejes. Cuando se mueven dos o tres ejes simultáneamente la trayectoria resultante es una línea recta entre el punto inicial y el punto final. La máquina se desplaza según dicha trayectoria y según el avance programado (F). El control calcula los avances de cada eje para que el avance de la trayectoria resultante sea la F programada. Si un eje no pudiera alcanzar la velocidad de avance programada, el control no continuaría con el movimiento y se generaría una alarma (máxima velocidad de avance superada).

En el capítulo introductorio de nuestra herramienta educativa (Figura 4-3), se ha realizado una primera explicación sobre la velocidad de avance. Es en esta lección es cuando debemos ampliar y profundizar nuestros conocimientos ya que requiere aplicarlos bien durante la programación. La velocidad del movimiento de los ejes está controlada por el valor de la velocidad de avance (F). Este valor F puede estar en diferentes unidades, en milímetros por minuto (G94) o en milímetros por revolución del husillo (G95). El valor de la velocidad de avance puede estar programado en la línea de comandos actual, o en una línea anterior, ya que siempre utiliza el valor más reciente de F hasta que se ordena otro valor de F.

## INTERPOLACIONES LINEALES (G01)

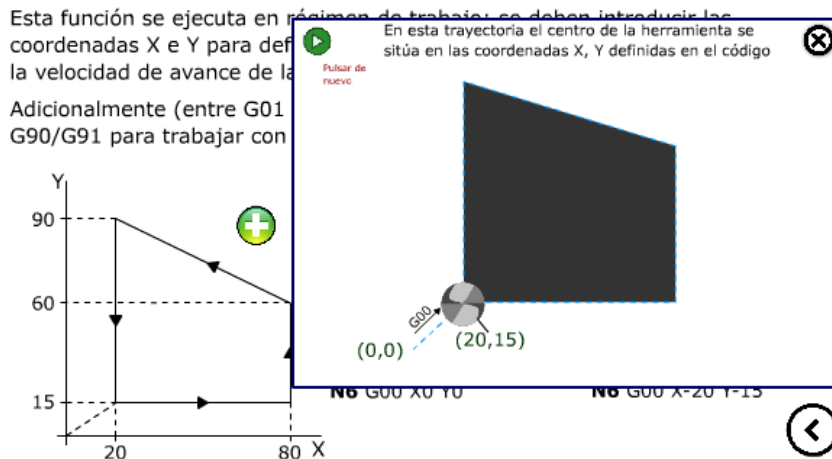


Figura 4-9. Diagrama Interpolaciones Lineales

A modo de ejemplo, y para que los conocimientos de la lección se puedan ilustrar de la manera más comprensible posible, con el programa Adobe Animate, se ha generado una sencilla ilustración que pueda representar de manera fiel la trayectoria que seguiría la herramienta en un ciclo de mecanizado G01 real (Figura 4-9). La ilustración, se ha programado para que en cada punto crítico (cambios de trayectoria) realice una pausa para que el alumno pueda observar detenidamente en qué consiste exactamente la función G01.

### 4.3.3 Diapositiva de Interpolaciones Circulares (G02, G03)

Las interpolaciones circulares, es otra de las nociones elementales que debemos entender y manejar de manera correcta a la hora de programar cualquier mecanizado. Podríamos definir una interpolación circular como el movimiento que el husillo realiza en régimen de trabajo y que une dos puntos de un perfil describiendo entre ellos una trayectoria circular.

Existen dos tipos de interpolaciones circulares, y como podemos ver en la Figura 4-10 son las interpolaciones en sentido horario (G02) y en sentido antihorario (G03). Las interpolaciones circulares son también modales e incompatibles entre sí, por tanto, seguirán activas hasta que no se programen o anulen con otras.

## INTERPOLACIONES CIRCULARES (G02/G03)

Para realizar una interpolación circular tenemos dos opciones:

**G02:** interpolación circular en sentido horario



**G03:** interpolación circular en sentido antihorario

Al igual que la interpolación lineal, se debe programar el avance F.

El punto final se puede programar en coordenadas **cartesianas absolutas** o

**incrementales:**

**Absolutas**

N1 G00 X15 Y20  

N2 G03 G90 X75 Y80 I0 J60 F800

N3 G03 X135 Y20 I60 J0

**Incrementales**

N1 G00 X15 Y20

N1 G03 G91 X60 Y60 I0 J60 F800

N2 G03 X60 Y-60 I60 J0

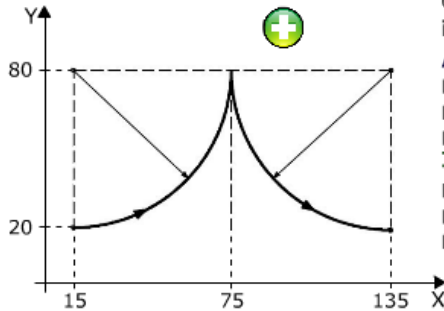


Figura 4-10. Interpolaciones Circulares (G02, G03)

Al igual que ocurre con el resto de instrucciones, podemos programar las interpolaciones circulares en coordenadas absolutas o incrementales. Independientemente del tipo de programación incremental o absoluta que hayamos elegido, y de si es un giro horario o antihorario, siempre que programamos esta función, debemos indicar las coordenadas finales del arco. Así es como se ha indicado en la pestaña auxiliar que el alumno puede ver pulsando sobre el símbolo “más” (Figura 4-11). A lo largo de toda la aplicación, se van a utilizar con regularidad este tipo de pestañas “pop-up”, ya que son una herramienta eficaz para añadir información muy específica y así ampliar todos los conocimientos que se pretenden transmitir en la lección

## INTERPOLACIONES CIRCULARES (G02/G03)

Para realizar una interpolación circular tenemos dos opciones:

Las coordenadas X e Y que se programan, se refieren siempre al punto final de la trayectoria circular.

El punto inicial lo habremos programado previamente, mediante G00 ó G01.



Se programa de igual manera con G02 como con G03.



ar el avance F.

se puede programar en **cartesianas absolutas** o **es:**

**Absolutas**

N1 G00 X15 Y20  

N2 G03 G90 X75 Y80 I0 J60 F800

N3 G03 X135 Y20 I60 J0

**Incrementales**

N1 G00 X15 Y20

N1 G03 G91 X60 Y60 I0 J60 F800

N2 G03 X60 Y-60 I60 J0

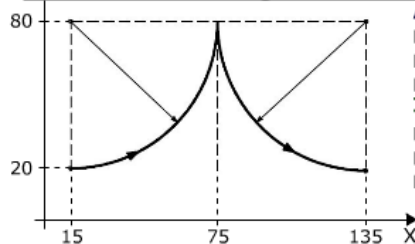


Figura 4-11. Pestaña auxiliar Interpolaciones Circulares



La interpolación circular, dependiendo de la información disponible en el plano, se puede programar de diversos formatos:

- Programación del centro del arco de circunferencia.
- Programación del radio.

En nuestra herramienta educativa, nos vamos a centrar únicamente en explicar el modo de programar a través del arco de la circunferencia.

En el plano X/Y, podemos programar una interpolación circular mediante las coordenadas incrementales I y J. Los parámetros I, J se usan para situar el centro del arco. Es importante destacar (y así lo podemos ver en la Figura 4-12), que estos parámetros incrementales son las coordenadas del centro al punto inicial del arco. Por lo tanto, las coordenadas I, J tienen valor y signo, y es básico programar correctamente estas coordenadas, ya que la información que recibiría el control sería errónea y el resultado de nuestro mecanizado no será el deseado y podríamos exponer la máquina a riesgo de colisión con el tocho.

Por lo tanto, para programar correctamente el código de interpolación circular, debemos añadir el punto final del arco (mediante las coordenadas X, Y, Z), y mediante las coordenadas incrementales (I, J, K).

En la aplicación, esta información ha sido añadida a través de una pestaña auxiliar como podemos ver en la Figura 4-12.

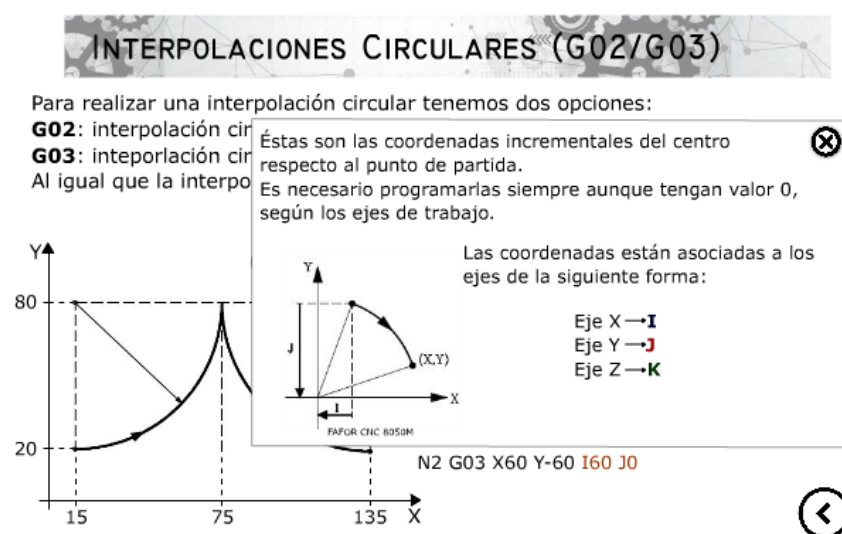


Figura 4-12. Parámetros I, J

Adicionalmente, y de la misma manera que se ha hecho en la lección sobre las interpolaciones lineales (G01), a través de una pestaña adicional, en el programa Adobe Animate se ha elaborado un diagrama que representa el movimiento que realiza la herramienta al interpretar el esquema de la Figura 4-13. Al igual que ocurre con el diagrama creado anteriormente, el alumno podrá pausar la trayectoria y observar detenidamente el recorrido de la herramienta, a la vez que se muestran las funciones de la trayectoria y los puntos recorridos.

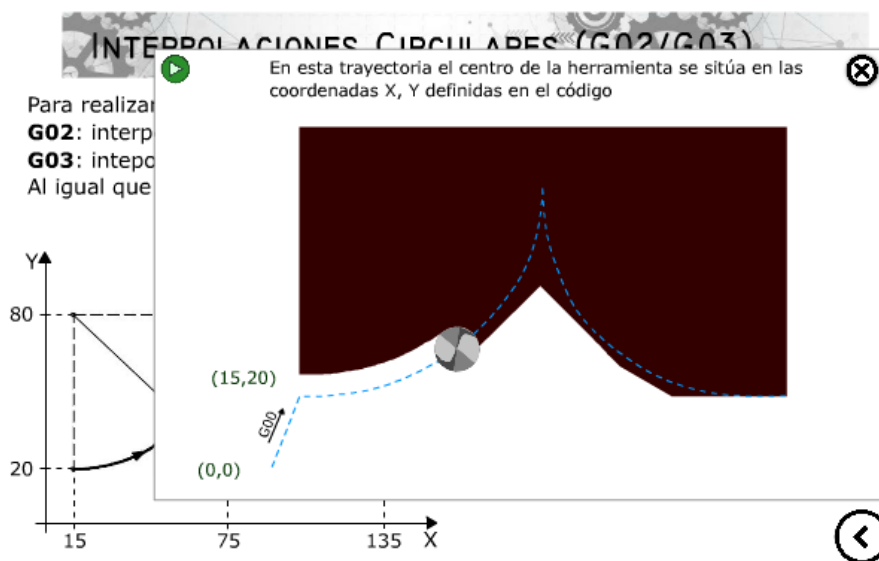


Figura 4-13. Diagrama de Interpolaciones Circulares

## 4.4 Diapositiva de Decalajes

En el capítulo de introducción de nuestra herramienta educativa (Figura 4-14), en la pestaña “Cotas según X, Y, Z”, podemos observar los ejes sobre los que va a trabajar cualquier fresadora, ya sea CNC o analógica.

## ESTRUCTURA DE UN BLOQUE

**N T G X Y Z F S M**

### Cotas según X, Y, Z

**N** El sistema de referencia de la máquina se compone de las coordenadas **X, Y, Z**.  
**N** También se denominarán X, Y, Z los 3 ejes que  
**F** tendrá la pieza a mecanizar.  
**C**  
**V** El CNC permite hacer uso de coordenadas absolutas  
**V** y de coordenadas relativas o incrementales, a lo  
**F** largo de un mismo programa.

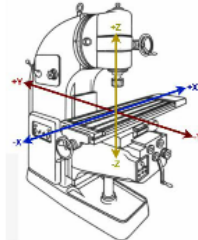


Figura 4-14. Cotas según X, Y, Z

Todas las funciones y códigos explicados hasta ahora, carecerían de algún sentido si previamente no definimos un punto de referencia sobre el que basar toda nuestra programación.

## DECALAJE

En cualquier máquina dirigida por control numérico, se han definido los siguientes puntos de referencia y origen:

**Origen de máquina:** es el origen del sistema de coordenadas y está fijado por el fabricante, no se puede modificar.

**Origen de pieza:** es el punto de origen de la pieza, se utiliza este punto para la programación del resto del código (coordenadas).

**Origen de herramienta:** está impuesto por el fabricante con respecto al origen de máquina y es el punto de referencia del cabezal.



En todo momento debemos especificar la altura y el diámetro de la herramienta, para trabajar con precisión.

**TCP:** Punta o centro de la herramienta (Tool Center Point)

"sandvik.coromant.com"



Figura 4-15. Decalaje

Como ya se ha mencionado anteriormente en el capítulo Decalajes de origen, el cero pieza es el punto más importante desde el punto de vista del programador. Es por eso, que es necesario dedicar una diapositiva para explicar en profundidad la importancia de escoger y programar bien el origen de la pieza (Figura 4-16).

La finalidad de esta diapositiva, es que el alumno comprenda la función del decalaje y cómo establecer correctamente el "origen pieza", cómo cambiar los orígenes en caso de

necesidad, y qué implica trabajar con uno u otro “origen pieza”. El cero pieza es el origen de coordenadas que utilizaremos como referencia para delimitar los puntos del contorno de nuestro programa, por eso lo situaremos donde mejor nos convenga, y podremos cambiarlo a lo largo de nuestro programa todas las veces que lo consideremos oportuno.

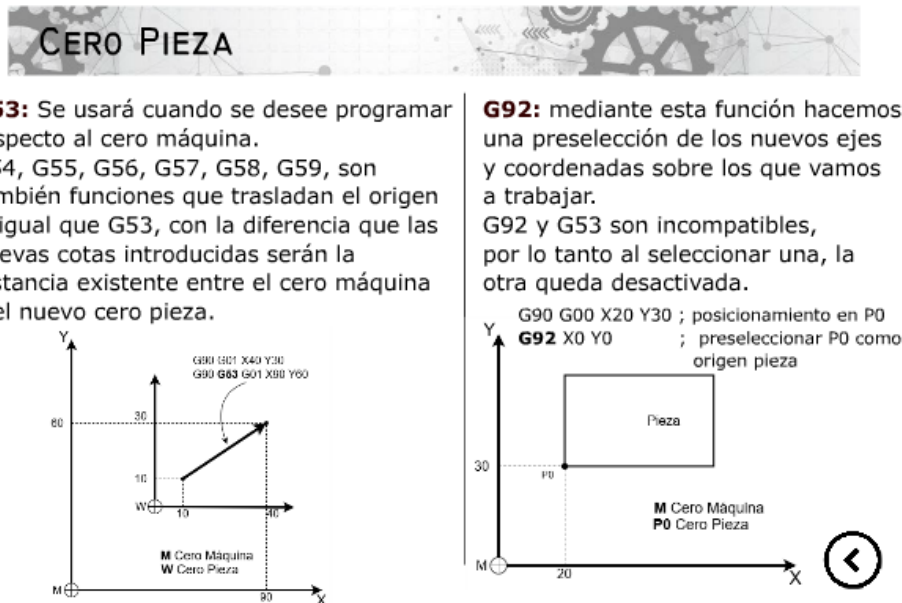


Figura 4-16. Cero pieza

Las funciones “G53” a “G59” no son las únicas que podemos utilizar para establecer nuestro origen pieza. Existe la función “G92”, que permite hacer la preselección del nuevo cero pieza que queremos establecer y así modificar sobre la marcha el origen y establecerlo donde nos sea más cómodo, es decir, el control asume las cotas de los ejes programados a continuación de G92, como nuevos valores de los ejes. Este código no es tan utilizado como los anteriores, pero puede resultar de gran utilidad a la hora de mecanizar piezas con geometrías complejas.

Como se ha mencionado al principio de este capítulo, disponemos de la opción de establecer nuestro cero pieza realizando un calibrado con la herramienta. Para poder ejecutar este paso, es condición necesaria previa, haber realizado un calibrado de la propia herramienta. Esto se debe hacer cada vez que añadimos o quitamos una herramienta a nuestro portaherramientas.

Es por ello, que el control dispone de una tabla específica que contiene toda la información relativa a nuestras herramientas (Figura 4-17). Esta es la “tabla de correctores” en la que se almacenan diferentes parámetros que pueden involucrar a la herramienta, como ya se ha explicado en capítulos anteriores. Estos datos son básicos e imprescindibles para trabajar

con precisión y sacar el máximo rendimiento que nos ofrecen las modernas máquinas CNC.

A la hora de programar, lo primero que debemos escoger es con qué herramienta vamos a realizar nuestro mecanizado. Es por ello que no sólo debemos seleccionar el número de herramienta (T), sino que también el corrector (D) que la acompaña.

**DECALAJE**


El corrector D de una herramienta indica la posición de la herramienta en la tabla de correctores.

Sirve para almacenar los datos de radio de la herramienta (**R**), longitud (**L**), desgaste del radio (**I**) y desgaste en longitud (**K**).

Así, podremos utilizar otros códigos que nos ofrece el CNC como son las compensaciones de radio y longitud de herramienta.

Los números de T y D no tienen por qué coincidir, pero parece lógico asignar a cada herramienta el mismo número del corrector D.

*Ejemplo: T03 D03*



TOOL NO.	RADIUS	LENGTH	RADIUS WEAR	LENGTH WEAR
T001	0.000	0.000	0.000	0.000
T002	0.000	7.983	0.000	0.000
T003	0.000	0.000	0.000	0.000
T004	0.000	0.000	0.000	0.000
T005	0.000	0.000	0.000	0.000
T006	0.000	0.000	0.000	0.000
T007	0.000	0.000	0.000	0.000
T008	0.000	0.000	0.000	0.000
T009	0.000	0.000	0.000	0.000
T010	0.000	0.000	0.000	0.000
T011	0.000	0.000	0.000	0.000

TCP sandvik.coromant.com

Figura 4-17. Pestaña adicional Decalaje

## 4.5 Diapositiva de Compensaciones

La compensación del radio de la herramienta es una de las herramientas más potentes que nos pueden ofrecer las máquinas CNC modernas. Es por esto que es vital explicarla para que el alumno pueda comprender su funcionamiento y cómo programar correctamente este código.

Se puede definir la compensación del radio de la herramienta como la función que se encarga de restar/sumar el radio de la herramienta para que las dimensiones finales de la pieza sean las deseadas (Figura 4-18).

Es importante destacar que las funciones de compensación del radio de la herramienta son modales, por lo tanto, seguirán activas hasta que no se anulen. Para ello disponemos del código "G40", cuya única función es anular o apagar la compensación del radio. Este código se suele programar al final del programa o al final de cada bloque.

Gracias a la compensación de radio, a la hora de programar, únicamente nos debemos centrar en la labor de programar la geometría de un perfil, liberándonos de la tarea de realizar cálculos geométricos difíciles. Sin la compensación de radio de herramienta, las trayectorias programadas son seguidas por el centro de la herramienta. La programación de un perfil irregular y con peculiaridades sin la compensación del radio de la herramienta puede llegar a ser complicada, por lo que se deja esta tarea al control, el cual automáticamente con el dato del radio de la herramienta activa, calcula la trayectoria compensada, que es paralela a la real.

Además, la trayectoria compensada depende del sentido de avance de la herramienta, así se distinguen compensación de radio a izquierdas y a derechas. Es por ello que en nuestra herramienta educativa nos vamos a centrar a explicar las dos compensaciones más importantes y utilizadas en cualquier programación. Estas son las funciones “G41” y “G42”.

### COMPENSACION DEL RADIO DE LA HERRAMIENTA

A la hora de trabajar correctamente, debemos tener en cuenta el radio de la herramienta para poder obtener las dimensiones deseadas de la pieza. Para hacer más sencilla la programación y no tener que calcular las coordenadas sumando o restando el radio, podemos utilizar las compensaciones del radio de la herramienta, G41 y G42.

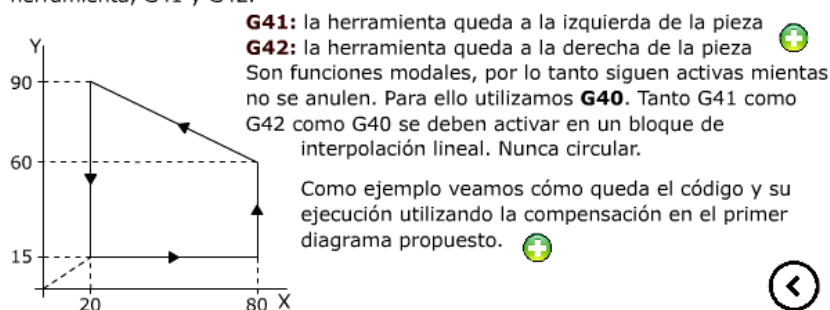


Figura 4-18. Compensación del radio de la herramienta

Dada la importancia de la compensación de la herramienta para la correcta programación, en la Figura 4-19 se muestra al alumno las diferencias entre el código G41 y G42, para que así sea mucho más sencillo comprender en qué casos debemos programar cada código.

Es básico saber diferenciar estos dos códigos, debido a que nos exponemos a escribir erróneamente el programa de mecanizado de una pieza, y finalmente no obtendríamos las dimensiones de mecanizado deseadas.

## COMPENSACION DEL RADIO DE LA HERRAMIENTA

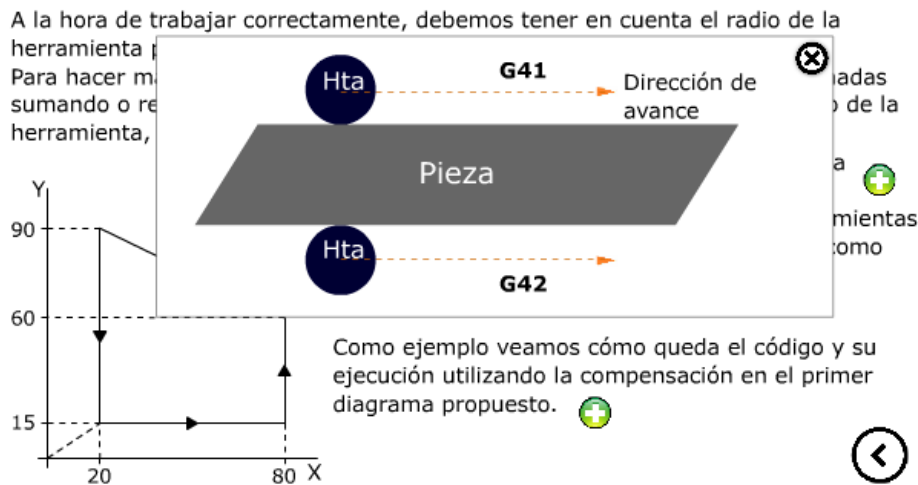


Figura 4-19. Pestaña auxiliar compensaciones de herramienta

### 4.6 Diapositiva de Ciclos fijos

En el mundo profesional, no siempre nos van a pedir realizar la programación sobre piezas sencillas como pueden ser contorneados o planeados, utilizando únicamente las funciones y códigos que se han explicado hasta ahora. Es muy probable que nos pidan realizar algunos mecanizados en la fresadora, como taladrados, roscados, vaciado de cajas circulares o rectangulares, es necesario programar movimientos repetitivos tales como posicionamientos, mecanizados y retrocesos.

Es por ello, que en los diferentes lenguajes de programación CNC, se han establecido una serie de ciclos fijos que nos pueden ayudar a simplificar nuestro código a la vez que ayudar a acortar considerablemente la longitud de todo el bloque ya que como se ha mencionado, no necesitaremos programar una y otra vez los mismos movimientos repetitivos. Así, podemos definir un ciclo fijo como el conjunto de secuencias repetitivas con las que se realizan operaciones básicas, que son memorizadas y controladas por una función preparatoria.

Una de las características de los ciclos fijos, es que pueden ser programados para trabajar en cualquier plano. Por lo tanto, al programar cualquier ciclo fijo, éste se ejecuta en el plano seleccionado, realizando el mecanizado en el eje perpendicular a dicho plano.

En los ciclos fijos se programan todos los parámetros que definen la operación a realizar, los cuales se identifican por letras seguidas de un valor numérico. En los ciclos fijos se

programan todos los parámetros que definen la operación a realizar, los cuales se identifican por letras seguidas de un valor numérico. Como podemos ver en la Figura 4-20, los parámetros más importantes que debemos tener siempre en cuenta son el plano de partida, el plano de referencia y la profundidad de mecanizado.

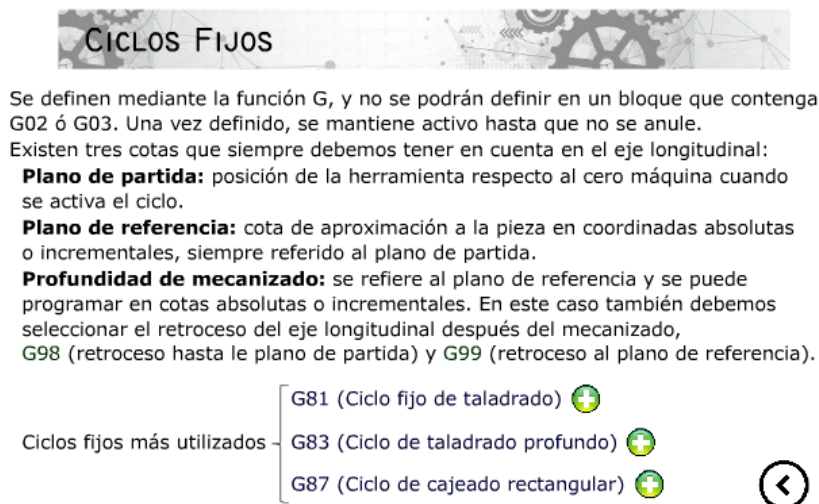


Figura 4-20. Ciclos Fijos

La ejecución de un ciclo fijo no anula las funciones "G" modales anteriores a la llamada al ciclo. Excepto la compensación de radio (G41 y G42) que deja activo un G40. Por ello es preferible trabajar sin compensación de radio. Todos los ciclos fijos son modales, y se ejecutan en todas las coordenadas programadas a partir del bloque del en el que lo programamos y hasta su anulación.

En nuestra aplicación, nos hemos centrado en 3 ciclos fijos; estos son el ciclo fijo de taladrado (G81), el ciclo de taladrado profundo (G83) y el ciclo de cajeado rectangular (G87). Y, como se ha venido haciendo hasta ahora, utilizaremos las pestañas auxiliares para explicar en profundidad y detalle cada ciclo.

- G81: Ciclo fijo de taladrado. Este ciclo realiza un taladrado en los puntos indicados hasta alcanzar la cota final programada. Este ciclo es comúnmente empleado para realizar taladrados de profundidad media o larga. En el capítulo de ejemplos prácticos se hará una comparación entre la ejecución y programación de este ciclo "G81" con realizar el taladrado utilizando únicamente la programación estándar con "G01".



- G83: Ciclo fijo de taladrado profundo (Figura 4-21). Esta función es menos usual pero más completa. Se considera taladrado profundo cuando la profundidad del agujero supera de 4 a 5 veces el diámetro de la broca. La función ejecutará un desplazamiento rápido de los ejes del plano principal hasta alcanzar la posición del agujero. Desplazamiento rápido del eje perpendicular al plano principal, desde el plano de partida hasta el plano de referencia. Taladrado del agujero hasta la profundidad (I) a la velocidad de avance programado (F). Retroceso rápido de la herramienta (Z) hasta el plano de referencia. Desplazamiento rápido del eje (Z) hasta una distancia de 1mm menor que la última profundidad (I) alcanzada. Taladrado del agujero hasta la 2ª profundidad (I) a la velocidad de avance programado (F). Retroceso rápido de la herramienta (Z) hasta el plano de referencia. De esta forma se repetirán las secuencias hasta alcanzar la profundidad total del agujero, es decir, (IxJ). El resto tiene un funcionamiento igual que el definido para G81.

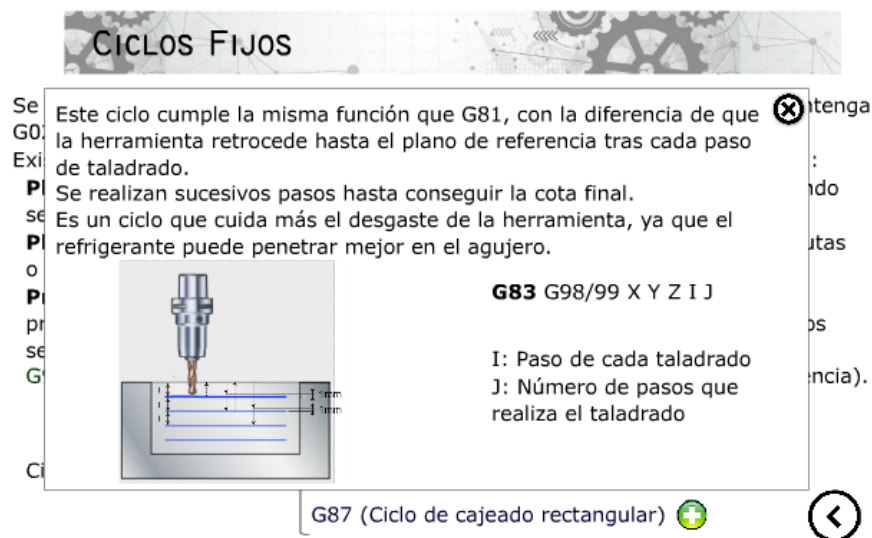


Figura 4-21. Pestaña auxiliar Ciclo G83

- G87: Ciclo fijo de cajera rectangular. Este ciclo realiza una cajera de forma rectangular con unas dimensiones dadas de largo, ancho y profundidad, en el punto indicado hasta alcanzar la cota final programada. Adicionalmente, el ciclo permite programar el avance de fresado, programar una última pasada de acabado con su correspondiente velocidad de avance (F). Con el fin de alcanzar un buen acabado en el mecanizado de las paredes de la cajera, el control aplicará en cada una de las profundizaciones, una entrada y salida tangenciales a la última pasada de

fresado. G87 es capaz de desbastar y acabar una cajera que tenga forma cuadrada o rectangular de forma automática mediante los parámetros indicados en el ciclo y de los datos de la herramienta utilizada.

- G80: Anulación del ciclo fijo. Este código cumple la misma función que “G40”, al anular el ciclo y dar por terminada su ejecución en el bloque en el que ha sido programado.

## 4.7 Ejemplos prácticos

Se han planteado 3 ejemplos prácticos con el fin de aplicar todos los conocimientos aprendidos durante la lección, y demostrar gráficamente el recorrido de la herramienta en régimen de trabajo. Para añadir más interactividad a nuestra aplicación, el alumno tiene la posibilidad de pulsar sobre cada línea de código y así comprobar el movimiento real de la herramienta.

Las tareas sugeridas son sencillas y completamente realizables si se ha seguido toda la lección, es por ello que estos ejemplos prácticos se muestran en los últimos pasos de la lección.

### 4.7.1 Ejemplo práctico 1

En este primer ejemplo, se va a pedir escribir el programa que nos permita realizar un planeado, un perfilado, una cajera central y mecanizar los 4 agujeros que tiene nuestra pieza.

Como podemos ver en la Figura 4-22, en el enunciado se muestran los planos necesarios para recabar toda la información necesaria para ejecutar correctamente nuestra programación.

Los diferentes bloques están claramente diferenciados, y se han colocado en el orden correspondiente para ejecutar un mecanizado real de la pieza, empezando por el planeado y acabando por el taladrado.

## EJEMPLO PRÁCTICO I

Como ejemplo práctico se va a proponer escribir el programa que permita realizar un planeado, perfilado, hacer una cajera central y realizar taladros profundos en un bloque de dimensiones iniciales 120 x 120 x 50 mm.

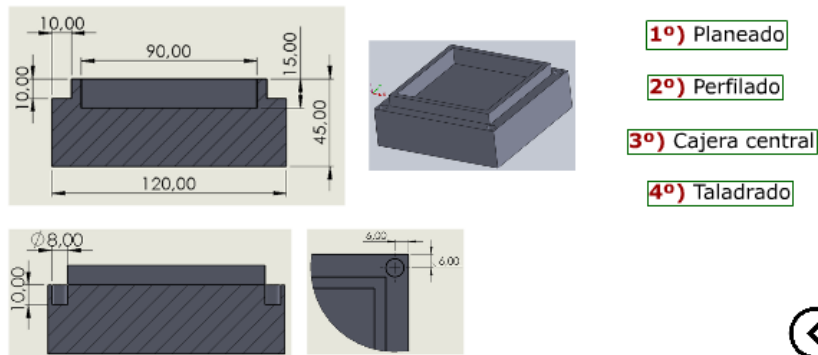


Figura 4-22. Ejemplo práctico 1

### 4.7.1.1 Planeado

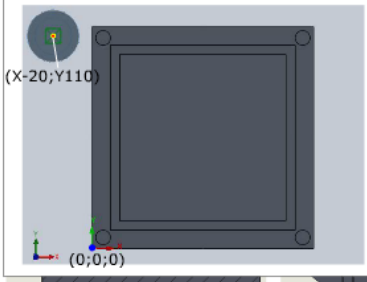
Cuando el alumno pulsa sobre el botón de planeado, se mostrará una ventana auxiliar que nos mostrará la ejecución del código en bloques, a la vez que el enunciado según el cual realizaremos nuestra programación.

El objetivo último de estos ejemplos es mostrar gráficamente cómo se mueve la herramienta y cómo el control transmite las instrucciones del código a la máquina. Es por ello, que cada línea de código tiene interactividad en nuestra aplicación, de modo que si el alumno pulsa sobre la línea "N70 X135", en la parte inferior izquierda de la pestaña se le mostrará el desplazamiento de la herramienta hasta ese punto. En la Figura 4-23 podemos ver la situación inicial de la herramienta, fuera del tocho y antes de comenzar el mecanizado.

### EJEMPLO PRÁCTICO I

El planeado consiste en crear una superficie de trabajo en el plano XY.

- Rebajar 5mm en programación incremental
- Calibramos como Z0 el plano superior del tocho
- Utilizaremos una fresa de 30mm de diámetro
- Velocidad de avance 100 mm/rev
- Velocidad angular 1000 rpm y giro a derechas



```

N10 T1 D1 M06
N20 G54
N30 G90 G95 S1000 M03
N40 G00 X-20 Y110 Z20
N50 G91 G00 Z2 M08
N60 G01 Z-7 F100
N70 X135
N80 Y-12
N90 X-135
N100 Y-12
N110 X135
N120 ; Repetir el mismo proceso desde N80 hasta N120 para completar todo el planeado
N130 G01 G90 Z20
N140 G00 X-30
N150 M30
  
```

Figura 4-23. Ejemplo práctico 1-Planeado

Para ampliar la información que se transmite desde la aplicación, no se muestra únicamente el movimiento de la herramienta vista desde el plano X/Y, cuando el alumno pulsa sobre la línea “N60 G01 Z-7 F100”, la vista pasa a ser desde el plano X/Z. De esta manera, se puede comprobar el viaje de la herramienta en el eje Z, algo que no se puede apreciar únicamente desde el plano X/Y (Figura 4-24).

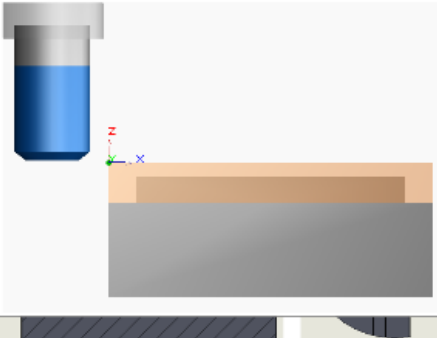
En el mundo de la programación CNC, no existe una única manera u orden de programar el mecanizado de una pieza concreta. Esto se debe a que el código nos presenta diferentes maneras de conseguir el mismo mecanizado. En el caso de nuestros ejercicios propuestos, vamos a seguir el mismo patrón de escritura en todos los ejercicios.

Primeramente, en la línea “N10” vamos a seleccionar nuestra herramienta de trabajo. Por ser el planeado el primer código que debemos escribir, vamos a seleccionar la herramienta T1 y el corrector D1. Adicionalmente debemos programar la función auxiliar M06, para que la selección y cambio de nuestra herramienta sea efectivo.

### EJEMPLO PRÁCTICO I

El planeado consiste en crear una superficie de trabajo en el plano XY.

- Rebajar 5mm en programación incremental
- Calibramos como Z0 el plano superior de la pieza
- Utilizaremos una fresa de 30mm de diámetro
- Velocidad de avance 100 mm/rev
- Velocidad angular 1000 rpm y giro a derechas



```

N10 T1 D1 M06
N20 G54
N30 G90 G95 S1000 M03
N40 G00 X-20 Y110 Z20
N50 G91 G00 Z2 M08
N60 G01 Z-7 F100
N70 X135
N80 Y-12
N90 X-135
N100 Y-12
N110 X135
N120 ; Repetir el mismo proceso
desde N80 hasta N120 para completar
todo el planeado
N130 G01 G90 Z20
N140 G00 X-30
N150 M30

```

Figura 4-24. Ejemplo práctico 1-Vista eje Z

En todos nuestros casos, en la línea “N20” estará programado el origen de coordenadas G54. Como se ha explicado en capítulos anteriores, este origen de coordenadas se debe establecer de manera manual desde la máquina.

Tal y como nos indica el enunciado, debemos utilizar la programación incremental, pero para realizar la primera aproximación de la herramienta al tocho, utilizaremos la programación absoluta “G90”, nuestra velocidad de avance (F) se debe establecer en mm/rev, por lo que debemos programar el código “G95”. Nos dicen que la velocidad angular es de 1000 rpm, por lo que escribimos S1000; y con el giro a derechas por lo tanto utilizamos la función auxiliar M03.

El siguiente paso que va a hacer estándar en todos nuestros ejercicios, será programar el acercamiento de la herramienta hasta el tocho. Para programar este movimiento siempre vamos a utilizar la función de desplazamiento rápido G00. Para evitar riesgos de colisión con el tocho tomaremos una distancia prudencial en este caso X-20 Y110 Z20.

Es en la siguiente línea de nuestra programación (N50) cuando vamos a activar la programación incremental (G91). Realizaremos un desplazamiento en el eje Z hasta alcanzar una altura de 2 mm por encima del tocho de partida. Activaremos el chorro refrigerante para disminuir el desgaste de la herramienta mediante la función M08.

A continuación, en la línea N80, cumpliremos con el objetivo inicial de rebajar 5 mm el tocho de partida. Es por ello que, partiendo de nuestra posición previa Z2, debemos bajar

hasta la posición Z-7 para conseguir esos 5 mm de rebaja que nos pide el enunciado. Por último, programaremos la velocidad de avance de 100 mm/rev.

Es a continuación de estos pasos iniciales obligatorios que debemos dar, cuando podemos pasar describir las coordenadas que realizará nuestra herramienta.

Para acabar la programación de nuestro planeado, debemos retirar la herramienta a una posición segura, es por eso que en las líneas N130 y N140 vamos a llevar primero la herramienta hasta la posición Z20 para a continuación salirnos hasta X-30.

Por último, todos los bloques se van a cerrar de la siguiente manera: utilizando la función auxiliar M30 que representa el fin de programa.

#### 4.7.1.2 Perfilado

Seguidamente el alumno puede cerrar la pestaña del planeado, y clicar sobre el perfilado. Se seguirá el mismo formato presentado anteriormente en el planeado, se presenta un enunciado donde nos pone las condiciones que debemos seguir a la hora de programar nuestro código. Para ilustrar mejor el recorrido de la herramienta, se muestran las coordenadas que nos van a ser necesarias para la programación (Figura 4-25).

El primer paso que debemos tomar para realizar el perfilado es cambiar la herramienta, en este caso utilizaremos la herramienta T2 y su corrector D2. Al igual que el planeado, primero vamos a seleccionar nuestro origen G54, que al no ser modificado será el mismo que en el planeado. Una de las diferencias más importantes que vamos a tener con respecto al planeado, es que el enunciado nos dice que debemos programar el avance en milímetros por minuto. Por lo tanto, utilizaremos la función G94.

En este caso vamos a programar un perfil por lo tanto el acabado debe ser preciso, Es por ello que necesitaremos utilizar las funciones compensatorias del radio de la herramienta. La pieza queda a la derecha de la herramienta por lo que necesitaremos utilizar la función G41.

## EJEMPLO PRÁCTICO I

El perfil de la pieza tiene unas dimensiones de 100x100x10, mecanizar bajo las siguientes condiciones en el plano XY:

- Herramienta de 10mm de diámetro y profundidad de pasada máxima son 4mm
- Coordenadas absolutas y giro a derechas
- Velocidad de avance 200mm/min
- Velocidad angular 2000 rev/min

```

N10 T2 D2 M06
N20 G54
N30 G90 G17 G94 S2000 M03
N40 G00 G41 X60 Y120 Z2
N50 G01 Z-8 F200 M08
N60 Y110
N70 X110
N80 Y10
N90 X10
N100 Y110
N110 X60
N120 Z-12
N130 ; Repetir de N60 a N100 hasta llegar a Z-15
N240 G40 G01 X61,5 Y 114
N250 G00 Z2 M09
N260 Z20
N270 M30

```

Realizar los pasos...

Figura 4-25. Ejemplo práctico1-Contorneado

Una vez que hayamos mecanizado nuestro perfil o contorno, debemos apagar la compensación del radio de la herramienta (G40). También, debemos apagar el refrigerante mediante la función M09.

### 4.7.1.3 Cajera central

Para mecanizar la cajera central, vamos a utilizar uno de los ciclos fijos previamente explicados. Utilizaremos el ciclo fijo “G87”, específicamente creado para mecanizar cajeras cuadradas o rectangulares.

Observamos que el punto de entrada de nuestra herramienta, será el centro geométrico de la pieza, es decir “X60 Y60” (Figura 4-26). Situaremos la cota del plano de referencia (Z) en -5, ya que previamente habremos realizado el planeado, y 5 mm es la cantidad de material que habremos eliminado del tocho inicial. Vamos a establecer que la profundidad de mecanizado en cada pasada (I) sea de 2 mm.

Las letras J y K representan la distancia del centro al borde la cajera según abscisas y ordenadas respectivamente, en nuestro caso, estos parámetros son idénticos al ser una cajera cuadrada, por lo tanto, serán de 45 mm.

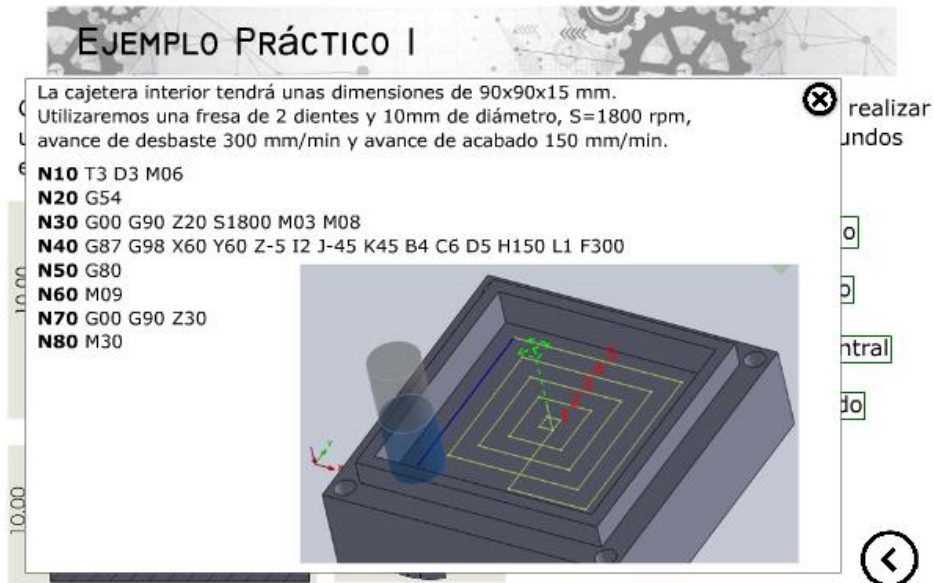


Figura 4-26. Ejemplo práctico 1-Cajera central

#### 4.7.1.4 Taladrado

Para el último apartado de este ejercicio, se pedirá realizar el taladrado de los cuatro agujeros de nuestra pieza. Como se ha explicado previamente, el ciclo fijo “G81” nos permite programar el taladrado de diferentes puntos. Sin embargo, para programar un taladrado no es condición necesaria programar siempre un ciclo fijo, también es posible realizar un taladrado mediante las funciones de régimen de trabajo (G01).

Es por ello, que, en nuestra herramienta educativa, se van a explicar las dos maneras de programar dicho taladrado. En la Figura 4-27, se va a diseñar, primeramente, la programación utilizando únicamente la función de desplazamiento rápido (G00) y la función de interpolación lineal en régimen de trabajo (G01) para mecanizar los agujeros.

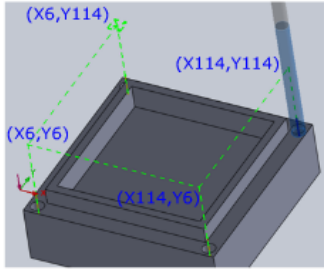
Esta forma de programación, implica escribir un código más largo a diferencia del ciclo fijo, pero en cambio, obtenemos un código mucho más sencillo de entender debido a que implica funciones muy básicas y bien conocidas.



## EJEMPLO PRÁCTICO I

Vamos a realizar 4 agujeros de diámetro 8mm y profundidad 10mm:

- Utilizar una broca de 8mm
- Velocidad de avance 200mm/min
- Velocidad angular 2000 rpm y giro a derechas



```

N10 T4 D4
N20 G54
N30 G00 Z20
N40 G00 G90 X6 Y6 S2000 M03 M08
N50 G01 Z-25 F200
N60 Z-10
N70 G00 X6 Y114
N80 G01 Z-25 F200
N90 Z-10
N100 G00 X114 Y114
N110 G01 Z-25 F200
N120 Z-10
N130 G00 X114 Y6
N140 G01 Z-25 F200
N150 Z-10
N160 M09
N170 G00 Z30
N180 M30
    
```

realizar  
undios  
jo  
o  
ntral  
do  
◀

Figura 4-27. Ejemplo Práctico 1-Taladrado

Con el objetivo de que el alumno pueda aprender y saber diferenciar ambas formas de programar un taladrado, se muestra cómo se debe programar mediante el ciclo fijo “G81”. Podemos ver en la **¡Error! No se encuentra el origen de la referencia.**, en la línea “N30”, e establecemos nuestras condiciones iniciales, es decir desde donde parte la herramienta, velocidad de giro de la herramienta y sentido del giro.

Seguidamente, procedemos a programar nuestro ciclo fijo “G81” (Figura 4-28): elegimos que la herramienta retroceda hasta el plano de partida (G98), en este caso Z0. A continuación programamos cuál será nuestro primer punto de taladrado, en este caso “X6 Y”. Situamos nuestro plano de referencia en Z0, inicialmente calibrado como el plano superior del tocho de partida, es por ello que, a la hora de programar nuestra profundidad de taladrado, debemos tener en cuenta la altura de tocho inicial. Entonces, nuestra profundidad de taladrado (I) será de -25. Finalmente, no programaremos un tiempo de espera (K) entre taladrado y taladrado, y por defecto será 0.

Para programar todos los agujeros de nuestra pieza, debemos indicar sus coordenadas a continuación del ciclo, como podemos ver en las líneas “N50”, “N60”, y “N70”.

Finalmente, para apagar el ciclo de taladrado, debemos utilizar la función “G80”.

## EJEMPLO PRÁCTICO I

Vamos a realizar 4 agujeros de diámetro 8mm y profundidad 10mm:

- Utilizar una broca de 8mm
- Velocidad de avance 200mm/min
- Velocidad angular 2000 rpm y giro a derechas

```
N10 T4 D4
N20 G54
N30 G90 G00 X0 Y0 Z20 S2000 M03 M08
N40 G81 G98 X6 Y6 Z0 I-25 K0 F200
N50 X114 Y6
N60 X114 Y114
N70 X6 Y114
N80 G80
N90 M09
N100 G00 Z30
N110 M30
```

realizar  
undos  
lo  
o  
ntral  
do  
◀

Figura 4-28. Ejemplo práctico 1-Taladrado mediante Ciclo Fijo

### 4.7.2 Ejemplo práctico 2

El segundo ejemplo práctico que se va a proponer, será un perfil de geometría sencilla, donde el alumno podrá poner en práctica programar sencillas interpolaciones lineales y circulares.

En la Figura 4-29, podemos observar que en el perfil propuesto ya vienen dadas todas las coordenadas necesarias para realizar nuestra programación.

Con este ejercicio, se pretende aumentar gradualmente la dificultad. Ya que, en el anterior ejercicio el perfil a programar era un sencillo cuadrado, donde la programación no exigía aplicar interpolaciones circulares, únicamente lineales.

La característica de este ejercicio es, que cada bloque de código se ha convertido en un botón. Cuando el alumno pulsa sobre cada bloque, podrá observar cómo el perímetro que conforma nuestro perfil, se ilumina en color rojo y señala con claridad la parte que se está mecanizando. El objetivo principal de este ejercicio es, que el alumno comprenda de manera clara y visual qué movimiento o acción produce cada línea de código, a la vez que aclarar hacer comprender al alumno cómo programar correctamente los parámetros “I” y “J” de las interpolaciones circulares.

## EJEMPLO PRÁCTICO 2

Pincha sobre las diferentes líneas de código:

```

N10 G55
N20 T01 D01 M06
N30 G90 G00 G41 X0 Y0 Z2 S2000 M03
N40 G01 Z0 F150
N50 X27,5 Y0
N60 G03 X57,5 Y0 I15 J0
N70 G01 X70 Y0 F150
N80 X70 Y-30
N90 G02 X60 Y-40 I-10 J0
N100 G01 X10 Y-40
N110 G02 X0 Y-30 I0 J10
N120 G01 X0 Y0
N130 G00 G40 Z20
N140 G00 G53 X0 Y0 Z0
N150 M30
    
```

Figura 4-29. Ejemplo práctico 2

El enunciado del ejercicio no indica si debemos programar en coordenadas absolutas o incrementales, es por ello que seleccionaremos programar en coordenadas absolutas (G90). Así mismo, el perfil que debemos mecanizar queda a la derecha de la herramienta, por lo que utilizaremos la compensación de radio “G41” para que únicamente debamos programar las coordenadas del perfil, y no realizar las operaciones de restar o sumar el radio de la herramienta para calcular el perfil.

El primer arco que debemos calcular es la semicircunferencia que se encuentra entre los puntos “P1” y “P2”. Sabiendo las coordenadas de cada uno de estos puntos, la dificultad radica en obtener los parámetros “I” y “J”. Teniendo en cuenta que estos parámetros se calculan a partir del primer punto del arco (en este caso P1), los parámetros de nuestro perfil quedarán de la siguiente manera (Tabla 3):

Tabla 3. Parámetros I,J

Arco	I	J
P1-P2	15	0
P4-P5	-10	0
P6-P7	0	10

Podremos obtener los signos de cada parámetro, colocando un eje de abscisas y ordenadas imaginario sobre cada punto inicial. De la siguiente manera (Figura 4-30):

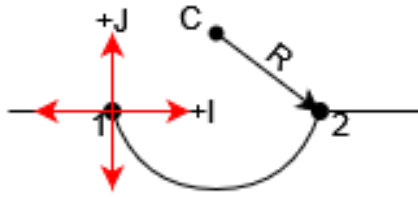


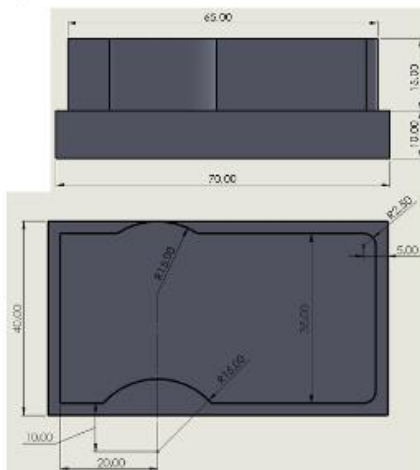
Figura 4-30. Parámetros I y J

### 4.7.3 Ejemplo práctico 3

Dado que en los dos ejemplos prácticos anteriores se ha abordado la programación de una figura completamente rectangular (Ejemplo práctico 1) y de una figura con bordes redondeados sencillos (Ejemplo práctico 2), en este ejercicio (Figura 4-31), se pretende plantear realizar la programación de unas geometrías de mayor complejidad, donde necesitaremos de cálculos adicionales para obtener todas las coordenadas que conforman el perfil.

#### EJEMPLO PRÁCTICO 3

En este segundo ejemplo práctico se va a proponer mecanizar la siguiente pieza a partir de un tocho de 70x40x30 mm.



El orden de operaciones será el siguiente:

1º Planeado

2º Contorneado

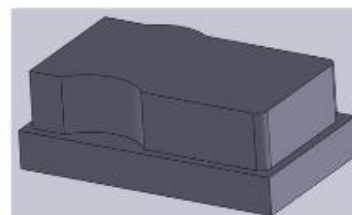


Figura 4-31. Ejemplo práctico 3

#### 4.7.3.1 Planeado

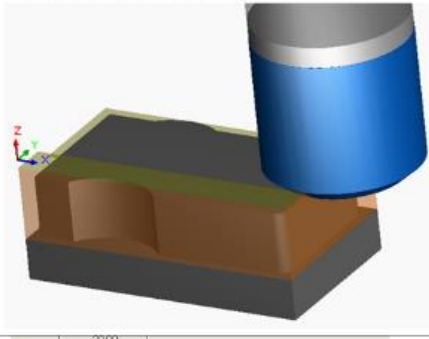
El planeado planteado para esta pieza, se puede realizar únicamente con dos pasadas si utilizamos una herramienta lo suficientemente grande, por ello utilizaremos una fresa de

30 mm de diámetro para simplificar la programación a la vez que acortar notablemente el tiempo de mecanizado.

**EJEMPLO PRÁCTICO 3**

Con profundidad de pasada máxima de 3mm, en el plano XY rebajar 5mm el tocho inicial:

- Utilizar una fresa de 30mm y giro a derechas
- Calibramos como Z0 el plano superior del tocho de partida
- Velocidad de avance de 150 mm/min
- Velocidad angular 2500 rev/min



```

N10 T1 D1 M06
N20 G54
N30 G90 G17 G94 S2500 M03
N40 G00 X-30 Y28 Z3
N50 G01 Z-2 F150 M08
N60 X90 Y28
N70 X90 Y10
N80 X-30 Y10
N90 G00 Z3
N100 X-30 Y28
N110 G01 Z-5
N120 ; Repetir los mismos parámetros de N60 a N110
N130 G00 Z20 M09
N140 M30
    
```

Figura 4-32. Ejemplo práctico 3-Planeado

Como podemos observar en la Figura 4-32, durante la programación del bloque, se ha seguido el planteamiento estándar que establecimos en el Ejemplo práctico 1. En este caso, el enunciado nos indica que la profundidad máxima de pasada es de 3 mm, por lo tanto, serán necesarias 2 pasadas hasta alcanzar la cota de “Z-5”, y así nos lo indica la línea “N130”, ordenando repetir los parámetros desde N60 a N110.

#### 4.7.3.2 Contorneado

Para ejecutar precisamente el contorneado de la pieza, necesitaremos calcular las coordenadas de los arcos. Según nuestros planos de partida, que podemos ver en la Figura 4-31, podemos obtener todas las coordenadas necesarias para programar nuestro contorneado.

Situaremos el punto de inicio de nuestra herramienta en la coordenada “X2,5 Y2,5”. Previamente, habremos activado la compensación del radio de la herramienta, para así poder programar con precisión todo nuestro perfil. Dado que previamente en el planeado hemos rebajado la altura del tocho inicial en 5 mm, y en este anuncio nos dicen que la profundidad máxima de pasada son 5 mm, deberemos programar “Z-10”.

En este caso, el enunciado no nos indica qué tipo de programación debemos acometer (incremental o absoluta), por lo que somos libres de escoger el que nosotros deseamos, en este caso utilizaremos la programación absoluta (G90).

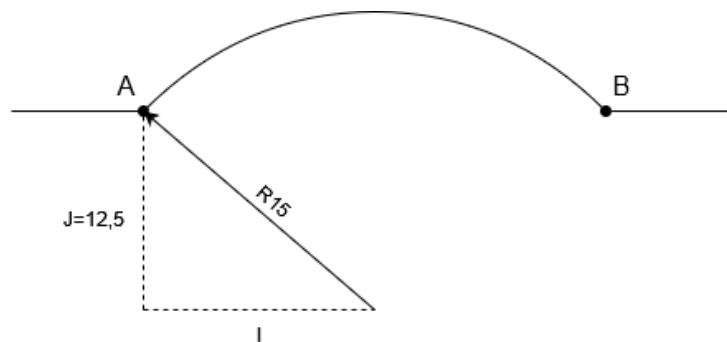
En la línea “N70”, es cuando deberemos calcular el punto inicial del primer arco de circunferencia que nos encontramos. El arco tiene un radio de 15 mm, y en el eje de abscisas se desplaza 20 mm con respecto a la pared de nuestro perfil. En el eje de ordenadas, realizando una sencilla resta, podemos obtener en centro del arco. Éste se halla a una distancia de 12,5 mm con respecto al perfil que queremos mecanizar.

Es ahora, cuándo debemos calcular los parámetros I y J, imprescindibles para programar la primera interpolación circular que nos encontramos. En este caso, será una interpolación circular en sentido horario (G02).

En la Figura 4-33 , se muestra la interpolación circular que deberemos calcular. En concreto, los puntos A y B. Para ello, deberemos utilizar la Ecuación 1 para obtener el parámetro “I”:

*Ecuación 1. Cálculo del parámetro I*

$$I = \sqrt{R^2 - J^2} = \sqrt{15^2 - 12,5^2} = 8,29$$



*Figura 4-33. Cálculo de las coordenadas A y B*

Una vez obtenido el parámetro “I”, disponemos de toda la información necesaria para obtener nuestros puntos “A” y “B” (Tabla 4):

*Tabla 4. Coordenadas Ejemplo Práctico 3*

Puntos	X	Y
A	14,21	37,5
B	31,21	37,5

Como podemos en la línea “N80” (Figura 4-34), vamos a programar nuestra interpolación circular: las coordenadas que debemos programar son las del punto final del arco, es decir, el punto “B”. Seguidamente y para completar toda la información que necesita nuestra función, debemos añadir los parámetros “I” y “J”. Como se ha explicado previamente, estos parámetros tienen signo, por lo tanto debemos ser consecuentes y programar “J-12,5”.

**EJEMPLO PRÁCTICO 3**

El contorno de la pieza sobresale 15mm, por lo que tendremos las siguientes condiciones:

- Utilizar una fresa de 8mm y giro a derechas
- Profundidad máxima de pasada 5mm
- Velocidad de avance de 125 mm/min
- Velocidad angular 3000 rev/min

las siguientes condiciones:

```

N10 T2 D2 M06
N20 G54
N30 G90 G17 G94 S3000 M03
N40 G00 G41 X2,5 Y2,5 Z3
N50 G01 Z-10 F125 M08
N60 X2,5 Y37,5
N70 X14,21 Y37,5
N80 G02 X31,21 Y37,5 I8,29 J-12,5
N90 G01 X62,5 Y37,5
N100 G02 X67,5 Y35 I0 J-2,5
N110 G01 X67,5 Y7,5
N120 G02 X65 Y2,5 I-2,5 J0
N130 G01 X31,32 Y2,5
N140 G03 X11,32 Y2,5 I-11,18 J-10
N150 G01 X2,5 Y2,5
N160 Z-15
N170 ; Repetir de N60 hasta N140
para llegar a Z-20
N180 G40 G00 Z20 M09
N190 M30
                
```

Figura 4-34. Ejemplo práctico 3-Contorneado

De la misma manera con la que hemos calculado nuestra primera interpolación en la línea “N80” podremos hallar las coordenadas y parámetros de las siguientes interpolaciones. En este caso, debemos ser cuidadosos en la en línea “N140”, ya que es una interpolación circular de sentido anti horario (G03).

## 4.8 Autoevaluación

Para completar nuestra plataforma educativa, se van a proponer una serie de ejercicios de autoevaluación (en concreto 7), donde el alumno podrá poner en práctica todos los conocimientos adquiridos durante la lección. En algunos ejercicios, junto con la opción correcta, se va a añadir una información adicional de por qué esa opción es correcta.

El orden de dificultad planteado, es progresivo. Empezando por los ejercicios más sencillos y terminando con ejercicios que exigen poner en práctica varios de los conocimientos aprendidos a la vez.

## 4.8.1 Ejercicio 1

Este primer ejercicio consistirá, en responder correctamente a un ejercicio tipo test de tres opciones, donde sólo una de ellas será la correcta (Figura 4-35).

El enunciado del ejercicio dice lo siguiente: dado el siguiente diagrama: ¿cómo se escribiría el código en coordenadas incrementales (G91)?

**AUTOEVALUACIÓN** ➔ siguiente

**Ejercicio 1**  
Dado el siguiente diagrama: cómo se escribiría el código en coordenadas incrementales (G91)?

a)	b)	c)
G00 X20 Y10	G00 X20 Y10	G00 X20 Y10
G01 X40 Y0	G01 X60 Y10	G01 X40 Y0
X20 Y20	X80 Y30	X20 Y20
X-20 Y20	X60 Y50	X-20 Y20
X-40 Y0	X20 Y50	X-40 Y0
X0 Y-10	X20 Y40	X20 Y50
X10 Y0	X30 Y20	X20 Y40
X0 Y-20	X30 Y20	X30 Y20
X-10 Y0	X20 Y20	X-10 Y0
X0 Y-10	X10 Y20	X0 Y-10
G00 X-20 Y-10	G00 X0 Y0	G00 X0 Y0

⏪

Figura 4-35. Ejercicio 1

Como podemos observar en la Figura 4-33, la única opción que cumple los requisitos del enunciado es la opción "a)". Las opciones "b)" y "c)" presentan diferentes errores que no responden correctamente a lo que nos pide el enunciado. Podemos observar que la opción "b)" realiza una programación en coordenadas absolutas, y que la opción "c)" mezcla errores de programación en coordenadas absolutas y en coordenadas incrementales.

Tal y como se ha explicado al principio de este capítulo, el alumno debe poder comprobar si su respuesta es correcta o no. Por lo tanto, al pulsar la opción a la aplicación mostrará que ha respondido de manera correcta. Tal y como se puede ver en la Figura 4-36.

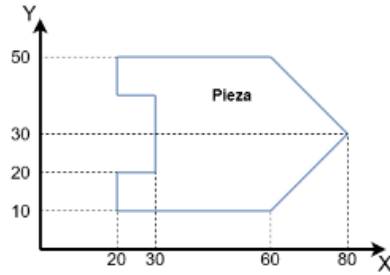


**AUTOEVALUACIÓN** siguiente

**Ejercicio 1**

Dado el siguiente diagrama: cómo se escribiría el código en coordenadas incrementales (G91)?

a)	b)	c)
G00 X20 Y10	G00 X20 Y10	G00 X20 Y10
G01 X40 Y0	G01 X60 Y10	G01 X40 Y0
X20 Y20	X80 Y30	X20 Y20
X-20 Y20	X60 Y50	X-20 Y20
X-40 Y0	X20 Y50	X-40 Y0
X0 Y-10	X20 Y40	X20 Y50
X10 Y0	X30 Y20	X20 Y40
X0 Y-20	X30 Y20	X30 Y20
X-10 Y0	X20 Y20	X-10 Y0
X0 Y-10	X10 Y20	X0 Y-10
G00 X-20 Y-10	G00 X0 Y0	G00 X0 Y0



**CORRECTO!**



*Figura 4-36. Ejercicio 1-Opción correcta*

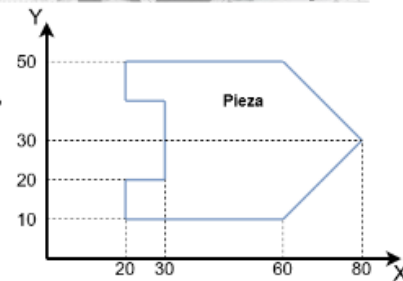
Por el contrario, si el alumno no sabe la respuesta correcta a la pregunta, y selecciona la opción “b)” o “c)”, entonces la aplicación le mostrará que su respuesta es incorrecta como podemos ver en la Figura 4-37.

**AUTOEVALUACIÓN** siguiente

**Ejercicio 1**

Dado el siguiente diagrama: cómo se escribiría el código en coordenadas incrementales (G91)?

a)	b)	c)
G00 X20 Y10	G00 X20 Y10	G00 X20 Y10
G01 X40 Y0	G01 X60 Y10	G01 X40 Y0
X20 Y20	X80 Y30	X20 Y20
X-20 Y20	X60 Y50	X-20 Y20
X-40 Y0	X20 Y50	X-40 Y0
X0 Y-10	X20 Y40	X20 Y50
X10 Y0	X30 Y20	X20 Y40
X0 Y-20	X30 Y20	X30 Y20
X-10 Y0	X20 Y20	X-10 Y0
X0 Y-10	X10 Y20	X0 Y-10
G00 X-20 Y-10	G00 X0 Y0	G00 X0 Y0



**INCORRECTO**



*Figura 4-37. Ejercicio 1-Opción incorrecta*

## 4.8.2 Ejercicio 2

Este segundo ejercicio, sigue el mismo patrón que el Ejercicio 1. El enunciado, pide describir el perfil que se muestra en la imagen adjunta (Figura 4-38). En esta ocasión se pide que la programación sea en coordenadas absolutas (G90).

La dificultad añadida de este ejercicio está, en saber distinguir correctamente de entre las tres opciones presentadas, en cuál se han programado de manera correcta las coordenadas absolutas y las interpolaciones circulares pertinentes.

AUTOEVALUACIÓN

←
→  
anterior siguiente

**Ejercicio 2**  
 Dado el siguiente diagrama: cómo se escribiría el código en coordenadas absolutas (G90)?

a)

```
G00 X10 Y10
G01 X50 Y10
G03 X60 Y20 I10 J0
G01 X60 Y40
X20 Y40
G02 X10 Y30 I10 J10
G01 X10 Y10
G00 X0 Y0
```

b)

```
G00 X10 Y10
G01 X50 Y10
G03 X60 Y20 I0 J10
G01 X60 Y40
X20 Y40
G03 X10 Y30 I0 J-10
G01 X10 Y10
G00 X0 Y0
```

c)

```
G00 X10 Y10
G01 X50 Y0
G02 X60 Y20 I0 J10
G01 X60 Y40
X0 Y40
G03 X10 Y30 I0 J10
G01 X10 Y10
G00 X0 Y0
```

←

Figura 4-38. Ejercicio 2

Como podemos observar en la opción “a)” se ha realizado de manera adecuada la programación en coordenadas absolutas, pero se ha cometido un error en la sexta línea, ya que como vemos se ha programado el código “G02”, cuando debería ser “G03” al ser un giro antihorario. Además, en la misma línea se ha cometido un segundo error, dado que el parámetro “J10” es erróneo, puesto que debería ser “J-10”.

La opción “b)”, dispone de marea correcta todos los códigos necesarios. La programación está en coordenadas absolutas, así como se ha utilizado adecuadamente la interpolación circular “G03” a la vez que los parámetros “I” y “J” son los correctos.

Por último, la opción “c)” es la segunda respuesta incorrecta. En la línea tres, se programa erróneamente el código “G02” en lugar de ya mencionado “G03”. A continuación, en la línea número cinco, las coordenadas “X0 Y40” no son verdaderas, debiendo ser éstas

“X20 Y40”. Y para acabar, al igual que ocurre en la opción “a)”, el parámetro “J10” es incorrecto.

### 4.8.3 Ejercicio 3

La finalidad del ejercicio número tres es, indicar qué funciones de compensación del radio de la herramienta son las correctas. Se presentan diferentes herramientas, y en cada una de ellas se indica la trayectoria (el sentido) que siguen. El alumno debe indicar si dicha herramienta, su trayectoria, y la compensación indicada son verdaderas o falsas (Figura 4-39).

La primera pregunta, la herramienta T01 sigue la trayectoria entre los puntos 1 y 2. La compensación que debería acompañar a esta herramienta es la “G41”. Por lo tanto, podemos indicar que la primera pregunta es “Verdadero”.

Con la segunda pregunta, seguimos trabajando el perfil exterior de la pieza presentada, y con el mismo sentido de trabajo (G41). Pero en este caso, se indica que la compensación que debemos utilizar es “G42”, por lo tanto, esta opción es “Falso”.

**AUTOEVALUACIÓN** ← anterior siguiente →

**Ejercicio 3**  
Indica qué funciones de compensación son las correctas:

- 1) Herramienta T01; Trayectoria 1-2, G41  
Verdadero Falso
- 2) Herramienta T02; Trayectoria 2-3, G42  
Verdadero Falso
- 3) Herramienta T03; Trayectoria 3-4, G40  
Verdadero Falso
- 4) Herramienta T04; Trayectoria 4-1, G42  
Verdadero Falso
- 5) Herramienta T05; Trayectoria 12-5, G41  
Verdadero Falso
- 6) Herramienta T06; Trayectoria 8-9, G42  
Verdadero Falso

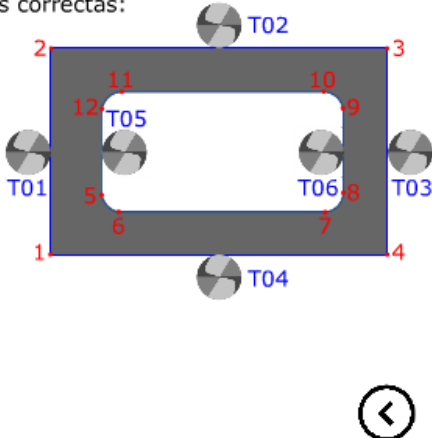


Figura 4-39. Ejercicio 3

En la pregunta número 3, podemos distinguir a simple vista que la opción que debemos marcar es “Falso”, ya que el código “G40”, sirve para apaga o anula las funciones de compensación del radio de herramienta.

La pregunta número 4, el alumno también deberá marcar como correcta la opción “Falso”, ya que se comete el mismo error que en la pregunta 2, la compensación que correspondería programar sería “G41”, y no “G42”.

En la pregunta 5, pasamos a trabajar en el perfil interior de la pieza mostrada. En este caso, seguiremos la trayectoria 12-5, es decir, el punto inicial de la herramienta es el punto 12, y el final, el 15. En este caso corresponde programar la función “G41”, ya que la pieza mecanizada queda a la derecha de la herramienta. Por lo tanto, la respuesta es “Verdadero”.

Por último, en la pregunta 6, la herramienta “T06” seguirá la trayectoria ascendente desde el punto 8 hasta el punto 9, por lo que, el alumno deberá marcar la opción “Falso”, debido a que no procede utilizar el código “G42”, sino “G41”, dado que como pasa en la pregunta anterior, el tocho queda a la derecha de la herramienta.

#### 4.8.4 Ejercicio 4

Durante los ejercicios 4 y 5, se seguirá utilizando el mismo método de preguntas “test” y tres opciones. En este caso, se realizarán preguntas teóricas para comprobar si el alumno ha seguido completamente las lecciones y ha comprendido todo el contenido impartido en ellas.

La pregunta que realiza este ejercicio es: mediante las funciones M, ¿es posible definir el perfil de una pieza? (Figura 4-40)



##### **Ejercicio 4**

Mediante las funciones M, ¿es posible definir el perfil de una pieza?

- a) Depende de la función M
- b) Sí
- c) No

##### **Ejercicio 5**

Indica cuál es el bloque correcto:

- a) N30 G01 X15 Y-10 F200 S1000
- b) N30 G01 Y-10 X15 S1000 F200
- c) M30 G01 X30 Y23



Figura 4-40. Ejercicio 4

La única respuesta que cabría dar en este caso es, que las funciones auxiliares M, se utilizan controlar los diferentes estados de los que dispone la máquina. Como puede ser, definir el sentido de los giros del cabezal, programar paradas de programa, o realizar cambios de herramienta, entre otras muchas funciones ya explicadas con anterioridad en capítulos previos.

**AUTOEVALUACIÓN** anterior siguiente

**Ejercicio 4**  
Mediante las funciones M, ¿es posible definir el perfil de una pieza?

a) Depende de la función M  
b) Sí  
c) No

**CORRECTO!**  
Las funciones auxiliares se utilizan para definir el funcionamiento de la máquina y no la geometría.

**Ejercicio 5**  
Indica cuál es el bloque correcto:

a) N30 G01 X15 Y-10 F200 S1000  
b) N30 G01 Y-10 X15 S1000 F200  
c) M30 G01 X30 Y23

Figura 4-41. Ejercicio 4-Opción correcta

Por lo tanto, cuando el alumno marque la respuesta correcta “c) No”, aparecerá un mensaje indicando que se ha señalado la respuesta correcta (Figura 4-41) y adicionalmente se ha añadido un razonamiento de por qué esa respuesta es la correcta.

#### 4.8.5 Ejercicio 5

El ejercicio 5, sigue el mismo patrón que el Ejercicio 4. En este caso, la pregunta que nos plantea es: indicar cuál es el bloque correcto.

El objetivo de este ejercicio, es que el alumno haya comprendido cómo funciona la estructura de programación de un bloque. Esta lección es la introductoria, debido a la importancia que posee entender el orden correcto que debe seguir cada código.

Esto es especialmente relevante, porque el código de programación FANUC, exige seguir este orden preestablecido en todo momento.

Como podemos observar en la Figura 4-42, el único bloque que cumple con las exigencias es la opción “a) N30 G01 X15 Y-10 F200 S1000”, por lo tanto, el alumno la debe señalar como la opción correcta.

**AUTOEVALUACIÓN**

**Ejercicio 4**  
Mediante las funciones M, ¿es posible definir el perfil de una pieza?

a) Depende de la función M  
b) Sí  
c) No

**Ejercicio 5**  
Indica cuál es el bloque correcto:

a) N30 G01 X15 Y-10 F200 S1000  
b) N30 G01 Y-10 X15 S1000 F200  
c) M30 G01 X30 Y23

**CORRECTO!**  
Ya que sigue el orden  
N\_G\_X\_Y\_Z\_F\_S\_T\_D\_M

Figura 4-42. Ejercicio 5

#### 4.8.6 Ejercicio 6

En el sexto ejercicio, nos piden señalar el bloque correcto dadas unas condiciones previas. Estas condiciones son programar una velocidad de giro de 2000 rpm y con giro de cabezal en sentido horario, programar un avance de 250 milímetros por minuto, y también es condición necesaria utilizar el refrigerante a chorro.

**AUTOEVALUACIÓN**

**Ejercicio 6**  
¿Cuál es el bloque correcto si queremos mecanizar una pieza bajo las siguientes condiciones?

- Velocidad de giro: 2000 rpm y giro a derechas
- Avance: 250 mm/min
- Usamos el refrigerante

a) N10 G94 F2000 S250 M04  
b) N20 G94 F250 S2000 M03 M08  
c) N30 G95 F250 S2000 M03 M08  
d) N40 G95 S2000 F250 M04 M08  
e) N50 G95 F250 S2000 M03

Figura 4-43. Ejercicio 6

Dadas las siguientes cinco opciones (Figura 4-43), podremos descartar la opción “a)” ya que el código “M04”, indica programar el sentido de giro del cabezal en sentido antihorario, es decir “a izquierdas”.

La opción “b)” sería en este caso, la correcta. Dado que se utiliza el código “G94” para señalar que el avance se programa en milímetros por minuto, se respeta el orden necesario que exige el código FANUC, se emplea “M03” para indicar que el sentido de giro del cabezal es en sentido horario, y también se enciende el chorro refrigerante mediante “M08”.

Por lo tanto, las siguientes tres opciones, serán incorrectas. Vemos que en todas ellas de inicio se programa el código “G95”, que indica que la función de avance (F) será en milímetros por revolución, algo que las condiciones iniciales del ejercicio no piden. Adicionalmente podemos ver, que en la opción “d)” se utiliza de nuevo el código “M04”, que no se corresponde con nuestras exigencias, y que en la opción “e)” falta por programar el refrigerante a chorro “M08”.

#### 4.8.7 Ejercicio 7

Por último, el ejercicio número 7 nos plantea programar correctamente una interpolación circular, desde un punto A hasta un punto B. Este ejercicio, por ser el último, pretende ser el de mayor dificultad, ya que exige que el alumno deba realizar calculos para obtener todos los parámetros necesarios para hallar la respuesta correcta.

Como ya se ha demostrado anteriormente en los Ejemplos prácticos, los cálculos para obtener los parámetros de una interpolación circular son los siguientes (Ecuación 2):

*Ecuación 2. Parámetro I-Ejercicio 7*

$$I = \sqrt{R^2 - J^2} = \sqrt{12^2 - 5^2} = 10.909$$

Por ello, y siguiendo las indicaciones de la Figura 4-30. Parámetros I y J, podremos obtener los signos de nuestros parámetros “I” y “J” (Tabla 5) si colocamos los ejes imaginarios en el punto A.

Tabla 5. Ejercicio 6-Coordenadas

Puntos	X	Y
A	54,091	60
B	77	55

Los parámetros “I” y “J” quedarían de la siguiente manera (Tabla 6):

Tabla 6. Ejercicio 7-Parámetros I y J

Arco	I	J
A-B	10,909	-5

Es por ello, como podemos observar, la única opción que representa fielmente todos los cálculos realizados, es la opción “b)” (Figura 4-44).



**Ejercicio 7**

Dada la siguiente interpolación circular, selecciona el bloque correcto desde A a B:

- a) G02 X77 Y55 I5 J10,909
- b) G03 X77 Y55 I10,909 J-5
- c) G02 X77 Y55 I12 J5
- d) G03 X75,908 Y55 I10,909 J5

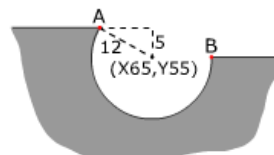


Figura 4-44. Ejercicio 7

En el resto de opciones, hay errores de concepción básicos, que el alumno debe saber distinguir perfectamente. Como pueden ser, confundir el código “G03” con el “G02” como ocurre en las opciones “a)” y “c)”, equivocarse con los parámetros “I” y “J”, como puede ser la opción “c)”, o programar erróneamente las coordenadas finales del punto “B” tal y como ocurre en la opción “d)”.



## 5 Conclusiones

Durante los últimos años se han producido profundos cambios en la forma de concebir la docencia universitaria. Es aquí donde la innovación docente se ha convertido en el tema de referencia en los círculos docentes, pero innovar no significa hacer sólo cosas distintas, sino cosas mejores. Es por eso que innovar en la docencia se ha convertido en una exigencia institucional. Una manera de hacer innovación universitaria es a través de Internet y de las distintas aplicaciones interactivas, que nos permiten mostrar de una manera más didáctica con ejemplos prácticos y figuras animadas.

Este trabajo se ha centrado en profundizar en el mundo de las Tecnologías de la información y comunicación (TICs), con la intención última de transmitir todos los conocimientos y lecciones a través de Internet. La premisa sobre la que se basa esta modalidad es que el aprendizaje y la educación, puede generarse en cualquier momento, en cualquier lugar, dentro y fuera de las instituciones educativas. Para ello, se han propuesto ejercicios que tratan de explicar de la manera más didáctica posible en qué consiste la programación básica CNC, y entender cómo funciona la estructura de un programa. Además, se han propuesto ejercicios de autoevaluación con el fin de que el alumno pueda poner a prueba los conocimientos que ha adquirido a lo largo de toda la lección.

Por lo tanto, esta modalidad educativa está centrada en la motivación del alumno con el apoyo de las tecnologías de la información y facilitando el aprendizaje autónomo, el estudiante pueda gestionar su aprendizaje en función de sus propias necesidades.

El objetivo principal de este trabajo es la creación de una herramienta o aplicación educativa que trata sobre el tema de “Control Numérico” de la asignatura de Tecnología Mecánica del grado de Ingeniería Mecánica. Se han enseñado y explicado todos los pasos necesarios para su creación mediante el programa Adobe Animate. Se han mostrado todas las funcionalidades y opciones del programa, así como el código JavaScript necesario para realizar toda la programación sin la cual, nuestra aplicación estaría carente de interactividad y posibilidad de interacción.

La situación actual en la que se encuentra la educación universitaria tanto en España como en Europa, por complicaciones provocadas por el virus Covid-19, y donde en

ocasiones se alterna la educación presencial junto con la educación telemática, son precisamente herramientas así las que pueden servir de instrumento indispensable para que el profesor pueda transmitir adecuadamente todos los conocimientos que considera necesarios.

A nivel personal, considero que la aplicación desarrollada puede ser de gran utilidad para que futuros estudiantes de Tecnología Mecánica puedan profundizar sus conocimientos en Control Numérico, y puedan entender y programar la ejecución de un mecanizado básico en una máquina CNC. Se ha hecho un esfuerzo en que todos los códigos se hayan explicado de la manera más didáctica posible y dentro de las limitaciones que ofrece el programa Adobe Animate.

Asimismo, estoy contento con el trabajo personal realizado, ya que durante la realización de esta aplicación no sólo he utilizado y afianzado mis conocimientos en Control Numérico, sino que también he ampliado mis conocimientos básicos sobre programación JavaScript y desarrollo de páginas web. Adicionalmente, he necesitado utilizar otros programas como pueden ser AutoCAD o SolidWorks para diseñar las piezas que se ponen de ejemplo en la aplicación, con los cuales he adquirido más soltura y he aprendido nuevas funciones que desconocía. Todos los manuales y tutoriales que he consultado han requerido emplear bastante tiempo y esfuerzo, pero personalmente considero que este tiempo ha sido bien invertido y que todos los conocimientos que he adquirido durante este proceso me van a ayudar en mi futuro profesional.

## 6 Referencias

- [1] “4 puntos básicos sobre la evolución de la docencia en el último siglo”  
<https://www.realinfluencers.es/2018/11/21/4-puntos-basicos-sobre-la-evolucion-de-la-docencia-en-el-ultimo-siglo/>
- [2] “Cambios en la docencia. Ordenadores, aliados o enemigos”  
[http://educativa.catedu.es/44700165/aula/archivos/repositorio/0/147/html/cambios\\_en\\_la\\_docencia.html](http://educativa.catedu.es/44700165/aula/archivos/repositorio/0/147/html/cambios_en_la_docencia.html)
- [3] “Innovación universitaria, Ignacio Sánchez”  
<https://www.latercera.com/opinion/noticia/innovacion-universitaria/706732/>
- [4] “Control Numérico e historia”  
[https://es.wikipedia.org/wiki/Control\\_num%C3%A9rico](https://es.wikipedia.org/wiki/Control_num%C3%A9rico)
- [5] “Qué es el mecanizado CNC, definición y características”  
<https://umesal.com/que-es-el-mecanizado-cnc/>
- [6] “Introducción a la tecnología CNC”  
<https://www.demaquinasyherramientas.com/mecanizado/introduccion-a-la-tecnologia-cnc>
- [7] Fagor CNC 8050M, “Manual de Operación”, Ref 9701 (cas)
- [8] “Adobe Animate CC”  
[https://es.wikipedia.org/wiki/Adobe\\_Animate\\_CC#Adobe\\_Animate\\_y\\_la\\_animaci%C3%B3n\[4\]E2%80%8B](https://es.wikipedia.org/wiki/Adobe_Animate_CC#Adobe_Animate_y_la_animaci%C3%B3n[4]E2%80%8B)
- [9] “Tutorial en línea” - Adobe Animate  
<https://helpx.adobe.com/es/animate/user-guide.html/es/animate/using/welcome.ug.html>
- [10] “Compilador”  
[https://es.wikipedia.org/wiki/Compilador#Proceso\\_de\\_compilaci%C3%B3n](https://es.wikipedia.org/wiki/Compilador#Proceso_de_compilaci%C3%B3n)
- [11] “Tecnologías de la información y la comunicación”  
[https://es.wikipedia.org/wiki/Tecnolog%C3%ADas\\_de\\_la\\_informaci%C3%B3n\\_y\\_la\\_comunicaci%C3%B3n#Educa%C3%B3n](https://es.wikipedia.org/wiki/Tecnolog%C3%ADas_de_la_informaci%C3%B3n_y_la_comunicaci%C3%B3n#Educa%C3%B3n)
- [12] [https://jmp technological.com/manual\\_cnc/funcion\\_G91.php](https://jmp technological.com/manual_cnc/funcion_G91.php)

- [13] [https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/PPFM/MCN/MCN03/es\\_PPFM\\_MCN03\\_Contenidos/website\\_111\\_estructura\\_de\\_un\\_programa\\_cnc.html](https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/PPFM/MCN/MCN03/es_PPFM_MCN03_Contenidos/website_111_estructura_de_un_programa_cnc.html)
- [14] <http://r-luis.xbot.es/cnc/codes01.html>