

Universidad de Holguín “Oscar Lucero Moya”

Facultad de Ingeniería

Centro de Estudios CAD/CAM



DESARROLLO DE UN SISTEMA DE SIMULACIÓN VIRTUAL DE MAQUINADO SOBRE ENTORNOS WEB.

Tesis presentada en opción al Título Académico de Máster en
Diseño y Fabricación Asistido por Computadora para la Rama
Metal - Mecánica

Olben Falcó Salcines.

Holguín, 2007.

Universidad de Holguín “Oscar Lucero Moya”

Facultad de Ingeniería

Centro de Estudios CAD/CAM



DESARROLLO DE UN SISTEMA DE SIMULACIÓN VIRTUAL DE MAQUINADO SOBRE ENTORNOS WEB.

Tesis presentada en opción al Título Académico de Máster en
Diseño y Fabricación Asistido por Computadora para la Rama
Metal - Mecánica

Autor: Ing. Olben Falcó Salcines.

Tutor: P.T. Ricardo Lorenzo Avila Rondón, DrC.

Holguín, 2007.

DEDICATORIA

“A mi Padre y a mis hermanos.”

AGRADECIMIENTOS

A mi Padre y a mis hermanos. A mis amigos y a mis enemigos.

SÍNTESIS.

La presente investigación trata el problema de la simulación del proceso de maquinado en piezas rotacionales sobre entornos Web. Para ello parte de los modelos de piezas importados de los sistemas CAD y de la interpretación de los programas CNC introducidos por usuarios remotos. Enmarcado en un entorno de laboratorio virtual, se implementa el análisis y tratamiento de la gráfica sobre Web, la interacción remota con Bases de Datos de Máquinas–Herramienta, y la integración con otros sistemas CAD/CAPP/CAM. La lectura y representación de ficheros de intercambio (DXF, IGES, STEP), así como la descripción de tecnologías Web existentes orientadas a la manufactura remota sobre Web, son aspectos importantes a tratar. En la investigación se muestran los avances y limitantes principales asociados al tema en cuestión, para la explotación de estos sistemas, orientados principalmente a las pequeñas y medianas empresas (SMEs).

Como resultado concreto se desarrolló un simulador que asiste al módulo CAPP y permite la simulación geométrica del maquinado basado en tecnología Applets. El sistema implementado muestra las trayectorias de las herramientas así como la representación de los modelos pieza en 3D. La verificación de la gramática de los programas pieza y la posibilidad de edición por parte de los usuarios son temas descritos. La aplicación desarrolla una estructura básica para el análisis léxico, sintáctico y semántico de los programas pieza a simular. El desarrollo del sistema se basa en el empleo de las técnicas de programación orientada a objeto (OOP) en Java, y la utilización de bibliotecas gráficas GL4Java.

INDICE

INTRODUCCIÓN.....	7
CAPÍTULO 1 ESTADO ACTUAL Y DESARROLLO DE LOS SISTEMAS DE MANUFACTURA SOBRE WEB.....	11
1. Introducción.....	11
1.1 Sistemas de manufactura enmarcados sobre entornos Web y su campo de aplicación en el ciclo de desarrollo del producto.	12
1.1.1 Ejemplos de sistemas de manufactura basados en Web.....	14
1.1.2 Campo de aplicación.....	16
1.1.3 Características de los sistemas de manufactura sobre Web.	17
1.2 Tecnologías y arquitecturas informáticas empleadas para el desarrollo de los sistemas de manufactura virtuales orientados a Web.....	19
1.2.1 Lenguajes de programación y lenguajes de marca, para desarrollo de aplicaciones de Manufactura vía Internet.....	20
1.2.2 Tecnologías de modelación para objetos distribuidos.....	22
1.2.3 Representación gráfica sobre Web.	23
1.3 Importancia del proceso de simulación.	31
1.3.1 Simulación de procesos.	34
1.3.2 Simulación geométrica.....	35
1.3.3 Características de las aplicaciones CAM.	38
1.4 Laboratorios virtuales y educación a distancia.	39
1.4.1 Requerimientos técnicos actuales de los laboratorios virtuales.	43
1.4.2 Ejemplo eficiente de un laboratorio virtual.	44
1.5 Conclusiones del Capítulo.	46
CAPÍTULO 2 CONCEPCIÓN DE UN SISTEMA PARA LA SIMULACIÓN VIRTUAL DEL MAQUINADO EN OPERACIONES DE TORNEADO.....	47
2. Introducción.....	47
2.1 Factores considerados para la simulación geométrica del proceso.	47
2.1.1 Integración CAD/CAPP.....	48
2.1.2 Diagrama del Modelo de Datos.....	49

2.1.3 Estructura del programa pieza.	52
2.1.4 Análisis y representación de trayectorias.	61
2.1.5 Operaciones básicas y selección de las herramientas de corte.	69
2.2 Estructura y representación del modelado geométrico.	71
2.2.1 Sistemas de coordenadas.	71
2.2.2 Windows y Viewport.	72
2.2.3 Elementos geométricos simples.	74
2.2.4 Representación poliédrica de los modelos 3D.	77
2.3 Modelo de integración para la simulación del maquinado sobre Web.	81
2.4 Conclusiones del Capítulo.	82
CAPÍTULO 3 IMPLEMENTACIÓN DEL SISTEMA PARA LA SIMULACIÓN	
VIRTUAL DEL MAQUINADO SOBRE WEB.	83
3. Introducción.	83
3.1 Arquitectura propuesta del sistema de simulación sobre Web.	83
3.2 Requerimientos funcionales y no funcionales del sistema.	85
3.2.1 Requerimientos funcionales.	85
3.2.2 Requerimientos no funcionales.	86
3.3 Modelo de clases del sistema.	87
3.4 Implementación	88
3.5 Interfaz del sistema.	89
3.5.1 Descripción de los componentes del sistema.	90
3.6 Conclusiones del Capítulo.	92
CONCLUSIONES	93
BIBLIOGRAFÍA	95

INTRODUCCIÓN.

El desarrollo que ha tenido Internet, sobrepasa todas las expectativas desde su creación. Su versatilidad de usos es tal, que no solamente sus beneficios son aplicables a casi todas las ramas de la ciencia, sino que además permite globalizar el conocimiento, al desafiar la privatización e individualidad de unos pocos, para el beneficio de muchos. Ahora la información distribuida en diferentes localizaciones, puede ser accedida y compartida por los usuarios desde cualquier parte del mundo. Internet también ha favorecido a las compañías de manufactura las cuales han estado inmersas completamente en una atmósfera de competencia. Este entorno a cambiado la forma tradicional y manera de acometer las tareas en todas las esferas dentro del ciclo de vida del producto, al permitir un trabajo colaborativo, donde se tiene como principal característica la eliminación de las distancias, así como el ahorro de tiempo para lograr el intercambio de información con interacción multiusuario en entornos virtuales.

Inmersos en este ambiente de desarrollo, los sistemas CAD/CAPP/CAM, presentan una notable y novedosa evolución. Con la aplicación de ellos se obtiene un alto grado de automatización, en la obtención de productos; sin embargo aún existen muchos problemas por resolver. Las limitaciones para la producción de piezas complejas, la necesidad de optimización de trayectorias de maquinado, así como la búsqueda de una integración de los factores involucrados en el proceso de manufactura, la falta de una verificación y control de las posibles colisiones entre los diferentes dispositivos utilizados en el maquinado, la omisión de chequeo sintáctico en los programas pieza y la utilización de aplicaciones distribuidas que asistan el proceso de manufactura remota vía Internet, son algunas de las tareas que ocupan el campo de investigación de los ingenieros. Ante las dificultades mencionadas aparecen los sistemas de simulación como útil adjunto, donde se comprueba exhaustivamente el proceso real, a través de su representación virtual.

Sin embargo a pesar de que prácticamente el mundo industrial es consciente del valor de la simulación de los procesos de corte y su extensión a un marco distribuido sobre Internet, éste aún permanece en sectores muy reducidos asociados principalmente a las grandes compañías, encontrándose en general en un ámbito académico y de investigación. No obstante su protagonismo aumenta en todos los sectores industriales, fundamentalmente gracias al desarrollo de nuevos sistemas de computo más rápidos y fiables que permiten el desarrollo de cálculos cada vez más complejos.

Además aún cuando el alto desarrollo de las Máquinas Herramienta en el sector industrial alcanza índices de productividad sin precedentes mediante el denominado mecanizado de alto rendimiento, muchas pequeñas y medianas empresas (SMEs) se les dificulta el uso de esta alta tecnología y a su vez se empeñan en sobrevivir en un mercado de alto rigor competitivo. Para esto, implementan tecnologías flexibles de manufacturas que permitan elevar la productividad aún con equipamiento convencional [1]. Hoy día las pequeñas y medianas empresas, muestran una apertura a la utilización de las aplicaciones de simulación, con el objetivo de aumentar la eficiencia en el maquinado [29, 30]. La rápida evolución industrial plantea nuevos retos y nuevas problemáticas, es necesario desarrollar ayudas y aplicaciones para permitir a los técnicos desarrollar nuevas tecnologías de forma rápida. Obtener medios de manufactura de bajos costos y crear tecnologías CNC flexibles, tiende a ser la solución de muchos países en vías de desarrollo.

En respuesta a este desafío surge la **necesidad** de desarrollar entornos virtuales sobre Internet, que asistan a las SMEs, donde se permita acortar los tiempos de respuestas relacionados con la planificación, desarrollo y eficiencia en el proceso de manufactura y donde se pueda prestar el mejor servicio al mayor número de usuarios. El presente trabajo muestra el desarrollo de una aplicación de simulación de maquinado que asiste a un sistema CAPP.

Dentro de todo este contexto se establece el siguiente **Problema científico**:
¿Como aumentar la eficiencia del proceso de maquinado en las pequeñas y medianas empresas?

Para dar solución a este problema se plantea la siguiente **Hipótesis**: El desarrollo de un sistema de simulación virtual para el maquinado, sobre entornos Web, permitirá a las pequeñas y medianas empresas un aumento de la eficiencia del proceso de maquinado.

Dada esta situación, nuestro **objeto de investigación** se centra proceso de maquinado. Teniendo como **campo de acción**, la simulación del maquinado, en piezas rotacionales.

Para validar la hipótesis anterior, como **Objetivo General** del trabajo se propone: Desarrollo de un entorno virtual, para la simulación del proceso de maquinado sobre Web, en piezas rotacionales.

Con vista a cumplimentar el objetivo general, se han trazado los siguientes **objetivos específicos**.

1. Implementación del tratamiento de la gráfica según tecnologías 3D existentes sobre la Web. (VRML, Java3D, OpenGL).
2. Análisis e interpretación de las sentencias introducidas por el usuario mediante el uso de técnicas de procesamiento de lenguaje de comando CNC, en las que se incluyen: reconocimiento de palabras, así como análisis y significado de estas.
3. Interacción remota de la aplicación con las Bases de Datos de herramientas de corte y tecnologías del proceso de maquinado según pieza.
4. Creación de un entorno sobre la Web basados en Applets, donde se garantiza la interacción con usuarios remotos, así como con otros sistemas.

Para el cumplimiento de los objetivos presentados anteriormente, en esta investigación se utilizarán los **métodos** siguientes:

- Método de análisis y síntesis de la bibliografía existente de la temática en cuestión.
- El método inductivo deductivo, que se utiliza en la aplicación de los conocimientos a las condiciones de la investigación.
- Método Sistémico, donde modelamos nuestro objeto, de manera que integremos los componentes y sus relaciones.

Justificación y viabilidad de la investigación.

La creación de un ambiente virtual de simulación del maquinado, sobre la Web, a bajo costo propia para países en vías de desarrollo, donde las SMEs del sector metal mecánico tienen un limitado acceso al uso de tecnología CNC y las Máquinas-Herramienta utilizadas en los procesos de manufactura, debido al alto costo de dicha tecnología la cual es extranjera en su gran mayoría, propiciando la especialización del personal o el uso de equipamiento manual, lo cual lleva a un deterioro en la manufactura. Todo ello elevado a un entorno de sistema de manufactura remoto y globalizado vía Internet, donde se tiene el acceso a un mayor número de usuarios y se acortan los tiempos de respuestas a pedidos; podría ser la solución factible y económica que no solo asistiría a la industria sino que también tendría un carácter académico, para el entrenamiento a nivel de laboratorios virtuales dentro del proceso de aprendizaje. La presente investigación, esta orientada a la solución de uno de los problemas de mayor envergadura en el sector de manufactura de pequeñas y medianas empresas (SMEs), encarando los requerimientos de los mercados internacionales donde la calidad, tiempo y costo son la clave para llegar a ser competitivos.

CAPÍTULO 1

ESTADO ACTUAL Y DESARROLLO DE LOS SISTEMAS DE MANUFACTURA SOBRE WEB.

1. Introducción.

A pesar del alto desarrollo alcanzado por las aplicaciones CAD/CAM y su impacto en la automatización del proceso de producción, las compañías de manufactura y principalmente, las pequeñas y medianas empresas (SMEs), en todas partes del mundo se empeñan por mantenerse en un mercado altamente competitivo, donde la integración colaborativa entre entidades, resulta una vía indispensable para lograr el éxito. Por otro lado los nuevos conceptos de comercio y en especial los actuales paradigmas de manufactura, imponen nuevas estrategias. Aquí el producto es el ente principal y comienza a desarrollarse en un ambiente colaborativo mediante la interacción de información en tiempo real entre equipos de ingenieros, fabricantes, proveedores y clientes. En este entorno uno de los facilitadores más importantes es la virtualización de los componentes del proceso de manufactura como lo aborda J. Mandar, (2002) [2] en su artículo. Esta nueva manera de fabricación, aumenta la productividad, la capacidad de respuesta rápida a los cambios según las necesidades, reduce los tiempos en el ciclo de vida del producto y favorece la integración de los datos entre las compañías involucradas. De modo que la manufactura virtual, vista esta como la actividad central dentro de la cadena de desarrollo del producto y favorecida por los beneficios de Internet, resulta un campo de vital importancia e investigación. Este capítulo describe los principales resultados alcanzados en este tema y se enfatizan los sistemas de simulación sobre entornos Web orientados a la manufactura. A su vez se analizan las tecnologías informáticas usadas para la implementación de estas aplicaciones y se resalta la importancia de la simulación del maquinado, así como su desempeño en el proceso actual de enseñanza y aprendizaje mediante el uso de laboratorios virtuales y entornos de e-Learning.

1.1 Sistemas de manufactura enmarcados sobre entornos Web y su campo de aplicación en el ciclo de desarrollo del producto.

La tendencia actual en el ámbito del desarrollo del producto esta dirigida al logro del mayor grado de integración. Con el surgimiento de los sistemas CAPP se alcanzaba reutilizar mucha de la información incluida en la etapa inicial y se planificaba el proceso de producción al convertir la descripción del proceso de diseño en ordenes u operaciones de manufactura. Con la aparición del CAPP, la brecha existente entre los sistemas CAD y CAM era disminuida y la capacidad de automatización era ahora alcanzable. La figura siguiente muestra esta secuencia y presenta algunas tecnologías empleadas.

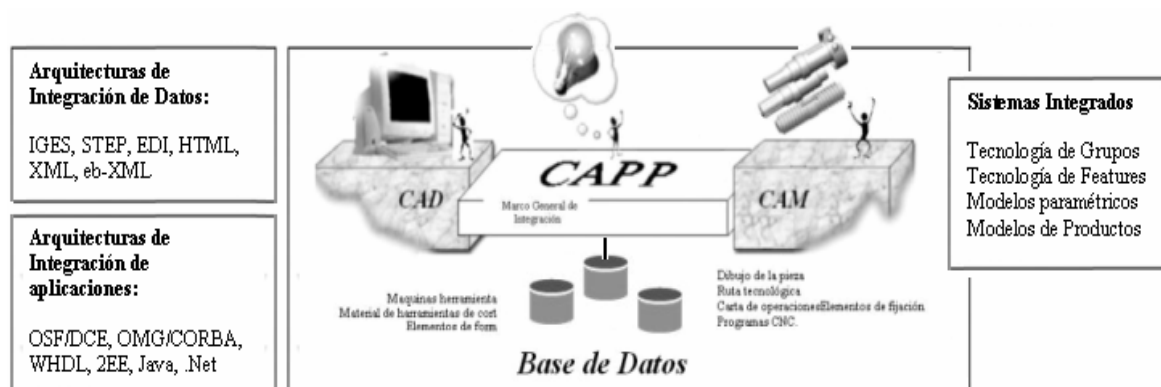


Figura 1.1 Integración CAD/CAPP/CAM.

Esta integración permite el éxito de los negocios en las compañías de manufactura, donde en gran medida depende de la habilidad de identificar las necesidades de los clientes y cuidadosamente crear productos que satisfagan esa necesidad y a su vez estos puedan ser producidos a bajo costo en el menor tiempo posible. Sin embargo, no es suficiente prever una integración local, la demanda actual exige que esta sea globalizada. Para alcanzar los máximos beneficios en este sentido, las aplicaciones basadas en Web sobresalen significativamente. Todos los aspectos del desarrollo del producto, en todo su ciclo de vida, pueden ser perfeccionados, al mejorar la capacidad industrial en cuanto a diseño y manufactura. La utilización de sistemas basados en Web vía Internet, que

asiste al proceso de manufactura distribuida, son las herramientas que favorecen las nuevas filosofías de fabricación, donde la integración se obtiene sin precedentes. Según Peng, (2002) [3] el mayor beneficio que se alcanza con el uso de las tecnologías basadas en Web en aplicaciones industriales es actualmente, el logro de un diseño cooperativo del producto su administración general y tareas con fines de entrenamiento. Las diferentes etapas y actividades asociadas dentro del ciclo del producto asistido por computadoras, han dejado de tratarse de forma independiente y secuencial, para analizarse y automatizarse en un entorno, distribuido y colaborativo entre diferentes disciplinas, lográndose obtener artículos de alta calidad con el menor costo y tiempo de producción. Estos productos se insertan con mayor posibilidad de éxito en un mercado altamente competitivo. La Figura 1.2 muestra este nuevo entorno.

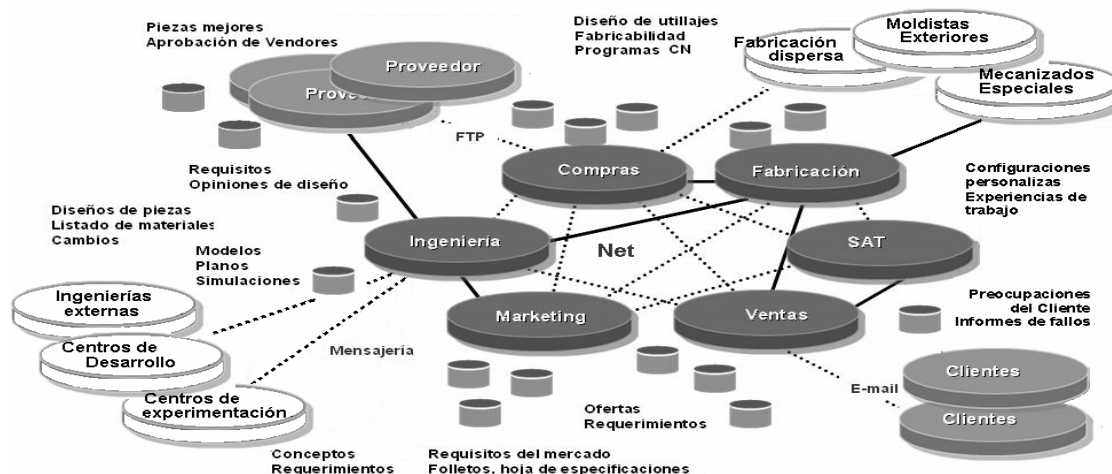


Figura 1.2 Concepción del ciclo de vida del producto colaborativo (Cortesía, RETDIC, 2004).

Las nuevas tecnologías de diseño y manufactura (PDM, PLM, CIM, DFA, CPC, JIT, e-Manufacturing, Ingeniería Concurrente), se desarrollan ahora en conjunto, implementadas en plataformas de multimedia, laboratorios y sistemas virtuales con conexión remota vía Internet, en donde no solo se conjugan aspectos puramente tecnológicos sino también administrativos, comerciales y académicos.

En la actualidad se aprecia una evolución novedosa y acelerada de los sistemas que asisten a la fabricación de los productos basados en Web.

1.1.1 Ejemplos de sistemas de manufactura basados en Web.

Varios sistemas de manufactura basados en Web, se han desarrollado ya en la pasada década, con miras a la automatización en las diferentes etapas del ciclo de vida del producto. Entre ellos se desarrollan aplicaciones CAD/CAM, sistemas de administración de bases de datos, sistemas expertos asociados a bases de conocimientos y sistemas de manufactura con tecnologías de realidad virtual. Todo esto inmerso en un entorno globalmente integrado, mediante el uso de las tecnologías Web. El uso de cada uno de estos sistemas, se orienta a automatizar y optimizar las diferentes etapas de desarrollo del producto. A modo de ejemplo se presenta una serie de sistemas e investigaciones que muestran el uso de los sistemas sobre Web. Principalmente se mencionan aquellos que incorporan módulos de simulación del proceso de maquinado.

- **CyberCut**; (Smith e Wright, 1996) [4]. Es un sistema de fabricación experimental vía Internet, el cual emplea una arquitectura abierta con Máquinas CNC. Tiene incorporado módulos CAD-CAPP-CAM, implementados en Java™, e incorpora servicios de prototipado rápido así como técnicas para el maquinado de precisión basadas en sensores. Los diseñadores pueden usar los Web-browsers para diseñar los componentes, enviar la planificación del proceso a un controlador remoto y ejecutar el proceso en una máquina fresadora de tres ejes. Fue desarrollado en la Universidad de Berkeley (<http://cybercut.berkeley.edu>).
- **WebCAD**; (Mohole *et al*, 2002) [5]. Sistema de Diseño asistido por ordenador (CAD) con explícitas reglas de restricción de diseño con miras a la fabricación CNC en máquina fresadoras. El sistema restringe la libertad del diseñador de tal manera que la pieza diseñada pueda fabricarse en una máquina fresadora CNC de tres ejes. Este sistema tiene conectividad con CyberCut.

- (Sang-Hoon, *et al.*, 2002) [6]. Desarrolla un sistema del mecanizado virtual vía Internet usando CORBA. El sistema muestra la capacidad del VRML y Java, para usarse en el control y monitoreo remoto del proceso de maquinado. El sistema está compuesto por tres módulos; un módulo de análisis de los ficheros NC, simulación del maquinado, y el módulo que controla y monitorea la máquina, cada uno de ellos integrado con CORBA y tecnología Applet. El sistema está hecho en una estructura Cliente-Servidor y opera sobre Internet.
- (Ferreira, J *et al.*, 2004) [7]; Implementan un procedimiento, para la integración de equipamiento automatizado, en un sistema de manufactura flexible (FMS) y su empleo en la manufactura remota de las piezas vía Internet usando tecnologías CGI y VRML. El sistema incorpora módulos CAPP, generación de código de los programas NC y conectividad con bases de datos en MySQL.
- **TeleRP:** (Jiang; Fukuda, 2000) [8]; Sistema sobre Web de prototipado rápido para el análisis y monitoreo remoto de piezas. En este sistema, la selección de un sitio de manufactura de prototipado rápido válido y factible, registrada en el servidor Web para la tele-Manufactura de RP depende fundamentalmente de dos factores: la capacidad del sitio de manufactura y las restricciones del tiempo de entrega concerniente a todas las peticiones dentro del sitio.
- (Zhang *et al.*, 2000) [9]; Desarrolló un marco de trabajo para el intercambio de datos STEP, vía Internet, para empresas virtuales. El sistema implementa un servicio Web, para la traducción de Datos a STEP, formato crucial para lograr la comunicación de los datos del producto entre sistemas CAD/CAPP/CAM/CAX heterogéneos. De esta manera la combinación de Internet con STEP, provee un eficiente y efectivo mecanismo de soporte para las compañías y empresas virtuales.
- **WebMachining** (Ferreira, J *et al.*, 2004) [10]; Es un sistema CAD/CAPP/CAM orientado a la manufactura remota de piezas rotacionales, vía Internet. Su metodología es concebida a partir del paradigma de modelado basado en elementos de formas a fin de permitir la integración de las actividades CAD, CAPP y CAM en un entorno Web.

Al considerar los ejemplos que se mencionan, así como otros descritos en el artículo de Yan, H. y Xue (2003) [11], donde se aborda el desarrollo actual de los sistemas de manufactura en este entorno, se delimitan las principales áreas de explotación y características generales que presentan, tomándolas como referencia para el desarrollo de aplicaciones similares.

1.1.2 Campo de aplicación.

El mayor aporte alcanzado por esta nueva generación de sistemas, radica principalmente en el entorno distribuido en que se ejecutan [3]. Se potencia el envío de información y control remoto de instrucciones de producción hacia máquinas inteligentes (SMS), (Jerard *et al.*, 2006) [12], y se facilita la manufactura de piezas sin tener necesidad de contar con las máquinas y talleres de maquinado in situ. Los siguientes aspectos consideran algunas de las tareas que se pueden realizar por estos sistemas.

- Integración CAD/CAPP/CAM distribuida.
- Control y monitoreo remoto del proceso de manufactura.
- Análisis y optimización de tecnologías del proceso, lo cual incluye la selección de las Máquina-Herramientas, optimización de trayectorias de maquinado, análisis de la rugosidad superficial.
- Generación y análisis de los programas de control numérico.
- Simulación del proceso de maquinado según tecnologías de representación virtual sobre Web.
- Comunicación e intercambiabilidad de los datos del producto mediante ficheros de intercambio.
- Conectividad distribuida de diseño y manufactura con otros equipos de trabajo.
- Incorporación de tecnologías CIM, PDM, FEM, Ingeniería concurrente, y Prototipado Rápido.
- Soporte de entrenamiento académico para multi-usuarios en entornos de laboratorios virtuales.

Este último aspecto sobre el uso de estos sistemas como medio académico y soporte de entrenamiento, por su importancia e impacto, se aborda en un epígrafe específico.

1.1.3 Características principales de los sistemas de manufactura sobre Web.

En el trabajo realizado por M. Larrauri (2000) [13], se revisó un grupo de sistemas de simulación y verificación de programas NC, con el objetivo de analizar y enumerar las principales características que los describen. Sin embargo a pesar de los criterios mencionados en el artículo, la tendencia evolutiva de los sistemas de simulación ha impuesto nuevos requisitos y aspectos fundamentales a la hora de evaluar los nuevos sistemas, considerando su uso en un contexto de manufactura remota. Los siguientes criterios se consideran los aspectos fundamentales a la hora de evaluar estos.

- Capacidad de simulación en un entorno multiusuario sobre Web mediante los Web-Browser.
- Conectividad con bases de datos distribuidas.
- Capacidad de representación visual por video de las operaciones reales mediante servidores remotos. Lo que garantiza la monitorización a distancia mediante sensores y tecnologías WebCam.
- Simulación en tiempo real.
- Herramientas de administración del proyecto de maquinado, mediante árboles descriptivos.
- Inclusión de plantillas personalizadas para la descripción del lenguaje de comandos de los programas NC.
- Capacidad de simulación de trayectorias descritas por curvas B-Spline y superficies NURBS.
- Simulación con multi-ejes.

- Empleo de nuevas tecnologías aparte del estándar VRML para la representación gráfica en 2D y 3D sobre la Web en Internet, como Java3D y las bibliotecas GL4Java.
- Análisis y representación de la calidad superficial de los modelos maquinados.
- Implementación con superlenguajes de programación Java, .Net y C++.
- Módulos de modelación y representación geométrica, optimización de variables que intervienen en el proceso y técnicas de procesamiento de lenguajes (Parsers).
- Implementación de gráficos, algoritmos genéticos, y técnicas de Programación Orientada a Objetos (OOP).

Por otro lado el hecho que el ambiente de trabajo de estas aplicaciones sea principalmente sobre los Web-browsers, impone altos requerimientos informáticos, así como el uso de nuevas tecnologías web. Según Yan, H. y Xue (2003) [11], la tendencia del desarrollo de los sistemas de manufactura y sistemas CAPP basados en Web, se resume en dos categorías:

- I. **Mejoras de las capacidades de cómputo sobre la Web:** Inicialmente las tecnologías Web fueron desarrolladas para mostrar multimedia, incluyendo textos, imágenes, sonidos y video clips, situados en sitios remotos, empleando los navegadores Web. Sin embargo con la aparición de Java, se creaban aplicaciones sobre Web mediante Applets descargados localmente del lado del Cliente, aumentando la capacidad de computo e interactividad de los usuarios conectados a los Servidores. De igual manera las tecnologías Web/Internet, como (HTML, DHTML, XML), del lado del Cliente y las herramientas de programación para los servidores tales como (ASP, DCOM, RMI, CORBA, ICE) y las herramientas recientes como Java y .NET, que incluye superlenguajes como C++ y C#, han favorecido el desarrollo de los sistemas de manufactura sobre Web, como lo ejemplifica el sistema de maquinado virtual vía Internet que usa CORBA de Sang, (2002) [6].

II. Desarrollo de las funciones de modelado y representación gráfica de los datos sobre Web: De igual manera el tratamiento de la gráfica sobre la Web ha sido potenciada gracias al empleo de tecnologías de modelación geométrica, tales como IGES, STEP, VRML, y Java3D, las cuales no solo mejoran las capacidades de la representación tridimensional geométrica de los modelos, sino que permiten la implementación y control por parte de los usuarios, con el uso de las técnicas de programación orientadas a objetos y de bibliotecas OpenGL, para obtener de forma dinámica e interactiva, una representación virtual de escenas sobre la Web.

El epígrafe siguiente aborda mediante ejemplos las tecnologías informáticas fundamentales empleadas por estas aplicaciones.

1.2 Tecnologías y arquitecturas informáticas empleadas para el desarrollo de los sistemas de manufactura virtuales orientados a Web.

El presente tópico muestra algunas de las tecnologías informáticas empleadas en el desarrollo de aplicaciones de manufactura sobre Web. No solamente se requiere un conocimiento del proceso de manufactura en sí, sino que es necesario tener dominio de las tecnologías informáticas más recientes, con el objetivo de lograr la mayor efectividad y eficiencia de computo, como el sistema CyberEye creado por Zhuang *et al.* (2000) [14], que incorpora técnicas de HTML, ASP, Java, ActivX, COM+, y Conectividad con ODBC (Open Database Connectivity) en el desarrollo de plataformas para el diseño colaborativo del producto.

Para su análisis, dada la gran variedad de herramientas, y tecnologías de servicios Web existentes, se agrupan tres categorías (1) Lenguajes de programación, (2) Tecnologías de modelación para objetos distribuidos y (3) Representación gráfica sobre WWW.

1.2.1 Lenguajes de programación y lenguajes de marca, empleados en el desarrollo de aplicaciones de Manufactura vía Internet.

Existen dos plataformas que dominan y compiten por prevalecer con el dominio del mercado con respecto a la implementación sobre la Web. J2EE (Java 2 Enterprise Edition) creada por *Sun Microsystems* y .NET, de *Microsoft Corporation*. Estas son las tecnologías más usadas por las compañías de negocios para desarrollar los servicios Web basados en XML, y precisamente con ellas las nuevas aplicaciones de manufactura encuentran un potencial desarrollo. Ambas tecnologías son más que dos lenguajes de programación, como lo describen Chad y Ed, (2001) [15], en su artículo y ambas presentan puntos en común.

- Con ambas plataformas se crean servicios Web y ofrecen sistemas de bajo costo jBoss/Linux/Cobalt para J2EE, o Windows/Win32 hardware para .NET.
- La portabilidad en ambas tecnologías es presentada, pero en contextos distintos; Java ofrece portabilidad entre máquinas, es decir no depende ni del Hardware ni del sistema operativo. Mientras que .Net logra la portabilidad entre diferentes lenguajes.
- La capacidad de extensión de ambas soluciones son teóricamente ilimitadas y ambas presentan sus propias ventajas y desventajas, por lo que establecer una comparación resulta difícil, cuando se comparan ambas plataformas. Según lo analiza y describe Helena L. (2003) [16].

No obstante la mayoría de las aplicaciones o sistemas orientados a la manufactura vía Internet, se han desarrollado en Java y no en Microsoft.Net. El uso de la tecnología Applets, potencia los entornos de simulación y representación en la Web, la portabilidad en cuanto a multiplataforma que Java soporta, logra el objetivo de comunicación en el versátil entorno que se desempeñan las tareas de manufactura por lo que se aprecia una mejor acogida por parte de los desarrolladores a base de tecnología J2EE.

Aparte de estos superlenguajes, existen otros empleados en el desarrollo de tales aplicaciones. A modo de ejemplos se hace mención de algunos de ellos.

- **HTML** (**H**ypertext **M**arkup **L**anguage) y **DHTML** (**D**ynamic **H**TML): Son los lenguajes que garantizan la creación de las páginas Web.
- **XML** (**E**xtensible **M**arkup **L**anguage); Permite que la información requerida sea organizada en una estructura arborea de datos y sea extraída y mostrada usando los navegadores-Web. Esta nueva tecnología Web mejora las capacidades del modelado de los datos en el desarrollo de los sistemas de manufactura, como se muestra en el sistema colaborativo de modelado de ensambles, desarrollado por Chen *et al.* (2001) [17], donde un esquema de STEP basado en XML es usado para la modelación de los productos ensamblados.
- **Servlet** : Se define como un programa independiente de plataforma que aporta la misma funcionalidad a la programación en el lado del servidor que tradicionalmente han realizado la interfaz **CGI** (**C**ommon **G**ateway **I**nterface), interconectado con herramientas de programación **JSP** (**J**ava **S**erver **P**ages) o **ASP** (**A**ctive **S**erver **P**ages), entre otras.
- **JDBC**: es una interfaz de acceso a **RDBMS** (**R**elational **D**atabase **M**anagement **S**ystem) independiente de la plataforma y del gestor de bases de datos que se utiliza.

Una típica solución de J2EE se muestra en la figura siguiente.

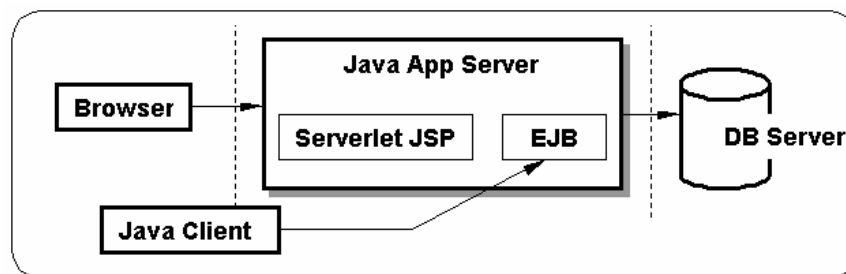


Figura 1.3. Configuración de una solución de J2EE.

1.2.2 Tecnologías de modelación para objetos distribuidos.

Para lograr el desarrollo colaborativo del producto sobre Web, es necesario el empleo de protocolos de comunicación que garanticen la capa de comunicación e integración de los sistemas en entornos heterogéneos. Entre estos se mencionan.

- **CORBA** (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture): Garantiza la integración de la información utilizada en los sistemas sobre Web, esta información normalmente es heterogénea, autónoma y distribuida y permite el empleo de varios lenguajes de programación. CORBA se destaca en su uso de asistencia a los sistemas de manufactura vía Internet, [6, 18, 19] No obstante presenta la limitante de tener un alto grado de complejidad para su uso y está orientado a grandes empresas.
- **RMI** (**R**emote **M**ethod **I**nvocation) [46]; RMI es una manera robusta y efectiva de construir aplicaciones distribuidas en la cual todos los programas participantes deben ser escritos en Java. Debido a que sus diseñadores asumen esto, RMI es sorprendentemente simple y fácil de usar. A su vez es un ambiente ideal para que los programadores de Java, aprendan como desarrollar una aplicación distribuida.
- **SOAP** (**S**imple **O**bject **A**ccess **P**rotocol) [47]; Esta tecnología junto a los **Web Services** permite la integración de componentes de software multi-lenguaje y están disponibles sin los problemas de licencias, No obstante ambas presentan limitantes a la hora de desarrollar aplicaciones comerciales.
- **ICE** (**I**nternet **C**ommunications **E**ngine); Tecnología desarrollada por (ZeroC, Inc., 2003-2004) [32], se presenta como una elección prometedora para el desarrollo distribuido sobre web, puede en síntesis ser considerado como un mini-CORBA.

Como resultado parcial de las tecnologías descritas, para su selección, es importante analizar los siguientes aspectos.

- Microsoft.NET: No soporta plataformas que no sean de Microsoft Corp.
- CORBA: Alto grado de complejidad para su implementación y no es una solución económica, para las pequeñas y medianas empresas (SMEs).
- Web Services: La principal desventaja son las severas ineficiencias, la necesidad de usar licencias de la plataforma a desarrollar, y carencia de estándares.
- SOAP: Impone serias limitaciones a las aplicaciones en términos de eficiencia de ancho de banda y sobrecarga del CPU, así como cuestiones referentes a la seguridad.
- RMI: Solo permite implementación en código Java.
- ICE: Aún no es un estándar.

Estos aspectos son críticos a la hora de hacer una elección en cuanto a que tecnología usar, puesto que estas opciones no lucen como un escenario de vencedores, todas presentan ventajas y desventajas. Lo cierto es que podemos escoger en una plataforma que solo corre en arquitecturas de Microsoft, o seleccionar una plataforma compleja de implementar, o una solamente una para programas de Java o elegir una plataforma que es ineficiente y carente de estandarización. Para el desarrollo de nuestro trabajo se optó por la tecnología RMI, para la solicitud y chequeo de sintaxis al servidor Web.

1.2.3 Representación gráfica sobre Web.

El desarrollo y alcance de las nuevas formas de gráficos que han emergido en la actualidad sobre el WWW, ha permitido un impulso en la evolución de los actuales sistemas de manufactura. Desarrollar formatos de ficheros estándares y compactos, así como herramientas de representación visual para los productos a elaborar en 2D y 3D, ya se consideran aspectos indispensables de estos sistemas.

A diferencia de las muchas aplicaciones publicitarias, informativas y de comercio sobre la Web, donde se manipula gran cantidad de información visual en imágenes, logos, multimedia y otros ejemplos visuales estáticos; las aplicaciones de Manufactura sobre Web, encuentran un importante desafío a la hora de satisfacer las necesidades de representación gráfica del producto, que generalmente requiere que sea una gráfica dinámica e interactiva. Es necesario pensar que la información visual es transportada a través de la red y considerar la relación entre el tamaño de la información transferida y el procesamiento que se lleva a cabo localmente.

El CGRM (Computer Graphics Reference Model), reconoce cinco ambientes diferentes para el tratamiento de la gráfica por computadora, (A. Mumford, 1999) [20]. Construcción, Virtual, Visual, Lógico y Realización. Mucha de la información que es generada por una aplicación y es renderizada, tendrá un progreso a través de un número de transformaciones específicas en diferentes direcciones, con el objetivo de producir una imagen mostrable. El siguiente diagrama muestra los entornos y nombres de las composiciones que pueden ser almacenadas.

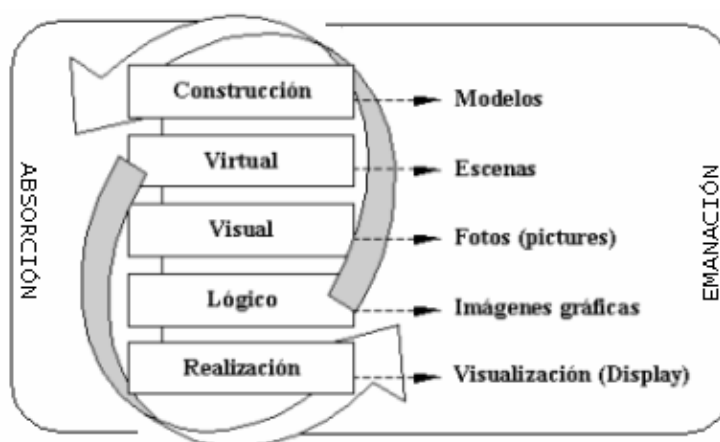


Figura 1.4. Entornos de tratamiento para la gráfica por computadora.

En estos ambientes representativos de la gráfica por computadora, se emplean una gran cantidad de formatos específicos para el modelado y representación en las aplicaciones de manufactura sobre Web. Estos sistemas requieren la representación visual de los modelos equivalentes a los productos o piezas a manufacturar, así como representaciones de simulación y virtualización del proceso. Los siguientes formatos sobresalen en su empleo.

- **IGES**, (Initial **G**raphics **E**xchange **S**pecification) es una especificación para el intercambio de Datos CAD publicado en 1980, por la NBS (National Bureau of Standards), de EUA. IGES fue originalmente desarrollado para el intercambio de datos de modelos tales como: modelos alámbricos en 2D y 3D, textos, datos dimensionales, y clases limitadas de superficies. La norma actual ha sido extendida y desarrollada, concerniente al soporte de nuevas entidades, sintaxis claridad y consistencia. En la actualidad IGES se ha convertido en un estándar de intercambio de datos entre diferentes sistemas CAD. Sin embargo, el tamaño de los ficheros y consecuentemente el tiempo de procesamiento, son algunos problemas prácticos. El registro de información en más de una sección con punteros bi-direccionales, causa errores de implementación en el procesamiento.

Entre las aplicaciones de manufactura sobre Web que usan estos ficheros está: El sistema ATS (Artefact Transport System), desarrollado por Nidamarthi, (2001) [21], donde diferentes bases de datos CAD, son primeramente convertidas a formato IGES y luego traducidas dentro de modelos VRML, para la colaboración entre diseñadores usando navegadores-Web.

- **STEP** (The **S**Tandard for the **E**xchange of **P**roduct Model Data); Es una norma internacional (ISO 10303), para la representación e intercambio de información de los modelos del producto. Esta incluye un lenguaje de especificación de datos, EXPRESS, que describe la representación de los datos. STEP, define también métodos de implementación, de transferencia física de ficheros, y

ofrece diferentes recursos, geométricos y de representación. El objetivo de STEP, fue definir una norma que incluya todos los aspectos de un producto durante su tiempo de vida, y a su vez ofrecer un mecanismo independiente, que describa la información de los productos entre los diferentes sistemas CAx, durante su ciclo de vida. Los métodos son usados para el intercambio de los datos y la representación ofrece una definición de la información del producto para aplicaciones heterogéneas. La gran utilidad de STEP es que no solamente define la forma geométrica de los productos, sino que también incluye topología, características, especificación de tolerancias, materiales, elementos de manufactura, etc. Toda esta información es necesaria para definir completamente un producto y utilizarla con propósitos de diseño, análisis, manufactura, prueba, inspección, y soporte. El uso de este formato crece constantemente, la mayoría de los proveedores de sistemas CAD/CAM implementan e incorporan intercambio con STEP, ya que este incluye el ciclo de vida del producto en términos de almacenamiento e intercambio de datos.

Definitivamente STEP es el resultado más importante y de mayor alcance jamás establecido en el dominio de la ingeniería y sin duda será el estándar de intercambio para la ingeniería asistida por computadoras. Su versatilidad de información integrada lo hace el formato ideal para su uso en las aplicaciones de manufactura orientadas a la Web. Ejemplo de ello es el sistema CyberView desarrollado por Kim *et al.* (1998) [22], donde un esquema de base de datos orientado a objeto (OODB) fue introducido para modelar una base de datos de diseño geométrico basado en STEP. También Kim y Jeong, (2001) [11, 23], implementaron un sistema que convierte modelos de STEP creados desde los sistemas CAD a un modelo de PDM (Product Data Management) basado en Web. El sistema presenta dos módulos; un visor que muestra los datos STEP usando un navegador-Web y un editor/traductor que asocia los esquemas STEP con las bases de datos PDM.

Existen muchos otros formatos de representación de modelos como DXF, STL, ACS y OBJ, los cuales simplifican su lectura y representación, pero básicamente se centran en la descripción geométrica y topológica de los modelos de piezas o productos. A parte de la representación de modelos existen varias tecnologías para la administración y representación de escenas o mundos virtuales en 3D. Algunas de las más empleadas son.

- **VRML (Virtual Reality Modeling Language);** Es una norma internacional ratificada por (ISO 14772) y en esencia permite la descripción geométrica y comportamiento de escenas. Esta tecnología permite especificar mundos virtuales dinámicos en 3D, mediante las cuales los usuarios pueden interactuar simultáneamente con ayuda de un navegador de VRML, como el Cosmo Player 2.1.1 o el visor de Microsoft VRML 2, muchos de estos programas son plug-ins que se incorporan a sistemas como Netscape o Internet Explorer.

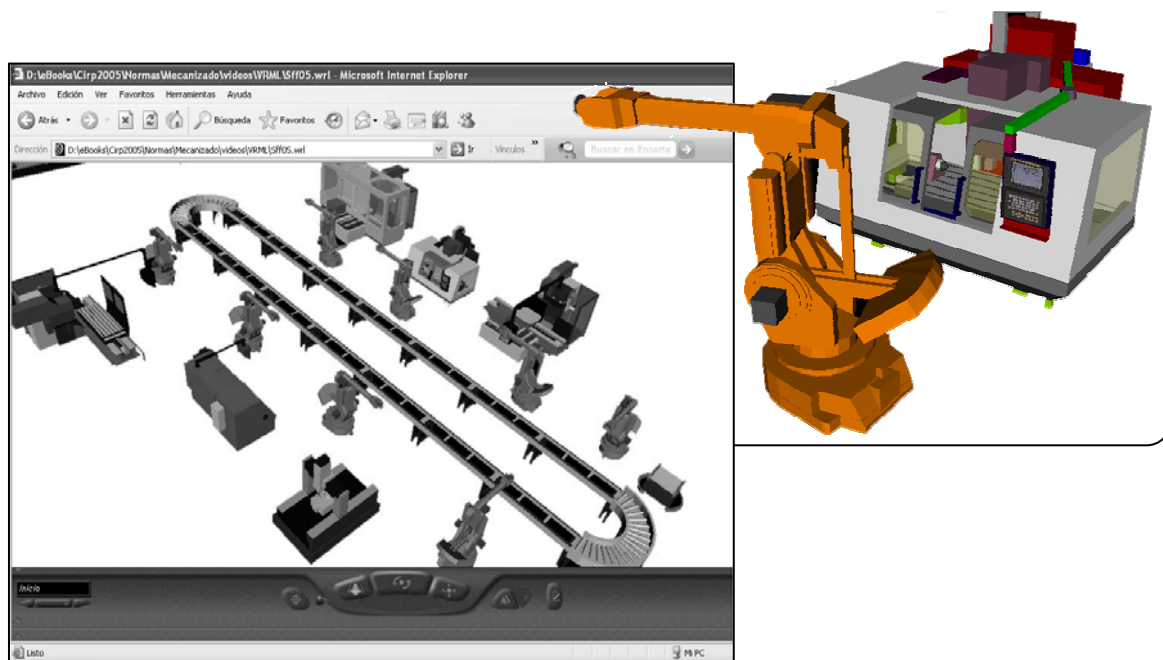


Figura 1.5. Sistema de Manufactura Flexible en VRML sobre Internet Explorer.

Las escenas pueden estar distribuidas sobre Internet y a su vez conectadas con otras escenas vía URL. De este modo VRML puede ser visto como un lenguaje orientado a la extensión de la grafica 3D sobre la Web, a diferencia de HTML que fue diseñado para texto. VRML también permite la construcción de modelos CAD simplificados con tamaños reducidos de memoria con el objetivo de ganar velocidad en la transferencia de los ficheros en Internet. Los documentos de VRML tienen extensión *.wrl y la mayoría de los paquetes CAD profesionales incorporan traductores a este formato, donde no solamente la topología del modelo puede ser exportada sino también representaciones de análisis de movimientos. Este formato es el más difundido en el campo de aplicación, para aplicaciones de manufactura sobre Web por su facilidad y conectividad con Java [6, 7, 10, 25, 26, 40].

Un ejemplo de su uso es el realizado por Sandy R. (1999) [24], donde desarrolla una metodología de integración metodológica para la simulación de sistemas de manufactura. No obstante unos de los obstáculos principales al aumento del uso del VRML, es la dificultad de integrar los modelos producidos por diferentes sistemas. No son del todo compatibles con los navegadores Web y además presentan dificultades en la integración con escenas complejas debido al incremento de funcionalidad analítica requerido para el desarrollo de estos sistemas.

- **Java 3D.** Es una adición al lenguaje Java para la representación de gráficos tridimensionales. Los programas escritos en Java 3D tienen la misma portabilidad de Java, así como los mismos beneficios sobre Internet. Java 3D es una biblioteca de clases de alto nivel que provee una interfaz más simple que otras bibliotecas gráficas, y se emplea en el desarrollo de juegos y animaciones, administración de escenas, detección de colisiones simples, etc. Java 3D está basado en tecnologías existente como DirectX y OpenGL, y puede también reproducir modelos VRML. Aunque su uso no esta tan difundido como el VRML, Java 3D presenta una mejor definición de la modelación. El

desarrollo de RoboLab, (2004) [27], laboratorio virtual remoto orientado a la robótica, establece comparaciones entre ambas tecnologías. Java 3D, gana en calidad de visualización. Sin embargo, a pesar de tales ventajas, Java 3D tiene también debilidades importantes [44]. Algunas de estas se mencionan a continuación.

- Aunque Java 3D se ha empleado en el desarrollo de juegos, el rendimiento de Java 3D es lento. Su principal razón es porque Java en sí es lento.
- Java 3D es una API de alto nivel, que dificulta la estructuración a bajo nivel en caso requerido.
- Sun Microsystems no esta interesada en brindar soporte a Java 3D. Además no está incluido en el Java 1.4.1 SDK
- A pesar que Java 3D esta disponible para Windows, por razones de licencia no está disponible para todos los sistemas operativos. Actualmente no está disponible para Mac OS X.

Además de VRML y Java 3D existe un gran número de modalidades para la programación con Java orientada al 3D. Estas se dividen principalmente en tres categorías. Unión de Java a OpenGL, APIs de escenas gráficas y empleo de motores de juegos. En el desarrollo del presente trabajo se empleó el uso de Java con OpenGL. Esta tecnología se describe seguidamente, así como su comparación con respecto o otras.

- **GL4Java.** Conocido también como “OpenGL para Java”, y como se ha comentado anteriormente es la adaptación y extensión de las librerías OpenGL al lenguaje de programación JAVA. Similares a las que utiliza C o C++. Esta alternativa a Java 3D, es una de las uniones mas populares y puede ser usada con los paquetes de Java, AWT y SWING con conexión a OpenGL 1.3 [44]. GL4Java utiliza toda la funcionalidad de OpenGL, el cual desde el punto de vista del programador es una API para interactuar con dispositivos gráficos y aceleradores 3D. GL4Java ha sido desarrollado bajo

el modelo de código libre desde 1997 y se ha convertido desde ese entonces en una API robusta y estable.

Al establecer una comparación entre GL4Java y Java 3D se destacan los siguientes aspectos:

- OpenGL provee un modelo procedural de gráficos. Este modelo contiene algoritmos y métodos gráficos que han sido usado históricamente por los programadores.
- OpenGL es optimizado en muchas formas, tanto de hardware como en software.
- OpenGL es el estándar contra el cual los proveedores miden sus tecnologías gráficas y está actualmente disponible para más plataformas que Java 3D.
- OpenGL brinda a los programadores acceso a bajo nivel para el proceso de renderizado, el cual Java 3D explícitamente evita.
- No obstante se aclara que esta flexibilidad impone el precio de que su programación sea más difícil que Java 3D.

A pesar del tal desarrollo informático, y aunque muchos sistemas de manufactura basados en Web han sido desarrollados, los expertos plantean aspectos de fortaleza y debilidades. La implementación de los sistemas reales son complejos debido a la gran cantidad de variables que intervienen en el proceso de manufactura, como son los tiempos, avances, velocidades, y estrategias de maquinado. Se considera además su adaptación al entorno específico, por lo que el desarrollo de interfaces de usuario amigables son aspectos críticos a considerar. Se requiere de una buena infraestructura y equipamiento de conectividad de Internet, para satisfacer los altos requerimientos de los sistemas de manufactura basados en Web. Para que el desarrollo de los sistemas de manufactura y se extiendan hacia las pequeñas empresas, estos deben salirse de los entornos puramente académicos y aplicarse a entornos industriales [3].

1.3 Importancia del proceso de simulación. Su empleo en la manufactura asistida por ordenador.

Los sistemas de simulación, se presentan como una herramienta necesaria en el estudio, análisis y modelado de los procesos de corte. Mediante el empleo de estos sistemas se puede inferir y estimar el comportamiento y efecto que tendrán los diferentes elementos que intervienen en el proceso real. Estos sistemas son de gran utilidad y soporte a los sistemas CAPP y para la programación de control numérico CNC, así como para el aumento del rendimiento en las operaciones de maquinado. En principio este aumento puede ser logrado por dos factores principales: Aumentando el potencial para los tipos de operaciones consideradas; (Mejorando la M-H, potencia, velocidad, y exactitud), o al lograr una mejor selección y control de los parámetros del proceso: en el caso del corte; mediante la combinación óptima entre las máquinas, herramientas y condiciones de corte [31]. Es precisamente en este segundo aspecto donde los sistemas de simulación juegan su rol fundamental. Comúnmente el proceso de simulación dentro del ciclo de vida del producto se encuentra luego de la etapa de la planificación, ejecutada por los sistemas CAPP, como se ilustra en la siguiente figura.

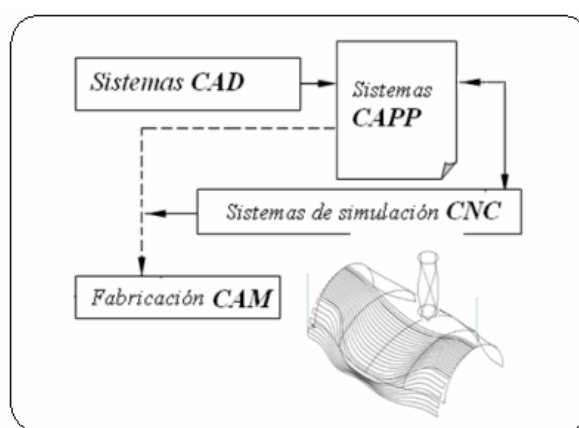


Figura 1.6. Sistemas de simulación en el ciclo de vida del producto.

A diferencia del maquinado convencional, la manufactura en las máquinas CNC, impone la necesidad del empleo de herramientas de simulación. Los accidentes que ocurren en las máquinas convencionales no son comparables con aquellos

que pudieran ocurrir en máquinas CNC, debido a los altos regímenes de corte alcanzados en estas máquinas, asociado también a los altos costos comprometidos, tanto en máquinas, como herramientas y dispositivos que describen el entorno de esta tecnología de avanzada. En las máquinas convencionales los errores generalmente están relacionados con la manipulación del operario, mientras que las tecnologías CNC, los problemas más frecuentes se traducen a errores en líneas de código. De manera que al realizar la simulación de los bloques en el programa, se detectarían fácilmente errores asociados principalmente a las trayectorias de las herramientas. No obstante la utilización de la simulación abarca mucho más.

Mediante el empleo de los simuladores se hacen evidentes las numerosas ventajas sobre la solución de problemas a través de experimentos con los Sistemas Reales. Entre estas ventajas se destacan:

1. *Tratabilidad*: En la mayoría de los casos es imposible, por razones técnicas, económicas, experimentar ampliamente con los Sistemas Reales. En el contexto del mecanizado se requeriría contar con Máquinas-herramienta de alto costo de adquisición o de talleres especializados orientados a operaciones de CNC, (Control Numérico Computarizado).
2. *Eficiencia*: La Modelación y Simulación permiten acortar el tiempo de diseño y desarrollo de Sistemas Reales, así como el ahorro de medios y recursos. Definitivamente mediante la simulación se evita el riesgo de los gastos de materiales y de herramientas, así como los errores de manipulación en las máquinas.
3. *Entrenamiento*: La Simulación permite el entrenamiento y formación de personal para el uso de Sistemas Reales complejos, sin necesidad de poseer dichos sistemas. Finalmente muchos de estos sistemas sirven de asistencia académica en la preparación de estudiantes y tecnólogos sin tener que contratar servicio técnicos especializados o adquisición de equipamiento.

Atendiendo a su campo de acción la simulación en el ámbito de la manufactura se agrupa en dos tipos fundamentales.

- I. **Simulación de procesos:** ej. Efecto de la temperatura y de las fuerzas de corte en la formación de la viruta; Deformación de la herramienta [34].

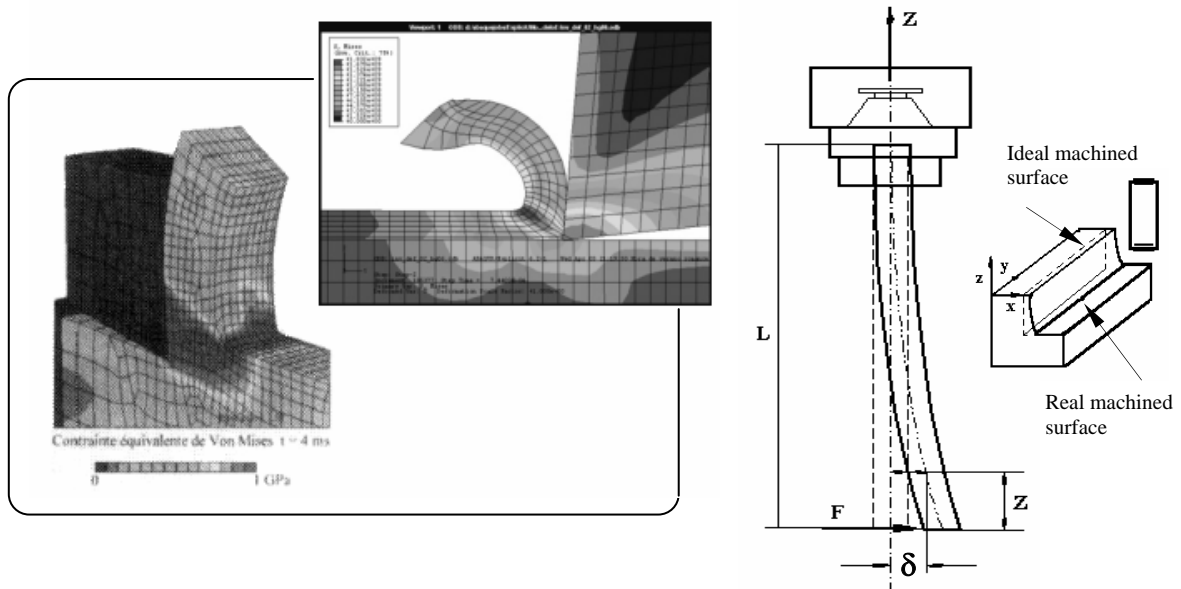
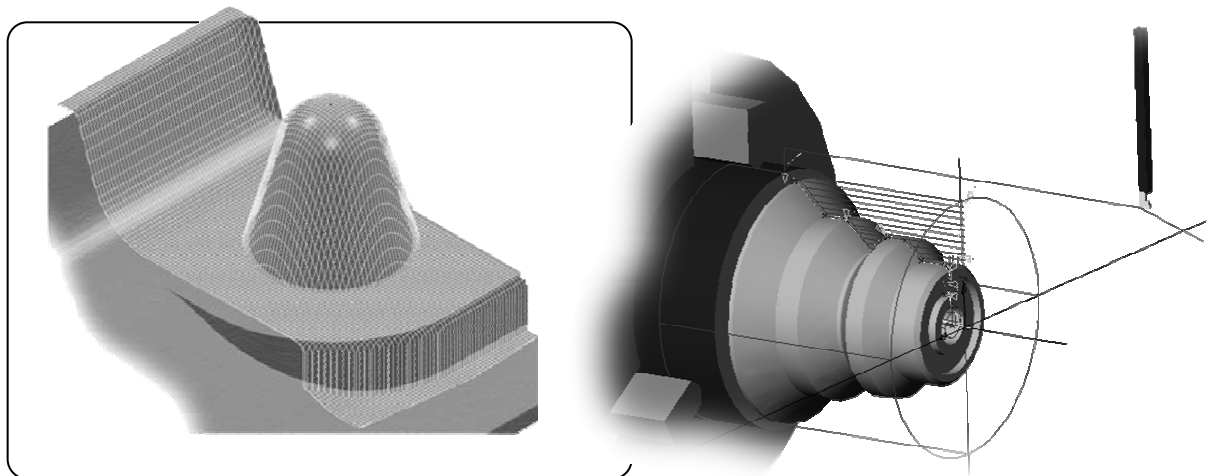


Figura 1.7. **a)** Simulación en corte oblicuo mediante un modelado euleriano y análisis de la deflexión no lineal de la herramienta durante el corte. **b)** Simulación geométrica de trayectorias de la herramienta.

- II. **Simulación geométrica:** ej. Simulación de las trayectorias de la herramienta.



1.3.1 Simulación de procesos.

En el caso de la simulación de procesos se destaca el uso de la simulación numérica así como el uso de diferentes tipos de modelos numéricos. Este tipo de simulación numérica del proceso de corte aún para el mecanizado ortogonal se encuentra en una etapa incipiente. Esta temática es exhaustivamente abordada por Arrazola *et al.*, (2002), en su artículo “Fundamentos del Proceso de Corte y Situación Actual de su Modelado” [33], donde se exponen las áreas de interés industrial hacia la simulación en el maquinado. Entre las áreas principales se encuentran:

- Duración de vida de la herramienta.
- Fuerzas y potencias necesarias en la operación a realizar.
- Calidad superficial y tolerancias dimensionales de la pieza.
- Tipología de la viruta, forma y longitud.
- Tensiones residuales e inestabilidad dinámica del proceso.
- Deformación de la herramienta durante el corte.
- Formación de rebabas o rotura frágil del material a la salida de la herramienta.

El artículo sugiere el empleo de la simulación numérica mediante el uso de los métodos de elementos finitos, como la vía más factible para el modelado del proceso de corte. Este a su vez, dada la complejidad de los fenómenos que ocurren durante la formación de la viruta impone el uso de formulaciones matemática no lineales para el estudio de las variables involucradas y describe diferentes modelos numéricos empleados por los sistemas de simulación existentes.

Aunque este tipo de simulación se acerca más a la realidad, y se obtienen resultados comparables con los experimentales o industriales, impone también elevadas exigencias. Entre ellas se mencionan dos fundamentales:

- **Conocimiento detallado de la mecánica del proceso de corte:** Este es uno de los problemas mayores, ya que resulta compleja la medición de ciertos parámetros fundamentales: deformaciones, velocidades de deformación, calentamiento o enfriamiento, temperaturas, y tensiones. La dificultad en este estriba en la accesibilidad a la zona donde se producen los diferentes fenómenos que han de ser representados. Todo ello hace que los ensayos experimentales para poder determinar dichos parámetros en determinadas condiciones resulten complejos y costosos. En muchos casos los resultados se alejan de las condiciones para las cuales se había diseñado el ensayo.
- **Conocimiento de las leyes de comportamiento:** La ley de comportamiento del material, ley de comportamiento en fricción. La principal dificultad aquí estriba en que dichas leyes son obtenidas en ensayos mecánicos los cuales, en muchos casos, no permiten reproducir las condiciones en las que posteriormente se va a aplicar. Como consecuencia, en algunos casos para la obtención de modelos que se puedan aplicar dentro de las condiciones que se dan en el mecanizado, deberán de realizarse extrapolaciones no exentas a incertidumbres, lo que conllevará a la obtención de resultados alejados de la realidad.

1.3.2 Simulación geométrica.

Por otro lado la simulación geométrica, direcciona su representación y análisis desde un enfoque macroscópico del proceso. Este tipo de simulación tiene también una amplia utilidad y aporta grandes beneficios en el desarrollo de la manufactura. Entre las principales tareas se encuentran:

- Optimización y representación de las trayectorias de la herramienta.
- Representación de trayectorias sobre curvas complejas (spline, Besier).
- Detección de las colisiones entre los diferentes elementos considerando la cinemática de la máquina.
- Detección de colisiones en el maquinado con multi-ejes.

- Estrategias de maquinado.
- Alta capacidad de representación de elementos involucrados (pieza, herramienta, máquina, utillaje, etc.) y simulación de programas NC de diferentes controles (FANUC, FAGOR, SIEMENS).
- Cálculo del material removido.
- Detección de penetración de la herramienta en la pieza final.
- Reconocimiento de áreas de material sobrante.

La simulación geométrica del maquinado, virtualiza el proceso antes de ejecutarse en los puestos y máquinas reales, lo que garantiza la revisión de los códigos de programación NC, los posibles errores de colisiones, lo cual redundaría en pérdidas considerables de herramientas y daños en las máquinas, Esta simulación favorece la exactitud de los cálculos ingenieriles de la tecnología para el maquinado de las piezas. Por otro lado la realización del maquinado de piezas con un alto grado de complejidad, y por ende de mayor valor añadido, es cada vez más frecuente en la industria. Para efectuar esta tarea no basta con modernizar las Máquinas Herramienta, se requiere de módulos adicionales capaces de detectar, mostrar y ofrecer soluciones, a las posibles interferencias entre los utillajes que intervienen en el mecanizado durante simulación de las trayectorias, lo que posibilita el ahorro de un gran número de horas máquinas y costos de producción. Independientemente de que las máquinas modernas CNC, cuentan con simuladores incorporados en sus controles, se requiere de aplicaciones de simulación adicionales, situados en las computadoras, con miras a la automatización de todo el proceso, lo cual favorece también al emplearse tecnologías DNC (Control Numérico Directo), donde el proceso del taller CNC es controlado por terminales (PC), distantes de las máquinas. Con el desarrollo de los medios de computo e informática (Hardware & Software), varias casas matrices y empresas de manufactura emprendieron la tarea de crear potentes aplicaciones CAD/CAM. Estas aplicaciones no solo asisten a los usuarios, sino que también son generativas, y optimizan los procesos con gran eficiencia, rapidez y exactitud.

Atendiendo a esto las ventajas que proporciona la simulación se hace evidente a la hora de mejorar la eficiencia de los procesos. La evaluación de diferentes estrategias del proceso en corto tiempo, la eliminación de prototipos así como la facilidad de no depender de los medios físicos, entre otras, podría determinar la simulación como herramienta indispensable. Sin embargo ha pesar de todas las ventajas mencionadas existen algunas razones por los que la simulación como técnica no se ha utilizado como era de esperar. Según Jose L. *et. al.* (2002) [43], algunas de las desventajas principales son:

- Modelos pobres en detalles relevantes para la simulación.
- Simulaciones excesivamente lentas.
- Interfaces de usuario complejos.
- Falta de personal calificado, para aplicar los modelos eficientemente.
- Datos incorrectos de entrada a la simulación y por lo tanto resultados inseguros e inapropiados.
- Modelos rígidos no capaces de adaptarse a los cambios en el sistema productivo o máquina.

Todos estos hechos han reducido el uso de la simulación como herramienta eficiente y de utilidad para el desarrollo de los procesos de producción. No obstante los avances fundamentales de las tecnologías de la información, hacen que hoy en días se realicen simulaciones de gran interés y utilidad lo que justifica su utilización e investigación en este campo.

1.3.3 Características de las aplicaciones CAM.

Entre las aplicaciones CAM más notables del mercado actual que incorporan módulos de simulación geométrica se encuentran: (Catia V.5, Pro/E, IDEA, DellCAM, VeryCUT, SurfCAM, EdgeCAM, ESPRIT, Unigraphics). Todas estas aplicaciones cumplen los requisitos, de las herramientas de simulación. A consideración de los expertos, los sistemas de simulación CNC que pretendan tener éxito en el mercado, deberán cumplir los siguientes requisitos mínimos:

- El software deberá poseer una interfase que le brinde la mayor facilidad posible a sus usuarios desde el punto de vista operacional, con especial atención al entorno de edición gráfico, en 2D y 3D.
- Eficiente capacidad de análisis y edición de los comandos léxico, sintáctico y semántico, del lenguaje de comandos CNC, el cual tendrá todo el tratamiento de errores y posibles soluciones alternativas.
- Capacidad de simulación de diferentes operaciones tecnológicas: Torneados, Fresados, Taladrados, etc.
- Previsión, detección e información visual de las colisiones producidas durante el mecanizado, considerando el entorno productivo.
- Interacción e intercambio de datos con otros sistemas CAD/CAM mediante el tratamiento de los ficheros de intercambio de datos.
- Entornos de edición gráfica (Zoom, Pan, Orbit, Select).

Aunque todos estos aspectos constituyen ventajas que facilitan el trabajo de los usuarios, se mencionan algunas limitantes y desventajas que presentan estas aplicaciones profesionales:

- Carecen de una integración y representación de todos los factores y variables que se involucran en el proceso de maquinado.

- La selección de herramientas, queda a solicitud de los usuarios y muchos de los sistemas no brindan consultas de selección de las herramientas de corte, acorde con la pieza a elaborar.
- Presenta una elevada curva de aprendizaje de los usuarios, lo cual incrementa los costos de capacitación.
- Son de difícil adquisición para pequeñas y medianas empresas, por causas de los costos por licencias.
- Imponen altos requerimientos de Hardware.
- La mayoría de las grandes aplicaciones existentes, no soportan conexión remota, ni están implementadas en ambientes Web.

Visto de este modo, aún existe la necesidad de continuar el desarrollo de aplicaciones, que si bien es cierto que no tendrán el alto nivel, de las ya existente, sí tendrán la capacidad de ser mejor configurables a las condiciones y posibilidades que tienen muchas empresas pequeñas y países en vías de desarrollo. De igual manera la creación de sistemas de simulación que corran sobre la Web, potenciará la colaboración empresarial y tendrá mayor alcance para todos. Ya no será imprescindible contar con los talleres de maquinado, ni siquiera con los paquetes profesionales, sino que el proceso de manufactura remota vía Internet, proveerá prestaciones de servicios mediante los navegadores Web a los tecnólogos e ingenieros y se podrá maquinar a distancia lotes de piezas que luego se enviarán a los clientes. Otro aspecto de aplicación de estos sistemas, es su empleo en un entorno académico de laboratorio virtual, y herramientas e-Learning con el objetivo de asistencia tanto a estudiantes como a ingenieros.

1.4 Laboratorios virtuales y educación a distancia.

Las herramientas de simulación también tienen una larga trayectoria en la capacitación de aptitudes prácticas y su popularidad se incrementa cada vez más. Situaciones concretas son reconstruidas en un contexto “real virtual” bajo condiciones de prueba y permiten a los estudiantes experimentar y solventar problemas de una manera más segura, y más sencilla, que en la vida real. El futuro de este campo es prometedor. Estas nuevas tecnologías están siendo

también se implementan en los laboratorios virtuales de diferentes especialidades, (Física, Química, Robotica, y Manufactura) los cuales puede aportar muchas ventajas en la docencia de asignaturas donde se aprecia el impacto que tiene el uso de esta nueva herramienta docente en el proceso de enseñanza-aprendizaje [27].

En la actualidad se desarrollan metodología para la implementación de laboratorios virtuales en el sector de manufactura y robótica, con miras a promocionar la enseñanza a distancia en el desarrollo de experimentos remotos vía Internet. Ejemplo de ello es la manipulación de un brazo robot Unimation PUMA260, mediante el empleo de tecnologías Java y VRML [36], donde los estudiantes realizan diferentes tareas complejas. La combinación de la teoría con la práctica solo es logable mediante la interacción del alumno con las Máquinas-herramienta CNC, robots y células flexibles, los cuales son medios muy costosos y no siempre están presentes en las universidades o escuelas técnicas. Por tanto al crear una infraestructura remota de enseñanza a distancia con interfaces que manipulen y monitoreen remotamente los dispositivos, se podría prescindir de ellos en forma física [35]. Para muchos centros de investigación y universidades, los laboratorios virtuales sobre Web resultan una alternativa imprescindible en la formación y capacitación de bajo costo para sus estudiantes.

Existen tres razones de importancia para el empleo de las herramientas de simulación como medio educativo.

1. **Seguridad.** El mundo real es peligroso. Los estudiantes no deberían arriesgar su vida (o la de otras personas) o asustarse en el proceso de aprender su oficio.
2. **Simplicidad.** Facilidad y compresión de tiempo que se logran con la versión "virtual" de la capacitación. En ocasiones es necesario dedicar largo tiempo a organizar y realizar ciertos experimentos. En otros, los resultados no son tan claros, debido a que se presentan demasiadas variables inmanejables. Las

simulaciones pueden ser una manera conveniente y convincente de sintetizar el mundo real.

3. **Economía.** Las simulaciones pueden ser menos costosas que aprender en el mundo real, especialmente ahora, cuando las computadoras son cada vez más baratas. Las simulaciones pueden ahorrar costosos laboratorios o implementos tales como metal, componentes electrónicos o electrodos de soldadura.

Tales beneficios han permitido la extensión hacia entornos de aprendizaje en el campo de la manufactura. Actualmente, las simulaciones por computadora de las máquinas CNC son muy comunes, bastante sofisticadas y poco costosas. Si se utilizan adecuadamente, pueden acelerar la capacitación y bajar los costos significativamente, pues las personas entrenadas pueden aprender mucho de ellas y requieren mucho menos supervisión.

En la actualidad tanto en países desarrollados como subdesarrollados, aumenta la necesidad del uso de las computadoras en el proceso educacional. Los desarrolladores de software aprovechan los beneficios de las tecnologías informáticas especialmente aquellas que se basan en la conectividad con Internet. De la misma manera tecnologías como VRML, VML, Java3D y GL4Java, se pueden usar como herramientas educacionales potenciales, las cuales se enlazan con Internet. Como se analizó anteriormente cada una de ellas permite la visualización tridimensional dinámica de modelos, mundos animados, sonidos y entornos interactivos sobre los navegadores Web.

Frecuentemente los estudiantes de ingeniería mecánica no tienen suficiente interacción con equipamiento de manufactura. Esto se debe a la dificultad de traer las máquinas reales al aula de clases. Por otro lado no resulta fácil enseñar a los estudiantes en los talleres propios, donde la actividad, los ruidos y riesgos dificultan la calidad de la clase. Además existe la necesidad de tener medios de enseñanza que de manera fácil le permitan aprender interactivamente a los estudiantes.

Este aspecto es resuelto al contar con entornos de manufactura virtual, que simulen el proceso. De esta forma los talleres son virtualmente traídos a las aulas. Esta conectividad se generaliza mediante el empleo de la Internet. El uso de Internet resulta un medio necesario y determinante cuando se desee hablar de avance en proceso educativo. Internet permite la conectividad de los estudiantes con diferentes universidades, grupos de investigación, comunidades virtuales y aplicaciones sobre Web que proveen la información y soporte necesario para su aprendizaje.

En este entorno académico muchas herramientas educativas pueden ejemplificar los resultados alcanzados. Ejemplo de ello es la aplicación de manufactura en VRML, para la enseñanza de estudiantes de ingeniería, usando Internet, desarrollada en la Universidad de Oklahoma [37]. Así como el proyecto desarrollado por cuatro Universidades donde se incluye el instituto ISPJAE de nuestro país (David E. *et al.*) [38]. En este proyecto se tuvo la experiencia del uso de una herramienta de ingeniería concurrente vía Internet, en un entorno e-Learning en el campo de CAD/CAPP/CAM. En este proyecto se conjugaron equipos de estudiantes y profesores de ingeniería mecánica e industrial así como grupos de expertos, con la finalidad de crear un entorno de Ingeniería Colaborativa. En este estudio se unifican las necesidades de los futuros ingenieros profesionales, combinando, la colaboración interdisciplinaria, el uso de la información, así como de tecnologías que le permiten a los estudiantes enriquecer su nivel educativo al solucionar problemas reales de importancia global.

Además del proyecto mencionado, se aprecia en nuestro país un acelerado incremento al uso de las técnicas informáticas aplicadas en el proceso docente educativo. Los proyectos de Universalización, Educación a distancia, y la Colaboración entre universidades, brinda nuevas oportunidades para el desarrollo de aplicaciones virtuales. Otro resultado que fortalece lo mencionado, es el proyecto entre el Instituto Tecnológico de Monterrey, (ITESM, México) en conjunto con la Universidad de Holguín, (UHo, Cuba) [29]. En este proyecto se trabaja en la creación de un entorno virtual vía Internet con el fin de integrar aplicaciones

CAD/CAPP/CAM y simulación para el maquinado. En este entorno virtual los estudiantes de ambos centros pueden realizar sus prácticas de manufactura de forma remota. Estos resultados no solo inciden en la educación, sino también dan inicio al desarrollo de nuevas tecnologías de manufactura que podrán servir al sector de pequeñas y medianas empresas (SMEs) de países en vías de desarrollo, tales como Cuba, México y otros.

1.4.1 Requerimientos técnicos actuales de los laboratorios virtuales.

Entre las características principales que deben tener los laboratorios virtuales, se encuentra, garantizar los entornos multiusuarios, donde se toma en cuenta la diversidad de las plataformas tecnológicas que los estudiantes poseen, o a las que pueden tener acceso. Lo anterior puede considerarse como una limitante para el desarrollo de los laboratorios virtuales, pero dada la naturaleza de la institución y sus estudiantes, es imprescindible pensar en computadoras con sistemas operativos como Windows 98, NT, 2000, XP, Linux, Mac. y navegadores de Internet como Netscape, Internet Explorer y Opera, entre otros. En este sentido optar por superlenguajes como Java y tecnologías Web, suele ser la solución de mayor provecho. En el caso de Java, debido a sus características de ser un lenguaje simple, orientado a objeto, distribuido, robusto, seguro, de arquitectura neutral, portátil, de alto rendimiento, multitarea y dinámico. La conectividad con Internet no es un requisito indispensable de manera que tales aplicaciones pueden funcionar en una intranet local o incluso como aplicación independiente.

Según Kappelman, citado por Monge-Nájera, (2002) [39], una aplicación de esta naturaleza debe cumplir algunos requisitos antes de ser considerada como un laboratorio virtual completo: ser autocontenido, interactivo, contener imágenes bidimensionales y tridimensionales, animaciones, video, audio, incluir ejercicios, autoinstalarse automáticamente, tener navegación no estrictamente lineal, un buscador y un subprograma para hacer anotaciones sin la necesidad de acudir a un procesador de textos externo. En laboratorios ubicados en páginas Web, habría con esta tecnología espacios donde el estudiante podrá digitar o copiar textos que considere necesarios para repasar los temas importantes o para formular

preguntas a su tutor. El buscador también puede implantarse por medio del uso de bases de datos, sistemas expertos o mediante el uso de otras técnicas de inteligencia artificial.

1.4.2 Ejemplo eficiente de un laboratorio virtual.

La mayoría de las características antes mencionadas han sido incorporadas en el proyecto AVML (Advanced Virtual Manufacturing Laboratory) uno de los laboratorios basados en Web, más completos que existen en el campo de la manufactura [41]. Los principales elementos del entorno de AVML, se resumen en: (a) Modelos virtuales en 3D del laboratorio como de las máquinas con textura fotorealista; (b) Simulador de maquinado para el control del movimiento de las herramientas y piezas; (c) Modelo semi-empírico de las operaciones de maquinado; (d) Base de conocimiento jerárquica para el entrenamiento del proceso; (e) Bases de conocimientos no estructuradas para las lecturas suministradas; (f) Presencia de un tutor virtual similar a un humano, con interacción mediante lenguaje natural o teclado.

Esta aplicación incluye el entrenamiento de estudiantes en las operaciones de maquinado en un entorno seguro, lo cual permite a los estudiantes e investigadores ver e interactuar con elevada precisión la simulación física en las máquinas herramienta, y optimizar el proceso de manufactura mediante herramientas CAPP, en la que también es posible probar diferentes tecnologías de maquinado. El entorno de entrenamiento virtual reduce los costos significativamente y aumenta la accesibilidad y seguridad en la capacitación de la manufactura avanzada.

La siguiente figura muestra la interfaz sobre el navegador Internet Explorer del proyecto AVML



Figura 1.8. Laboratorio virtual AVML.

A pesar de las ventajas tan claras que tienen todas estas aplicaciones, existe una serie de aspectos que por el momento pueden limitar su uso dentro del desarrollo de laboratorios virtuales. En primera instancia los laboratorios virtuales tienen la limitante del medio en que se presentan, es decir, al depender del computador, se tienen los elementos visuales y auditivos, pero los demás sentidos, tacto, gusto y olfato se reducen mucho, al menos por el momento. También cabe destacar que la mayoría de estas aplicaciones hoy día, carecen de un modelo evaluativo eficiente, que permita garantizar el progreso del aprendizaje en los estudiantes. Muchas plataformas de e-Learning tales como Moodle, optan resolver esta dificultad al tener arquitecturas abiertas, para la extensión e incorporación de nuevos módulos, con miras a mejorar su potencial. Los laboratorios virtuales se presentan como un medio imprescindible para garantizar la calidad de la docencia y aprendizaje.

1.5 Conclusiones del Capítulo.

Luego de revisar la literatura especializada, así como analizar el estado de desarrollo de los entornos de manufactura actuales, se puede concluir con:

1. Para lograr un entorno integrado que asista en el proceso del ciclo de desarrollo del producto no es suficiente contar con las aplicaciones CAD/CAPP/CAM tradicionales, se requiere el reforzar a estas de nuevas aplicaciones que se ejecuten sobre Web.
2. La mayoría de las aplicaciones de manufactura vía Internet, así como laboratorios virtuales, se implementan en Java, y se aprecia, además del estándar VRML, el uso de otras tecnologías como GL4Java, las cuales imponen sus mejoras y garantizan el desarrollo futuro de la representación grafica dinámica y tridimensional sobre Web.
3. La permanencia en el mercado de muchas medianas y pequeñas empresas (SMEs) dependen del uso de aplicaciones de manufactura virtual sobre Web, con las cuales pueden mejorar la eficiencia de su proceso de maquinado. Por lo que el desarrollo de estos programas encuentra un amplio campo de aplicabilidad.
4. El uso de e-Learning así como los laboratorios virtuales orientado a la manufactura, favorecen la capacitación de tecnólogos así como la preparación de estudiantes, al lograr obtener un profesional de mayor calidad, sin necesariamente contar con equipamiento altamente costoso, como son las máquinas CNC y talleres industriales.

CAPÍTULO 2

CONCEPCIÓN DE UN SISTEMA PARA LA SIMULACIÓN VIRTUAL DEL MAQUINADO EN OPERACIONES DE TORNEADO.

2. Introducción.

En el Capítulo 1 se estudió el estado actual de los sistemas de simulación sobre entornos Web orientados a la manufactura. A su vez se describen las tecnologías informáticas usadas en su implementación y se resaltó la importancia de la simulación, tanto en su uso industrial como académico, mediante el empleo de entornos de e-Learning y laboratorios virtuales. En el presente capítulo se estudian los factores que intervienen en el proceso de simulación geométrica para el maquinado, específicamente en la operación tecnológica de torneado. Se plantea el análisis sintáctico de los programas piezas a simular, así como el análisis para representar y estructurar la visualización en 3D de los modelos importados de los sistemas CAD. Finalmente se plantea un modelo mediante el cual se garantiza la integración CAD/CAPP/CAM.

2.1 Factores considerados para la simulación geométrica del proceso.

Como se planteó anteriormente muchos son las variables y factores tecnológicos que han de considerarse para la simulación geométrica del proceso de maquinado. El simulador ideal será aquel que incluya la mayor cantidad de estos, aunque siempre el proceso de simulación será en sí, una abstracción de la realidad. Esta representación, será altamente afectada en el caso de que la aplicación se desarrolle para entornos Web, como lo es en el presente caso, debido principalmente a las restricciones de velocidad, tratamiento gráfico y tecnologías propias que la Web impone. En este epígrafe se mencionan los factores considerados para el desarrollo de la presente aplicación.

Los factores a considerar se resumen a continuación:

- Integración CAD/CAPP.
- Análisis de los programas piezas.
- Análisis y representación de trayectorias.
- Operaciones básicas y selección de las herramientas de corte.

2.1.1 Integración CAD/CAPP.

La concepción del modelo de simulación en esta aplicación, está orientado a un funcionamiento modular en un entorno mucho más amplio, independientemente que su uso está previsto también como aplicación independiente. No obstante queda claro que los mayores beneficios y acercamiento a la realidad solo pueden ser concebidos en un entorno integrado CAD/CAPP, donde el simulador importará los modelos de Máquinas, Utillaje y Piezas, desarrollados por paquetes CAD, así como la carga de las herramientas seleccionadas por el sistema CAPP. El sistema CAPP no solo optimiza esta selección de herramientas sino que también genera la tecnología de maquinado para cada artículo o lote de piezas a maquinar, incluyendo la generación automática de programas NC. Esta información tecnológica es almacenada en la base de datos la cual sirve de banco de información al sistema.

Para garantizar de la correcta interfaz entre el CAD, el CAPP y, el CAM es necesario que el sistema pueda ser asociado a una base de datos. A su vez generar la planificación del proceso requiere de un conocimiento previo relacionado con la geometría de la pieza, material a maquinar, las herramientas de corte, el material de las herramientas, los parámetros de maquinado y características de las Máquinas - Herramienta. En este aspecto en particular el módulo de actualización juega un importante papel para conocer las variaciones de estas características en el tiempo, los datos relacionados a las dimensiones de la pieza, las tolerancias, bases dimensionales, asociadas a la cadena dimensional y a el acabado superficial de las diferentes superficies.

Inicialmente con esta información se crea una gran cantidad de entidades y relaciones, que al mismo tiempo deciden el tipo de operación con los datos correspondientes, que son necesarios para obtener la forma final de la pieza. La Base de Datos es única y centralizada y en el caso que se discute se encuentra en el lado del Servidor Web.

2.1.2 Diagrama del Modelo de Datos.

El modelo de datos que se utilizó para describir la estructura lógica de la información persistente almacenada por el sistema, fue creado a partir de las clases persistentes obtenidas previamente y permitió mantener las estructuras que garantizan en todo momento la integridad, consistencia y confiabilidad de los datos manejados por el sistema. La siguiente figura muestra parte de la estructura del modelo de datos. Este modelo es el utilizado por el sistema CAPP.

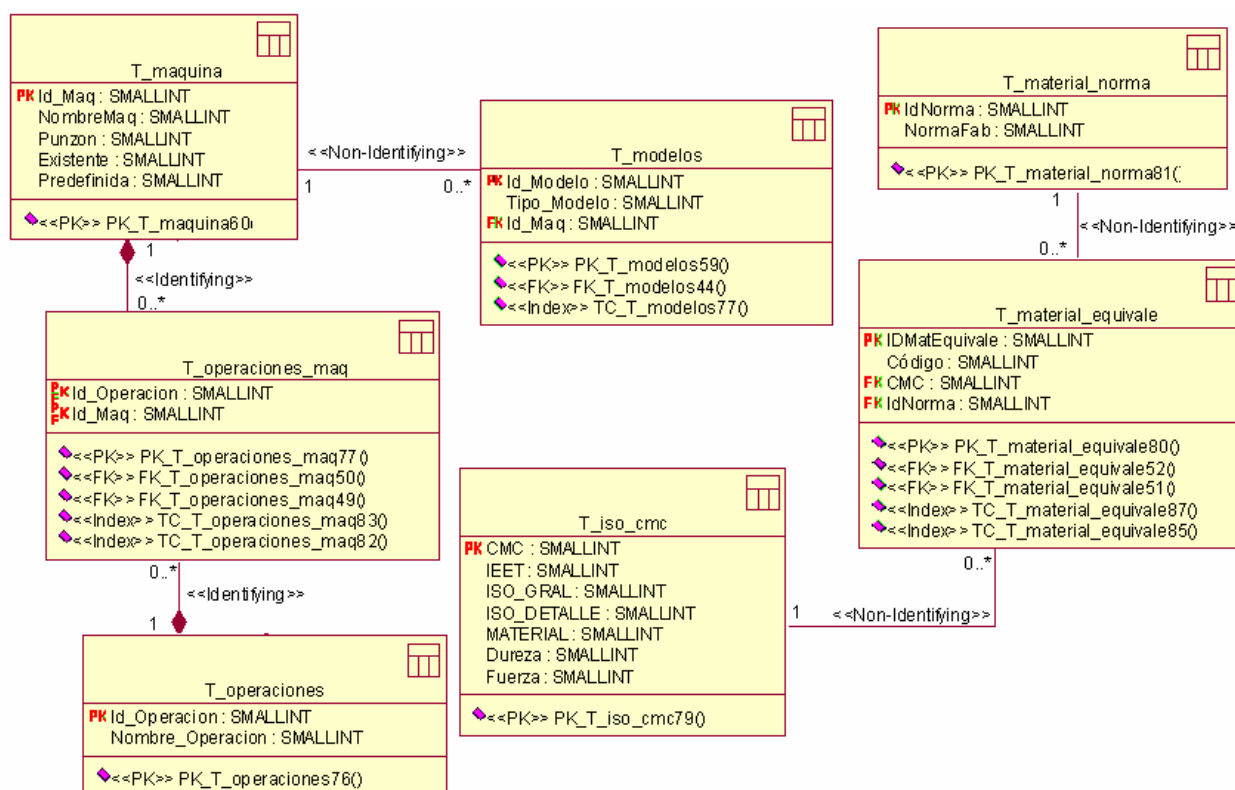


Figura 2.1. Diagrama del modelo de datos del sistema.

En la creación del modelo relacional de la Base de Datos se toma como punto de partida las entidades más generales que interactúan en las actividades CAD/CAPP/CAM, las cuales son los diferentes procesos o etapas por las que transita un artículo. Estas se pueden jerarquizar como se muestra en la Figura 2.2, de modo que sea más fácil tratar este tipo de información en el modelo relacional.

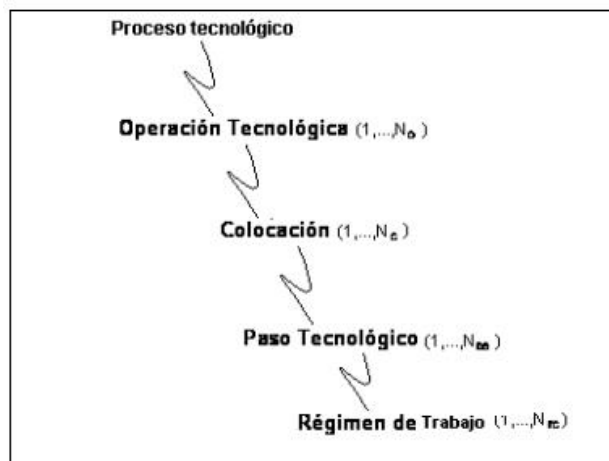


Figura 2.2. Jerarquía de tareas de los procesos tecnológicos en general.

El simulador reutiliza la información de la tecnología de maquinado generada por el CAPP, para efectuar su representación y le permite a los usuarios una mejor comprensión del proceso. De igual manera provee un entorno donde los usuarios pueden definir una ruta del proceso específica. Esta información es descrita en una estructura de datos de tipo árbol. Cada proyecto contendrá la información del modelo final, operaciones, herramientas y programas NC cargados.

En el caso concreto de las operaciones, como parte integral del proceso tecnológico que se realiza en un puesto de trabajo, se describe toda la información relacionada a la máquina empleada, la cual a su vez es seleccionada de la Base de Datos o como selección previa del CAPP. Así mismo se reflejan las diferentes transiciones u operaciones intermedias (pasos tecnológicos), de modo, que el simulador sea capaz de estructurar el proceso de maquinado.

En este punto, el análisis de las colocaciones y la representación de dispositivos de sujeción juegan un papel importante en el proceso de la simulación para el estudio de interferencias y colisiones. El siguiente esquema muestra una posible configuración del árbol jerárquico incorporada al sistema.

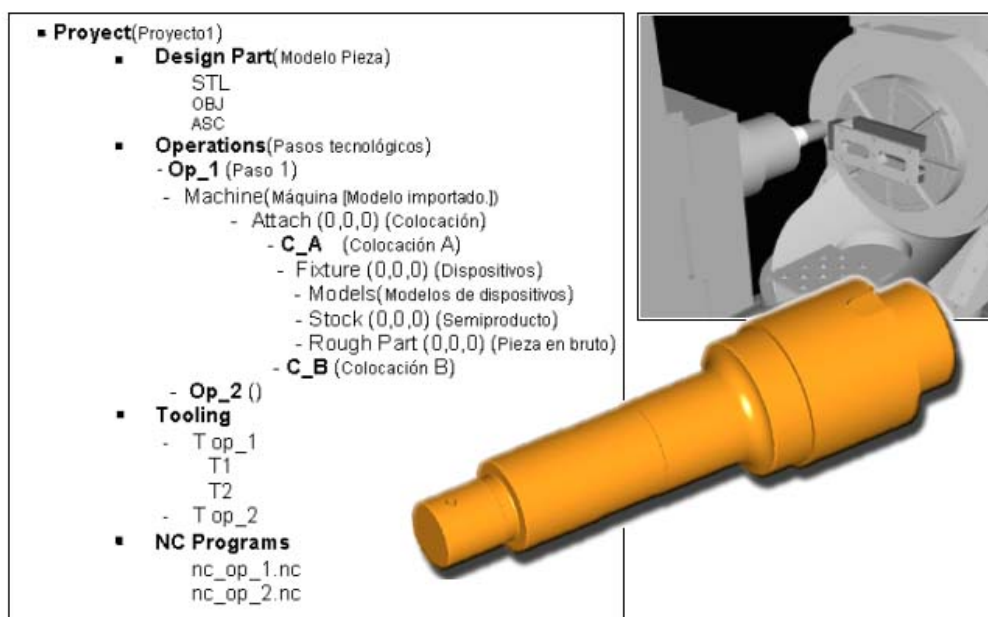


Figura 2.3. Árbol descriptivo del proyecto.

En esta estructura también se encuentra la información asociada al herramiental utilizado para las operaciones a realizar. El proceso de automatización en la selección de las herramientas se realiza por el sistema CAPP, las cuales son cargadas por el simulador. No obstante, los usuarios pueden actualizar la base de datos y cargar nuevas herramientas en este proceso. Es por ello de vital importancia contar con los modelos reales de las herramientas con la finalidad de representarlas y utilizarlas finalmente en el proceso de simulación. Este proceso de selección de herramientas dada su complejidad se aborda posteriormente. Por último se incorporan los ficheros que contienen los programas pieza a simular. Uno de los objetivos del proceso de simulación es precisamente comprobar la correcta programación CNC, la cual determina el resultado final del maquinado. El epígrafe siguiente describe la estructura general de dichos programas.

2.1.3 Estructura del programa pieza.

Un programa pieza es una secuencia de bloques que describe en detalles los movimientos de la máquina CNC, y debe de ejecutarse con la finalidad de manufacturar la pieza. Es importante mencionar que existen diferentes lenguajes de programación de máquinas CNC. Cada control NC, se diseña para aceptar un “dialecto” particular. De modo que un tipo de formato que trabaja correctamente en un control numérico puede ser cambiado ligeramente al correr en otro. Existen diferentes tipos de controles numéricos: FANUC, SINUMERIK, FAGOR. Cada uno se asocia a diferentes fabricantes, de hecho existen variaciones entre familias de un mismo fabricante. La literatura generalmente, debido a la diversidad existente, establece un formato estándar o código ISO-G el cual se asimila como base por la mayoría de los controles. En cada caso, el manual de programación de cada máquina CNC en particular, se consulta antes de realizar las operaciones de maquinado. Estos aspectos han de analizarse por los simuladores, de lo contrario producirían resultados incorrectos. Crear una serie de convenciones que admitan fácilmente introducir nuevos tipos de programas piezas escritos en *lenguaje de comando CNC* según el tipo de fabricante del control numérico, es una novedosa tarea, la cual incorporan algunas aplicaciones CAM, como el SURFCAM.

Los programas pieza a convención de este trabajo se dividen en tres partes fundamentales: Sentencia inicial, Sentencias intermedias y Sentencia final.

a) Sentencia inicial y programas secundarios, (apertura) Ejemplo. %..... y/o L.....
Signo % seguido de hasta 4 números. Sentencia L seguida de hasta 3 números enteros. Debe tenerse en cuenta que, desde un programa se puede llamar a un subprograma. Mientras que lo contrario NO es posible.

b) Sentencias intermedias (programa). Están formadas por palabras. Estas palabras se componen de un *DIRECCIONAMIENTO* (letra) y un *NÚMERO*. (Si el número no está presente se interpreta como cero). Ejemplo N25 G2 X
Una sentencia intermedia está compuesta por varias palabras, y termina con un L_F (final de línea).

c) Sentencia final (cierre). Ejemplo. M30 y M17

El programa pieza puede introducir al control por teclado en diálogo del operador, o desde un fichero formato texto (.TXT) a través de un cable de comunicación RS232. Estos mismo son los ficheros que carga el simulador.

El alfabeto general de las palabras que se utilizan para programar se describe a continuación.

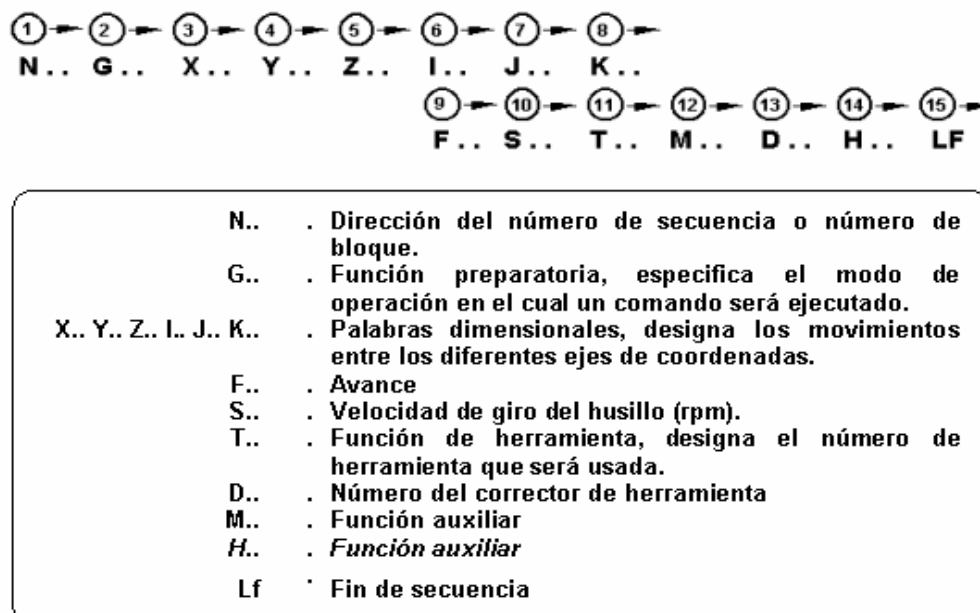


Figura 2.4. Alfabeto general para los programas pieza.

En el caso de las funciones dentro de un programa se clasifican en modales y no modales. Modal significa que, las funciones G una vez programadas permanecen activas mientras no sean anuladas mediante otra función G incompatible o mediante las funciones M02 o M30. En un mismo bloque se pueden programar varias funciones G, en el caso de que sean incompatibles, el CNC asume la última programada.

Para la descripción de las funciones G y M, véase Anexo 1.

2.1.3.1 Intérprete de lenguaje de comandos.

Como se mencionó anteriormente, la simulación de las trayectorias, así como el control de las herramientas y regímenes de corte para el maquinado de las piezas, depende de los programas NC cargados. Una vez realizado este paso, el programa de simulación debe ser capaz de interpretar y estructurar la información proveniente de dicho programa pieza. Para ello es necesario analizar e interpretar las sentencias introducidas por el usuario donde se usan técnicas de procesamiento de *lenguaje de comando CNC*, en las que se incluyen: reconocimiento de palabras, y el análisis y significado de estas. Esto conlleva a la definición del siguiente problema:

2.1.3.2 Definición del problema.

A partir de un texto de entrada en *lenguaje de comando CNC*, se debe interpretar el mismo, y simular sobre la base de lo que significa cada uno de los comandos. Estos comandos se pueden crear a partir del lenguaje específico asociado al control numérico (NC) de que se trate.

La estructura general del problema se puede representar mediante el siguiente diagrama.

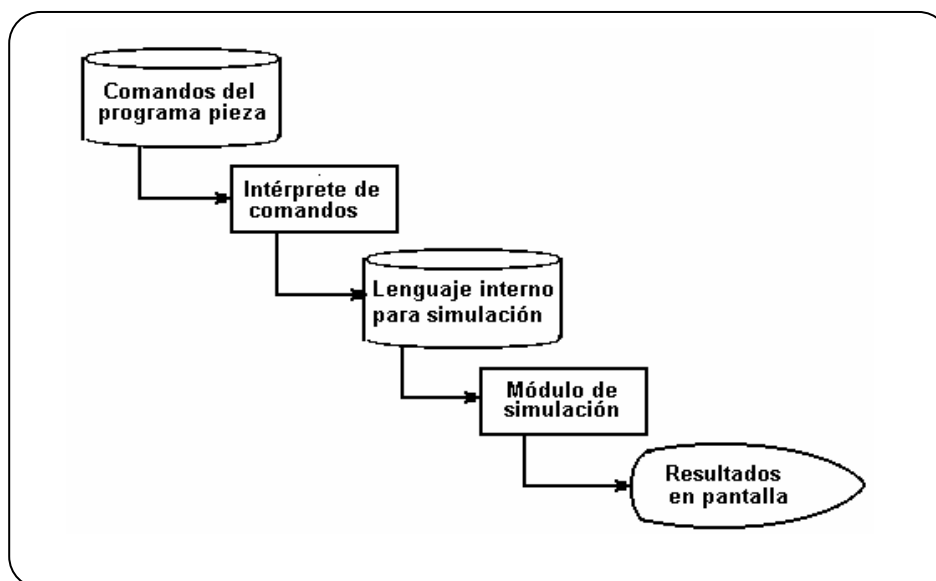


Figura 2.5. Secuencia de análisis del sistema de simulación.

Cuando el sistema CAPP genera el código NC, o el usuario introduce el texto de entrada para simular un programa pieza en lenguaje de comandos CNC, la aplicación analiza los comandos *léxica y sintácticamente*, a partir de una serie de reglas gramaticales y un diccionario de datos interno disponible que especifican el lenguaje que acepta el programa pieza.

El programa de simulación debe de analizar cada uno de estas funciones. De esta forma si los comandos generados por el CAPP o el usuario son correctos, el programa simulador los transfiere a un *lenguaje intermedio*. Posteriormente éste lenguaje intermedio se transforma en funciones que simulan la ejecución de las acciones correspondientes (operaciones de maquinado), o sea, las de movimiento de objetos en pantalla (pieza, herramientas, trayectorias). Este lenguaje intermedio está asociado a los intérpretes de los programas cargados en los ficheros. Un intérprete es un pequeño programa el cual convierte los programas NC a instrucciones o datos de movimiento. Una técnica empleada por los simuladores para tratar con el problema de los diversos fabricantes, es escribiendo intérpretes (postprocesadores) separados para diferentes máquinas.

2.1.3.3 Análisis léxico y sintáctico del lenguaje de comandos CNC.

La utilización del *lenguaje de comando CNC* presenta el inconveniente de la variedad de tipos de programa pieza asociados a cada fabricante de control numérico, lo cual dificulta el análisis de las sentencias o *comandos*. De igual manera analizar todas las posibles sentencias de un lenguaje asociado a un fabricante en particular, es una tarea complicada, si se quieren aceptar todas las posibles oraciones a introducir por el usuario. Sin embargo, el vocabulario de descripción del *lenguaje de comando CNC* y la representación gráfica es restringido, como consecuencia, son también, las reglas gramaticales y el "sentido común" para interpretar las líneas del programa pieza (frases u oraciones).

El alcance del problema de la simulación gráfica del programa pieza se limita a las siguientes características: El texto de entrada se expresa en un lenguaje, el cual;

se crea a partir de una *gramática* también limitada y estricta, conforme a un vocabulario permitido de tokens (palabras). La gramática, es específica y se refiere al conjunto de reglas que indican los patrones permitidos en que las palabras pueden aparecer. El significado del texto de entrada (sentencias del programa pieza en lenguaje de comando CNC) se representa luego en forma de sentencias en un lenguaje interno (*texto interno*).

Para poder elaborar el análisis de los comandos que aparecen en el programa pieza, se elabora la siguiente gramática, la cual especifica el lenguaje que se acepta por el programa.

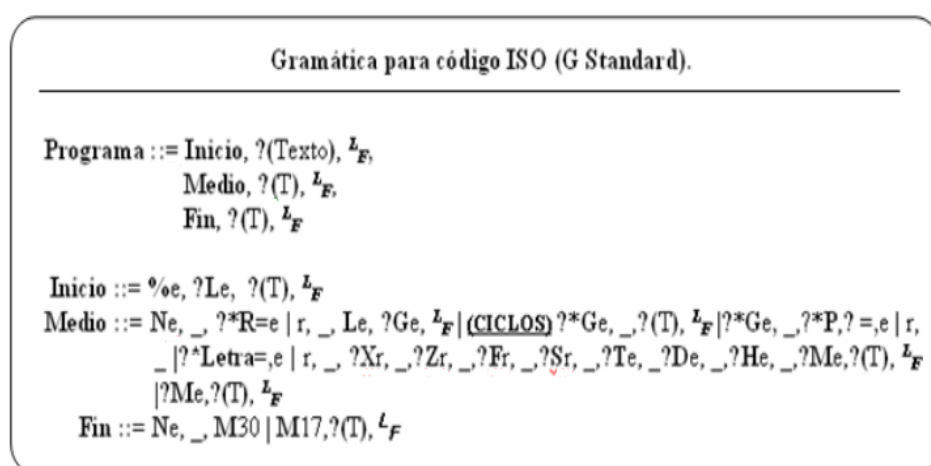


Figura 2.6. Definición de la estructura del programa según su gramática.

Esta gramática corresponde al siguiente glosario.

Glosario.

1. , seguido de
2. * posiblemente repetidos
3. ? posiblemente ausentes
4. | o exclusivo
5. ::= es
6. < > delimitadores en el texto de la gramática
7. _ espacio
8. e +/- entero
9. r +/- real

10.T texto de comentario

Una vez definida la gramática, se describen las diferentes partes del programa pieza. Para realizar el análisis de comprensión del programa pieza en lenguaje de comando CNC, se utiliza como base la siguiente estructura:

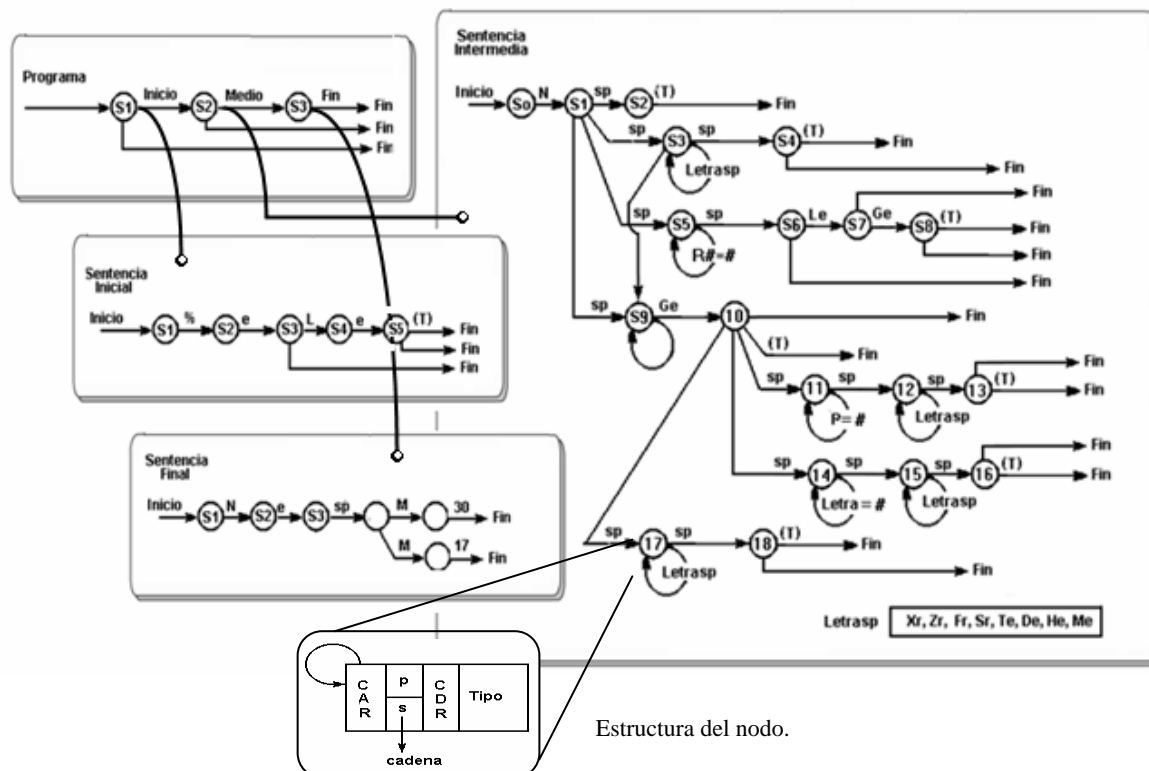


Figura 2.7. Estructura del programa.

2.1.3.4. Análisis semántico.

El *parsing* es el proceso de determinar si una cadena de componentes léxicos puede ser generada por una *gramática*. Es por ello que en el estudio de este problema, es útil construir un *árbol de análisis sintáctico* para poder garantizar que la traducción de los *comandos* sea correcta. O sea que, el *árbol de análisis sintáctico* indica cómo del símbolo inicial de una *gramática* se deriva una cadena válida del lenguaje.

Para realizar el análisis semántico de las líneas de programa pieza se implementa un diccionario de datos que contiene un determinado número de palabras o símbolos terminales junto a ciertos parámetros y sus diferentes tipos de interpretaciones (es decir sus características gramaticales). Esto permite analizar cualquier tipo de comando que se introduzca.

Se observa que cada nodo a insertar en el árbol sintáctico (cada elemento de la lista), contiene la información de éste y su clasificación según la gramática. Además, cuenta con punteros a los nodos anteriores y posteriores, para así formar la lista. Una vez que se realiza el análisis léxico de las sentencias y se obtienen los tokens (palabras), estos se analizan por el parser y se introducen en la lista, conforme ocurre el análisis.

En el proceso de *parsing* (análisis) se analizan de modo *recursivo* los *tokens* de entrada. Este proceso recursivo verifica si los tokens pueden ser generados por la gramática, y si es así, se llama a la función *MkCONS*, que crea e inserta un nuevo nodo en la lista, de lo contrario se ha producido una falla en el análisis.

A continuación se presenta un ejemplo válido para la gramática y su representación gráfica mediante un *árbol sintáctico*.

Ejemplo: *N25 G X10 Z2 D3 M8 ^L_F*

Interpretación de la sentencia anterior:

- N25* Número de línea o bloque.
- G* Desplazamiento rápido de la herramienta.
- X10* Coordenada sobre el eje X.
- Z2* Coordenada sobre el eje Z.
- D3* Tiene en cuenta los valores del corrector de la herramienta 3 para evitar un eventual choque de la herramienta con la pieza o el husillo.
- M8* Encendido del líquido refrigerante.
- ^L_F* Final de línea o sentencia.

La representación gráfica del árbol generado es la siguiente:

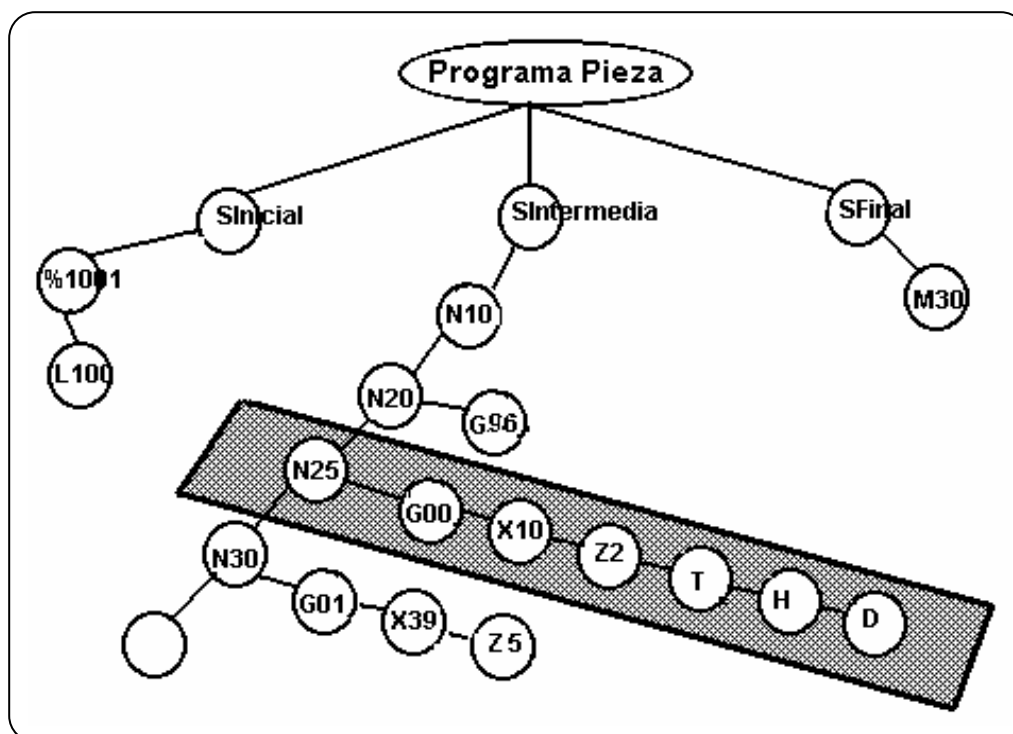


Figura 2.9. Programa pieza descrito en una estructura tipo árbol.

Para aplicaciones distribuidas y principalmente aquellos simuladores que se encuentren sobre Web, este proceso de análisis, generalmente se realiza en el lado del servidor. Los programas piezas cargados son enviados al servidor, y este chequea su validez, que luego envía reportes a los usuarios. La Figura 2.10 muestra un fragmento del código que chequea la gramática. Tanto la carga de los ficheros CNC como su comprobación gramatical, requiere del empleo de tecnologías Web Service o de las tecnologías para objetos distribuidos mencionados en el Capítulo 1. En el capítulo 3 se describe la arquitectura planteada para el desarrollo del simulador sobre Web.

```

comandos=new ArrayList();
StringTokenizer st = new StringTokenizer(code.toString(),"NGXZMFSIKR",true);
String token;
double X=100.0; //valor de X
double Z=100.0; //valor de Z
double I=0.0; //valor de X
double K=0.0; //valor de Z
double R=0.0; //valor de X

while (st.hasMoreTokens()) { //itera por todos los tokens
    token=st.nextToken();
    if (token.compareToIgnoreCase("G")==0){ //Comando G
        token=st.nextToken();
        if (token.compareToIgnoreCase("00")==0 || //Comando G00
            token.compareToIgnoreCase("0")==0){

            while(token.compareToIgnoreCase("N")!=0){//coger todos los parametros
                token=st.nextToken();
                if (token.compareToIgnoreCase("X")==0){ // parametro X
                    token=st.nextToken();
                    X=(Double.valueOf(token)).doubleValue();
                } else
                if (token.compareToIgnoreCase("Z")==0){ // parametro Z
                    token=st.nextToken();
                    Z=(Double.valueOf(token)).doubleValue();
                } //else otros parametros del G00

            } //del while
            comandos.add(new G00(X,Z)); //adiciono el comando a la lista
        } else
            ...
    }
}

```

Figura 2.10. Fragmento de código para el chequeo de comandos.

2.1.4 Análisis y representación de trayectorias.

El análisis y la representación de trayectorias es uno de los problemas principales que debe tratar la simulación geométrica. Para ello el simulador debe incorporar un entorno gráfico que contenga toda la información relacionada al sistema de coordenadas empleado y los puntos de referencias correspondiente al entorno real de trabajo, para el control de los movimientos de la herramienta.

2.1.4.1 Sistema de coordenadas cartesianas.

Lo primero a considerar son los ejes de movimiento en los tornos CNC. Solamente se consideran los dos ejes de movimientos principales cuando se trata de tornos CNC. Estos ejes son el Z y el eje X, como se observa en la Figura 2.11. Existen centros de torneados más complejos que tienen cuatro o más ejes. El eje Z coincide con el eje de rotación del husillo principal, y a su vez describe el movimiento longitudinal de la herramienta.

Los programas para tornos CNC, pueden ser escritos para mover la herramienta en diferentes modos: Absoluto, Incremental, o mixto. Además, el eje X puede ser programado en términos del diámetro o del radio de la pieza.

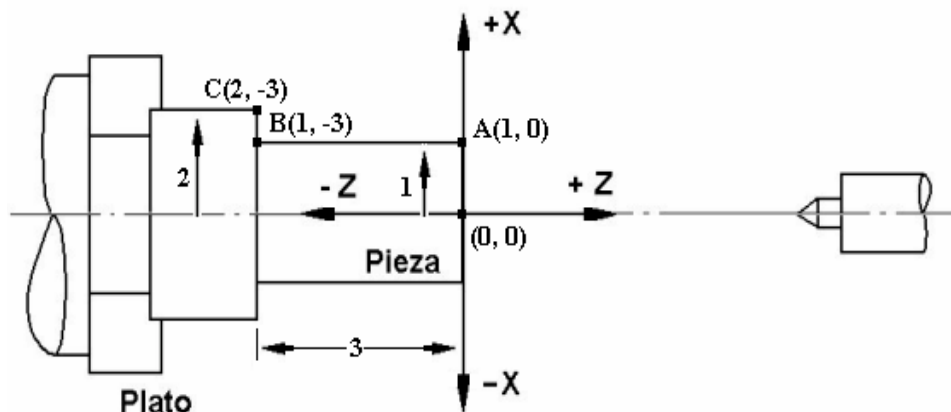


Figura 2.11. Programación por radio en el sistema de coordenadas cartesianas.

2.1.4.2 Puntos de referencia.

Existen tres importantes puntos de referencia o puntos de origen para los tornos CNC. (A): Punto de referencia (cero máquina), el cual es definido por el fabricante de la máquina; (B): Punto de cambio de herramienta; (C): Punto de origen del programa (cero pieza). Este último es el punto cero desde el cual son definidas todas las dimensiones del programa pieza. Tanto este como el punto de cambio de herramienta son configurados por los tecnólogos en el control de la máquina CNC, y mantienen el control de posicionamiento de la herramienta en cada instante del proceso. En el punto de cambio de herramienta, también se configuran dimensiones asociadas a la herramienta, según su tipo, longitud, ancho, referencia respecto al centro del radio de la punta o a sus aristas. Toda esta información debe ser incluida en el simulador con la finalidad de establecer el posicionamiento exacto en los movimientos de las herramientas.

La Figura 2.12 refleja cada punto de referencia.

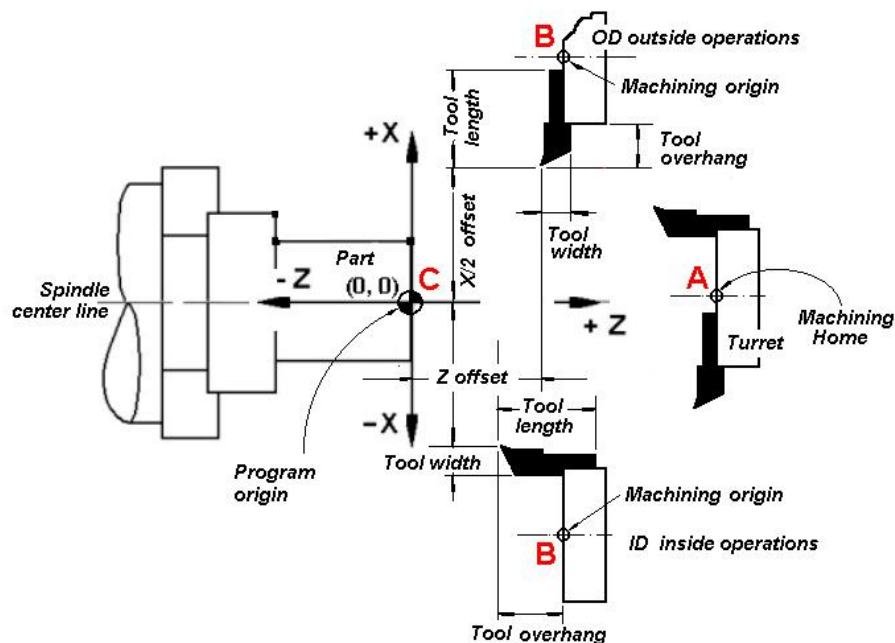


Figura 2.12. Puntos de referencia en los tornos CNC.

2.1.4.3 Interpolación lineal.

Para los tornos CNC, la interpolación lineal corresponde a los movimientos de la herramienta a lo largo de una línea recta programada con un avance específico. Este tipo de interpolación es usada para realizar operaciones de torneado de desbaste, refrentados, entre otras. En el caso de posicionamiento rápido, también la herramienta se desplaza linealmente pero sin remover material de la pieza.

El comando asociado para este caso es la función G00, que garantiza un movimiento rápido de la herramienta con el avance rápido establecido en la puesta a punto de la máquina. Para el caso de interpolación lineal la sintaxis general se describe en el siguiente recuadro, se asume que se programa por el centro del radio de la punta de la herramienta.

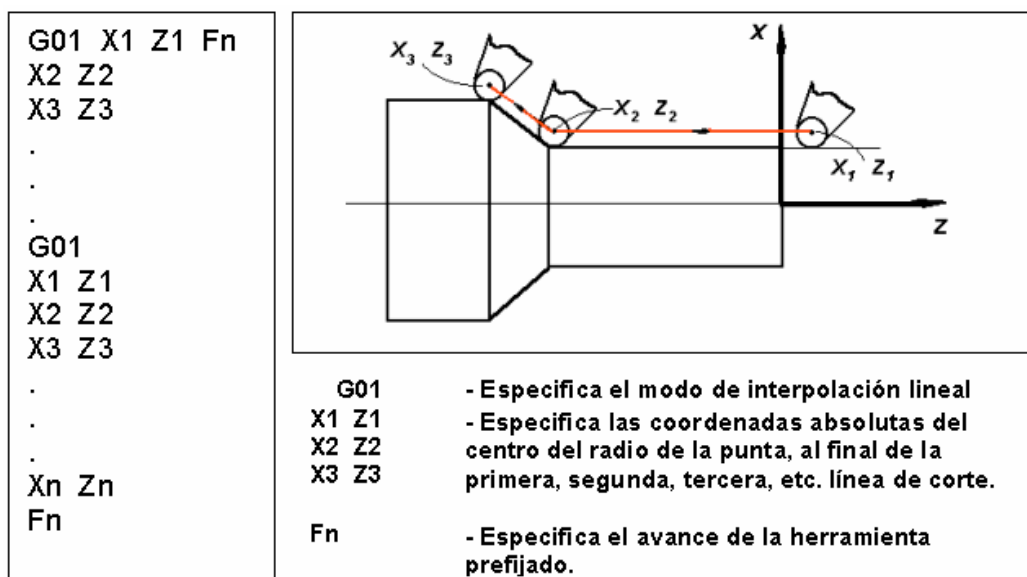


Figura 2.13. Sintaxis general de la interpolación lineal.

Para programar el contorno lineal de una pieza determinada es necesario calcular el perfil equidistante o el denominado Offsets. Como datos de entrada, serán las coordenadas de cada punto del perfil, y el radio de la punta de la herramienta. Para este cálculo es fundamental tener dominio del cálculo trigonométrico y análisis de geometría en general, véase un ejemplo de cálculo en el Anexo 2. La función G01 es modal e incompatible con G00, G02, G03, G33.

2.1.4.4 Interpolación circular.

El caso de la interpolación circular se asocia a los movimientos circulares de la herramienta tanto en sentido horario (G02), como antihorario (G03). Esta interpolación puede calcularse en función de las coordenadas del centro del arco, o en función del radio del arco. La Figura 2.14 muestra la sintaxis general de este comando.

Las funciones G02 y G03 son modales e incompatibles entre sí, a su vez los valores de I, K se programan con signo. Es necesario programarlos siempre, aunque tengan valor cero. El formato de un bloque para programar una interpolación circular en coordenadas polares es el siguiente:

N4 G02(G03) An In Kn Fn

An: Representa el ángulo con respecto al centro polar del punto final del arco.

G02 ó G03 Xn Zn In Kn Fn

G02: Interpolación circular en sentido Horario

G03: Interpolación circular en sentido antihorario.

Xn Zn: Especifica la coordenada absoluta X y Z del centro del radio de la punta de la herramienta al final de arco de trayectoria.

In Kn: Especifica la distancia incremental X y Z con signo $-/+$, del centro del radio de la punta, en el punto inicial del arco al centro del arco.

Fn: Especifica el avance de la herramienta.

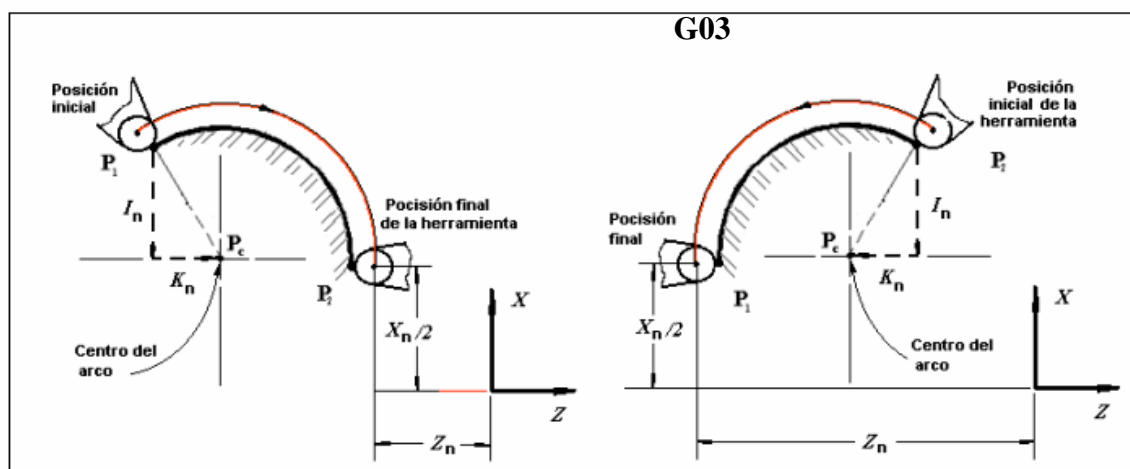


Figura 2.14. Sintaxis general de la interpolación circular.

La interpolación circular también puede programarse en función del radio del arco R en lugar de las coordenadas (I y K) del centro, Figura 2.15a. El formato es el siguiente:

G02(G03) Xn Zn (+/-)Rn Fn; Donde: X y Z coordenadas del punto final del arco de trayectoria y R es el radio del arco.

Si el arco de la circunferencia es menor que 180° , el radio R se programará con signo positivo, y si es mayor que 180° , el radio será negativo. La Figura 2.15b representa este convenio. Siendo P_0 el punto inicial del arco y P_1 el punto final, con el mismo valor de radio, existen cuatro arcos que pasan por ambos puntos.

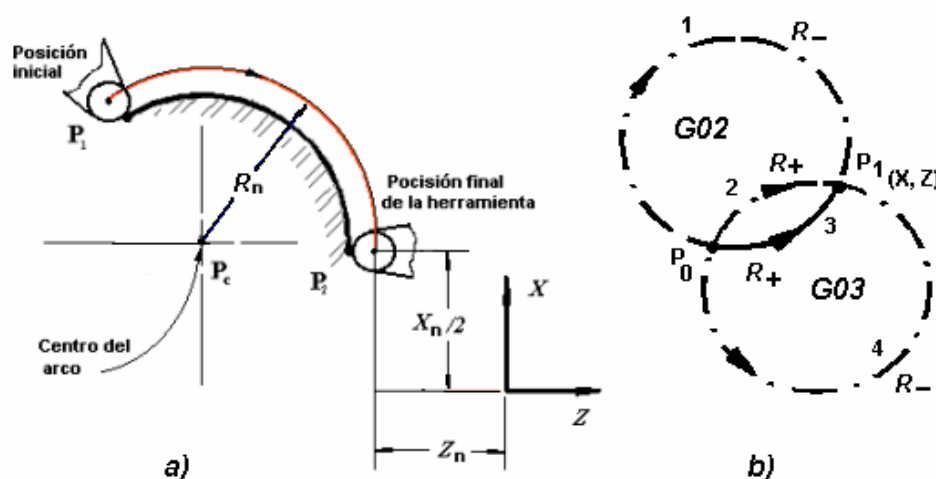


Figura 2.15. a) Interpolación circular en función del radio. b) Convenio de signos

Si se programa un círculo completo con la programación del radio, en el CNC se visualizará un error, dado que existen infinitas soluciones.

2.1.4.5 Trayectoria de la herramienta compensada.

En los trabajos habituales de mecanizado, es necesario calcular y definir la trayectoria de la herramienta teniendo en cuenta las dimensiones de la misma (longitud y radio), de forma que se obtengan las dimensiones de la pieza deseadas. Hasta ahora las trayectorias han sido calculadas a partir del centro del radio de la punta de la herramienta, pero se puede programar también en función de la arista de la punta. Cuando se programa en función de la arista de la punta de la herramienta, la trayectoria programada coincide favorablemente con la geometría de la pieza.

La restricción principal es que la geometría puede solamente consistir en líneas horizontales o verticales. De lo contrario se introducen errores para los contornos

inclinados o en forma de arcos. Por lo que resulta necesario aplicar una compensación adicional. Este fenómeno se observa en la Figura 2.16 a).

Los problemas con la programación según la arista de la herramienta son resueltos al aplicar la compensación del radio de la punta de la herramienta, Figura 2.16 b).

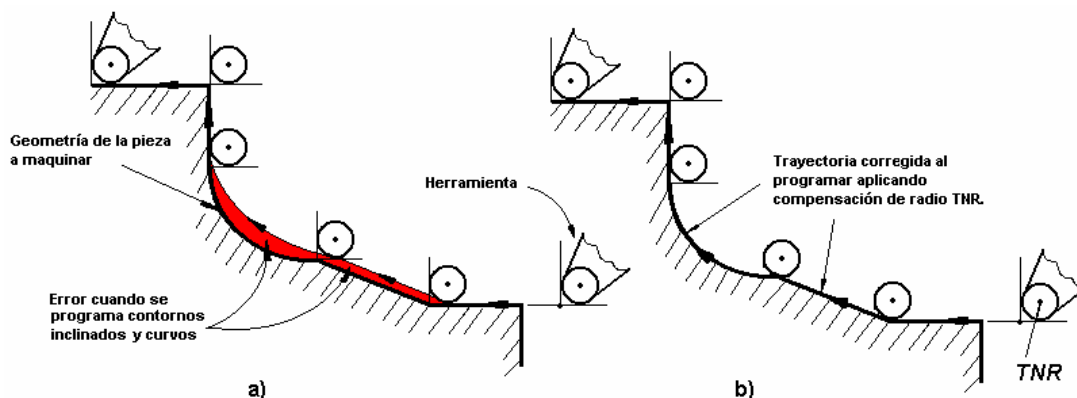


Figura 2.16. a) Error de maquinado al programar según la geometría de la pieza directamente sin compensar. b) Trayectoria con compensación de radio.

La compensación de longitud y radio de herramienta, permite programar directamente el contorno de la pieza sin tener en cuenta las dimensiones de la herramienta. El CNC calcula automáticamente las trayectorias a seguir, a partir del contorno de la pieza y de las dimensiones de las herramientas almacenadas en la tabla de herramientas. La tabla también almacena los códigos de forma (F) de cada herramienta. Esta es una información importante que debe definirse antes de usar la compensación.

La información relevante de la tabla de herramienta es la siguiente:

X, Z: Longitud de la herramienta; I, K: Corrección de la longitud herramienta.

R: Radio de la punta.

F: Vector de la punta de la herramienta.

Este vector relaciona un punto en correspondencia con las tangencias y el centro del radio de la punta de la herramienta. El vector se usa para indicar al control la posición correcta de la herramienta durante el uso de los comandos de compensación. Para observar el valor de cada vector en correspondencia a la forma de la herramienta véase Anexo 3.

Para la compensación de radio existen tres comandos o funciones preparatorias.

G40: Anulación de la compensación de radio de herramienta.

G41; G42: Compensación de radio a izquierda y derecha respectivamente. Véase Figura 2.17.

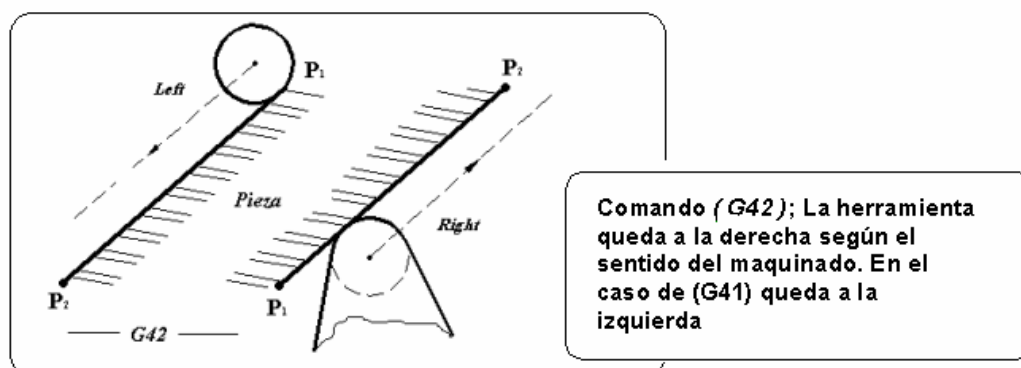


Figura 2.17. Comandos de compensación G41 y G42.

El sistema de simulación realiza los cálculos para determinar la trayectoria de las herramientas asociadas a cada comando G41 o G42, como mismo lo realiza el control CNC. Esta problemática se resume a un cálculo de puntos. Para la solución a la determinación de los puntos, que intervienen en la obtención de la trayectoria compensada, se tiene en cuenta tres casos en dependencia del ángulo α el cual siempre está en el lado opuesto al mecanizado y depende del sentido de mecanizado (**Right, Left**), del comando de compensación que esté activado (**G41 ó G42**) y de las entidades que definen el ángulo (**Line-Line; Line-Arc; Arc-Line; Arc-Arc**).

Para la determinación de los puntos, se implementaron funciones en Java y C++.

El ángulo (α), se calcula mediante la función *Angle3P*:

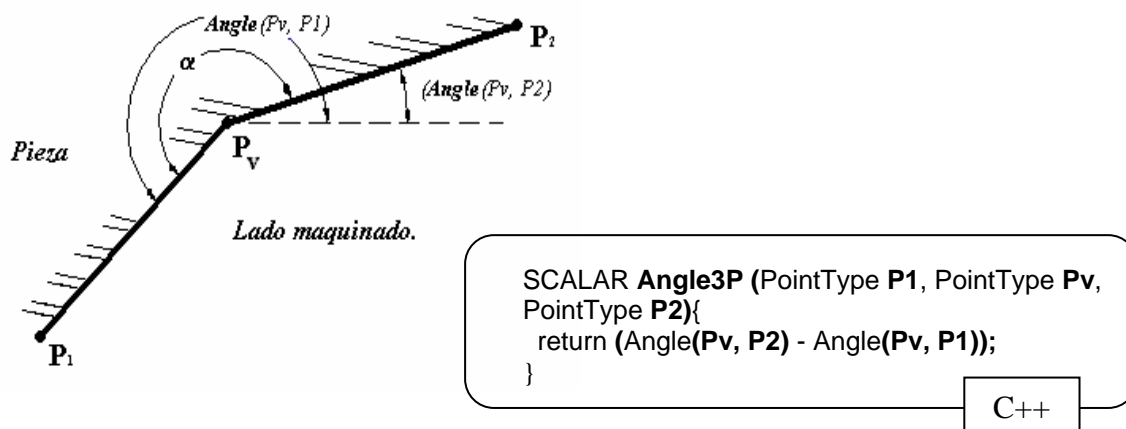


Figura 2.18. Ángulo de análisis para el cálculo de puntos.

Si la función retorna un valor negativo, el sentido de mecanizado es *derecho* y el comando de compensación es **G42** entonces: $\alpha = \text{fabs}(\alpha)$; si el valor retornado es positivo entonces; $\alpha = 360 - (\alpha)$;

Según el ángulo α calculado, se analizan tres casos diferentes.

- $(\alpha \geq 180^\circ)$
- $(90^\circ \leq \alpha < 180^\circ)$
- $(\alpha < 90^\circ)$

Para ver el análisis de estos casos se debe de consultar el Anexo 6.


2.1.5 Operaciones básicas y selección de las herramientas de corte.

Otro elemento de consideración es la capacidad de simulación de diferentes operaciones de torneado. En este aspecto es determinante la selección correcta de las herramientas a emplear. Este proceso de selección de herramientas es realizado por el sistema CAPP, del cual, el simulador CNC importa la lista de herramientas seleccionadas previamente, de la Base de datos. Para realizar la

simulación de las diferentes operaciones, cada herramienta ha de ser indexada y dicho código corresponde con el escrito en el programa CNC.

Entre las operaciones básicas que se realizan en el torneado se encuentran: Refrentado, Cilindrado, Ranurado, Tronzado, Barrenado, Taladrado y Roscado. Algunas de estas operaciones de corte son ejecutadas en las superficies exteriores de la pieza (OD), mientras que otras se realizan en las superficies interiores (ID). Acerca de los diferentes tipos de operaciones véase Anexo 5.

Con respecto a la selección de las herramientas de corte, varios son los factores influyentes y criterios tecnológicos que determinan su selección. Los mencionados a continuación, pueden ser considerados como los principales:

- Material de la pieza (PMK).
 - Perfil de la pieza.
 - Tipo de operación.
- 
 - Sistema de sujeción de la plaquita.
 - Tamaño y tipo del portaplaquitas.
 - Forma de la plaquita
 - Tamaño de la plaquita.
 - Radio de punta.
 - Calidad de la plaquita.
 - Datos de corte.

Todos estos criterios son manejados por el CAPP para la correcta selección de las herramientas, sin embargo mucha de esta información tecnológica ha de reutilizarse en el simulador, ya que involucra diferentes tipos de representación. De igual manera es crucial para visualizar en el entorno de simulación, contar con los modelos tridimensionales de las herramientas reales, lo cual incluye plaquitas y portaplaquitas.

Algunos de los tipos de portaplaquitas existentes y dirección de maquinado, según el tipo de operación se ilustran en la figura siguiente. Para mas detalles véase el Anexo 4.

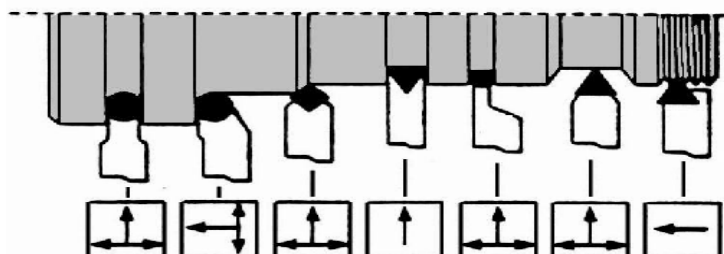


Figura 2.19. Diferentes formas o tipos de portaplaquitas y su dirección de maquinado.

2.2 Estructura y representación del modelado geométrico.

Otro aspecto a tratar en el simulador es el modelado geométrico de los elementos a representar. El modelado de sólidos es el procesamiento de la información geométrica y espacial de los objetos en tercera dimensión (3D o 2D). El modelado, es la construcción artificial de un objeto, para facilitar su análisis y estudio; los modelos contienen la información almacenada en archivos que permiten visualizar los objetos y son usados para representar formas que se utilizan en una escena.

2.2.1 Sistemas de coordenadas.

La representación de cuerpos sólidos de tres dimensiones (3D) en una computadora es como dibujar en dos dimensiones puntos de tres dimensiones. Lo que requiere de establecer un homomorfismo entre R^3 y R^2 . Para ello existen las proyecciones en perspectivas, que a continuación se describen:

Una típica representación gráfica de un sistema tridimensional, por computadoras, usa cinco sistemas de coordenadas diferentes, los cuales son reflejados en la siguiente figura.

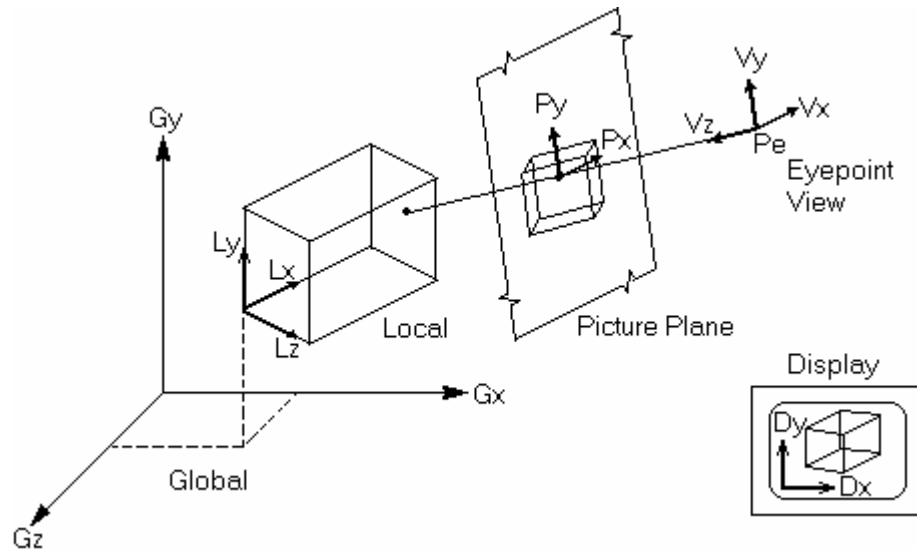


Figura 2.20. Sistemas de coordenadas para la visualización de los modelos.

2.2.2 Windows y Viewport.

Ambos conceptos han de manipularse para la representación tridimensional. Muchos de estos aspectos, así como las diferentes transformaciones matriciales que se le aplican a los modelos son incorporados como métodos de la Clase `gl4java.awt.GLAnimCanvas`.

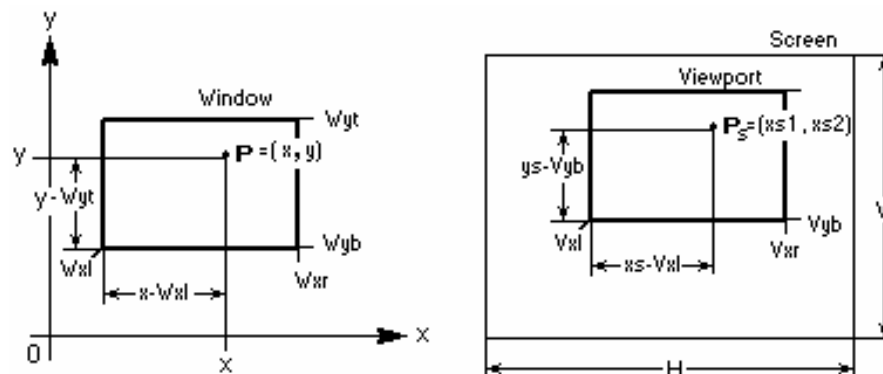


Figura 2.21 Representación conceptual del Windows y Viewport.

Window: Es una región del objeto que usted quiere visualizar y mapear en un sistema de coordenadas físico. Frecuentemente usted deseará solo mapear una porción del objeto en el plano, y para hacer esto simplemente encerrará esta porción en un área limitada rectangularmente denominada window.

Viewport: Es la región que nos permite visualizar físicamente en la pantalla del display, la imagen de lo que previamente ha sido enmarcado en la región de **window**, donde entonces podemos establecer una correspondencia semejante entre los puntos de ambas regiones.

Por medio de estas ecuaciones podemos reconciliar el sistema global, al sistema bidimensional de la pantalla.

$$x_s = \frac{x - W_{xl}}{W_{xr} - W_{xl}} (V_{xr} - V_{xl}) + V_{xl}$$

$$y_s = \frac{y - W_{yb}}{W_{yt} - W_{yb}} (V_{yt} - V_{yb}) + V_{yb}$$

$$\frac{(V_{xr} - V_{xl})}{W_{xr} - W_{xl}} \rightarrow \text{Factor de escala.}$$

La expresión anterior que no es más que la transformación del sistema de coordenadas lógico a físico la cual se obtiene del despeje de la siguiente ecuación.

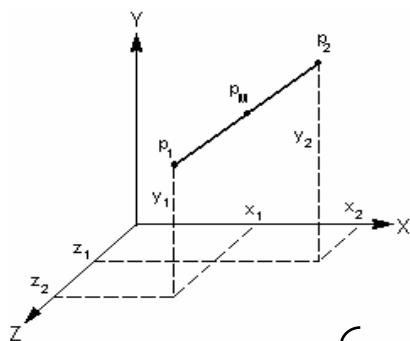
Si de la ecuación se despeja x entonces la conversión se realiza del sistema físico a lógico.

$$\frac{x_s - V_{xl}}{V_{xr} - V_{xl}} = \frac{x - W_{xl}}{W_{xr} - W_{xl}}$$

2.2.3 Elementos geométricos simples.

En la representación de los objetos más simples encontramos la línea y el círculo. Ambos elementos son empleados en la visualización de trayectorias.

La línea: Después del punto la línea es la próxima entidad más simple.



El gráfico muestra la definición de la línea recta en el espacio 3D.

La Línea es el lugar geométrico de los puntos que satisfacen las siguientes ecuaciones lineales, denominada

Ecuación paramétrica de la recta.

$$x = a_x u + b_x$$

$$y = a_y u + b_y$$

$$z = a_z u + b_z$$

$x, y, z \rightarrow$ Variables dependientes.

$u \rightarrow$ Variable paramétrica $u \in [0, 1]$

$a_x, a_y, a_z, b_x, b_y, b_z \rightarrow$ Constantes.

donde:

Si $u = 0$	Si $u = 1$
$x_0 = b_x$	$x_1 = a_x + x_0$
$y_0 = b_y$	$y_1 = a_y + y_0$
$z_0 = b_z$	$z_1 = a_z + z_0$

Sustituyendo en la ecuación anterior obtenemos.

$$x = (x_1 - x_0)u + x_0$$

$$y = (y_1 - y_0)u + y_0$$

$$z = (z_1 - z_0)u + z_0$$

Si se conocen los puntos extremos de un segmento de línea, con esta expresión se pueden determinar todos los puntos intermedios, o sea los que forman el segmento. La ecuación paramétrica le permite definir líneas en el espacio de dos o más dimensiones, y también le permite analizar la geometría o las relaciones espaciales entre puntos y líneas.

La mayoría de los compiladores del lenguaje C\C++ ó Java tienen incorporados a ellos bibliotecas gráficas, las cuales incluyen una gama de funciones a bajo nivel, que permiten la representación de las diferentes entidades gráficas. Entre ellas podemos mencionar a la función line, circle, arc. Sin embargo el conocimiento y dominio pleno de las ecuaciones, nos permite tener control y modelar a conveniencia el dibujo de la entidad, modificando su representación según la necesidad.

El siguiente gráfico muestra el comportamiento de la ecuación, para determinar la generación de los puntos intermedios de una recta, al conocer las coordenadas de sus extremos, al variar la variable paramétrica $u^{TM}[0, 1]$.

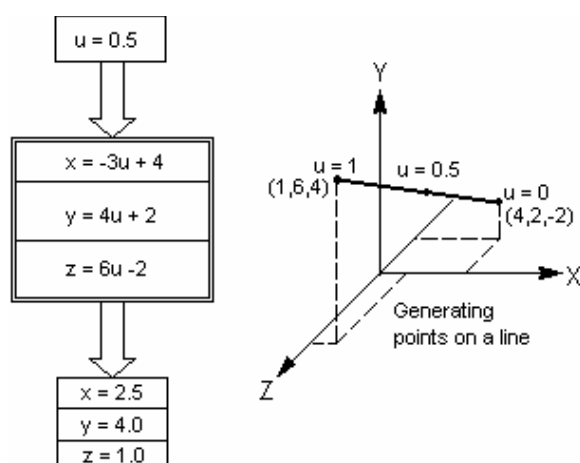


Figura 2.22. Cálculo de puntos mediante la ecuación paramétrica de la recta.

Usted puede ver en el gráfico, que al insertar los valores de $u = 0$ y $u = 1$, dentro de la ecuación paramétrica se obtienen precisamente, las coordenadas del punto inicial y final del segmento de línea. Luego si le introducimos al sistema otro valor de $u = 0.5$, obtenemos las coordenadas de un nuevo punto, **p (2.5, 4.0, 1.0)**, el cual pertenece al punto medio del segmento.

La representación paramétrica, antes desarrollada es una útil forma para el cálculo de los puntos de una línea. Una importante clase de puntos en una línea frecuentemente usado en la modelación geométrica y en la gráfica por computadoras, consiste en los puntos que tienen lugar en iguales intervalos a lo largo de la línea. La obtención de estos puntos se consiguen al variar la variable paramétrica u de la forma siguiente:

$$u = \frac{l}{D} \quad \text{Donde } D, \text{ es la longitud de la línea, multiplicada por un factor.}$$

El siguiente ejemplo muestra el código de una función que muestra el trazo de los puntos de una línea, utilizando la ecuación paramétrica de la recta en Java.

```
private JPoint parametricaLinea(double u, JPoint p0, JPoint p1){
    JPoint p=new JPoint();
        p.x=(p1.x-p0.x)*u+p0.x;
        p.y=(p1.y-p0.y)*u+p0.y;
        p.z=(p1.z-p0.z)*u+p0.z;
    return p;
}
```

De la misma manera mediante la ecuación paramétrica del círculo calculamos puntos para trayectorias circulares.

El parámetro t representa el ángulo de arco, y va de 0 a 2π

```
public JPoint parametricaCirculo(JPoint p0, double radio,
double t){
    double xp=p0.x+radio*Math.cos(t);
    double yp=p0.z+radio*Math.sin(t);
    return new JPoint(xp,0.0,yp);
}
```

2.2.4 Representación poliédrica de los modelos 3D mediante la estructura Winged-edge.

Los modelos consisten en información almacenada en archivos que permiten visualizar los objetos y son usados para representar formas que se utilizan en una escena (C. Moreno, 2000) [42]. Las formas simples pueden definirse a partir de expresiones analíticas, las formas más complejas requieren de métodos más complejos. En el caso de modelos geométricos; estos describen componentes con propiedades geométricas, como su distribución espacial y de forma, conectividad entre componentes (estructura), así como las relaciones entre los componentes (topología). Existen diversas técnicas de representación de modelos tratados en la gráfica por computadora (M. Patrikalakis, 2003) [41]. De igual manera existen diversas estructuras que pueden almacenar la toda la información necesaria para representar el modelo sólido.

En el desarrollo del presente trabajo, para la visualización de los elementos (piezas, Máquinas-Herramienta), se implementó una estructura del tipo **Winged-edge** [41], la cual se nutre de una representación poliédrica del modelo B-Rep o sólido a analizar. Un **Poliedro** es una porción de espacio limitada por polígonos planos, que tiene por elementos característicos las Caras, las Aristas y los Vértices:

Caras: son los polígonos que la limitan.

Aristas: son los lados de las caras.

Vértices: donde concurren tres o más caras.

En el caso de los poliedros se debe cumplir el Teorema de Euler como condición necesaria para validar este tipo de objeto.

Teorema de Euler: $\text{Vértices} + \text{Caras} = \text{Aristas} + 2$	{	Cada arista conecta 2 vértices Cada arista está compartida por dos caras A cada vértice llegan al menos 3 aristas Las caras no se pueden atravesar mutuamente
--	---	--

La fórmula de Euler generalizada es:

$$\mathbf{V} + \mathbf{F} - \mathbf{H} = 2 (\mathbf{C} - \mathbf{G})$$

$$\left\{ \begin{array}{l} \text{Vértices} + \text{caras} - \text{agujeros en las caras} = \\ \text{Aristas} + 2 (\text{n}^\circ \text{ de componentes separados} - \\ \text{género}) \end{array} \right.$$

De esta forma cada modelo sólido en 3D, se representa por la estructura siguiente.

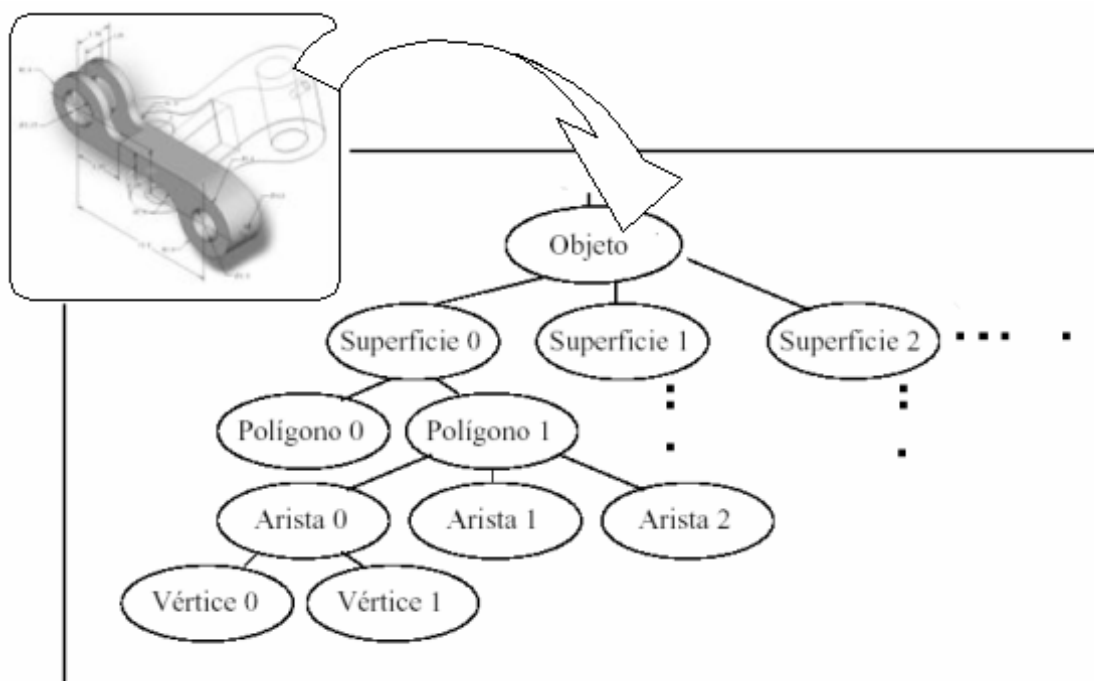


Figura 2.23. Estructura del modelo sólido.

En el caso de la Winged-edge, esta recoge la información mínima requerida para la representación de los modelos, donde se analiza el modelo topológicamente considerando las relaciones existentes entre los Vertices (V), Aristas(e) y Caras(f).

Para la mejor comprensión de esta estructura y a modo de ejemplo se analiza el siguiente tetraedro, del cual se extrae dicha información de forma estructurada.

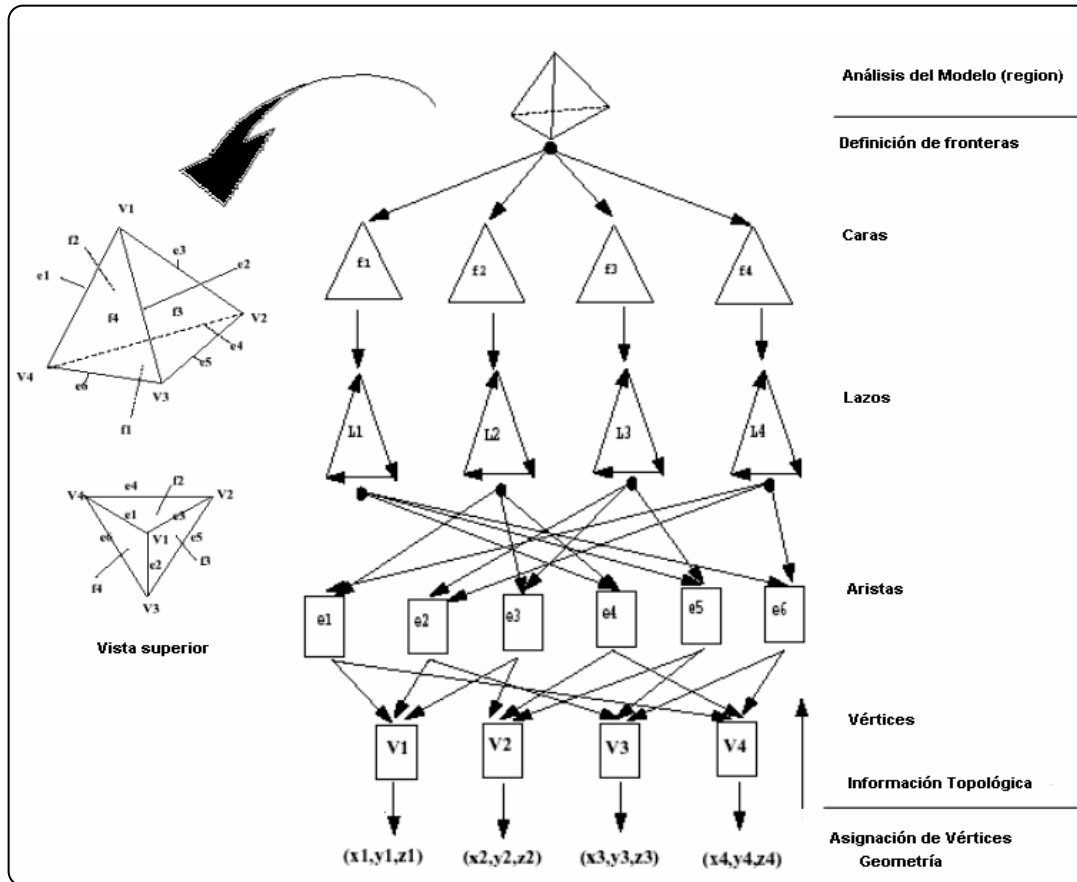


Figura 2.24. Representación de frontera de un modelo tetrahédrico.

Luego de este análisis la estructura Winged-edge es llenada a partir de la tabla de lados o aristas. De cada lado se resalta la siguiente información:

- Vértices de los lados ($V_{1,2}$).
- Cara derecha e izquierda ($f_{1,2}$).
- Lado predecesor y sucesor de la cara derecha (CCW_1, CW_1).
- Lado predecesor y sucesor de la cara izquierda (CCW_2, CW_2).

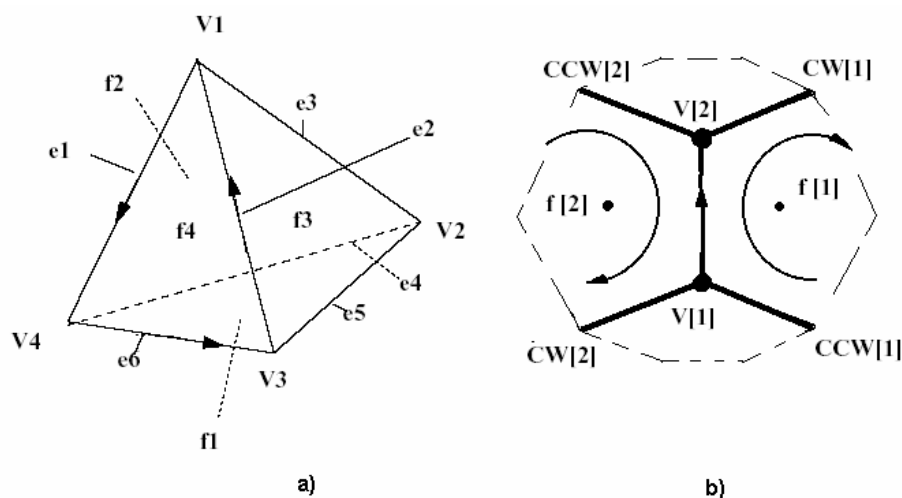


Figura 2.25. a) Modelo. b) Estructura de datos Winged-edge.

La siguiente tabla muestra las relaciones topológicas de cada lado.

Edge	V [1]	V [2]	f[1]	f[2]	CW[1]	CCW[1]	CW[2]	CCW[2]
e1	V1	V4	f2	F4	e4	e3	e2	e6
e2	V3	V1	f3	F4	e3	e5	e6	e1
e3	V1	V2	f3	F2	e5	e2	e1	e4
e4	V2	V4	f1	F2	e6	e5	e3	e1
e5	V2	V3	f3	F1	e2	e3	e4	e6
e6	V4	V3	f1	F4	e5	e4	e1	e2

Tabla 2.1. Descripción topológica del modelo.

Mediante la estructura Winged-edge es posible estructurar la información necesaria para la representación de los modelos importados. Los datos de los modelos son cargados de los diferentes ficheros que el simulador puede leer. En nuestro caso ficheros *.STL, *.OBJ, *.ASC. En estos ficheros se describe la topología del modelo atendiendo a sus vértices, caras, aristas y en algunos casos el vector normal a las caras. Una extensión necesaria debido a su importancia en el ámbito del modelado de sólidos es la lectura y representación de ficheros *.IGES y *.STEP.

2.3 Modelo de integración para la simulación geométrica del maquinado sobre Web.

El siguiente diagrama representa el modelo propuesto de integración, mediante el cual se garantiza un entorno multiusuario, para la simulación del proceso de maquinado. A su vez en el se muestran las diferentes entradas y salidas al sistema, así como las principales variables evaluadas.

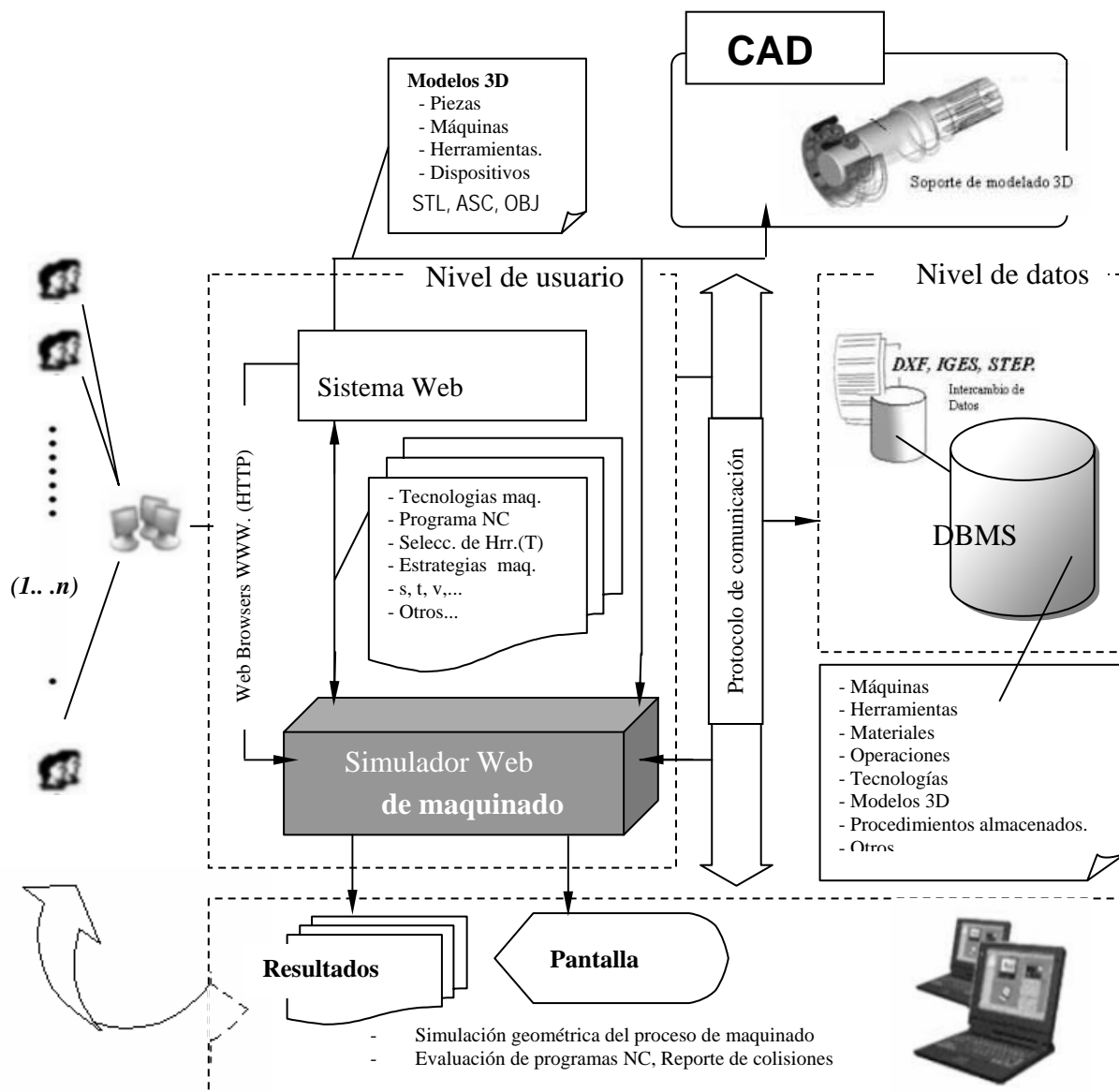


Figura 2.26. Modelo de integración para la simulación del proceso de maquinado.

2.4 Conclusiones del Capítulo.

A partir de los factores analizados en el desarrollo de la aplicación de simulación se establecen las siguientes conclusiones:

1. El hecho que la aplicación considere un entorno integrado con aplicaciones CAPP aporta una mayor comprensión del proceso de maquinado en cuanto a su simulación, y brinda la posibilidad de comprobar las tecnologías generadas.
2. Debido a la cantidad de fabricantes de controles NC existentes, la consideración de una gramática general para la simulación de los programas pieza, no es suficiente. Es preciso desarrollar plantillas donde el usuario pueda describir la gramática a utilizar, y el sistema sea capaz de interpretar.
3. La utilización de la estructura de datos Winged-edge, garantiza la representación de los diferentes modelos importados.
4. El modelo de propuesto para la simulación del proceso de maquinado, posibilita un entorno integrado multiusuario, con prestaciones tanto industriales como académicas.

CAPÍTULO 3

IMPLEMENTACIÓN DEL SISTEMA PARA LA SIMULACIÓN VIRTUAL DEL MAQUINADO SOBRE WEB.

3. Introducción.

En el presente capítulo se procede a describir concretamente las características y especificidades de la construcción del software. Para ello se inicia con la representación de la arquitectura del sistema, enmarcado en un entorno Cliente – Servidor. A su vez se analiza y describe el sistema desde su diagrama de clases y finalmente se detalla cada componente de la interfaz de usuario.

3.1 Arquitectura propuesta del sistema de simulación sobre Web.

La siguiente figura muestra la arquitectura del sistema de simulación, donde se resaltan los componentes y tecnologías usadas para su desarrollo.

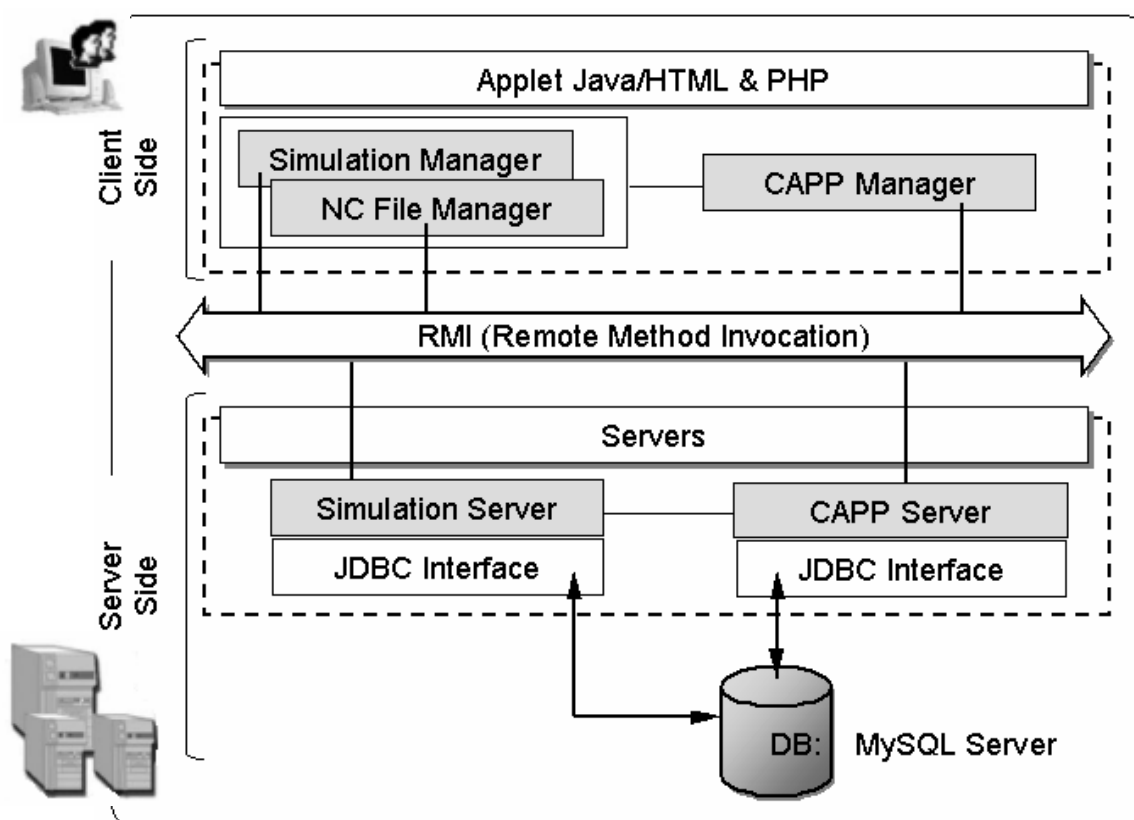


Figura 3.1. Arquitectura del sistema, visto en un entorno cliente - servidor.

En el lado del cliente los usuarios acceden desde un navegador Web con la interfaz del sistema mediante el uso de Applets. En este lado ambos sistemas, el CAPP y el simulador de maquinado interactúan. Normalmente la generación de tecnologías de maquinado así como los programas NC se obtienen desde el servidor CAPP, no obstante el simulador puede proporcionar la creación y edición de sus propios ficheros NC. En el lado del servidor de simulación se analiza el chequeo de sintaxis y se envían los resultados al cliente. También desde este entorno se garantiza la conectividad con el servidor de base de datos MySQL, donde se almacena toda la información tecnológica, Máquinas y Herramental necesario para el análisis del proceso de maquinado. La siguiente imagen muestra el entorno del sistema visto desde una arquitectura de tres capas.

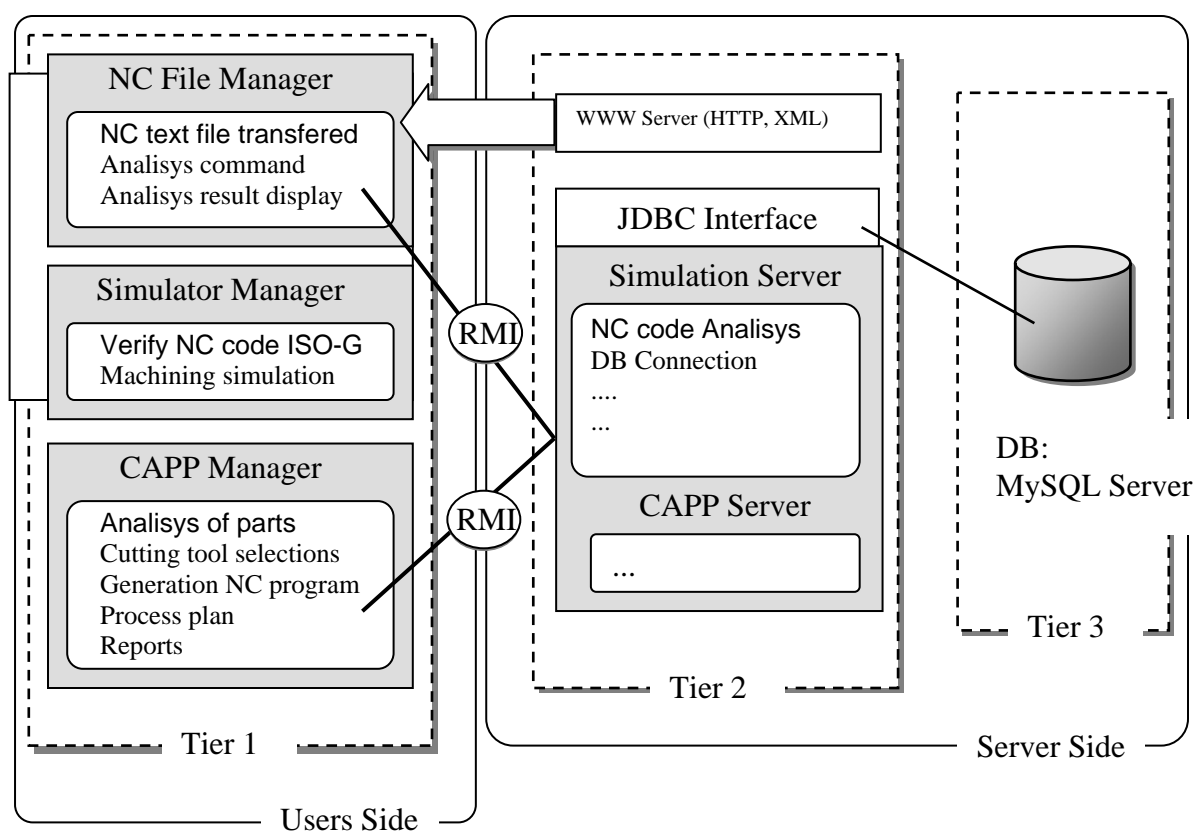


Figura 3.2. Arquitectura tres capas del sistema de simulación del maquinado.

3.2 Requerimientos funcionales y no funcionales del sistema.

3.2.1 Requerimientos funcionales.

Dentro de los requerimientos funcionales se incluyen las acciones que podrán ser ejecutadas por el usuario, las ocultas que debe realizar el sistema, y las condiciones extremas a determinar por este. “Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo.”[48]

A continuación se presentan los requerimientos funcionales determinados para la aplicación a elaborar.

R1. Visualizar el entorno de simulación donde se analiza el proceso de maquinado.

R2. Visualizar aquellos elementos considerados en la simulación, mediante los modelos importados (máquina herramienta, piezas, utillaje).

R3. Conectividad del sistema con el sistema CAPP y la base de datos tecnológica.

R4. Administración del árbol de operaciones, herramental y programas NC importados.

- Insertar las máquinas herramientas, desde la base de datos resaltando las características tecnológicas importantes para el ingeniero. Actualizar la base de datos con nuevas máquinas y herramientas desde el entorno cliente. Modificar las máquinas herramientas, adicionar o eliminar características tecnológicas.

R5. Chequeo sintáctico de los programas piezas importados.

- Posibilidad de edición de los programas NC. Entorno de múltiples documentos, con la finalidad de correr y comparar diferentes programas NC.

R6. Empleo de los comandos de simulación y edición gráfica, para el control y análisis del proceso: Inicio, Pausa, Paso a paso, Parada. Selección de vistas, Zoom, Pan, Orbit. Tratamientos de sombreados de los modelos importados. Wire, Hide, Shade, Render.

R7. Actualizar los usuarios que tienen acceso al sistema y dar los privilegios adecuados para realizar la operación que necesiten, incluye.

- Insertar usuario del sistema. Eliminar usuarios del sistema. Modificar usuarios del sistema.

R8. Control de la política de usuarios. Garantizar accesibilidad a los usuarios en dependencia de sus privilegios.

R9. Tratamiento de excepciones y errores. Se garantiza el control de errores con miras al correcto funcionamiento de la aplicación proveyendo robustez al sistema.

R10. Realizar la simulación del proceso de maquinado.

3.2.2 Requerimientos no funcionales.

“Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, como restricciones del entorno, de implementación, rendimiento”. **[48]**

A continuación se presentan los requerimientos no funcionales determinados para la aplicación a elaborar.

Apariencia o interfaz externa: La interfaz se ajustará a los estándares de aplicaciones de simulación para la Web. Estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad.

Usabilidad: El producto está diseñado con versatilidad para que pueda ser usado tanto por estudiantes como por ingenieros y tecnólogos en la industria.

Seguridad: Debe poseer mecanismo de seguridad, con el objetivo de que se pueda recuperar gran parte de la información ante alguna situación de fallo, por ejemplo: la caída del fluido eléctrico o de desperfecto técnico del equipo de cómputo.

Documentación: El manual de usuario constará de dos partes: una teórica, con una descripción detallada de los aspectos más relevantes sobre el proceso y la otra parte describiendo todas las funcionalidades que tiene el subsistema.

Requerimiento de Software: La aplicación cliente funcionará sobre un navegador Web, y la aplicación servidor correrá sobre un servidor Web conectado a un servidor de bases de datos

Requerimiento de Hardware: Para explotar el sistema se necesita como mínimo un Pentium a 200MHz de velocidad con 32 Mb de memoria con navegador Web instalado.

3.3 Modelo de clases del sistema.

Un diagrama de clases es una colección estática de los elementos declaratorios del modelo del sistema, como clases, tipos y sus relaciones, conectados unos a otros y a sus contenidos [49]. El diagrama de clases del diseño es una expansión técnica de los resultados del análisis. Las clases, relaciones, y colaboraciones del análisis son complementadas con nuevos elementos, donde se enfocan cómo implementar el sistema. Todos los detalles y restricciones del sistema deben ser considerados. La Figura 3.3 esboza las principales clases que existen en el escenario del sistema descrito en tres capas, las relaciones existentes entre ellas, sus métodos y atributos más representativos. Este modelo nos brinda una perspectiva útil a la hora de llevar a código fuente el programa.

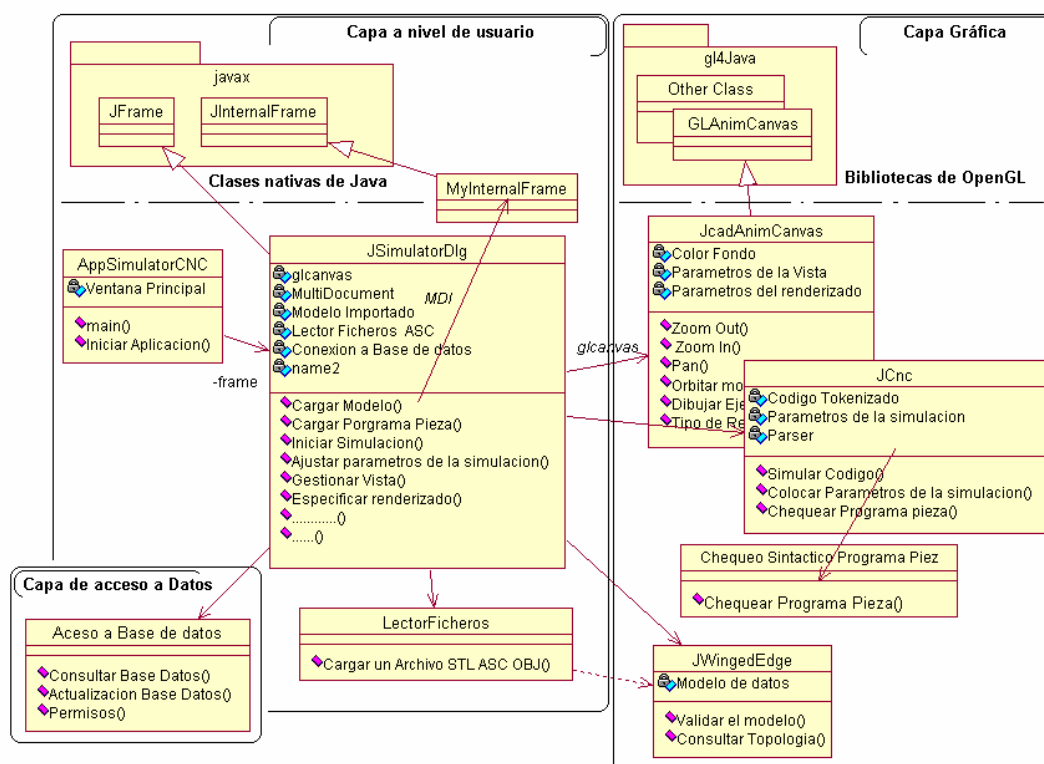


Figura 3.3. Modelo de clases del sistema de simulación en UML.

3.4 Implementación

Una vez identificadas las clases del sistema que han sido resultado del proceso de análisis y diseño, se deben implementar. En primer lugar, se analizan las necesidades tecnológicas que exige el sistema. Las cuales son:

- Un lenguaje orientado a objetos, puesto que el proceso es orientado a objetos.
- Un entorno de distribución de objetos, para aquellos objetos que colaboran entre sí y que están en diferentes aplicaciones.
- Lenguajes y servidores, para implementar el sitio Web.
- Acceso a una base de datos donde se almacena la información.

Al estudiar estos requisitos tecnológicos, Java se presenta como una elección para el lenguaje a utilizar, puesto que ofrece soporte para todos los requisitos.

Mediante RMI (Remote Method Invocation) se cubre la necesidad de disponer de un entorno para la distribución de objetos. RMI es una fácil solución de Java para la distribución de objetos. En cuanto a las necesidades tecnológicas para el sitio Web, se usa HTML para la implementación de las páginas estáticas, tecnología Applets para chequeo de formularios y, JSP para la implementación de las páginas de servidor.

Por último, para cubrir el acceso a bases de datos se utiliza JDBC (Java DataBase Connectivity), que es la solución de Java para conexión con bases de datos.

Finalmente, se destaca que todas las tecnologías empleadas se adaptan a un entorno basado en Java. Para el desarrollo de la aplicación se empleó el IDE (Integrated Development Environment) JCreator Pro v.4. Este IDE a pesar de no poseer capacidades de modo gráfico, se caracteriza por su ligereza ya que requiere máquinas poco potentes para ejecutarse, como Pentium 2 a 300 MHz con 64 mega bytes de RAM y permite desarrollar aplicaciones profesionales de alta complejidad.

3.5 Interfaz del sistema.

La interfaz principal de la aplicación, ha sido diseñada en conformidad con los requisitos estándares de las aplicaciones de simulación, mencionados en el Capítulo 1. La aplicación provee un entorno amigable e interactivo donde los usuarios no solo inspeccionan el proceso de simulación sino que pueden editar los programas pieza, interactuar con el entorno gráfico y a su vez importar y modificar el proceso de maquinado y evaluar diferentes estrategias del proceso. El entorno del sistema está dividido en tres áreas fundamentales; Administración de proyecto, entorno de simulación gráfico y entorno de edición de programas NC. A continuación la Figura 3.4. muestra el entorno de la aplicación y se señalan sus principales componentes, los cuales son descritos en el epígrafe siguiente.

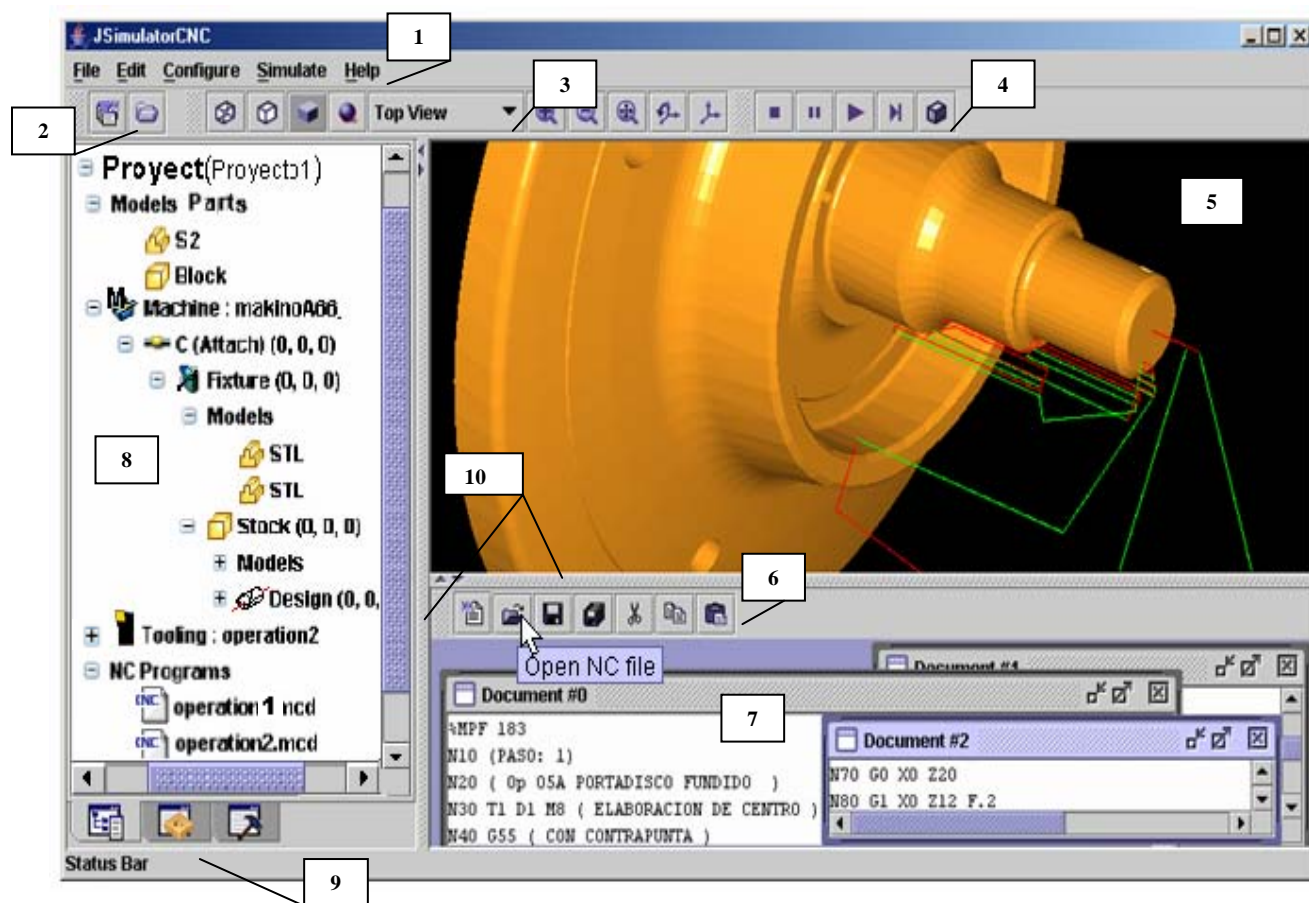


Figura 3.4. Interfaz de usuario de la aplicación ^{CN}Simulator.

3.5.1 Descripción de los componentes del sistema.

- 1- Barra de menú La barra de menús contiene los nombres de todos los menús de la aplicación. La barra de menú está compuesta principalmente por cinco menú desplegables (*File*, *Edit*, *Configure*, *Simulate*, *Help*). El menú *File*: esta asociado a todo el tratamiento con la apertura y salvos de los proyectos de maquinado. *Edit*: se relaciona con el tratamiento de edición de los programas piezas cargados. *Configure*: manipula la apariencia del entorno de simulación, así como el tamaño y color de fondo de la ventana gráfica. *Simulate*: este menú se vincula a los comandos para el control de la simulación del proceso. *Help*: brinda soporte de ayuda a los usuarios.
- 2- Barra de herramienta de proyecto. Está asociada a la carga y comienzo de los proyectos de maquinado a simular.
- 3- Barra de herramienta de visualización. Muestra los comandos asociados a la visualización de los modelos importados al entorno gráfico. Estos modos se resumen en vistas alámbricas, ocultas, sombreadas y suavizadas. En esta barra también se muestran comandos para los diferentes puntos de vistas y activación de funciones para uso del ratón como el Orbit.
- 4- Barra de herramienta de simulación. Esta barra controla el comportamiento de la simulación geométrica del maquinado, así como la carga de los modelos a maquinar.
- 5- Entorno de simulación geométrica del maquinado.
- 6- Barra de herramienta estándar de edición CNC. Permite la carga salva y modificación de los programas pieza a simular. Cada una de las barras descrita posee indicación de funcionalidad al situar el cursor sobre los comandos.
- 7- Entorno de múltiples documentos (MDI), asociado a la edición de los programas piezas cargados.
- 8- Este componente presenta tres fichas principales. (Ficha 1): Entorno de administración del proyecto de maquinado. (Ficha 2): Control y visualización de las herramientas y máquinas desde la base de datos (MySQL), remota en el servidor. (Ficha 3): Parámetros de simulación; velocidad, tipo y otros.

- 9- Barra de estado. La barra de estado muestra descripciones de comandos, mensajes, estado de la conectividad con el servidor y otras informaciones. Parte de esta información se refiere a lo que hace un comando determinado o a la acción que debería adoptar el usuario.
- 10- Controles *Spliters*: permite maximizar cada uno de los tres entornos de la aplicación con vista a un mejor aprovechamiento del área de trabajo, como se aprecia en la Figura siguiente.

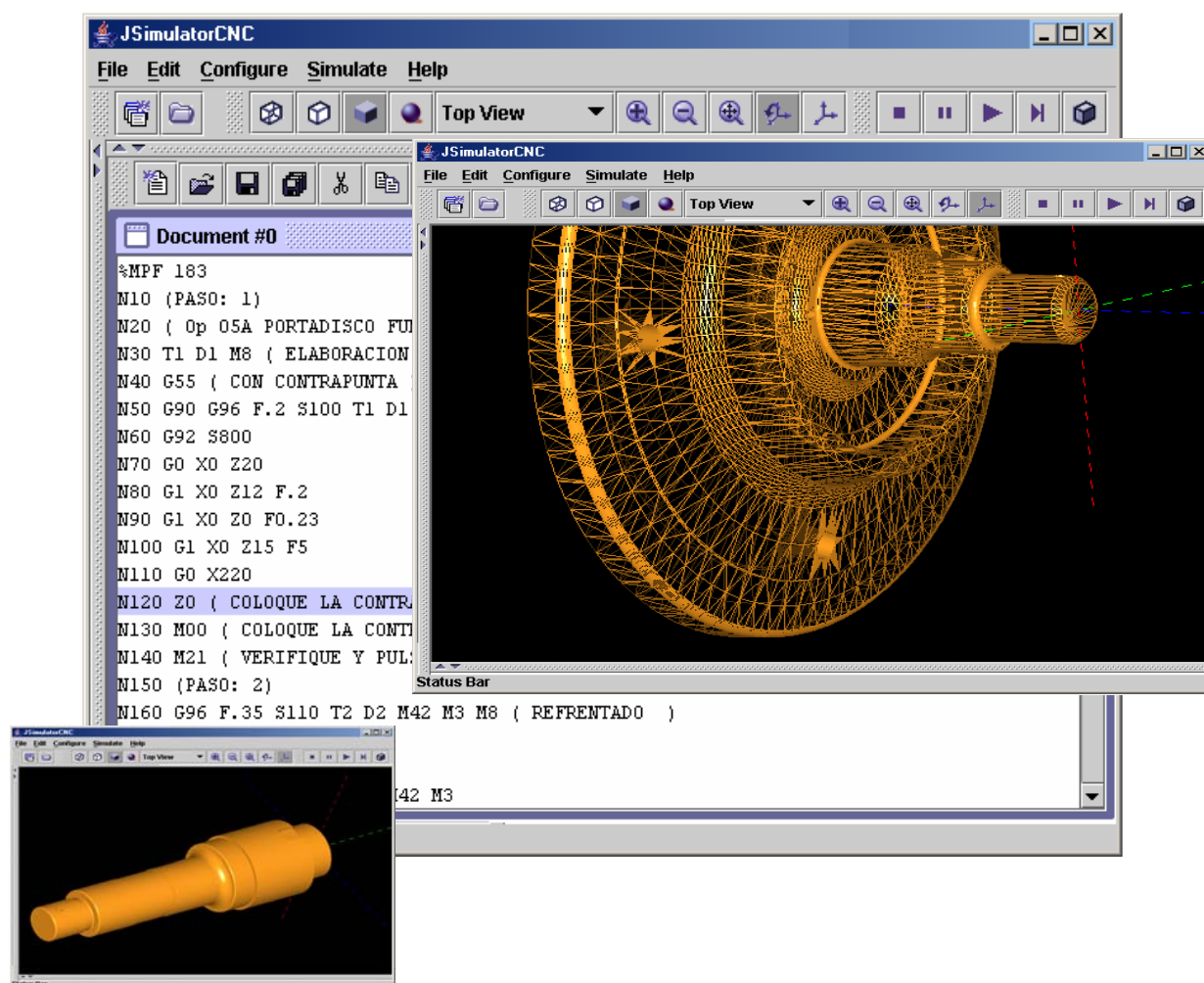


Figura 3.5. Vista de los componentes de edición y simulación, mediante el control *Splitter*.

3.6 Conclusiones del Capítulo.

En función de los resultados obtenidos, como conclusiones de este capítulo, se destacan las siguientes:

1. Se establece la creación de un entorno de simulación de maquinado sobre Web, definido por su arquitectura Cliente-Servidor.
2. La arquitectura propuesta posibilita la interacción remota multiusuario, en un entorno de integración CAPP/Simulación, orientado al proceso de maquinado.
3. La tecnología Applet de Java, garantiza el desarrollo de interfaces de alta complejidad sobre Web.
4. No necesariamente se requiere de hacer uso de IDE de alta complejidad, potencia y elevados costos, como JBUILDER, NETBEANS o ECLIPSE, para desarrollar aplicaciones de complejidad considerable como la descrita en este capítulo.

CONCLUSIONES

1. Para la implementación y representación de la gráfica 3D sobre Web, el empleo de las bibliotecas GL4Java, resulta una herramienta robusta y eficiente.
2. El uso de técnicas de procesamiento de lenguaje de comando CNC, es imprescindible para lograr un entorno de edición eficiente de los programas pieza que se introducen al sistema.
3. Para lograr un entorno integrado CAD/CAPP/CAM sobre entornos Web, el uso de un modelo de Datos único y centralizado en el Servidor, es de suma importancia para evitar redundancia de información empleada por los sistemas, garantizar la integridad de los datos y establecer el acceso multiusuarios del lado Cliente.
4. Se desarrolló una aplicación de simulación virtual para el maquinado de piezas rotacionales, con tecnología Applet de Java, la cual garantiza un entorno sobre Web multiusuarios.

RECOMENDACIONES

1. Extender la capacidad del sistema para leer ficheros de intercambio de datos IGES y STEP.
2. Establecer una plantilla genérica para la modelación de los programas pieza NC de diferentes fabricantes, lo cual garantiza capacidad de simulación del sistema atendiendo a la diversidad de controles CNC existentes.
3. Realizar pruebas de conexión del sistema, en un entorno de Internet y no solamente a nivel de Intranet.
4. Evaluar el comportamiento de la aplicación, en diferentes sistemas operativos y no solamente en la plataforma Windows.

BIBLIOGRAFÍA

1. Arturo Molina, G; Miguel de J. Ramírez. Proyecto Control Numérico Universal (CNU). Centro de Sistemas Integrados de Manufactura, ITESM, 2000.
2. Mandar Shridhar, Joshi; Shailendra Ravi. Virtual Manufacturing: An Important Aspect of Colaborative Product Commerce. *Journal of Advance Manufacturing Systems*. 1(1) 113-119, 2002.
3. Qingjin Peng. A survey and implementation framework for industrial-oriented web-based applications, *Integrated Manufacturing Systems*. 13(5) 319-327, 2002.
4. SMITH, C. S. and WRIGHT, P. K; CyberCut: A World Wide Web Based Design-to-Fabrication Tool. *Journal of Manufacturing Systems*. 15, 432–442, 1996.
5. A Mohole; P Wright; C Sequin. WebCAD: A computer aided design tool constrained with explicit 'design for manufacturability' rules for computer numerical control milling. *Proceedings of the Institution of Mechanical Engineers*. 216, 6, 2002.
6. Sang-Hoon Kong; Jaehong Park; Young-Geun Han; Gibom Kim; Kyo-II Lee. The Internet-based virtual machining system using CORBA, *Integrated Manufacturing Systems*. 13(5); ABI/INFORM Global. 340, 2002.
7. Joao Carlos E. Ferreira; Marcelo Texeira; Albeto J. Álvarez. A Procedure for Integrating Automated Equipment in a Flexible Manufacturing System an their use for the Remote Manufacture of Parts through the Internet. 2004.
8. JIANG, P. and FUKUDA, S. TeleRP — An Internet Web-based Solution for Remote Rapid Prototyping Service and Maintenance. *International Journal of Computer Integrated Manufacturing*. 14, 83–94, 2001.
9. Zhang Y.; Chun; Wang. An Internet based STEP data exchange framework for Virtual Enterprises. *ELSEVIER, Computer in Industry* 41, 51-63, 2000.
10. Joao Carlos E. Ferreira; Alberto J. Álvares. WebMachining: Implementation of a Collaborative CAD/CAPP/CAM System for E-Manufacturing Through the Internet. <http://WebMachining.AlvaresTech.com>, 2004.

11. Yan, H.; Xue, D. Recent research on developing Web-based manufacturing system: a review. *Int. J. Production Research.* 14 (15), 3601-329, 2003.
12. Robert B. Jerard; Barry K. Fussell, Min Xu, Chad Schuyler. Development of a Testbed for Research on Smart Machine Tools. University of New Hampshire, Durham, NH 03824; Proceedings of NSF Design, Service, and Manufacturing Grantees and Research Conference, St. Louis, Missouri. 2006.
13. M. Larrauri Gil; A. Martínez Lopez; M.L. Alvarez Gutierrez; S. Plaza Pascual; A. Bilbao Sagarduy. *Sistemas de verificación y simulación de programas de control numérico.* 2000.
14. ZHUANG, Y., CHEN, L.; VENTER, R. CyberEye: an Internet-enabled environment to support collaborative design. *Concurrent Engineering: Research and Applications.* 8, 213–229, 2000.
15. Chad Vawter; Ed Roman. J2EE vs. Microsoft.NET A comparison of building XML-based web services, June 2001.
16. L. Elena. ¿Una comparación entre Java y .NET?, Universidad Icesi. 2003.
17. Chen, L., Song, Z. and Liavas, B. Exploration of a multi-user collaborative assembly environment on the Internet: a case study. CD-ROM Proceedings of the ASME 2001. Design Engineering Technical Conferences and Computers and Information in Engineering. Conference, Pittsburgh, PA. 2001.
18. Wang, X. K. Research on a dispersed networked CAD/CAM system of motorcycle. *Journal of Materials Processing Technology.* 129, 658–662, 2002.
19. Xie, S. Q., Huang, H.; TU, Y. L. A WWW-based information management system for rapid and integrated mould product development. *International Journal of Advanced Manufacturing Technology.* 20, 50–57, 2002.
20. A. Mumford. *Graphics on the WWW.* 1999.
21. Nidamarthi, S.; Allen, R. H.; Sriram, R. D. Observations from supplementing the traditional design process via Internet-based collaboration tools. *International Journal of Computer Integrated Manufacturing.* 14, 95–107, 2001.
22. Kim, C. Y., Kim, N., Kim, Y., Kang, S. H. and O’Grady, P. Distributed concurrent engineering: Internet-based interactive 3-D dynamic browsing and markup of

- STEP data. Concurrent Engineering: Research and Applications. 6, 53–70, 1998.
23. Kim, S. H.; Jeong, Y. M. Development of the STEP-TO-DB converter for the Webbased PDM. CD-ROM Proceedings of the ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Pittsburgh, PA. 2001.
 24. Sandy R*; A. Godil; Qiming Wang G. Seidman A VRML Integration Methodology for Manufacturing Applications Information Technology Laboratory National Institute of Standards and Technology. 1999.
 25. Don Brutzman. The Virtual Reality Modeling Language and Java. Code UW/Br, Naval Postgraduate School. Communications of the ACM. 41(6), 57-64, 1998.
 26. Kian Huat T.; Tze Leong; G. Kurt. Understanding Machine Operations and Manufacturing using VRML. ASEE, 1999.
 27. Francisco A. Candelas, F. Torres, Pablo Gil. Laboratorio Virtual Remoto para Robótica y Evaluación de su Impacto en la Docencia. Revista Iberoamericana de Automática e Informática Industrial. 1(2), 49-57, Julio 2004.
 28. Visionary manufacturing challenges for 2020. T176 .V57, 1998.
 29. R. Avila; Ciro A. Rodriguez; O. Falco. Distributed Process Planning and Programming for Machining Operations in Small and Medium Enterprise. UHo, ITESM, 2004.
 30. Adamczyk Z. A new approach to CAM systems development for small and medium enterprises, Journal of Materials Processing Technology. 107(1), 173-180(8), November 2000.
 31. Kees van Luttervelt. Advancements in Modelling of Cutting Operations. XIII Machine Tool & Manufacturing Technology Congress. 25-27. San Sebastian, Spain. October 2000.
 32. Michi Henning; Mark Spruiell. Distributed Programming with Ice. July 2004
 33. J. Arazola; F. Le Maitré; F. Meslin. Fundamentos del Proceso de Corte y Situación Actual de su Modelado. 2002.

34. Ming C. Leu; Feng Lu; Denis Blackmore. Simulation of NC Machining with Cutter Deflection by Modelling Deformed Swept Volumes. *Annals of the CIRP*, 47(1). 1998.
35. A. José Álvares; João C. Espíndola Ferreira. Metodologia para implantação de laboratórios remotos via Internet na área de automação da manufatura. 2do COBEF, 2003.
36. Juan Manuel Fonseca. Visual Interactive Simulation for Distance Education. *SIMULATION*, 79(1). 2003.
37. Kian-Huat Tan; Tze-Leong Yew; Kart Gramoll. Understanding Machine Operations and Manufacturing using VRML. *Am. Soc. Of Engineering Educ*, (ASEE), Charlotte, NC. June 1999.
38. David E. Culler; José A. Pérez García. Using Internet Based Concurrent Engineering Tools to Educate Multinational Students about the Design, Process Planning and Manufacture of New Products. 34th ASEE/IEEE Frontiers in Education Conference. Savannah, GA. October, 2004.
39. Magdalena A. Álvarez; J. Monge-Nájera. Evolución Tecnológica de los Laboratorios Virtuales en la Universidad Estatal a Distancia. 2002.
40. Rabi Chawla; Amarnath Banerjee. A Virtual Environment for simulating Manufacturing Operations in 3D. *Proceeding of the Winter Simulation Conference*. 2001.
41. N. M. Patrikalakis. *Computational Geometry*. Massachusetts Institute of technology, Cambridge, MA 02139-4307, USA. 2003
42. Chacón Moreno. Estudio y análisis de la teoría de la multirresolución en el modelado de sólidos. Ingeniería, Universidad de las Américas-Puebla. Mayo, 2000.
43. José L. Ajuria; Juan T. Martín; Carlos Pilarte. Integración de un Sistema de Simulación y Verificación en el Diseño y Utilización de Máquina-Herramienta. *European Virtual Engineering, EUVE*, 2002.

44. Andrew Davison. Killer Game Programming in Java. USA, O'Reilly Media, Inc. Mayo 2005.
45. Sven Goethel. OpenGL[tm] for Java[tm] (formerly gl4java) Implementation Of A Native OpenGL-Interface to Java, X-Window and Windows (95/NT). Abril 200.
46. William Grosso. Java RMI. USA, O'Reilly Media, Inc. Octubre 2001.
47. Robert Englander. Java and SOAP O'Reilly. Mayo 2002.
48. Jacobson, I.; Booch, G. y Rumbaugh, J.; El Proceso Unificado de Desarrollo de software. Addison-Wesley. 2000.
49. Booch, Grady; Rumbaugh, James; Jacobson, Ivar. "El Lenguaje Unificado de Modelado". Addison Wesley. 1998

Anexo 1 – Resumen de las funciones G y M para las operaciones de torneado.
(Manual de programación del control FAGOR 8020).

Funciones G	Descripción	Modo
G00	Posicionamiento Rápido	Modal
G01	Interpolación Lineal	Modal
G02	Interpolación circular (sentido horario)	Modal
G03	Interpolación circular (sentido anti-horario)	Modal
G04	Temporización, duración programada durante k	No modal
G05	Trabajo en arista matada	Modal
G06	Interpolación circular desde el centro del arco coordenadas absolutas.	No modal
G07	Trabajo en arista viva	Modal
G08	Trayectoria circular, tangente a la trayectoria anterior.	Modal
G09	Trayectoria circular definida mediante tres puntos.	No modal
G14	Activación del eje C en grados	Modal
G15	Mecanizado en la superficie cilíndrica de la pieza.	Modal
G16	Mecanizado en la superficie frontal de la pieza.	Modal
G33	Roscado	Modal
G40	Anulación de compensación de radio	Modal
G41	Compensación de radio a izquierdas	Modal
G42	Compensación de radio a derechas	Modal
G49	FEED-RATE programable	Modal
G50	Carga de dimensiones de herramienta en tabla	No modal
G66	Ciclo fijo de desbastado siguiendo perfil de la pieza	No modal
G70	Programación en pulgadas	Modal
G71	Programación en milímetros	Modal
G83	Ciclo fijo de taladrado	No modal
G86	Ciclo fijo de roscado longitudinal	No modal
G88	Ciclo fijo ranurado longitudinal	No modal
G90	Programación de cotas absolutas	Modal
G91	Programación de cotas incrementales	Modal
G94	Avance F en mm/min	Modal
G95	Avance F en mm/rev	Modal
G96	Velocidad S en m/min.	Modal

Funciones M	Descripción
M00	Parada de programa
M01	Parada condicional del programa
M02	Final del programa
M30	Final del programa con vuelta al comienzo.
M03	Velocidad S en m/min.
M04	Arranque del cabezal a derechas (sentido horario)
M05	Parada del cabezal
M19	Parada orientada de cabezal
M08	Encendido del refrigerante
M09	Apagado del refrigerante
M06	Parada del programa (cambio manual de herramienta)

Anexo 2 – Ejemplo del cálculo. Determinación de las coordenadas absolutas X y Z del centro de la punta de la herramienta con radio 0.0625R, para torneear el perfil de la Figura A2.1. (Asumir el tipo de programación en diámetros).

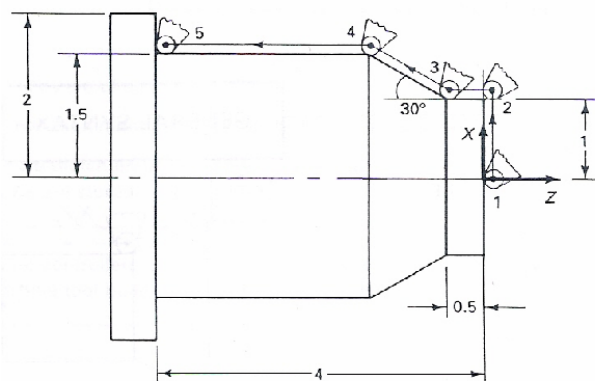
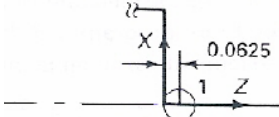
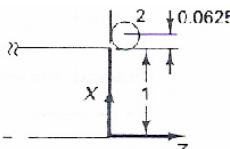
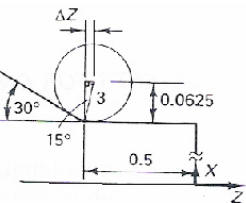
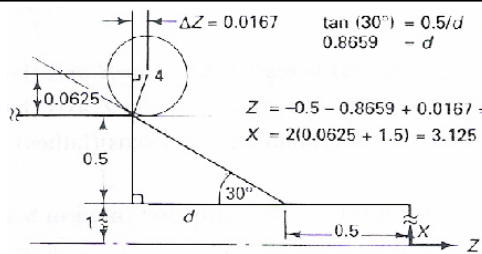
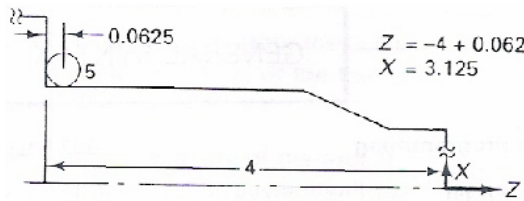
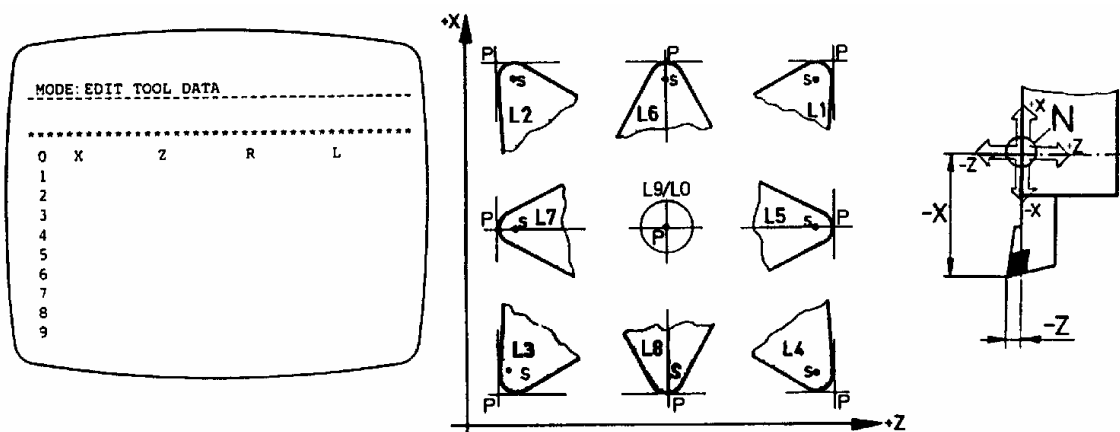


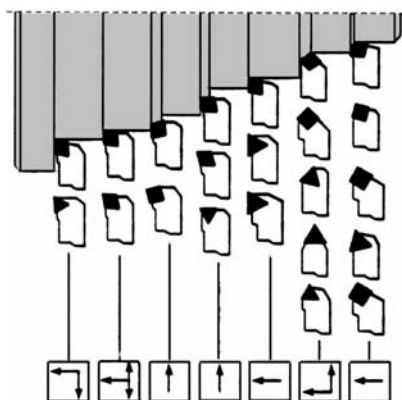
Figura A2.1

Posición	Cálculo	X	Z
①		0	0.0625
②	$X = 2(1 + 0.0625) = 2.125$ $Z = 0.0625$ 	2.125	0.0625
③	$\tan(15^\circ) = \Delta Z / 0.0625$ $0.0167 = \Delta Z$ $Z = -0.5 + 0.0167 = -0.4833$ $X = 2.125$ 	2.125	-0.4833
④	$\Delta Z = 0.0167$ $\tan(30^\circ) = 0.5/d$ $0.8659 = d$ $Z = -0.5 - 0.8659 + 0.0167 = -1.3492$ $X = 2(0.0625 + 1.5) = 3.125$ 	3.125	-1.3492
⑤	$Z = -4 + 0.0625 = -3.9375$ $X = 3.125$ 	3.125	-3.9375

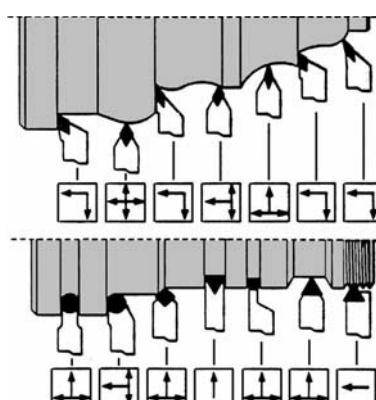
Anexo 3 – Valor del vector de la forma de la herramienta para la compensación de radio. (Tabla para la programación de la herramienta).



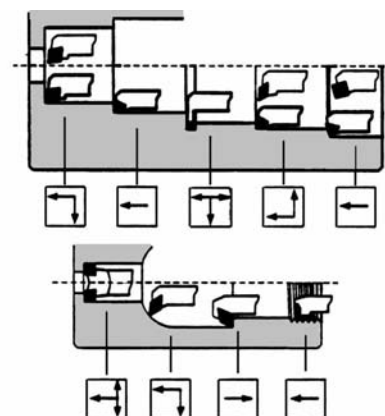
Anexo 4 – Tipos de portaplaquitas y dirección del maquinado.



Portaplaquitas de exteriores para desbaste.



Portaplaquitas de exteriores para acabado.

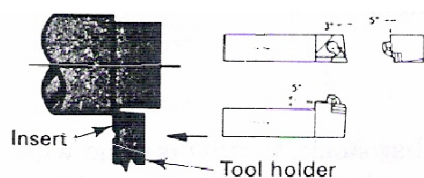


Portaplaquitas de interiores.

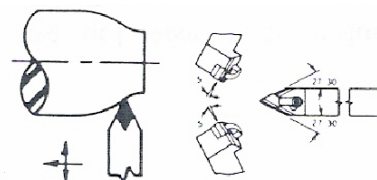
 $l_a = 0,4d$	 $l_a = 2/3 l$	 $l_a = 2/3 l$	 $l_a = 1/2 l$
 $l_a = 1/2 l$	 $l_a = 1/2 l$	 $l_a = 1/4 l$	 $l_a = 1/4 l$

Tamaño y forma de la plaquita.

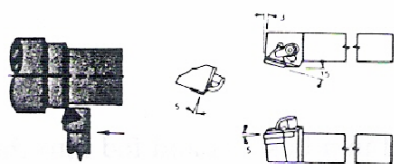
Anexo 5 – Algunos estilos y formas de las herramientas Kennametal Inc., acorde a las normas industriales, según el tipo de operación de maquinado.



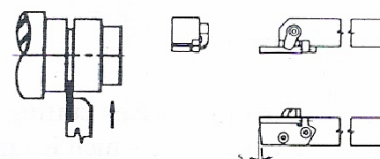
Style A for turning, facing or to a square shoulder.



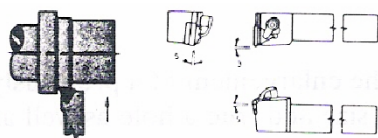
Style P for Profile machining. Insert centrally located.



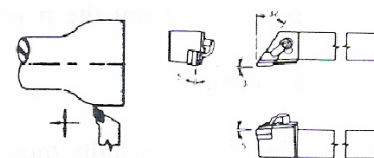
Style B for rough turning, facing or boring where a square shoulder is not required.



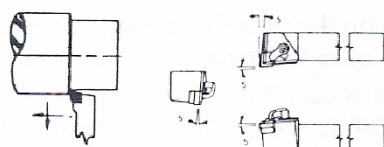
Style GCH for deep grooving.



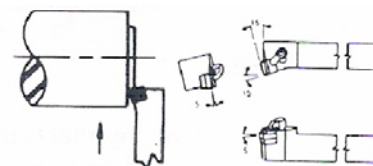
Style F for facing, straddle facing or turning with shank parallel to work axis.



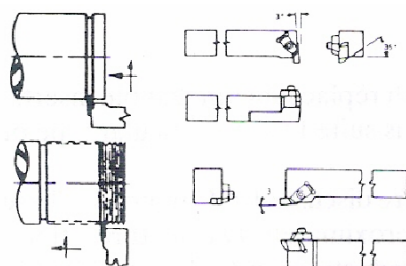
Style J for profiling and finish turning.



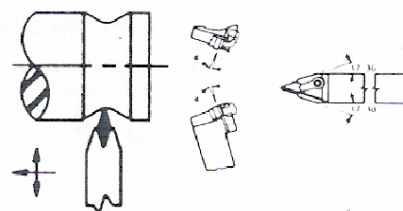
Style L for both turning and facing with same tool. 80° diamond insert.



Style K for lead angle facing or turning with shank parallel to work axis.



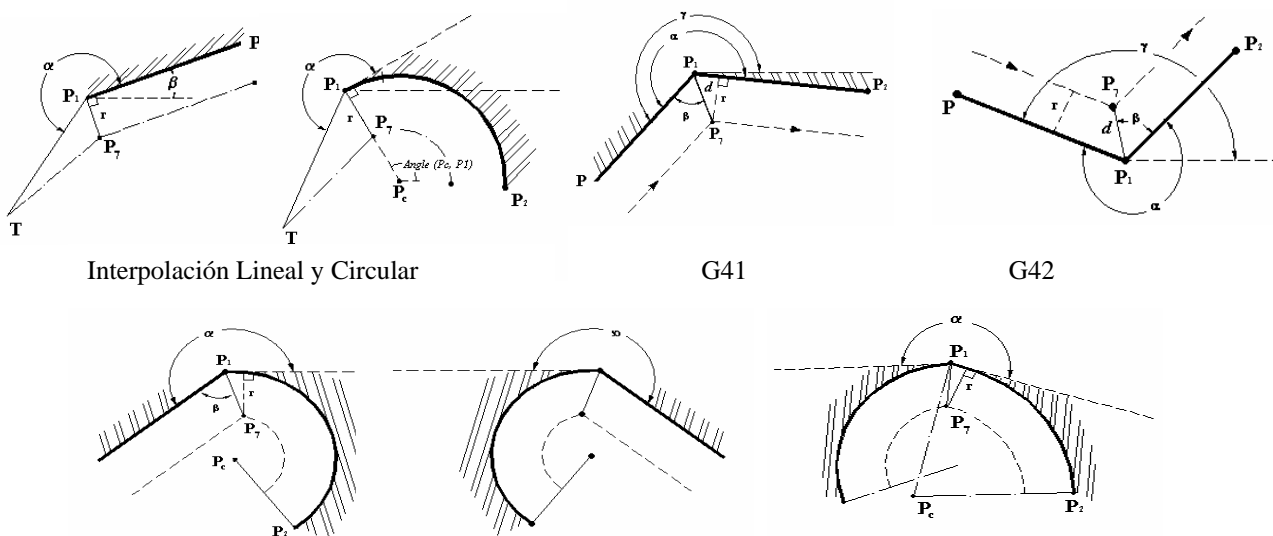
Style NE and style NS for threading and grooving operations. Inserts available for 60° V, Acme, Buttress and API threads.



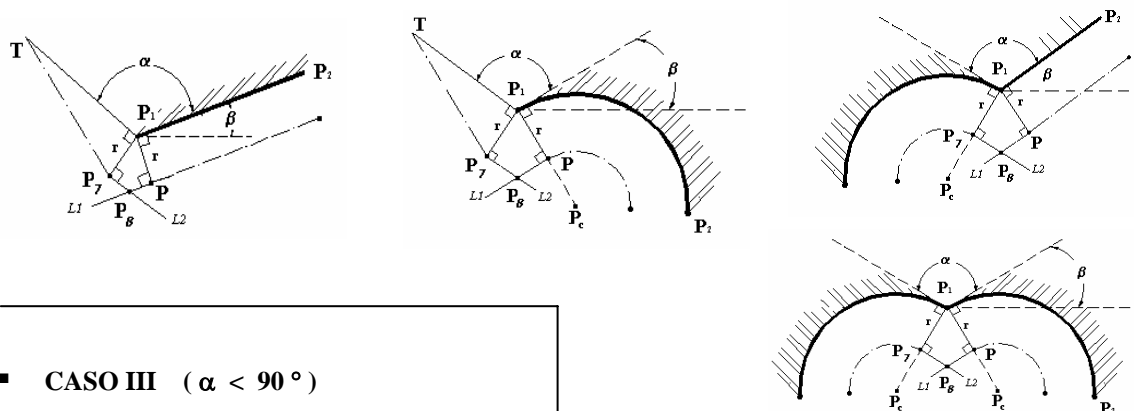
Style V for profile machining. 35° insert.

Anexo 6 – Casos para el cálculo de los puntos de las trayectorias compensadas.
(Entidades Line-Line; Line-Arc; Arc-Line; Arc-Arc).

▪ **CASO 1** ($\alpha \geq 180^\circ$)



▪ **CASO II** ($90^\circ \leq \alpha < 180^\circ$)



▪ **CASO III** ($\alpha < 90^\circ$)

