



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

88

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA PARA REQUERIR
PARTES AUTOMOTRICES IMPORTADAS DE JAPON
DE ACUERDO A PROGRAMAS MENSUALES DE
PRODUCCION DE PLANTAS ENSAMBLADORAS NISSAN

T E S I S

Y QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A
EDUARDO GUTIERREZ TORRES

Director de Tesis. M. I. Raymundo Hugo Rangel Gutiérrez

MEXICO. D. F.

TESIS CON
FALLA DE ORDEN

1994





UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mi madre, padre, familiares y amigos que siempre han confiado en mí y de alguna u otra manera siempre me han ayudado, Gracias.

Sinceramente :

José Antonio

Dedico esta tesis con mucho cariño amor y respeto a mis abuelos Don Rosendo Juárez Olvera (†) y Doña Aurora Padilla Vda. de Juárez que han sido como mis segundos padres.

AGRADECIMIENTOS :

Con amor a mis padres Josefina y Miguel quienes siempre me apoyaron en mis estudios y me motivaron a superarme y prepararme cada día, así como también a alcanzar las metas que me propongo.

Con cariño a mis tíos Carlos, Mario, Ernesto y Carmen por sus consejos, y porque de alguna manera siempre me brindaron su ayuda cuando en su momento lo necesite.

A mis hermanos Olga, Miguel, Daniel, Adoración, Leticia y Lorena por su paciencia y a quienes también dedico este trabajo como motivación, para que siempre se esfuercen por alcanzar sus metas no importa que tan difíciles les parezca y siempre recuerden que "El que persevera alcanza".

A mi hermano Agustín E. y su esposa Adela por las facilidades que nos otorgaron al prestarnos su equipo de cómputo y también como un motivo de superación.

A mi esposa Dulce María y mi pequeño hijo Héctor Arturo por su amor, ternura y comprensión al apoyarme en la culminación de mis estudios, Gracias.

Sinceramente :

Pedro Héctor

Dedico esta tesis a mis padres Alicia y Francisco, por su guía, apoyo y dedicación para que lograra estudiar una carrera profesional.

A mi amiga Ma. Elena Saucedo, por el gran apoyo y entusiasmo que me brindó en todo momento para la conclusión de mis estudios.

Sinceramente :

Eduardo

A mis tíos Reyna y Domingo que han sido como unos padres.

A mi madre María Dolores y a Javier (✧).

A mi esposa Clara María.

A mis hijos Sandra Elizabeth y Francisco.

Agradezco su cariño y apoyo y les dedico esta tesis.

Sinceramente :

Francisco Sánchez S.

I N D I C E

1. INTRODUCCION	4
1.1- ANTECEDENTES DEL SISTEMA	8
1.2- LENGUAJES DE CUARTA GENERACION	11
1.2.1- ¿QUE SON LOS LENGUAJES DE CUARTA - GENERACION?	12
1.2.2- LENGUAJES PROCEDURALES Y NO PROCEDURALES	15
1.2.3- EFECTOS DE LENGUAJES DE 4A. GENERACION.	23
1.2.4- CUANDO SON USADOS LOS LENGUAJES DE 4A. GENERACION.	26
1.2.5- DIALOGOS USADOS EN LOS LENGUAJES DE 4A. GENERACION.	28
1.2.6- LENGUAJES PARA PROTOTIPOS	36
1.2.7- EVOLUCION DE LOS LENGUAJES	51
2. METODOLOGIA.	
2.1 CONCEPTOS BASICOS DE PROGRAMACION NATURAL	55
2.1.1- DESCRIPCION Y CARACTERISTICAS	55
2.1.2- INSTRUCCIONES BASICAS DE PROGRAMACION NATURAL	59
2.2 CONCEPTOS BASICOS DE AMBIENTE DE BASE DE DATOS ADABAS	60
2.2.1- DESCRIPCION DE LA BASE DE DATOS ADABAS	60
2.2.2- DISEÑO DE UNA BASE DE DATOS EN AMBIENTE ADABAS	62

3. ANALISIS DE REQUERIMIENTOS	
3.1- DESCRIPCION GENERAL DEL SISTEMA	66
3.2- DIAGRAMA DE FLUJO DE DATOS DEL SISTEMA	71
4. DISEÑO DEL SISTEMA	
4.1 INTERFASES CON SISTEMAS EXTERNOS	72
4.1.1- INTERFASE CON EL SISTEMA PROGRAMACION	72
4.1.2- INTERFASE CON EL SISTEMA ANPICS	74
4.2 PROCESOS PRINCIPALES	76
4.2.1- PROCESO DE BOM CONVERSION	76
4.2.2- PROCESOS DE GROSS-CHECK Y EXPLOSION DE PARTES	82
4.2.2.1- PROCESO DE GROSS CHECK	82
4.2.2.2- PROCESO DE EXPLOSION DE PARTES	89
4.2.3- PROCESO DE ALTAS A LOS ARCHIVOS MAESTRO DE PARTES (P/F) Y AL MAESTRO DE CONTROL DE INVENTARIOS (MCI)	95
4.2.4- MODULO DE MANTENIMIENTO A RANES	99
4.5.5- AJUSTES - PARTES DE SERVICIO	104
4.2.6- PROCESO DE CALCULO DE ORDENES	106
4.2.7- TRANSMISION DE ARCHIVOS	111
4.2.8- RESPALDO Y ACTUALIZACION DEL SISTEMA	113
5. CONCLUSIONES	117

APENDICES

A)	DICCIONARIO DE TERMINOS DEL SISTEMA	119
B)	TRANSMISION VIA VAN (Red Local de Valor Agregado)	124
C)	LENGUAJE CONTROLADOR DE TRABAJOS (JCL)	130
D)	INFORMACION TECNICA DEL SISTEMA 9121 / IBM	133
E)	MANUAL DEL USUARIO	136
F)	BIBLIOGRAFIA	181

1. INTRODUCCION

La calendarización de la producción de unidades se lleva a cabo, en un inicio, en base a los requerimientos de ventas y capacidad de producción de las plantas, de donde se desarrollan los programas anual, mensual, decenal y diario de producción.

Una vez realizada dicha calendarización es necesario requerir mensualmente el material a cada uno de los proveedores con el fin de que éstos lo fabriquen y lo entreguen a las plantas de Nissan Mexicana en las fechas pactadas para su ensamble final.

Los proveedores de Nissan Mexicana son principalmente de tres orígenes. Proveedores Nacionales, proveedores de Estados Unidos y el principal Japón.

Japón es el principal proveedor ya que en un inicio los diseños primarios son desarrollados en Japón por lo que proveedores japoneses tiene la ventaja de producir material para pruebas durante el desarrollo de los modelos.

Una vez liberados los modelos se convoca a proveedores Nacionales o de Estados Unidos a participar en la fabricación de algunas partes de un determinado modelo.

Debido a las situación antes mencionada las partes de un modelo se podrían clasificar de la siguiente forma. (en función a su origen de fabricación).

Japón	70 %
Nacionales	20 %
U.S.A	10 %

Como se puede observar el mayor número de partes son de origen Japonés, por lo que es muy importante controlar con mayor exactitud y

eficiencia los requerimientos de material pedidos a Japón para la producción de cada mes.

Japón requiere de un pedido mensual de producción con un mes de anticipación, esto con el objetivo de realizar los últimos ajustes en las especificaciones de cada una de las partes que llegan a tener algún cambio de ingeniería, entendiéndose como tales a aquellos cambios administrativos o físicos en las partes. Cambios que se generan en optimizaciones al uso de las partes o por necesidades en los ensambles de las mismas.

El sistema actual de requerimiento genera muchos problemas que afectan directamente a la producción. Problemas como son los altos costos en control de inventarios, daño frecuente y generación de partes obsoletas principalmente.

Como primera etapa se contempla solo el requerimiento a Japón ya que es el proveedor con mayor aplicación, una vez liberado este proveedor se harán las adaptaciones necesaria para incluir a los dos proveedores restantes u otros que se llagaran a establecer.

Conforme se realiza la producción de unidades y al comportamiento del mercado se deben efectuar ajustes a los programas de producción y con esto a los programas de requerimiento a proveedores, ya que la producción real es diferente a la programada, y con dichos ajustes se busca cumplir con los volúmenes programados.

Como ya se mencionó anteriormente la producción real de unidades es diferente a la programada al inicio de la producción de cada modelo, aunque al finalizar la cuota de cada modelo se debe cumplir con los volúmenes establecidos en el programa anual. Al desarrollar un sistema de requerimientos de producción a nivel parte se logra tener un sistema más "amable" para poder soportar los constantes cambios de producción sin afectar a otros puntos críticos y delicados que influyen en la industria automotriz; puntos como son el costo de mantenimiento de inventarios, áreas de almacenaje, evitar material crítico o material sobrante que al final de la

producción se refleja como material obsoleto y este afecta a la producción real. Así mismo al requerir solo el material necesario para cumplir con la producción establecida facilita el control del mismo disminuyendo con esto los costos en el control de inventarios.

Considerando que al solicitar Nissan Mexicana material para producir un determinado tipo y número de unidades en el mes "n" de producción, Nissan Japón lo que hacia era determinar a nivel parte individual cuantas de cada una de ellas eran necesarias para cumplir con el volumen solicitado después el mismo volumen de partes individuales a sus proveedores y posteriormente enviarlas a Nissan Mexicana en forma de paquetes para ensamble (cajas). Una forma de optimizar el requerimiento de material a Japón era determinar el número y cantidad de partes individuales que se requerían para cumplir con la producción programada en un mes determinado. Con esto se podrían evitar o por lo menos disminuir en gran cantidad los problemas que hasta la fecha se habían presentado para Nissan Mexicana.

Para lograr esto fué necesario la creación de el mismo ambiente con el que Nissan Japón contaba. Esto es, tener en México una copia de sus archivos de diseño en donde se encuentran detalladas las especificaciones de cada una de las partes existentes.

El sistema se desarrollo en el equipo 9121 de IBM con el que cuenta la compañía, ya que era necesario el manejo de grandes volúmenes de información en cada uno de los sistemas con los que el sistema de Requerimiento de Partes tendría interfase con Japón.

El lenguaje de programación con el que se desarrolló el presente sistema es NATURAL, el cual es un lenguaje de cuarta generación con el que cuenta la compañía NISSAN MEXICANA S.A. DE C.V. La razón por la que se decidió utilizar este lenguaje de programación es la de que en este estaban desarrolladas varias aplicaciones y otras nuevas estan en proceso de desarrollo, además de poder aprovechar las características de un lenguaje de cuarta generación tales como las siguientes:

Es estructurado y con una sintaxis amigable para el analista.

Uso de comando apoyado en un lenguaje común (inglés) lo cual facilita el mantenimiento del mismo.

Facilita la interacción entre el analista y el equipo de cómputo.

Es un lenguaje fácil de aprender, fácil de usar y fácil de recordar (usa lenguaje natural común).

Es un lenguaje compatible y que soporta el sistema operativo "MVS" (Multiple Virtual Storage - Acceso Virtual Multiple).

No necesita un código extenso, al contrario es un código consiso.

Debido a su fácil manejo el mantenimiento puede llegar a ser dado por los mismos usuarios finales.

Mayor productividad en el desarrollo de los analistas-programadores.

1.1. ANTECEDENTES DEL SISTEMA

PANORAMA ANTERIOR

El sistema de requerimiento de partes para planta NISSAN estaba constituido de un sistema híbrido de requerimiento por número de parte y por paquete de partes CKD (Complete Knock Down - Módulo completo de ensamble).

La distribución del requerimiento de partes entre un sistema y otro se muestra en la siguiente tabla.

TIPO DE REQUERIMIENTO	VOL PARTES	DEST	FREC	TIEMPO
* CKD	70 %	JPN	MENS	90 DIAS
* NUMERO DE PARTE	30 %			
- A/E		LOCAL	MENS	15 DIAS
- V/R		USA	MENS	30 DIAS
- COMPLEMENTARIO		JPN	EVENT	***

donde:

A/E - Autorización de Entrega

V/R - Vendor Release (Autorización a Proveedor E.U.A..)

*** - Depende del medio de transporte

Bajo el sistema de requerimiento CKD (Paquete para ensamble) no se puede manejar un nivel de existencia óptimo para cada parte (basado en costo y/o uso), ya que no son requeridas individualmente, lo cual impacta directamente en el costo del mantenimiento de inventario de planta. Este problema se vuelve crítico a medida que los volúmenes de producción se incrementan.

Otro problema que produce el trabajar con este tipo de requerimiento se refleja en el manejo de materiales en planta, ya que para abastecer partes a la línea de producción es necesario realizar operaciones de desempaque y modulación en forma

desordenada, lo que se traduce en daños de partes. Este problema sería evitado si para cada parte se estableciera la norma de empaque más apropiada e individualmente, esto con el fin de abastecer a la línea de producción en una forma más ordenada y eficiente.

El abatimiento del costo del inventario requiere que las partes sean clasificadas por factores tales como:

- uso
- costo
- frecuencia

con el fin de mantener en planta los niveles de existencia más apropiados para cada parte.

Otro problema que genera el trabajar con el requerimiento de partes CKD es el de la falta de control físico de las partes en planta, ya que el desempaque de cajas no es inmediato, lo cual provoca que no se conozca la existencia y la ubicación de las partes en los almacenes.

Otro factor en contra del requerimiento CKD es que las áreas de almacenamiento de partes que se requieren van creciendo a la par que los incrementos de producción. Esta situación no es apropiada ya que en un momento determinado el área de almacenes puede ser mucho mayor a el área de producción. En la medida que los almacenes sigan creciendo, la inversión para lograr el control de los mismos representa gastos muy fuertes y el beneficio obtenido no se reflejará en la producción.

Es por ello que el sistema de requerimiento CKD debe ser sustituido por un sistema de requerimiento por número de parte, con el fin de abatir los problemas anteriormente mencionados.

OBJETIVOS:

Sustituir el requerimiento CKD por un sistema de requerimiento por número de parte.

Planeación y programación de requerimiento de partes para producción, a fin de que estos se encuentren en la cantidad exacta y en el tiempo requerido.

Abatir el costo de mantenimiento de inventarios al controlar el nivel de existencia de cada parte en función a factores como "flota" (niveles de seguridad) y frecuencia de embarque.

Eliminar operaciones de desempaque y modulación de partes para abastecer la línea de producción por medio de una norma de empaquetamiento para cada parte.

Reducir la generación de partes obsoletas.

Eliminar los pedidos complementarios de partes.

Incluir bajo el control del sistema la emisión de ordenes de partes adicionales.

1.2 LENGUAJES DE CUARTA GENERACION

En el último cuarto de siglo, los lenguajes para cómputo comercial han evolucionado lentamente; han sido de sintaxis diferentes, pero la construcción básica es similar (FORTRAN, COBOL, PL/I, Pascal, etc). Actualmente se está viviendo una inundación de nuevos lenguajes y herramientas, las cuales están produciendo dramáticos incrementos en la productividad del proceso de datos. Nunca antes las técnicas de aplicación ha producido cambios tan significativos.

Los nuevos lenguajes y herramientas son:

- Lenguajes de cuarta generación.
- Lenguajes de tercera generación.
- Lenguajes de alta productividad.
- Lenguajes no procedurales.
- Generadores de aplicación.

Estas herramientas en su conjunto constituyen una gran ventaja en el uso de la computadora; verdaderamente, sin este avance no seremos capaces de utilizar la gran cantidad de máquinas que aún se producirán.

Con el potencial de los lenguajes de cuarta generación, el poder de la computadora se hace accesible a cualquier persona; sin necesidad de un extenso entrenamiento en proceso de datos, su impacto producirá importantes desarrollos en el resto de la década.

El rango de estilo de esas herramientas es sorprendentemente amplio, algunos son poderosos, otros poseen excelentes gráficas, otros son adecuados para usuarios con poco entrenamiento, algunos son propios para profesionales, otros están estrechamente ligados a su propia base de datos.

Sin embargo, esta gran variedad y diversidad de productos en el mercado, provoca que el comprador interesado sea muy cauteloso. El campo de lenguajes de computadoras se ha convertido ahora en campo fértil de originalidad y nuevas ideas.

Muchas organizaciones han logrado altos índices en su productividad de proceso de datos; otros no han hecho el uso adecuado, por lo que tienen un bajo aprovechamiento de sus recursos. Muchas organizaciones ignoran el enorme potencial de esta nueva tecnología.

1.2.1 ¿QUE SON LOS LENGUAJES DE CUARTA GENERACION?

Los lenguajes de cuarta generación surgen de la gran necesidad de manipular a las computadoras de una manera mucho más fácil y rápida que en el pasado. Los usuarios finales deben ahora ser capaces de construir sus propias facilidades; ellos necesitan lenguajes que sean tan fácil de usar como sea posible, lenguajes que no tengan la necesidad de recordar nemónicos, formatos, secuencias y construcciones complejas.

La nueva generación de lenguajes de computadora, necesita entonces ser mucho más poderosa que la generación previa, de manera que los resultados puedan ser obtenidos mucho más rápido y ser más amigables al usuario.

El desarrollo de aplicaciones sin programadores, es tal vez la revolución más importante en computación desde la invención del transistor, por lo que se requieren lenguajes orientados al usuario.

CUATRO GENERACIONES DE LENGUAJES.

La primera generación de lenguajes de computadora fue el lenguaje de máquina (binario puro). La segunda generación de lenguajes, la cual se inició a mediados de los 50's, fue el lenguaje ensamblador simbólico; las direcciones simbólicas fueron usadas como direcciones físicas de máquina. El direccionamiento simbólico fue un gran paso, porque ahora cuando las localidades físicas de las variables o instrucciones tuvieron que ser cambiadas (y fueron constantemente cambiadas), el programador no tuvo que volver a proporcionar las nuevas direcciones físicas.

La tercera generación surge en los 60's y fueron llamados lenguajes de alto nivel. Algunos de ellos fueron para trabajo científico, como ALGOL y FORTRAN; otros fueron para trabajo comercial como COBOL, el cual fue el utilizado más comunmente.

La tercera generación de lenguajes necesitó de muchas líneas de código para sistemas típicos comerciales por lo que el tiempo para depurarlos es alto y las modificaciones de sistemas complejos son muy difíciles.

Los lenguajes de cuarta generación fueron creados para lograr los siguientes objetivos:

1.- Incrementar la velocidad de los procesos de construcción-aplicación.

2.- Hacer las aplicaciones fáciles y rápidas de cambiar, reduciendo así los costos de mantenimiento.

3.- Minimizar problemas de depuración.

4.- Generar código libre de defectos de expresiones de alto nivel de requerimientos.

5.- Hacer lenguajes amigables al usuario de manera que los usuarios finales puedan resolver sus propios problemas y saquen el mayor provecho posible de su computadora.

Los lenguajes de cuarta generación permiten que algunas aplicaciones sean generadas con menos líneas de código que las que se necesitarán con algún lenguaje de alto nivel; por lo que son identificados como lenguajes de alta productividad.

Muchos lenguajes de cuarta generación son dependientes de una base de datos y su diccionario o directorio de datos. El diccionario en algunos casos puede representar más que datos, puede contener formatos de pantalla, formatos de reporte estructuras de diálogo, asociaciones entre muchos datos, chequeos de validación, controles de seguridad, autorizaciones para leer o modificar

datos, cálculos que son usados para crear campos derivados, rangos permisibles y relaciones lógicas entre valores de datos. La extensión del diccionario que contiene reglas de negocios y lógica, es algunas veces referida como una enciclopedia.

El principal interés en la selección de un lenguaje de cuarta generación es la infraestructura que se necesita para soportarlo, la cual incluye base de datos, diccionario de datos o enciclopedia.

Los lenguajes de tercera generación usan principalmente construcciones Newman-like; esto es, expresan una secuencia de operaciones a ser ejecutadas en ramificaciones y ciclos. Sus operaciones básicas son ampliamente similares a un grupo de instrucciones de máquina; los lenguajes de cuarta generación son sorprendentemente diversos en sus construcciones.

Lenguajes de quinta generación.

El término "quinta generación" usualmente se refiere a sistemas que usan disciplinas que se originaron en el campo de la inteligencia artificial como son especialmente:

- 1.- Sistemas basados en el conocimiento.
- 2.- Sistemas expertos.
- 3.- Máquinas de inferencia o deducción
- 4.- Proceso de lenguajes humanos.

1.2.2 LENGUAJES PROCEDURALES Y NO PROCEDURALES

Algunos lenguajes de cuarta generación son referidos como lenguajes "no procedurales", un lenguaje procedural especifica como algo es realizado sin describir como.

Lenguajes como COBOL y PL/1 son procedurales ya que en su programación se indica detalladamente como cada acción es desarrollada, en un generador de aplicaciones en donde los usuarios llenan formas para decir que hacer, se trata de un lenguaje no procedural. El usuario dice que se va a hacer y no le concierne el procedimiento detallado de como hacerlo; la mayoría de los lenguajes generadores de reportes, paquetes de gráficas y generadores de aplicación son no procedurales.

Programación amigable.

La tercera generación de lenguajes no fue amigable al usuario, se asumió que el programador haría un esfuerzo sustancialmente profesional para aprender la sintaxis del lenguaje; esta sintaxis no tiene relación con la sintaxis del lenguaje humano, aunque está permitido el uso de palabras en inglés.

Funcionalidad limitada.

Los lenguajes de cuarta generación varían ampliamente en su potencia y capacidad; algunos son únicamente lenguajes de consultas; otros son generadores de reportes o generadores de gráficas; otros pueden generar aplicaciones completas; algunos son lenguajes de programación de muy alto nivel.

Algunos lenguajes pueden ser empleados por usuarios finales o por analistas de sistemas directamente auxiliando a usuarios finales. Mientras que los lenguajes de tercera generación podrán crear todas o la mayoría de las aplicaciones; algunos de los lenguajes de cuarta generación están diseñados solamente para una clase o rango específico de aplicaciones.

En los lenguajes de cuarta generación, mucho más que en los de la tercera, se tiene que seleccionar el lenguaje para adecuarlo a la aplicación.

El término "lenguajes de cuarta generación de función completa" se emplea para referirse a lenguajes que permiten a sus usuarios construir cualquier aplicación como en COBOL. Los lenguajes como MANTIS, IDEAL, NATURAL y APPLICATION FACTORY son lenguajes de cuarta generación de función completa; SQL, QBE y ADRS están diseñados para propósitos más limitados.

Un proceso de una transacción en un lenguaje de tercera generación, usa muchos accesos a base de datos; con un lenguaje de cuarta generación como lo es NATURAL, se requiere sólo un acceso a base de datos. Con PLI por política del lenguaje se utilizan muchos mapas de pantalla, con NATURAL sólo se utilizan dos o tres mapas de pantalla y es mucho más rápido de usar por operadores de terminal.

Opciones Establecidas.

En la mayoría de los lenguajes de cuarta generación, un compilador o intérprete asume inteligentemente acerca de lo que el usuario piensa necesitar. Por ejemplo, puede seleccionar automáticamente un formato para un reporte, poner el número de página, seleccionar diferentes tipos de cartas para desplegados gráficos, poner etiquetas en los ejes o subir encabezados, y contestar al usuario de modo amigable cuando necesita más información.

El software puede requerir muchos parámetros a ser especificados para una cierta operación, para esto proporciona al usuario una lista de tales parámetros con los valores que son los apropiados; por ejemplo un usuario puede mover el cursor a un campo o celda dados sobre una pantalla y seleccionar la opción FORMATO. El programa puede listar opciones en línea, tipos de formato, número de decimales y los selecciona por sí solo si el usuario no especifica ninguno. Estas son llamadas "opciones establecidas".

Monodiálogo y diálogo.

La mayoría de lenguajes de tercera generación son un monólogo, con el usuario escribiendo un programa; muchos lenguajes de cuarta generación emplean un diálogo con el usuario y la computadora interactuando. Esta clase de diálogos dan mucho más alcance para lograr resultados rápidos ya que se puede captar la mayoría de los errores mientras estos son generados. Con esto se simplifica el proceso de depuración.

Diálogos unidimensionales contra diálogos bidimensionales.

Un diálogo puede emplear interacción unidimensional ó bidimensional. Con la interacción unidimensional, la computadora muestra al usuario listas de opciones de las cuales él puede seleccionar.

Con la interacción de dos dimensiones, el usuario responde a una pantalla, puede apuntar, proporcionar datos, o mover registros en partes de ella; también el usuario puede introducirse a un espacio imaginario de dos dimensiones de gran tamaño usando la pantalla como una ventana, o ser capaz de saltar de manera visual de una gran base de datos a otra separadas en tablas de dos dimensiones.

Simplicidad.

Para muchos, los lenguajes de cuarta generación, representan una búsqueda de la simplicidad en la programación. En los 60's y 70's, la computación era demasiado difícil. Fue la competencia de técnicos altamente especializados quienes aprendieron formas complejas de codificar, cerrándose en un mundo en el cual una persona sin el entrenamiento debido no podía penetrar.

Los lenguajes de cuarta generación como el NATURAL, se basan en la filosofía de poder decirle a la computadora más directamente lo que uno desea hacer. Inicialmente tenía carencias, pero lentamente se transformó en un lenguaje que puede realizar cualquier cosa incluyendo las que el COBOL no puede hacer.

Los lenguajes de cuarta generación debido a su relativa simplicidad, permiten ahora al usuario

dedicar su esfuerzo mental en lo que realmente importa, que es el propósito de la aplicación.

Principios básicos en el diseño de lenguajes de cuarta generación.

Los principios básicos que se deben seguir en el diseño de los lenguajes de cuarta generación son los siguientes:

I.- El principio del trabajo mínimo. Se pretende poner a trabajar a las computadoras con el mínimo esfuerzo.

II.- El principio de la mínima experiencia. Deberemos ser más capaces de poner a trabajar a las computadoras tan fácilmente como sea posible, sin necesidad de un entrenamiento especial, para que el mayor número de personas pueda emplearlas.

III.- El principio de evitar el uso de sintaxis y mnemónicos complejos. La construcción de lenguajes deberá ser ideada para evitar que los usuarios tengan que aprender sintaxis y mnemónicos que sean difíciles de recordar.

IV.- El principio del tiempo mínimo. Se pretende hacer posible utilizar las computadoras eficientemente sin retrasos prolongados en el desarrollo de las aplicaciones.

V.- El principio del mínimo de errores. Las técnicas deberán ser ideadas de manera que la probabilidad de errores humanos sea minimizada.

VI.- El principio del mínimo mantenimiento. Los mecanismos de los lenguajes de cuarta generación deben ser aplicaciones fáciles de cambiar. (esto es debido a que el mantenimiento de los lenguajes de tercera generación es muy costoso).

VII.- El principio de resultados máximos. Se necesitan aplicaciones que sean tan poderosas, útiles

e interesantes como sea posible; se debe permitir a los usuarios emplear herramientas complejas que soporten decisiones.

Una de las más importantes características de los lenguajes de cuarta generación es la de que sus programas deben poder ser usados por los analistas de sistemas trabajando con los usuarios finales.

Características básicas de los lenguajes de cuarta generación.

Se presenta a continuación una lista de los criterios de selección para los diferentes tipos de lenguajes de cuarta generación.

Criterios de selección.

1.- ¿Está diseñado para cómputo de rutina o para toma de decisiones?

2.- ¿Está diseñado para usuarios finales o para profesionales en proceso de datos? (muchos lenguajes de cuarta generación son apropiados para ambos).

3.- ¿Requiere de la habilidad de un programador, o puede usarlo un analista quien no programa en lenguajes de tercera generación?

4.- De cuales de las siguientes características está provisto:

a).- Simples consultas.

b).- Consultas y actualizaciones simples.

c).- Consultas complejas.

d).- Consultas y actualizaciones complejas.

e).- Capacidad para crear rápidamente una base de datos.

f).- Operaciones inteligentes de base de datos, donde el cambio de un valor en la base de datos origina que otras operaciones ocurran automáticamente, tales como chequeos de validación, referencias cruzadas, y actualización de valores relacionados.

g).- Generación de formatos en pantalla para la captura de datos por capturistas y que incluyan chequeos de validación.

h).- Generación de pantallas de actualización de datos para capturistas y que incluyan chequeos de validación.

i).- Es un lenguaje procedural que ofrece capacidad de programación completa.

j).- Cuenta con gráficas para diseño de aplicaciones.

k).- Está provisto de hoja de cálculo.

l).- Puede manejar matrices multidimensionales.

m).- Cuenta con generadores de reportes.

n).- Puede generar gráficas.

o).- Soporta decisiones para preguntas

p).- Puede manipular gráficas.

q).- Cuenta con herramientas de análisis matemático.

r).- Cuenta con herramientas de análisis funcional.

s).- Cuenta con otras herramientas de soporte de decisiones.

t).- Cuenta con Manipulación de textos.

u).- Tiene correo electrónico.

5.- ¿Es en línea o fuera de línea?.

6.- ¿Corre en macro computadoras, en mini computadores o en computadoras personales?

7.- ¿Puede acceder macro computadoras o bases de datos remotas?

8.- ¿Es genuinamente fácil de usar?

9.- ¿Los resultados pueden ser obtenidos rápidamente?

Para que un lenguaje programación pueda ser denominado Lenguaje de Cuarta Generación, éste debe tener las siguientes características:

1).- Es amigable al usuario.

2).- Un programador no profesional puede obtener resultados con él.

3).- Emplea directamente un sistema administrador de base de datos.

4).- Pueden crearse programas para la mayoría de las aplicaciones con menos instrucciones que con un lenguaje de tercera generación.

5).- Un código no procedural es usado, donde es posible.

6).- Hace asunciones inteligentes por política acerca de lo que el usuario desea cuando es posible.

7).- Es diseñado por operación en línea.

8).- Fortalece el código estructurado.

9).- Facilita entender y mantener el código de otras personas.

10).- Los usuarios que no están involucrados con el procesamiento de datos pueden aprender una aplicación del lenguaje con una capacitación de pocos días.

11).- Los prototipos pueden ser creados y modificados rápidamente.

12).- Está diseñado para fácil depuración.

13).- Los resultados pueden ser obtenidos en menor tiempo que con un lenguaje de tercera generación en la mayoría de las aplicaciones.

1.2.3 EFECTOS DE LOS LENGUAJES DE 4A. GENERACION.

Efectos de los Lenguajes de 4ta. Generación en la productividad del proceso de datos.

Una razón principal para decidirse por los lenguajes de cuarta generación, es la de incrementar la productividad en la construcción de las aplicaciones. La experiencia en alcanzar este objetivo, es extremadamente variada. Varía enormemente con el tipo de sistema que se está construyendo, con los individuos involucrados y con las técnicas de administración de proyectos utilizadas. En algunos casos el incremento en la productividad es muy bajo o nulo, debido a la inadecuada aplicación de las técnicas de administración empleadas, o porque los usuarios de los lenguajes de cuarta generación han hecho un desorden debido a la falta de una metodología de diseño.

Es común encontrar un programa de lenguaje de cuarta generación para un sistema moderadamente complejo teniendo una décima parte del número de líneas de código que tendría su equivalente en COBOL.

Un programador de COBOL escribe un promedio de 20 líneas de código por día. Esto es, dividiendo el número total de líneas de COBOL en un sistema terminado entre el número total de días de programación, incluyendo la depuración y reescrituración. Con un lenguaje de cuarta generación se logran frecuentemente de 40 hasta 100 líneas de código por día.

Razones de baja productividad.

1.- Se requiere de mucho aprendizaje para manipular con destreza algunos lenguajes de cuarta generación. También se requiere invertir recursos (tiempo y dinero) en capacitación.

2.- Algunos generadores de aplicación son limitados en lo que pueden generar. Cuando son aplicados a un sistema inapropiado, pueden causar

problemas y por esto no lograr los resultados esperados.

3.- Para lograr una importante reducción en el tiempo de desarrollo, se necesitan adecuadas técnicas de administración y control.

4.- Se requiere una adecuada metodología de diseño para el lenguaje de cuarta generación. Está es esencial para lograr una considerable productividad y beneficios en el mantenimiento.

5.- No son usadas las características de la herramienta de prototipo interactivo.

6.- Algunos lenguajes de cuarta generación no tienen la capacidad necesaria para desarrollar sistemas complejos.

Técnicas recomendadas de administración y diseño en los lenguajes de cuarta generación para el desarrollo de alta productividad.

1.- Un lenguaje de cuarta generación que imponga un código completamente estructurado con el máximo uso de funciones establecidas y no procedurales.

2.- Un generador de sistemas en vez de un generador de aplicaciones.

3.- Herramientas gráficas para la automatización del diseño.

4.- Un editor de diagramas activo para especificaciones y código procedural.

5.- Un diccionario en línea.

6.- Un modelo de datos normalizado.

7.- Estaciones de trabajo dedicadas para el desarrollo, ligadas a un diccionario o enciclopedia central.

8.- Prototipos rápidos.

9.- Un ciclo de vida de desarrollo interactivo.

10.- Conversión de prototipos a código final, auxiliada por computadora si es necesario.

11.- Documentación automatizada.

12.- Biblioteca de módulos reutilizables.

13.- Proyectos de una persona donde sea posible.

14.- Entrenamiento y práctica con las herramientas.

15.- División de programas grandes en módulos autónomos pequeños, ligados vía un modelo de datos (y posiblemente una enciclopedia).

16.- Uso de funciones pre-programadas o paquetes donde sea posible.

17.- Técnicas de administración adaptadas a las funciones o paquetes anteriores.

1.2.4 CUANDO SON USADOS LOS LENGUAJES DE 4A. GENERACION

Los lenguajes de cuarta generación son más comunmente usados en procesos de datos comerciales que en aplicaciones de ingeniería, sin embargo pueden ser encontrados en todos los aspectos de computación existentes, puesto que a todos hace referencia, ya que está extendido a áreas nuevas y vitales. Muchos de ellos pueden manejar muy bien el procesamiento de datos.

El poder tecnológico computacional de nuestros días no ha sido usado como debería ser. Los procesos de datos aún presentan problemas.

Los manejadores no reciben la información que requieren de sus sistemas. Muchas decisiones que podrían hacerse con la ayuda de las computadoras son de hecho hechas con métodos inadecuados de información. Los sistemas son difícilmente cambiados por sus dueños y rara vez la implementación de nuevos procedimientos es habilitada.

La computadora es la máquina más flexible jamás inventada, capaz de aplicar una diversidad de aplicaciones. El tradicional ciclo de vida de un desarrollo de aplicación es lento y rígido. Este método ha sido costoso en muchas organizaciones están hechos con estándares y procedimientos, pero en muchas ocasiones los procedimientos no están trabajando.

Esta inhabilidad de usar correctamente las computadoras podría regenerar un mayor problema de organización. Los problemas originan crisis proporcionales. Los lenguajes de cuarta generación, generadores de aplicaciones, lenguajes de especificación y lenguajes soportadores de decisiones proporcionan la solución de muchos problemas.

Comparando los lenguajes SQL, ADS, MAPPER, EXPRESS, LOTUS, etc con un lenguaje de tercera

generación es como comparar motores eléctricos con motores de vapor. La arquitectura y mantenimiento de el viejo software hacen que las adiciones sean muy elaboradas y proveen mínimos avances en donde un importante cambio es necesitado. Los procesos manuales son muy poco referenciados como la "biblia" el desarrollo de procesos de datos, con las nuevas técnicas esto se vuelve obsoleto. Los antiguos procedimientos de procesamiento de datos como el manejador de pasos pueden dar libertad y flexibilidad con los nuevos métodos.

APLICACION DE ACUMULACION (BACKLOG)

En las grandes corporaciones la demanda de nuevas aplicaciones están originando los más rápidos procesamientos de datos con los que pueden abastecerse. El desbalance entre demanda y oferta es prácticamente muy malo. Por lo que la acumulación de aplicaciones necesarias es grande.

MANTENIMIENTO.

Los problemas de procesamiento de datos y desarrollo de programas (software) es malo por el problema del mantenimiento. El término mantenimiento es usado para referir reescritura de programas viejos para acomodar nuevos requerimientos hacer trabajos con cambios en sistemas fuentes. La reprogramación es muy poco usada por la separación de programas desarrollados o por que los problemas de interfase existen cuando los datos son pasados de un sistema a otro. Una necesidad de cambio en un conjunto de programas podría encadenar una reacción de cambios que tienen que hacerse a otros programas.

1.2.5 DIALOGOS USADOS EN LENGUAJES DE 4A. GENERACION

Los lenguajes de cuarta generación son diseñados principalmente para su uso interactivo. Con ellos el usuario emplea un diálogo de terminal ó de computadora personal. Sin embargo, algunos de los mecanismos de diálogo son muy diferentes a los utilizados en lenguajes tradicionales.

Los lenguajes de tercera generación fueron diseñados principalmente para ocupar programadores profesionales trabajando con hojas de codificación. La nueva generación puede ser diseñada para su interacción con una pantalla de gráficos con tiempos de respuesta muy cortos.

No todos los lenguajes de cuarta generación han utilizado interacción con pantallas, y ninguno de los primeros utilizó computadoras personales. Con algunos, el usuario llenaba una forma, con otros, utilizaba una terminal como máquina de escribir.

ORIGEN DEL DIALOGO

Quando una persona se comunica con una terminal de computadora, la interacción se construye a través de pares de mensajes, cada par siendo una frase o pregunta seguido de una respuesta a ésto. En algunos casos la persona origina el par; él dice algo y la computadora responde. En otros casos la computadora origina el par y la persona responde. Podemos entonces diferenciar los intercambios originados por la computadora como "OC" y los intercambios originados por la persona como "OP". Algunos diálogos consisten enteramente de uno de ellos. Algunos contienen una mezcla de ambos.

La ventaja de los intercambios OP es la de que la persona puede meter todo lo que el software puede interpretar en cualquier momento. Existe ahí un gran nivel de flexibilidad. La desventaja es la de que la persona tiene que saber que es lo que el software puede aceptar y tiene que saber precisamente como formatearlo.

La ventaja de los intercambios OC es la de que el software puede indicar al usuario que es lo que debe de hacer. Puede dirigir al usuario en una serie de pasos. Las respuestas del usuario son generalmente breves y directas. La desventaja puede ser la pérdida de flexibilidad. El usuario no puede meter cualquier comando en cualquier momento. La secuencia de pasos que constituyen un diálogo OC puede tomar más tiempo que la de los diálogos OP.

Un diálogo OC trabaja mejor con usuarios inexpertos que no han estudiado un lenguaje de programación. Es normalmente mucho más fácil construir una interfaz amigable al usuario en una máquina con un diálogo iniciado por la computadora que una con un diálogo iniciado por la persona.

DIMENSIONES DEL DIALOGO

Podemos distinguir entre diálogos de una y dos dimensiones. En los diálogos de una dimensión, los participantes intercambian mensajes como si lo hicieran con una terminal como máquina de escribir. En un diálogo de dos dimensiones, el usuario reacciona o construye una imagen de dos dimensiones en una pantalla. El puede señalar a algo en la pantalla, mover el cursor o la tecla espaciadora, recorrer la imagen, llenar en partes de la pantalla, manipular diagramas o en algunos casos amplificar una imagen de tal manera que llene la pantalla, adicionando detalles a lo que está pasando.

Es natural y fácil para los humanos el reaccionar a los espacios de dos dimensiones. Nosotros estamos acostumbrados a usar hojas de papel y diagramas. La retina de nuestro ojo es de dos dimensiones. Es natural apuntar hacia un espacio de dos dimensiones, por lo que no es difícil utilizar los mecanismos para mover la información en la pantalla de una computadora. El tiempo de respuesta, sin embargo, debe ser lo suficientemente rápido. Es difícil

interactuar con una pantalla de largos tiempos de respuesta.

Utilizando una pantalla de dos dimensiones, una tabla o matriz puede ser rápidamente construida, puede ser llenada, llamada y manipulada. Esto proporciona una manera fácil de solicitar, actualizar y manipular datos en una base de datos.

EFFECTO DE LOS TIEMPOS DE RESPUESTA EN FRACCIONES DE SEGUNDO

Los primeros lenguajes de cuarta generación fueron diseñados principalmente para ser usados con simples terminales. Mas tarde los lenguajes de cuarta generación fueron diseñados para ser usados con computadoras personales conectadas al sistema reemplazando a las terminales y en algunos casos con computadoras personales independientes. Las computadoras personales responden en un décimo de segundo cuando realizan operaciones simples como la selección de un menú, mover una ventana ó recorrer la pantalla.

Una terminal en una línea telefónica puede tomar de 2 a 3 segundos antes de que la respuesta comience, y luego varios segundos para pintar la pantalla.

El tiempo de respuesta de las computadoras personales (un décimo de segundo) y estaciones de trabajo tienen un mayor efecto en la elección de las estructuras de diálogo que son muy valiosas en los lenguajes de cuarta generación.

Los menús son comúnmente utilizados como bases de los diálogos amigables-al-usuario. En estos, al usuario le es presentada una lista en pantalla y el selecciona un concepto en ella. Cuando se presentan muchas opciones a elegir, éstas son divididas de manera jerárquica.

Si el tiempo de respuesta es medido en segundos y el barrido de la pantalla es a velocidades telefónicas, el uso de menús jerárquicos puede ser muy lento. Eso común para los diseñadores de lenguajes de cuarta generación el reconocer esto y reemplazar el uso de los menús por una entrada donde los usuarios teclean un "comando". El problema con esto es que los usuarios tienen que saber cuál es el comando. Este a veces es un comando complejo con muchas opciones.

En algunos casos al usuario se le da a elegir. Puede optar por una selección de menús cuando está aprendiendo y luego puede elegir utilizar los comandos cuando se vuelve más experto.

La habilidad de moverse rápidamente a través de menús o estructuras jerárquicas, a una velocidad de décimas de segundo hacen de las técnicas amigables-al-usuario buenas tanto para el profesional como para el usuario final. En este ambiente, el software no deberá forzar a nadie a emplear comandos de sintaxis compleja. Su mayor utilidad deberá estar hecha de menús, listas de comandos, recorrido en pantalla de alta velocidad, ventanas, un ratón, y de rápidas gráficas interactivas.

MENUS AJUSTABLES

Existen algunas maneras en las que puede ser acelerado el proceso de responder a los menús o a las listas de comandos. Una es incrementando la utilización de las teclas TAB, SPACE ó teclas de dirección, de tal manera que el cursor de edición se mueva rápidamente a través de la pantalla a la posición requerida. Otra manera es la de proporcionar medios para recorrer largas listas de información en pantalla en grandes saltos. La lista debe de ser arreglada como una matriz, y las teclas de dirección deben permitir el movimiento de arriba hacia abajo, de izquierda a derecha. Otro método es utilizando un "Ratón".

El usuario puede emplear sólo algunas de las muchas opciones en un menú. En este caso, él puede decirle al software de visualizar sólo aquellas en las que está interesado, acortando mucho los menús que él ve y por tanto acelerando el proceso de selección de los menús. Si él desea una opción del menú no visualizada, él puede solicitar por el menú completo.

Los menús y listas de comandos pueden ser, por tanto, ajustados a las necesidades del usuario con el propósito de simplificar y acelerar sus requerimientos.

USO DEL RATON

Muchas computadoras personales emplean ahora el uso de un ratón. Este ahora puede ser adicionado a una computadora que originalmente no fue diseñada para tenerlo.

Utilizando el ratón, los apuntadores y la barra cursora pueden moverse alrededor de una pantalla rápidamente. Apretando la tecla del ratón sobre una opción de un menú, puede producir que se presente un menú más detallado. Apretando otra opción de este menú puede aparecer otro menú aún más detallado si se requiere y así sucesivamente. Con un poquito de práctica, un usuario usando el ratón puede navegar a través de menús y diagramas rápidamente, recorrer grandes espacios en las pantallas, etc..

Un ratón es muy valioso cuando se requiere el uso de gráficas. Las gráficas pueden representar la lógica ó estructura que será construida en un programa de cuarta generación o base de datos. Con un ratón, el usuario puede apuntar a parte de un diagrama y luego visualizar un menú o panel de instrucciones.

En suma, un ratón es mucho más rápido que un teclado en la manipulación de grandes menús o diagramas, pero los usuarios necesitan de práctica para lograr buena velocidad con él.

Un ratón es muy valioso cuando se requiere el uso de gráficas. Las gráficas pueden representar la lógica ó estructura que será construida en un programa de cuarta generación o base de datos. Con un ratón, el usuario puede apuntar a parte de un diagrama y luego visualizar un menú o panel de instrucciones.

En suma, un ratón es mucho más rápido que un teclado en la manipulación de grandes menús o diagramas, pero los usuarios necesitan de práctica para lograr buena velocidad con él.

CORRIENTES EN LOS LENGUAJES DE CUARTA GENERACION

En relación a los conceptos que se han presentado referentes a las estaciones de trabajo, terminales y tiempos de respuesta, podemos encontrar varias corrientes en el desarrollo de los lenguajes de cuarta generación.

1.- Lenguajes fuera de línea, en donde el usuario tiene que llenar una serie de formas. Estos son limitados en comparación con los lenguajes orientados a terminales. Ejemplos son el MARK IV y el DMS.

2.-Lenguajes que pueden operar en una terminal como máquina de escribir y pueden procesar una serie de caracteres. Ejemplos son NOMAD, SQL, y DATATRIEVE.

3.-Lenguajes diseñados para unidades ciegas de visualización, en las que el usuario interactúa con una visualización de dos dimensiones. La interacción permite al usuario un diálogo más amigable que los diálogos de una dimensión que operan en las terminales como máquinas-de-escribir. Ejemplos son MAPPER, FOCUS, RAMIS, y QUERY-BY-EXAMPLE.

4.-Lenguajes diseñados para obtener respuestas en décimas de segundo en las computadoras personales ó redes locales. Los

5.-Lenguajes utilizando gráficas para el diseño de programas. El uso de diagramas es muy útil pues nos ayuda a visualizar y a diseñar estructuras complejas. Hoy en día se cuenta con editores gráficos con los que los diagramas pueden ser construidos, manipulados y editados. Estos editores pueden correr efectivamente en computadoras personales que cuenten con una pantalla gráfica y un ratón. Con un editor gráfico, un diseñador de sistemas puede diseñar la lógica de su aplicación por medio de una computadora que le proporcione varios tipos de ayuda. Si la técnica de diagramación es adecuada e incorpora herramientas tales como ayudas para el diseño de pantallas, ayuda para el diseño de reportes, y diccionario de datos, entonces puede ser generado el código de los diagramas.

USO DE TECNICAS DE INTELIGENCIA ARTIFICIAL

Trabajos originalmente clasificados como de inteligencia artificial han recaído al software que puede frasear la gramática compleja del Inglés. Esto es usado hoy en día en lenguajes de aplicación tales como INTELLECT y RAMIS ENGLISH. Para mejorar las aplicaciones complejas con el Inglés natural, el procesamiento del lenguaje necesita ser combinado con otras construcciones. Es de particular interés el combinar el procesamiento del lenguaje humano con gráficas.

El trabajo de inteligencia artificial ha sido también dejado a los sistemas basados en reglas en los que las reglas son almacenadas a través de datos en una "base de conocimiento". Aplicaciones complejas pueden ser construidas usando tales reglas para el procesamiento inferencial. La interpretación del lenguaje natural emplea tales técnicas, en donde las reglas para interpretar la gramática son almacenadas en la base de conocimiento junto con el vocabulario. El procesamiento de reglas es una característica deseable en la capacidad actual de los lenguajes de cuarta generación. Es conveniente que estos lenguajes se envuelvan con las herramientas de los lenguajes de quinta generación en las que el procesamiento inferencial, el procesamiento basado en reglas, las bases de conocimiento y el lenguaje natural humano juegan un papel muy importante.

El trabajo de inteligencia artificial ha sido también dejado a los sistemas basados en reglas en los que las reglas son almacenadas a través de datos en una "base de conocimiento". Aplicaciones complejas pueden ser construidas usando tales reglas para el procesamiento inferencial. La interpretación del lenguaje natural emplea tales técnicas, en donde las reglas para interpretar la gramática son almacenadas en la base de conocimiento junto con el vocabulario. El procesamiento de reglas es una característica deseable en la capacidad actual de los lenguajes de 4ta. generación. Es conveniente que estos lenguajes se envuelvan con las herramientas de los lenguajes de 5ta. generación en las que el procesamiento inferencial, el procesamiento basado en reglas, las bases de conocimiento y el lenguaje natural humano juegan un papel muy importante.

1.2.6 LENGUAJES PARA PROTOTIPOS

Los lenguajes de 4a. generación son utilizados tanto para la creación de prototipos como para la creación de la aplicación final. Los prototipos son extremadamente valiosos en el procesamiento de datos.

Con la más compleja ingeniería, se crea un prototipo antes de que se construya el producto final. Esto se hace para probar los principios, asegurar que el sistema trabaja, y obtener la retroalimentación necesaria en el diseño que permita que éste sea ajustado antes de que se realice la mayor inversión de dinero.

Sistemas complejos de procesamiento de datos necesitan de los prototipos más que la mayoría de los sistemas de ingeniería, debido a que hay mucho que aprender sobre su operación experimental, y probablemente se tendrán que realizar muchos cambios a la idea original. Los prototipos nos ayudan a resolver los problemas de los sistemas que no trabajan de la manera en que los usuarios finales realmente necesitan, y esto reduce significativamente las modificaciones que se tienen que hacer.

En un sentido, un sistema creado por el "ciclo de vida" tradicional del procesamiento de datos es un prototipo - no realizado para serlo y no considerado como tal-, pero conteniendo todas las imperfecciones de un prototipo. Estas imperfecciones son costosas de corregir, por lo que permanecen en el sistema, conduciendo a un mantenimiento costoso.

La razón por la que los prototipos de procesamiento de datos no fueron populares hasta la década de los 80, fué la de que el costo de la programación de un prototipo era tan alto como la del sistema de definitivo. Los lenguajes de cuarta generación permiten la creación de prototipos relativamente baratos.

Un analista de sistemas trabajando con un usuario final puede crear y demostrar diálogos

para requerimientos de base de datos, para la generación de reportes, y para la manipulación de la información en pantalla. Durante este proceso, el analista discute con el usuario final las necesidades y luego crea un espécimen del dialogo en una terminal. Esto puede tomar una hora ó una semana, dependiendo de la complejidad y del lenguaje que se está utilizando.

Al crear prototipos para ser usados por los usuarios externos, especialmente por los mandos mayores, es importante permitirles que ellos vean y utilicen la información que les será proporcionada cuando el sistema esté en operación, ya que ellos siempre realizan cambios.

Al usuario final se le muestra el diálogo y rápidamente es entrenado para poder utilizarlo. Por lo general él tiene varias sugerencias de cambios que le gustaría hacer, y el analista los realiza. El usuario puede añadir subtotales o columnas adicionales. El puede querer ejecutar ciertos cálculos. El analista puede mostrarle los diferentes tipos de tablas que pueden ser creadas: gráficas dispersas, gráficas de barras, gráficas con líneas de regresión, escalas lineales contra logarítmicas, etc. El usuario recuerda a un tipo de cliente diferente, o algún otro aspecto ó factores que sólo el usuario puede saber.

Durante este proceso, son descubiertas la mayoría de los malentendidos, ambigüedades y/o inconsistencias. Usuarios diferentes tienen diferentes interpretaciones de los mismos datos.

La operación de un sistema por varias semanas siempre cambia la percepción del usuario de lo que realmente quiere. La utilización del prototipo permite que esto ocurra tempranamente en el ciclo de desarrollo, y no después de la implementación, donde los cambios son costosos.

Así como el analista y el usuario continúen su discusión acerca de lo que es necesario, el prototipo ejecutable es un foco de debate; esto ayuda a asegurar que ambos están hablando de la

misma cosa. Las pantallas son impresas, el usuario final las toma y lleva a casa para pensar en ellas. El analista trabaja más adelante, mejorando la interacción de pantallas, adicionando nuevas características y mejorando los mensajes.

Finalmente, el usuario está satisfecho y lo requiere lo más pronto posible. En algunos casos lo puede tener muy rápidamente. La base de datos existe, y el prototipo puede convertirse en la aplicación final. En otros casos no se puede obtener aún porque se tendrá que hacer trabajo de diseño para mejorar la eficiencia de la máquina, la seguridad, auditoría, la telecomunicación con redes, o la creación de la base de datos.

En último caso el prototipo viene a ser el documento de requerimientos para la programación de la aplicación.

CONVERSION

Actualmente se desarrollan los prototipos y luego son convertidos a un lenguaje de tercera generación. Sin embargo, el prototipo puede convertirse en el sistema final.

Una razón para no convertirlo a COBOL o a otro lenguaje de tercera generación, es la de que tan pronto como es convertido, se vuelve caro de mantener, mientras que si se mantiene en el lenguaje del prototipo, las modificaciones son fáciles de hacer.

El argumento más común para la conversión es la eficiencia de la máquina. La eficiencia de la máquina es una cuestión concerniente sólo para aplicaciones de servicio pesado. En una instalación típica de procesamiento de datos, el 50% de las aplicaciones consumen un total del 2% de los ciclos de máquina. Sus costos de máquina son mucho menores que sus costos de mantenimiento en COBOL. Más aún, algunos generadores de aplicaciones que son apropiados para la realización de prototipos son diseñados

para proporcionar buena eficiencia de máquina, algunas veces mejores que las aplicaciones de COBOL, donde su construcción de bloques es rígidamente escrita en código de máquina.

Los equipos de trabajo de procesamiento de datos que sienten la urgencia de convertir los prototipos a COBOL deben calcular el costo de la conversión, más cerca de cinco años de mantenimiento y luego comparar estos costos con los ahorros de máquina y los costos de mantenimiento en los lenguajes de prototipos. Existe otro costo intrínseco al tener la aplicación en lenguajes de tercera generación: Su reducida capacidad de cambio, lo que puede resultar en la pérdida de oportunidades de negocios o en grandes ineficiencias.

Otras razones importantes para optar por usar el prototipo son, entre otras, la de alcanzar mejor seguridad, mayores controles de integridad, contar con protección contra caídas del sistema, rentabilidad, y auditoriedad. Pueden ser otras razones la habilidad de tener mayores bases de datos, más terminales, mejorar la red, ó mayores volúmenes de transacción.

Cuando se selecciona la técnica del prototipo, se debe tener en mente el realizarlo para su utilización en el sistema final.

TRES CLASES DE DESARROLLO DE APLICACIONES

Existen tres clases de desarrollo de aplicaciones:

1.- Desarrollo tradicional con ciclo de vida tradicional. Este tiene los problemas de bajo desarrollo, largas reservas, mantenimiento costoso, e inconcordancia con los requerimientos del usuario.

2.- Usando un lenguaje de 4ta generación como una herramienta para el prototipo. El usuario final interactúa con el prototipo y el analista lo modifica hasta que sea un modelo

adecuado para programar la aplicación con un lenguaje de tercera generación. Algunas veces se crean muchas versiones del prototipo. La herramienta para el prototipo alivia muchas de las necesidades de los requerimientos de documentación. Pueden ser creados módulos de código por la herramienta de prototipos para no tener que programar toda la aplicación de nuevo.

3.- Utilizando un lenguaje de 4ta generación para desarrollar la aplicación completa. En este caso, el prototipo se convierte en la aplicación. Un usuario final y un analista de procesamiento de datos pueden trabajar juntos para crear las aplicaciones.

El tercero de esta clasificación de desarrollos puede ser utilizado para la mayoría de las aplicaciones comerciales, las cuales muchas de ellas no necesitan estar muy involucradas con la eficiencia de la máquina, porque no son ejecutadas frecuentemente. Solamente una pequeña parte son para aplicaciones de grandes volúmenes.

Para aplicaciones de carga pesada, se requiere una gran eficiencia de la máquina. Esta puede ser alcanzada de una de las tres maneras siguientes:

1.- Primero, ya hemos acentuado que algunos lenguajes de cuarta generación son diseñados para alcanzar gran eficiencia de máquina. Un especialista que conoce todos los trucos con un generador de aplicaciones puede alcanzar muchos más resultados que el mismo analista. El analista puede crear una aplicación para carga pesada en DMS o ADF y luego pasarla a un DMS o ADF "acróbata" para optimizarla.

2.- Algunas rutinas pueden ser "comedoras de tiempo". Estas pueden ser aisladas y programadas en un lenguaje más eficiente si el generador cuenta con SALIDAS adecuadas.

3.- La aplicación completa puede ser reprogramada por eficiencia. En lugar de escribir la documentación de requerimientos, es

utilizada la versión del generador para guiar al equipo de programación. Algunas veces el generador crea solamente parte de lo que es requerido, por lo que es necesario un generador híbrido de entrada/salida además de la escritura de las especificaciones.

La tabla de la página siguiente muestra las características típicas de estos tres aspectos del desarrollo de aplicaciones. En organizaciones muy grandes, es posible que coexistan estos tres tipos de desarrollos. Un buen ejecutivo de procesamiento de datos organizará su operación de manera que éstas puedan coexistir. En algunas organizaciones, el departamento de procesamiento de datos utiliza solamente el ciclo de vida de desarrollo tradicional, y los usuarios finales se saltan al departamento de procesamiento de datos para crear sus propias aplicaciones utilizando más rápida y eficientemente los nuevos lenguajes.

PROTOTIPO PARCIAL DE SISTEMAS

Algunos esfuerzos en la creación de prototipos crean una versión piloto de una aplicación especial. Algunos sólo encuentran una faceta de una aplicación. Los prototipos PARCIALES han demostrado ser particularmente valiosos en algunos sistemas. Este puede presentarse en una gran variedad de formas:

Prototipo de Diálogo

El prototipo simula la interacción esperada con la terminal. Esta es la forma más común del prototipo parcial. Este permite al usuario final ver lo que estarán recibiendo, jugar con él, sugerir omisiones, mejorar la comprensión, reaccionar con el diálogo, y finalmente aceptar su desarrollo. Varios productos de software pueden ser utilizados como simuladores de diálogo.

TRES TIPOS DE DESARROLLOS DE APLICACIONES

	DESARROLLO CONVENCIONAL DE APLICACIONES	INGENIERIA DE LA INFORMACION Y LENGUAJES DE CUARTA - GENERACION	
		USO DE UNA AYUDA PARA PROTOTIPOS SEGUIDO DE PROGRAMACION	DESARROLLO DE APLICACIONES SIN PROGRAMADORES PROFESIONALES
ANALISIS DE REQUERIMIENTOS	Operacion consumidora de tiempo, generalmente retrasada por grandes reservas de aplicaciones	La imaginacion del usuario es estimulada. El puede trabajar en la pantalla con un analista para desarrollar los requerimientos	La imaginacion del usuario es estimulada. El puede desarrollar sus propios requerimientos o trabajar con un analista
DOCUMENTACION DE ESPECIFICACIONES DEL SISTEMA	Documentacion lenta, aburrida, generalmente inadecuada	Producida con una ayuda para prototipos; precisa y adecuada	Desaparece
APROBACION DEL USUARIO	El usuario no esta seguro de lo que esta aceptando. El no puede percibir todos los detalles	El usuario ve los resultados y puede modificarlos muchas veces antes de aprobarlos	No hay aprobacion formal. Los ajustes y modificaciones son un proceso comun
DATOS	Por lo general, disenados separadamente para cada aplicacion	Planeado con analisis de la informacion y modelo de datos	Planeado con analisis de la informacion y modelo de datos; puesto a disposicion en sistemas de datos
CODIFICANDO Y PROBANDO	Lento, caro, por lo general tardado debido a la logistica	El prototipo es convertido a un codigo mas eficiente; relativamente rapido y libre de error.	Rapido, no caro, desaparece en mayor grado
DOCUMENTACION	Tediosa, consumidora de tiempo	Puede ser automatizada parcialmente. Puede ser creado y activado en linea respuestas de AYUDA y entrenamiento interactivo	Grandemente automatica; entrenamiento interactivo y respuestas de AYUDA son creadas en linea
MANTENIMIENTO	Lento, caro, por lo general tarde	Menos lento, menos caro, menos tarde; solo son necesarias algunas modificaciones finales	Un proceso continuo, estando el usuario y el analista haciendo ajustes

El diseño del diálogo terminal afecta grandemente la utilidad y la percepción por los usuarios del sistema. Muchos sistemas han fallado a causa de los deficientes diálogos terminales. El prototipo de diálogo ayuda a construir un diálogo de prototipo que puede ser probado, criticado, y mejorado antes de la implementación final.

Entrada de Datos

Un grupo de usuarios puede ejecutar las entradas de datos. El subsistema de entrada de datos puede ser desarrollado en prototipo, ajustado independientemente y puede acoplarse a un sistema existente. El prototipo de entrada de datos es realizado para revisar la velocidad y la exactitud de la entrada de datos. Pueden ser efectuadas, además, revisiones de la integridad y de validación.

Algunos sistemas han sido divididos en "extremo frontal" y en "extremo posterior". El extremo frontal es interactivo. El extremo posterior consiste de múltiples corridas de actualización en lote. El extremo frontal puede hacerse en un prototipo independiente utilizando software tal como MAPPER, FOCUS, RAMIS, ó NOMAD. El extremo posterior puede permanecer en la forma de programas en COBOL.

Sistema de Reportes

Los reportes pueden ser proporcionados a los usuarios antes de la implementación completa del sistema. Estos pueden ser ya sea en lote o en línea. Por lo general se tienen que hacer muchos ajustes en el subsistema de reporte. Pueden ser usados generadores de reportes tales como RPG, NOMAD, ó ADRS.

Sistema de Datos

Una base de datos prototipo puede ser implementada con un número pequeño de registros.

Usuarios y analistas interactúan con él, generando reportes ó presentando información en pantalla, que puede serles muy útil. Esta interacción resulta en requerimientos para diferentes tipos de datos, nuevos campos, ó en diferentes maneras de organizar los datos.

Con algunas herramientas de prototipos, los usuarios y analistas tienen la habilidad de construir sus propios archivos, manipularlos, y de presentar la información en pantalla. Tales herramientas son utilizadas para explorar que tanto los usuarios emplearán la información, y qué es lo que deberá de estar en la base de datos.

Lógica y Cálculos

Algunas veces la lógica y los cálculos en una aplicación son muy complejos. Los usuarios técnicamente especializados tales como ingenieros, actuarios, etc. pueden usar lenguajes como el APL para la construcción de los modelos matemáticos que necesiten. Estos pueden ser incorporados posteriormente a grandes sistemas, o por lo menos, acoplados a otras aplicaciones, bases de datos, o a muchas terminales. El usuario puede utilizar sus prototipos en APL para revisar la exactitud de los resultados.

Paquete de Aplicaciones

Un paquete de aplicaciones es desarrollado con un pequeño grupo de usuarios para determinar si éste es satisfactorio.

En el caso de que se tenga la necesidad de realizar varias modificaciones, éstas pueden lograrse antes de que el paquete de aplicaciones sea encadenado a otras aplicaciones o pueden ser puestas en un volumen para su uso.

Concepto

Algunas veces no se sabe el concepto de una aplicación. Este necesita de pruebas y refinamiento antes de que se invierta mucho dinero en la construcción del sistema. La prueba puede ser realizada con la ayuda de un manejador de sistemas rápido de implementar. En este caso, se utilizan entradas de datos y formatos de reporte standard de manera que los conceptos pueden ser probados y refinados sin mucho trabajo. Posteriormente, se realizan reportes específicos de la aplicación y pantallas.

DIFERENTES EQUIPOS

En algunos casos el prototipo se realiza en un equipo diferente al que será utilizado en la aplicación final. Esto puede ser cuando por ejemplo todavía no está disponible el equipo final. También porque puede ser mucho más fácil experimentar con una computadora pequeña que con el complejo sistema final. Es por esto necesario adaptar ayudas al prototipo para que éste sea fácilmente adaptable al sistema final.

DATOS REALES

En el prototipo, el analista crea pequeños archivos con datos ficticios para ilustrar lo que el sistema realizará. En otros casos, los datos ficticios no son lo suficientemente buenos. Los usuarios necesitarán actualizar datos reales o explorar bases de datos complejas con objeto de tomar experiencia de lo que el sistema propuesto hará por ellos.

Si se requieren datos reales, los prototipos pueden estar conectados a un sistema en línea de datos ó se le puede proporcionar un archivo de datos que hayan sido sacados del sistema de datos real. Esto último es generalmente más seguro y flexible.

En algunos desarrollos de prototipos, el usuario pregunta por información de varios

tipos. El analista debe entonces encontrar donde existe esa información, capturarla, y reconstruirla en el sistema manejador de datos ó herramienta de prototipo. Algunas veces esta información existe en archivos en lote, algunas veces incorporados a bases de datos, ó algunas veces pueden ser obtenidos de fuentes externas.

Cuando se utilizan datos reales, el prototipo se convierte en un sistema que los usuarios aprovechan. Entonces se transforma en un sistema real de trabajo.

PILOTOS DESECHABLES

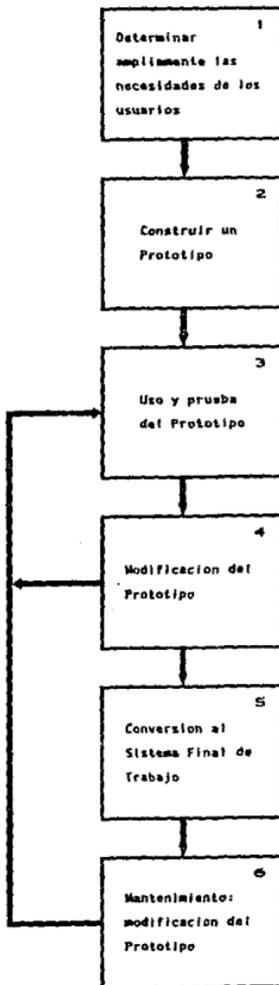
Las áreas de desarrollo ó los grupos de usuarios finales pueden crear sistemas piloto y mostrarlos a otros usuarios para que ellos los apliquen.

Con el uso de pilotos desechables, los diseñadores de sistemas descubren los requerimientos y refinamientos que los usuarios finales utilizarán finalmente. Es un proceso de prueba y error: construir un piloto, observarlo en uso, evaluarlo, revisarlo si eso fuese necesario, ó reemplazarlo con una versión diferente.

EL PROCESO DE DESARROLLO CON EL USO DEL PROTOTIPO

El proceso del prototipo, ilustrado en la página siguiente, presenta un ciclo diferente en el desarrollo del procesamiento de datos, en comparación con el ciclo tradicional. Debido a que los prototipos varían mucho en su carácter y uso, existen diferentes versiones de este ciclo de vida.

CICLO DE VIDA DE DESARROLLO DE LOS PROTOTIPOS



El primer paso es la clara determinación de que es lo que los usuarios finales necesitan. Por lo general los usuarios hacen peticiones. Un analista las estudia, determina qué datos requiere, y determina que podrá ser realizado en el prototipo. Para algunos sistemas esto puede ser informal, sin ninguna especificación escrita. Para sistemas grandes, esto puede involucrar detalladas descripciones escritas de qué datos son necesarios, diagramas de flujo, ó diagramas de acción mostrando como será utilizada una base de datos.

En el segundo paso, se crea un prototipo de trabajo. Es importante que esto sea hecho rápidamente. El prototipo puede utilizar las facilidades estándares disponibles ó un sistema manejador de datos con el propósito de demostrar algo a los usuarios. En un principio, éste puede demostrar solamente las principales funciones y todavía no incluir los detalles periféricos, como los requerimientos de auditoría, etc..

El tercer paso puede variar de una pequeña demostración inicial a un grupo de usuarios a una directa y refinada prueba con muchos usuarios. En esta etapa, algunos cambios solicitados por los usuarios pueden ser hechos inmediatamente en pantalla. Otras peticiones son sólo anotadas para su posterior modificación. Con algunas herramientas para prototipos, los usuarios finales que se encuentran con terminales situadas en locaciones múltiples pueden emplear el prototipo y pueden registrar sus peticiones ó comentarios en la terminal.

Para demostrar la gran utilidad del prototipo en la dirección de una empresa, los ejecutivos son invitados a hacer preguntas de lo que a ellos les gustaría que el sistema respondiera. El analista que desarrolló el prototipo modificará entonces el sistema lo más posible para satisfacer sus requerimientos.

Si en el paso 5 ocurre una conversión a un software diferente, la versión original deberá conservarse, ya que puede necesitarse para un futuro mantenimiento, regresando al paso 3, tal y como se muestra en la figura.

regresando al paso 3, tal y como se muestra en la figura.

PROTOTIPO DE LENGUAJES

Los lenguajes de cuarta generación utilizados para el desarrollo del prototipo, son generalmente los mismos que aquellos usados para el desarrollo de las aplicaciones. El analista de sistemas que construye prototipos puede ser un cliente del centro de información que soporta a los usuarios de estos lenguajes.

La más importante característica del prototipo de software es la de que éste debe poder permitir desarrollar al prototipo rápidamente. El desarrollo del prototipo requiere de un medio rápido y barato para efectuar ensayos de prueba. La segunda característica más importante es la de que los usuarios, ó los analistas trabajando con ellos, deberán poder hacer ajustes fácil y rápidamente.

Esas características están también a la cabeza de los requerimientos de los lenguajes de cuarta generación, diseñados para minimizar el personal requerido para la creación, mantenimiento y aplicación de los sistemas.

A pesar de que los requerimientos entre el desarrollo de un prototipo de lenguaje y un lenguaje de desarrollo son muy parecidos, difieren en que el lenguaje de desarrollo puede ser usado para obtener una buena eficiencia de la máquina, mientras que esto no concierne al lenguaje de prototipos. El lenguaje de desarrollo puede dar un tipo de formato fijo en pantalla; el desarrollador de prototipos puede proporcionar mayor flexibilidad en el diseño y en la experimentación con diferentes pantallas. El lenguaje de desarrollo puede utilizar un sistema manejador de base de datos tradicional, mientras que el lenguaje de prototipos puede utilizar un sistema manejador de datos fácil-de-implantar.

A pesar de esas diferencias, existe una buena razón para hacer al lenguaje de prototipos el mismo que el lenguaje de desarrollo: el prototipo puede convertirse en el sistema final. Esto tiene mejores ventajas: La primera, es que esto salva mucho tiempo de desarrollo. La segunda, es que da continuidad entre el prototipo y el sistema final ya que da la oportunidad de que los usuarios se vuelvan familiares con el sistema final. La tercera, es que da la posibilidad de modificar el sistema final en la misma manera en que se modifica al prototipo.

1.2.7 EVOLUCION DE LOS LENGUAJES

En la mayoría de los trabajos escritos que narran el futuro de los lenguajes de programación, se piensa que los futuros lenguajes serán extensiones de los lenguajes tradicionales. Dada la situación actual que se vive en el campo de la programación y de las computadoras, existen varias razones para dudar de esto, las cuales se presentan a continuación:

Los analistas para el desarrollo de aplicaciones usaran microcomputadoras. En un futuro las microcomputadoras serán más poderosas, con un poder comparable a una macro-computadora de los años 70's.

El uso de los monitores gráficos será muy sofisticado, debido a las grandes inversiones que se realizan en el campo de los videojuegos.

El mayor énfasis en el desarrollo de los lenguajes será en los lenguajes de especificaciones en lugar que los lenguajes de programación. Podrán ser construidas especificaciones computables mediante paquetes como el de Diseño Apoyado por Computadora "CAD" y luego convertidas rápidamente al código de máquina.

El uso de gráficos sofisticados será muy importante para el diseño de especificaciones.

Se generalizará al máximo el uso de sentencias o comandos estándares y con esto se minimizará el tiempo y esfuerzo en el desarrollo de aplicaciones.

Aparecerán lenguajes "inteligentes" que permitan un trato más amable con usuarios finales y serán capaces de desarrollar y dar mantenimiento a sus aplicaciones en forma rápida y fácil.

Se implementarán las "bases de datos inteligentes".

Se pondrán en uso las "Aplicaciones Inteligentes" para el desarrollo de los "Sistemas de Aplicación" más convencionales.

ESPECIFICACIONES COMPLEJAS.

Uno de los grandes problemas actuales es la creación de especificaciones de sistemas complejos, ya que estas generalmente llegan a ser inconsistentes, ambiguas y con omisiones, principalmente si se recurre a un grupo de analistas en lugar de un solo analista.

Es importante hacer uso de las aplicaciones que internamente desarrollan un pseudocódigo o incluso parte del código del lenguaje de programación al estar desarrollando las especificaciones de un sistema; tal es el caso de las llamadas "Herramientas CASE". Al hacer uso de este tipo de herramientas las especificaciones sin importar lo complejo o el número de analistas que participen en el desarrollo del sistema no serán ambiguas, ni inconsistentes y tampoco omitirán parte de las sesiones del sistema.

DIAGRAMAS.

Para representar alguna secuencia o especificación compleja muchas veces nos apoyamos en diagramas. Una gran variedad de técnicas en diagramación han sido desarrolladas, algunas para ser amigables con los analistas, otras para ser más rigurosas y algunas otras para ser convertidas directamente en código fuente.

Los Analistas de sistemas en un futuro deben hacer uso de las nuevas "herramientas de diagramación" que lleguen a ser desarrolladas, ya que el objetivo de estas es el de llegar a un determinado detalle donde el flujo de datos y la relación de bases de datos sea

rigurosamente checado por el mismo paquete, y una vez checado el flujo de bloques a detalle este pueda ser convertido fácilmente en especificaciones y en código ejecutable.

INTERACCION NATURAL.

Anteriormente la relación Máquina-Hombre era muy fría y especial puesto que las computadoras sólo se encontraban en laboratorios matemáticos o de ingeniería, y el lenguaje con el que se comunicaban era muy abstracto y con una sintaxis muy rigurosa. Actualmente esto ha ido mejorando, principalmente con la inclusión de las computadoras personales en el mercado cotidiano.

Es probable que en el futuro la comunicación con los lenguajes expertos sea por medio de un lenguaje cotidiano, donde el software de estas máquinas sea capaz de eliminar una posible ambigüedad entre ella y el usuario final, así como tener la capacidad de aprender nuevas sentencias que se convertirán en la fuente de cierta información (acceso a información de su base de datos inteligente). Es esta la quinta generación de lenguajes que los científicos, principalmente Japoneses tratan de desarrollar.

SISTEMAS EXPERTOS.

Los sistemas expertos de la quinta generación de lenguajes serán desarrollados para el uso de otros sistemas expertos, así como de personas expertas y personas cotidianas. Sistemas que tendrán las capacidad de auxiliar al usuario final en realizar millones de comparaciones por segundo para finalmente dar una opinión muy certera a la serie de preguntas que se la hagan.

SISTEMAS EXPERTOS PARA SISTEMAS EXPERTOS

Las personas que necesitan particularmente la ayuda de los sistemas expertos son aquellas que desarrollan aplicaciones y sistemas complejos para computadoras. Este trabajo se torna cada vez más complejo ya que día a día queremos que se desarrollen herramientas nuevas y automáticas que permitan el desarrollo de aplicaciones más completas y de uso fácil y amable, es decir un sistema experto. Para desarrollar este tipo de herramientas los analistas tienen que trabajar con configuraciones más elaboradas, bases de datos distribuidas y a redes de computadoras complejas ya que un sistema experto debe guiar al usuario final a resolver problemas del mismo sistema complejo o a resolver errores del mismo usuario.

Algunas de las características de los sistemas expertos son las que se mencionan a continuación:

- El uso de un lenguaje natural para comunicarse con la computadora como lo es el inglés.

- Asistencia en como operar la computadora

- Asistencia en como operar la aplicación (software)

- Diagnóstico de problemas (faltas) del sistema o errores del usuario mismo.

- Asistencia en la generación de especificaciones y el grado de las mismas para poder generar códigos ejecutables automáticamente.

- Asistencia en la creación de gráficos y los programas para generar los mismos.

2 METODOLOGIA

2.1 CONCEPTOS BASICOS DE PROGRAMACION NATURAL

Los lenguajes de cuarta generación varían de los de tercera en su poder y capacidad. Algunos son únicamente aplicadores de consultas (Query), otros solamente son reporteadores o generadores de gráficas, otros pueden generar aplicaciones completas y algunos otros son lenguajes de programación de muy alto nivel. Muchos pueden ser usados directamente por usuarios finales o por analistas de sistemas interrelacionados con sus usuarios.

Un lenguaje de cuarta generación puede describirse como aquel lenguaje que proporciona una alta productividad.

2.1.1 DESCRIPCION Y CARACTERISTICAS

NATURAL es un lenguaje de cuarta generación que es utilizado para el desarrollo en bases de datos, se le ha llamado así puesto que su sintaxis y gramática asemejan mucho al idioma inglés, es decir, programar en lenguaje NATURAL es como si se estuviera conversando con la computadora en inglés. Natural es un lenguaje de cuarta generación para el manejo de la base de datos ADABAS. Con estas herramientas de programación se desarrolló el presente sistema.

Un lenguajes de cuarta generación como ya hemos mencionado tienen las siguientes características principales :

- Código mínimo
- Tiempo mínimo de procesamiento.
- Mantenimiento mínimo

- Maximización de resultados
- Simplificación de trabajo

Los lenguajes de cuarta generación son utilizados más comúnmente en procesamientos de datos comerciales que en aplicaciones científicas o de ingeniería, puesto que manejan una gran cantidad de datos. Tienen una enorme aplicación para el desarrollo de prototipos y aplicaciones finales, así como un alto grado de rendimiento de CPU y puede trabajar tanto en ambiente de sistemas micro como macro.

AMBIENTE OPERATIVO

El lenguaje de programación Natural versión 2.1 es el manejador de la base de datos ADABAS, la cual trabaja en ambiente multiusuario. Puede realizar tanto tareas en línea (ON LINE) como tareas en lote (BATCH), este último tipo se controla por medio de un lenguaje controlador de trabajos (JCL), esto tiene una gran utilización, ya que no es necesario bloquear una terminal de la computadora mientras se esta ejecutando una tarea que requiere mucho tiempo de procesador central (CPU).

Existen dos modos de programación que son el modo reporte y el modo estructurado. El primero es un tipo de programación no estructurado que tiene que ver mucho con la versión anterior al natural 2.1 y por lo tanto tiende a dejar de utilizarse. El modo estructurado tiene una gran aplicación pues hace a los programas "entendibles" y como beneficio podemos mencionar una gran facilidad para dar mantenimiento a los programas fuente, además de que es posible realizar por medio de este una programación modular.

Una aplicación de NATURAL para una base de datos es una librería creada por el administrador de esta, para que residan en ella todos los programas que necesita un sistema de desarrollo o de producción. La seguridad de los sistemas es importante por lo que NATURAL 2.1 cuenta con un sistema de seguridad para el usuario en función a una clave de usuario y una clave de acceso.

Una vez que el programador accesa a la aplicación le presentará una pantalla de inicio (llamada Main Menu), con la cual se le indica el tipo de ambiente de desarrollo con el que se desea trabajar, cada uno de estos son aplicaciones independientes que se relacionan entre sí, siempre y cuando existan integrados al sistema en general. EL ambiente que se utiliza para programación es el llamado "Facilidades de Desarrollo", con este ambiente se invoca el editor del lenguaje, con el cual se puede dar mantenimiento a los "objetos", que son los tipos de programas específicos con los que cuenta NATURAL para el desarrollo de sistemas. Aquí nosotros podemos: crear, editar, borrar, salvar, catalogar y ejecutar objetos, los cuales se encuentran clasificados como sigue:

M MAP (MAPA)

Nos permite declarar cualquier tipo de pantallas para captura o menús de selección.

**G GLOBAL DATA
 (AREA DE VARIABLES GLOBALES)**

Se declaran variables de tipo global que pueden ser requeridas en cualquier momento por cualquier programa dentro de la aplicación.

**L LOCAL DATA AREA
 (AREA DE DATOS LOCALES)**

Se declaran variables y vistas de archivos que puedan ser requeridos por un conjunto de programas para un módulo común.

**A PARAMETER DATA AREA
 (AREA DE PARAMETROS)**

Se declaran parámetros que son enviados por un programa o subprograma para la ejecución de una subrutina o subprograma.

P PROGRAM (PROGRAMA)

Es el código fuente de un programa principal de algún sistema.

N **SUBPROGRAM** (**SUBPROGRAMAS**)

Es una especie de programa con menor código, es decir, programas que no son muy grandes y que pueden aplicarse en diferentes programas.

S **SUBROUTINE** (**SUBRUTINAS**)

Se declaran subrutinas de acceso común para una gran variedad de programas que las requieran dentro del sistema.

H **HELPROUTINE**
(**RUTINAS EN MAPAS PARA AYUDA**)

Son una especie de mapas donde se declaran enunciados de ayuda a un campo o bien a una aplicación.

C **COPY CODE** (**CODIGO DE COPIA**)

Se declaran porciones de programa que pueden ser comunes en un conjunto de ellos.

2.1.2 INSTRUCCIONES BASICAS DE PROGRAMACION NATURAL

COMANDOS

DIRECTOS : Son como su nombre lo indica comandos de acción inmediata no pertenecientes al editor de NATURAL. Dentro de los principales tenemos: LIST, EDIT, HELP, READ, LOGON, FIN, ETC.

DE EDITOR : son los comandos de ayuda en edición de programas o mapas, entre los principales tenemos insert, copy, join, skip, split, etc.

INSTRUCCIONES: son de dos tipos las llamadas FUNCIONES DE BLOQUE y de LAZO.

Dentro de las primeras encontramos las instrucciones siguientes:

IF ELSE
DECIDE ON FIRTS VALUE
DECIDE FOR FIRST CONDITION
FOR
REPEAT

Y en las segundas tenemos las instrucciones:

READ
READ WORK FILE
FIND

2.2 CONCEPTOS BASICOS DEL AMBIENTE DE BASE DE DATOS ADABAS

2.2.1 DESCRIPCION DE LA BASE DE DATOS ADABAS

Para trabajar y explotar al máximo una base de datos hay que saber como se diseña y como se conforma, con el objetivo de realizar un buen rendimiento. El rendimiento de un sistema es medible mediante el tiempo y los requerimientos de recursos que se utilizan para su ejecución. Esto es importante por las siguientes razones:

- Los sistemas pueden competir por los recursos con otros sistemas que tienen más tiempo restringido.

- Algunas funciones del sistema pueden ser completadas con un tiempo específico de estructura.

EL rendimiento puede, sin embargo, ser el objetivo más importante, por lo que los programas ejecutores tienen que ser diseñados para ofrecer el máximo, además de :

- Flexibilidad.
- Independencia de datos.
- Accesibilidad a información
- Consideraciones de seguridad.
- Espacio en disco.
- Fácil de esquematizar en usuarios ocurrentes al sistema.

Una metodología sencilla para lograr el mejor rendimiento sobre la base de datos es:

- El diseño de la base de datos.

- Definir opciones de alternativas cuando sea cargada por primera vez una base de datos.

- Definir la lógica de funciones de aplicación (por ejemplo cuando se usan accesos directos o una combinación de accesos secuenciales y ordenamientos).

Hablar de la base de datos ADABAS es relacionarse con todo lo que existe a su entorno, pero principalmente con su buen funcionamiento logrando el mejor rendimiento, que en parte se logra por un buen diseño de los archivos que la conforman.

Es posible diseñar un archivo de base de datos ADABAS con sus campos y tipos de formatos durante el diseño conceptual. Su estructura puede soportar cualquier aplicación requerida, pero sin un buen diseño del archivo no puede ser fácil la manipulación iterativa y no puede lograr también el mejor rendimiento por las razones siguientes:

- El número de llamadas a ADABAS pueden incrementarse. Cada llamada a ADABAS requiere interpretación y validación en modo multi-usuario.

- El acceso por medio de un índice de direccionamiento para almacenamiento de bloques para cada uno de los archivos es lento.

2.2.2 DISEÑO DE UNA BASE DE DATOS EN AMBIENTE ADABAS.

Además de los campos comunes y corrientes existentes en cualquier base de datos tenemos los propios de ADABAS, éstos son : Campos de valor Múltiple y Grupos Periódicos.

Un grupo periódico es un conjunto de campos que se declara cuando es necesario realizar almacenamientos masivos dentro de un mismo registro, es decir, se declarará en forma de pila. La tabla siguiente muestra un ejemplo de su aplicación:

NUMERO ORDEN	DIA ORDEN	CLIENTE	DIA DE REQUERIMTO	CODIGO	CANTIDAD
12345	1-Dic	NISSAN	1-Enero	SB001	200
			3-Enero	SB002	100
			5-Enero	SB003	500

El grupo periódico estaría conformado por: Día de Requerimiento, Código y Cantidad. Por lo que si convertimos esta necesidad a archivo ADABAS quedaría de la siguiente manera.

			5-Enero	SB003	500
			3-Enero	SB002	100
12345	1-Dic	NISSAN	1-Enero	SB001	200

NO-ORD FEC-ORD CLIENTE FEC-REQ CODIGO CANTIDAD

En el ejemplo se muestra la información convertida a un registro formateado en ADABAS al

cual llamaremos "Ordenes". Cada registro de Ordenes contiene un grupo periódico que permite almacenar un determinado número de ocurrencias. Como se mencionó el grupo periódico comprende los campos FEC-REQ, CODIGO Y CANTIDAD. Pueden especificarse hasta 99 diferentes ocurrencias con la cantidad deseada.

Un campo múltiple esta conformado por una serie de ocurrencias sobre si mismo, es decir, cuando tenemos un campo para el cual existe una gama de valores. Este puede ser declarado como campo múltiple y permite el ingreso hasta de 199 ocurrencias, la estructura de datos con la que podemos hacerlo es semejante a la de un vector.

TIPOS DE FORMATO DE CAMPO :

- FORMATO NUMERICO
- FORMATO ENTERO
- FORMATO ALFANUMERICO
- FORMATO CHARACTER
- FORMATO FECHA
- FORMATO LOGICO
- FORMATO EMPACADO

DESCRIPTORES :

Los descriptores son usados para seleccionar registros de un archivo basándose en una búsqueda de uso específico o bien durante una lectura lógica secuencial controlada. El uso de descriptores esta relacionado a la estrategia de acceso para uso de un archivo, un espacio adicional en disco y procesos de "candado", particularmente en los procesos en que se necesitan hacer actualizaciones. El uso eficiente para la declaración de campos descriptores tiene que ver mucho con el rendimiento del sistema. Algunas reglas para tal objetivo, puede describirse como sigue:

El valor distributivo del descriptor puede ser usado para seleccionar un pequeño porcentaje del total de registros en un archivo.

Los descriptores no deben ser definidos cuando se realicen pocas búsquedas.

Si dos de tres descriptores son usados en una combinación frecuente, es conveniente usar un super-descriptor.

Si el criterio de selección para un descriptor siempre envuelve a un sólo rango de valores, se puede usar un super-descriptor.

Si el criterio de selección de un descriptor nunca involucra la selección de valores nulos y grandes números de valores vacíos que estén relacionados al descriptor, el cual puede ser descrito con supresión nula, lo que ahorra espacio en disco.

Si un campo es actualizado muy frecuentemente no debe ser definido como descriptor, ya que se pierde el concepto de "llave de acceso".

Los archivos con un alto grado de volatilidad no deben contener un gran número de descriptores.

SUPER-DESCRIPTORES:

Un super-descriptor es un descriptor que es creado de una combinación de hasta cinco campos que pueden ser descriptores o no descriptores. Los super-descriptores son más eficientes que las combinaciones ordinarias entre descriptores, esto es porque ADABAS tiene solamente acceso a una lista invertida. Así mismo pueden ser usados como un descriptor común.

Los valores de búsqueda con los que son utilizados los superdescriptores pueden ser declarados con el mismo formato del

superdescriptor, para que el acceso sea más directo.

ESPACIO EN DISCO USADO.

El uso eficiente del espacio en disco es un muy importante para la eficiencia de la base de datos, pues nos simplifica mucho los accesos y nos permite hacer uso eficiente de los recursos, así como aumenta la velocidad de respuesta en línea a los usuarios finales.

Algunas aplicaciones requieren un gran espacio de datos, pero es necesario identificar en que porcentaje se utilizan, para que se les asigne el espacio óptimo de almacenamiento a cada archivo.

El uso eficiente de espacio en disco puede ser hecho mientras consideramos otros objetos del sistema.

Una de las técnicas para la liberación de espacio es la siguiente:

COMPRESION

Esta nos libera espacio lógico reacomodando los datos; hay tres tipos de compresión: el almacenamiento fijo en los cuales el campo no es comprimido en su totalidad, la compresión ordinaria la cual causa en ADABAS el borrado de campos alfanuméricos vacíos y la supresión de ceros en campos numéricos y finalmente la supresión de valores nulos, que incluye la compresión ordinaria y la supresión de valores nulos de campos adyacentes.

3 ANALISIS DE REQUERIMIENTOS

3.1 DESCRIPCION GENERAL DEL SISTEMA

El sistema requerimiento de partes es un sistema compuesto de diferentes procesos que en su conjunto realizan la selección de números de parte que serán utilizadas en la manufactura de vehículos automotores. Estas partes son controladas por áreas funcionales o departamentos dentro de la empresa, tales como lo son los departamentos de Especificación de Partes, de Control de Producción, de Almacén, etcétera.

Como ya se mencionó uno el principal objetivo es el calcular la cantidad requerida por número parte para un periodo de producción. Para su mayor comprensión se ha dividido en módulos que van obteniendo y procesando la información necesaria para el cálculo del requerimiento de partes. Los módulos en los que se ha dividido el sistema son los siguientes:

BOM CONVERSION

Este módulo obtiene la especificación de uso de todas las partes por medio de una interfase con el sistema ANPICS, la selección de la información se hace en base a parámetros definidos por el área de especificaciones del departamento de control de producción. De este proceso se obtiene un reporte de errores de conversión y con el cual los analistas de partes pueden corregirlos dando el mantenimiento necesario al archivo BOM.

GROSS CHECK

Este proceso se utiliza para verificar la información obtenida durante el BOM CONVERSION y consiste en hacer el cálculo del GROSS (que es la cantidad requerida en base a un programa de producción, sin considerar ajustes por inventarios especiales).

El cálculo obtenido se compara contra el GROSS del mes anterior y se obtiene un reporte en el que

se comparan las partes requeridas del mes anterior contra las del mes actual, esto para identificar las partes con diferencia en cantidad de uso, las partes "adoptadas" (partes que se van a pedir por primera vez) y las partes "canceladas" (partes que se dejan de utilizar).

INTERFASE CON EL SISTEMA DE PROGRAMACION

Este módulo selecciona el programa de producción del mes firme. El programa es una relación de los vehículos a nivel color interior-exterior que se producen en los días hábiles del mes firme; este programa se utiliza para el cálculo del requerimiento a nivel número de parte.

ALTAS A LOS ARCHIVOS PART-FILE Y MCI

Este proceso da de alta las partes que son adoptadas (partes nuevas) tanto en el archivo PART-FILE (Archivo Maestro de Partes) como en el archivo MCI (Maestro de Control de Inventario).

CREACION Y CANCELACION DE RANES

Frecuentemente existen discrepancias entre los requerimientos hechos a Japón, por concepto de pedidos de partes canceladas o no adoptadas, o bien se pide una norma de empaque (SNP) distinta a la usada en Japón, estas diferencias hacen necesario cancelar ranes emitidos o bien crear nuevos ranes. Estas discrepancias son resueltas en este módulo.

EXPLOSION DE PARTES

Calcula el GROSS por número de parte, en base al programa de producción real de la planta y a la especificación de "uso" (partes por carro) para cada una de las partes del automóvil, esto es, para cada parte a usar se verifica si el automóvil en los que está especificada se encuentra dentro del programa de producción, si es así, se multiplica el volumen a producir por las partes que son

utilizadas para su producción y el resultado es acumulado para cada parte, de esta manera se obtiene un volumen total para cada número de parte.

CALCULO DE REQUERIMIENTO

En este módulo se realiza el cálculo del requerimiento por número de parte y se generan las órdenes de compra a Japón; este módulo se subdivide en :

PREPARACION DEL PART-FILE WORK

Se obtiene un archivo copia del maestro de partes en donde se hace un recorrido del volumen requerido en el mes anterior para mantener un acumulado histórico de requerimientos (se conservan cinco meses de este concepto) y se obtienen los meses firme, tentativo 1 y tentativo 2, que se calcularon en el proceso de explosión de partes. También se calcula el último inventario teórico y se acumulan las cantidades de ajuste para las partes que lo requieran.

CALCULO DE ORDENES.

A través de este proceso se obtiene la cantidad lotificada que se requiere por día para satisfacer la producción, considerando las cantidades de inventarios y ajustes especiales. La cantidad requerida calculada se distribuye en ranes (número de autorización de recibo de lote), que son lotes económicos desde el punto de vista de producción, embalaje y transporte, a estos ranes se les asigna una fecha de entrega en planta (conocida como DUE DATE).

COMPARATIVO DE REQUERIMIENTOS

Este proceso compara la cantidad requerida del primer mes tentativo del requerimiento anterior contra la cantidad requerida del mes firme actual a fin de analizar diferencias entre la proyección del mes anterior y el requerimiento actual (mes firme) y hacer ajustes en caso necesario antes de enviar el requerimiento a Japón.

TRANSMISION A JAPON

Esta es la culminación de los procesos, mediante el cual se envían los archivos de cantidades firmes y tentativas requeridas por parte llamado "Detalle 1" y el desglose de los ranes de dicho requerimiento llamado "Detalle 3". La transmisión se realiza vía V.A.N. (Ver apéndice B)

ACTUALIZACION DE ARCHIVOS MAESTROS

Este proceso desliza la información del requerimiento de partes enviado al archivo Maestro de Control de Ranes (RAN-ASN-RECIBO), así como actualiza la información sobre el inventario teórico, material requerido, cantidad y ranes. De esta forma el requerimiento se registra y se esta en condiciones de hacer el seguimiento de material en tránsito, al recibir y al controlar el inventario de partes.

El menú principal del sistema se muestra a continuación:

NISSAN MEXICANA S.A. DE C.V. SISTEMA DE REQUERIMIENTO DE PARTES			
A	MANTO.A PARAM. CONVER.	J	EXPLOSION DE PARTES
B	BOM CONVERSION	K	CALCULO DE ORDENES
C	DESLIZAMTO.PROG.PROD.	L	REPORTE DE DETALLES
D	REPORTE DE DESLIZAMIENTO	M	PREPARACION DE REP.COM
E	PREPARACION DE GROSS C.	N	REPORTE COMPARATIVO
F	PROCESO DE GROSS CHECK	O	TRAMSMISION A JAPON
G	REPORTE DE ALTAS AL P/F	P	RESPALDO DE ARCHIVOS
H	ALTAS AL PART FILE	Q	ESTATUS DE PROCESOS
I	MANTENIMIENTO A RANES	OPCION : _	

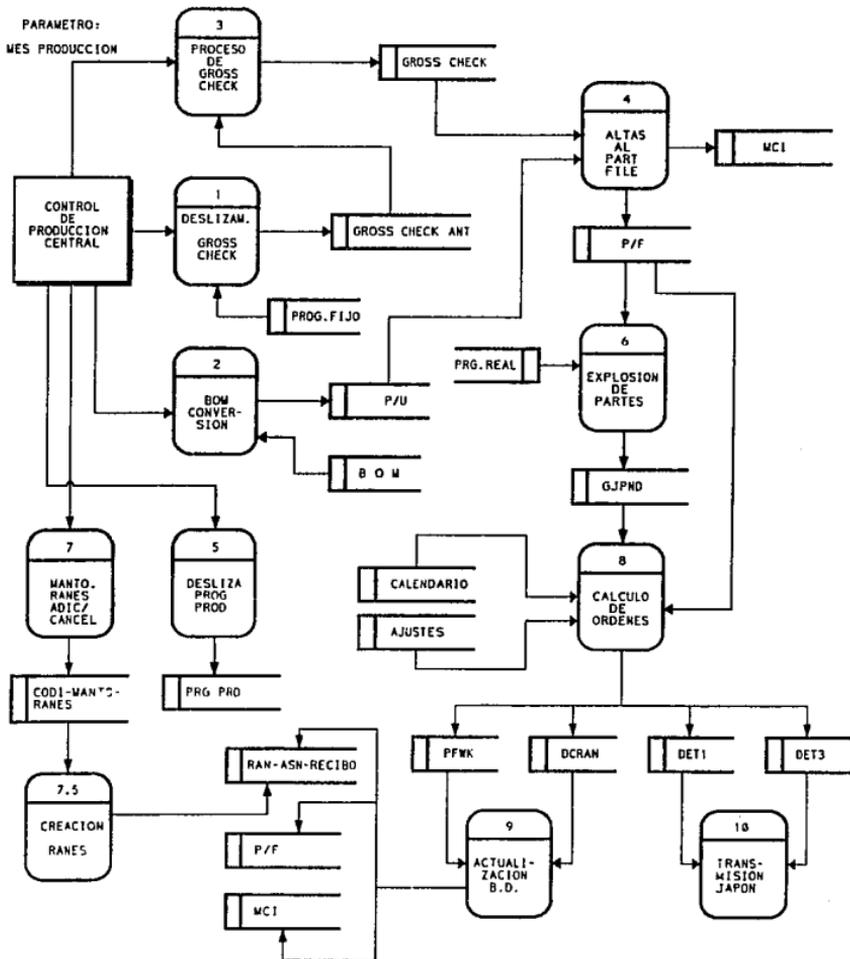
Cada una de estas opciones se describirán detalladamente. El presente menú es el que controla todo el sistema, también se presentan pantallas alternas en algunos procesos para introducción de parámetros de control o bien para mejor presentación como lo es el caso de mantenimiento a ranes cuya pantalla es la siguiente:

MANTENIMIENTO A RANES

- [A] MANTENIMIENTO A RANES ADICIONALES
- [B] MANTENIMIENTO A RANES CANCELADOS
- [C] REPORTE DE RANES ADICIONALES
- [D] REPORTE DE RANES CANCELADOS
- [E] CREACION DE RANES ADICIONALES
- [F] CANCELACION DE RANES

OPCIONES : _

3.2 DIAGRAMA DE FLUJO DE DATOS DEL SISTEMA



4 DISEÑO DEL SISTEMA

4.1 INTERFASES CON SISTEMAS EXTERNOS

4.1.1 INTERFASE CON EL SISTEMA DE PROGRAMACION.

El sistema de Programación es por medio del cual el departamento de Planeación y Programación de la producción controla los volúmenes y tipificaciones de cada uno de los modelos que se producen durante una determinada "cuota". Entendiendo por cuota el periodo de vida de un modelo (automóvil), por lo general una cuota comprende doce meses pero en algunas ocasiones es extendida a trece o catorce meses.

En este sistema se tiene toda la información de un automóvil a nivel unidad. Información como lo es el tipo de transmisión (manual o automática), su tipificación (austero, típico, lujo, o superlujo), tipo de carburador, el destino para su venta, si es de dos puertas o cuatro, los equipos opcionales. Información que se puede clasificar como administrativa o informativa.

Relacionada a la información anteriormente mencionada este sistema también cuenta con parámetros con los cuales las unidades pueden ser clasificadas y determinadas para requerir a los proveedores el material que en ella se utiliza, párametros como lo son el "USAGE" o "NUMERO DE APLICACION" el cual es un número de tres dígitos único para un determinado tipo de automóvil. Parámetros como el color de los interiores y el color exterior de la unidad y el volumen de unidades a producir en un determinado mes.

Estos parámetros son los mínimos necesarios para llevar a cabo el Requerimiento de Partes a Japón. Esto es explicado a un mayor detalle en el módulo de "Explosión de Partes" que será presentado en capítulos posteriores.

En este sistema primeramente es capturado el Programa Master (Programa Maestro de producción a nivel clave de tipificación -End-Item-). En este programa se tienen todos los tipos de automóviles que serán fabricados en la "cuota", a nivel mensual en lo que respecta a tipificaciones, destinos y colores.

La información que se captura en el sistema de Programación es ajustada previamente entre los departamentos de Ventas, Control de Producción Plantas, Control de Producción Central y Dirección General principalmente. Como Resultado de estos acuerdos o juntas se genera un programa anual de producción, llamado "Programa Master" (Programa Maestro) el cual debe ser cumplido al finalizar la "cuota" en condiciones regulares.

La información que se tiene en el Programa Maestro es a nivel "End-Item", es decir a nivel tipificación-destino.

4.1.2 INTERFASE CON EL SISTEMA ANPICS.

El sistema ANPICS (All Nissan Parts Internal Control System - Sistema Interno de Control de Partes Nissan) es el sistema mediante el cual los departamentos de Diseño y Especificaciones de partes en Planta mantienen actualizada la información en el maestro de especificaciones "BOM" de las partes que conforman a un automóvil.

El "BOM" es un archivo de la base de datos en la cual se tiene toda la información e historia de cada una de las partes que son utilizadas en la implementación de los automóviles que Nissan Mexicana fábrica. Este archivo es de suma importancia ya que en el se encuentra la toda la información necesaria para implementar cada uno de los automóviles que en esta compañía se fabrican.

Cada uno de los modelos son diseñados primeramente en Japón. Una vez liberada la especificación de dicho modelo se manda una copia del archivo a México con el objetivo de actualizar nuestro archivo maestro de especificaciones (BOM). En este archivo el modelo u auto se encuentra segmentado por módulos y estos módulos a su vez en un conjunto de partes individuales. La información que se tiene en este archivo maestro de especificaciones es un número de parte único con el cual es identificada cada una de las "n" partes de los "m" modelos que se fabrican en Nissan Mexicana, otros datos importantes son la descripción, la clave de origen de fabricación, las aplicaciones o carros en las que se utiliza dicha parte, la cantidad de cada parte que es utilizada en cada uno de los módulos, el color interior o exterior de la parte si lo tiene, la fecha en la que empezará a utilizarse o en su defecto la fecha en la que dejará de utilizarse, tomando como referencia la producción en México.

Por medio de este sistema se mantiene actualizada la especificación de cada una de las partes al mismo nivel que Japón.

La Interfase del sistema de Requerimiento de Partes a Japón con este sistema consiste principalmente en un acceso al archivo maestro de Especificaciones "BOM" y seleccionar todas y cada una de las partes que utilizaremos en los modelos a producir. Toda la información seleccionada es almacenada en un archivo secuencial de trabajo del sistema de Requerimiento de Partes el cual se llama "P/U" (Part Usages - Partes a ser Usadas).

Una vez obtenido el P/U el sistema de Requerimiento de Partes no vuelve a acceder al "BOM", ya que trabaja con el P/U. El proceso con el cual se genera el P/U es llamado "Bom Conversion" (conversión del BOM - Selección de las partes a ser utilizadas del archivo BOM), el cual es detallado en capítulos subsecuentes.

4.2 PROCESOS PRINCIPALES

4.2.1 PROCESO DE BOM CONVERSION

OBJETIVO :

Seleccionar los registros necesarios del archivo BOM para ejecutar el cálculo de ordenes. Deslizar la información para el control de partes, en función del código del proceso (lugar donde la parte es ensamblada), así como identificar las partes de cada modelo producido.

A través de este proceso se genera un archivo de Uso de Partes (P/U -Part Usage-), el cual contendrá información de especificación de uso para el control de partes en planta.

El proceso selecciona las partes en función de parámetros, en los cuales queda especificado el código de modelo, la planta y la fecha del mes de producción.

El proceso sera efectuado hasta que se garantice que el nivel de información del archivo P/U requerido para asegurar que el cálculo de ordenes sea exacto. Para esto se requerirá de un listado de errores de los parámetros de conversión donde se reporta el número de parte y el código de analista, como el bloque de la parte y el código de ensamble.

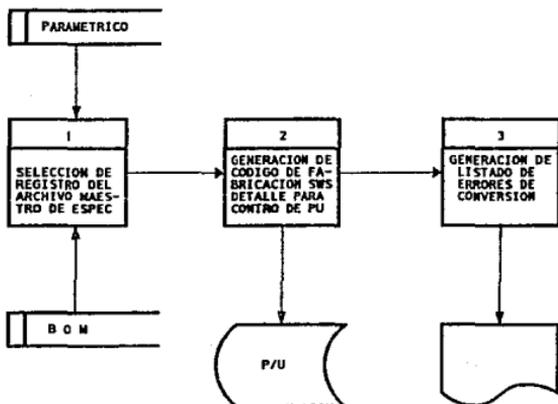
Condiciones:

Los parámetros y el BOM serán accesados en forma secuencial y ordenados por código de planta y código modelo.

El proceso debe ser realizado tantas veces sea necesario hasta minimizar los errores de conversión.

Los reportes se enviarán a las plantas interesadas para que se realice el mantenimiento al archivo BOM.

FUNCIONES PRINCIPALES:



CONDICIONES NECESARIAS:

A) Los parámetros de conversión deben estar actualizados ya que se realiza la selección de partes en base a ellos.

B) El mantenimiento al P/F se realiza posteriormente a este proceso

TIEMPO DE REALIZACION:

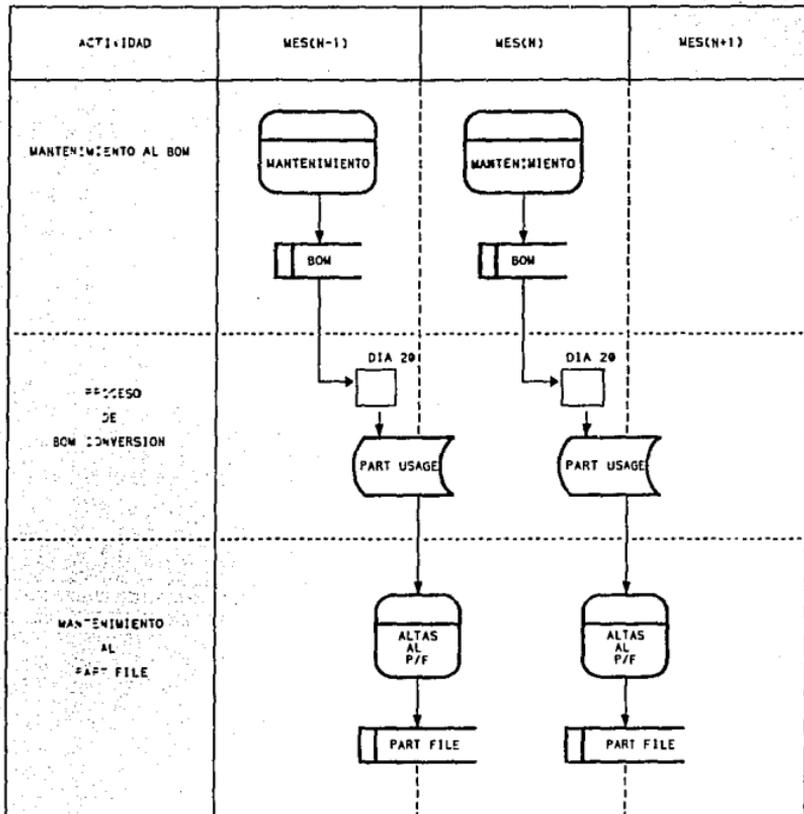
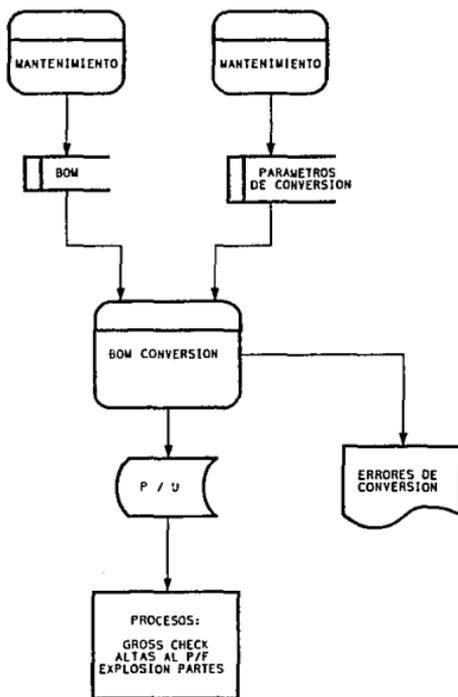


DIAGRAMA DE FLUJO:



ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

CONDICIONES DE PROCESO

1.- Selección de registros del BOM.

a) El código de planta-modelo a convertir deberá estar en los parámetros de conversión.

b) La lectura del BOM se realiza por planta-modelo únicamente a los registros que la misma planta y código de modelo.

c) El BOM CONVERSION es aplicado a registros que cumplen las siguientes condiciones:

- Que a la fecha del proceso no tengan fecha de abolición.

- Si tienen fecha de abolición esta deberá ser mayor que el mes de proceso.

- El campo STATUS (Arregement Status) deberá ser igual a espacios en blanco y el campo "cambio de diseño" (Design Change Symbol) diferente a "D".

d) Será aplicado a los registros con proceso de clasificación = 1, 2, 3, 6, 7, u 8.

e) Se aplicará a los registros en donde el campo MFG (código de manufactura) sea igual a blancos en su segunda ocurrencia.

f) No se aplicara a registros con el campo partes por automóvil (NO-OFF) = '****' y con el campo RAW-MATERIAL-USAGE = ' ' o a '000000'.

2.- GENERACION DEL CODIGO DE FABRICACION

a) El código de fabricación se genera de acuerdo al proceso de clasificación y a la fuente de clasificación que son campos del archivo BOM bajo las condiciones presentadas en la siguiente tabla.

Código Fabric.	Fuente de clasific.	Proceso de clas.	Código Proceso
A	INDIFERENTE	1, 6	
B	3	2, 3, 7	91
C	INDIFERENTE	2, 3, 7	
D	3	2, 3, 7	92
E	3	2, 3, 7	
F	INDIFERENTE	B	
G	3	B	92

3.- NUMERO DE PARTES POR CARRO.

Se establece como '0001' siempre en todos los casos.

4) Se genera el campo "Secuencia de trabajo" (subject Work Secuense), que indica la etapa de la línea en donde se ensambla dicha parte. Este campo contiene cinco posiciones y es de formato alfa-numérico en donde la:

Posición 1 = Planta Externa
 Posición 2 = Planta Interna
 Posición 3 y 4 = Código de proceso
 Posición 5 = 0

4.2.2 PROCESOS DE GROSS CHECK Y EXPLOSION DE PARTES

4.2.2.1 PROCESO DE GROSS CHECK

OBJETIVO:

Verificar que la especificación en el BOM esté actualizada con los últimos cambios que se originan por el departamento de diseño.

ESPECIFICACION DEL PROCESO :

1.- Se utiliza un programa de programación fijo con el fin de que las variaciones sean originadas por cambios en la especificación y no por el programa de producción.

2.- Cargar el programa de producción fijo a memoria.

Leer en forma secuencial el programa de producción fijo y cargar la información del programa en arreglos. la llave manejada para el programa de producción es el año modelo, el código de modelo, el color interior y color exterior y el número de usage.

3.- Procesar cada una de las partes del archivo Partes a Usar (P/U).

Leer secuencialmente el archivo de Partes a Usar (PU) y revisar que el código de origen de compra de la parte sea igual a 'C' o a 'F', que la fecha de adopción sea diferente a la fecha de abolición y que la segunda sea mayor o igual a la fecha del mes de requerimiento.

Si el registro cumple con la condición anterior, hacer un barrido al programa de producción fijo que se encuentra en memoria para identificar si algunos de los registros del programa de producción coinciden con las especificaciones de la parte. Para cada uno de los registros del programa de producción que coincidan, se acumula su programa diario en un arreglo de trabajo de tal manera que al concluir el barrido de programa de producción, se tenga en dicho arreglo la cantidad requerida de la parte por día.

Después de cálculo anterior se ajusta la cantidad requerida de la parte a nivel diario de acuerdo al número de piezas por carro que se utilizan por modelo, en otras palabras se multiplica el número de piezas por carro por la cantidad requerida por día.

El siguiente paso consiste en ajustar la cantidad requerida por día con base a la fecha de abolición y si esta se encuentra dentro del periodo de requerimiento, es necesario volver a cero las cantidades requeridas para los días con fecha igual o mayor a la fecha de abolición.

Otro ajuste similar al anterior debe hacerse en caso de que la parte tenga fecha de adopción, de tal manera que si la fecha de adopción se encuentra dentro del periodo de requerimiento, las cantidades requeridas para los días anteriores a la fecha de adopción deberán volverse cero.

Como último paso se revisa si el arreglo de trabajo en el que se guardan las cantidades requeridas de la parte es mayor a cero para almacenarlo en un archivo secuencial de salida llamado GROSS. Para poder realizar el proceso de comparación de archivos es necesario que los archivos anterior y actual de Gross Check se encuentren ordenados por llave del Part File (que se encuentra conformada por los campos: año modelo, número de parte, secuencia de trabajo, diferencia de proveedor y diferencia de parte).

Dada la condición anterior se procesan de la siguiente forma: leer secuencialmente el archivo de gross check anterior, leer una parte del archivo del mes actual, si sus llaves son iguales, esto indica que se trata de la misma parte y entonces se comparan las cantidades requeridas, si son iguales se imprime

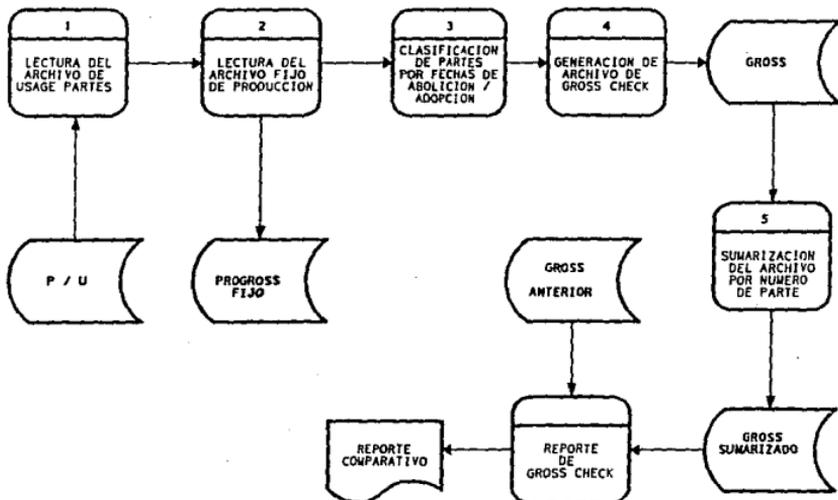
una línea en el reporte, si hay deferencia en las cantidades se imprime la línea incluyendo un mensaje de "DIF.CANT".

En caso de que la llave del Part File de la parte del mes anterior fuera menor que la llave de la parte del mes actual, se imprime una línea con los datos de la parte del mes anterior y con el mensaje "PTE. ABOLIDA".

Por otra parte en caso de que la llave de la parte del mes anterior sea mayor que la parte del mes actual, se imprime una línea en el reporte con los datos del mes actual con el mensaje "PTE. ADOPTADA".

Dado lo anterior este proceso además de permitirles comparar el Gross Check calculado del mes actual con el anterior y confirmar la corrección de la explosión, también indica que partes fueron adoptadas y abolidas, durante el periodo comprendido entre el requerimiento actual y el anterior.

FUNCIONES PRINCIPALES



CONDICIONES NECESARIAS

A) Es responsabilidad del usuario mantener el archivo PROGRESS FIJO actualizado con los "end-items" que son necesarios para la ejecución del proceso.

B) Se toma el último archivo de Partes a Usar (P/U) por lo que es necesario procesar la conversión del BOM antes de este proceso.

TIEMPO DE REALIZACION

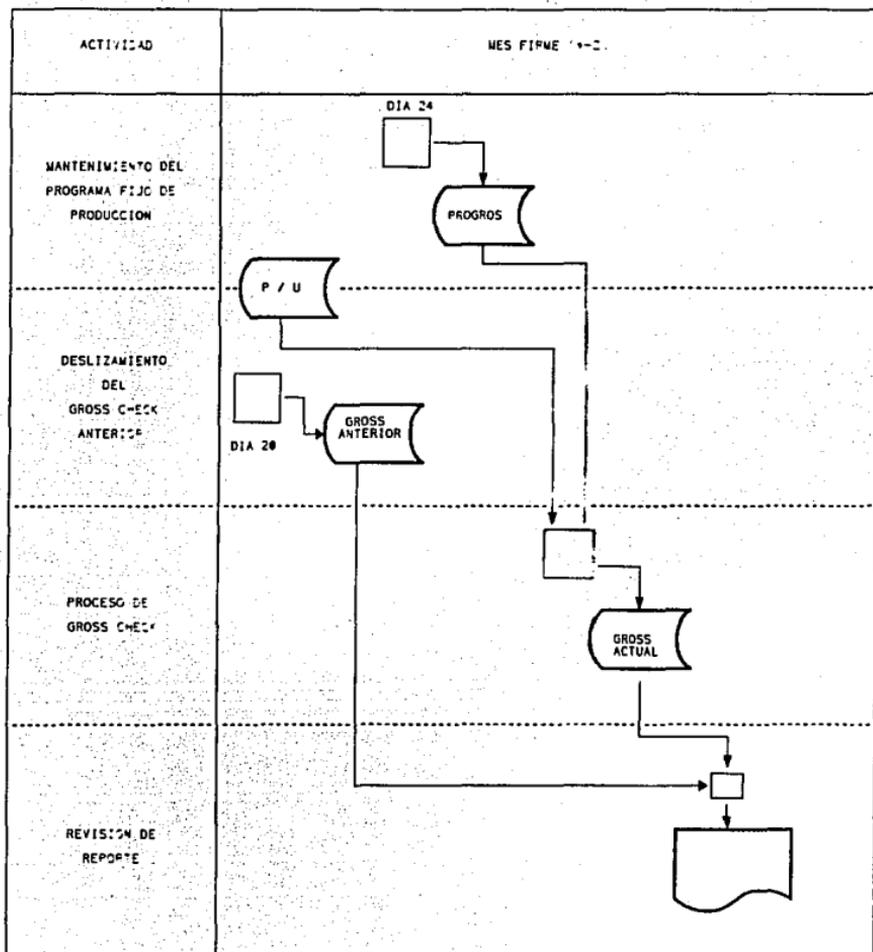
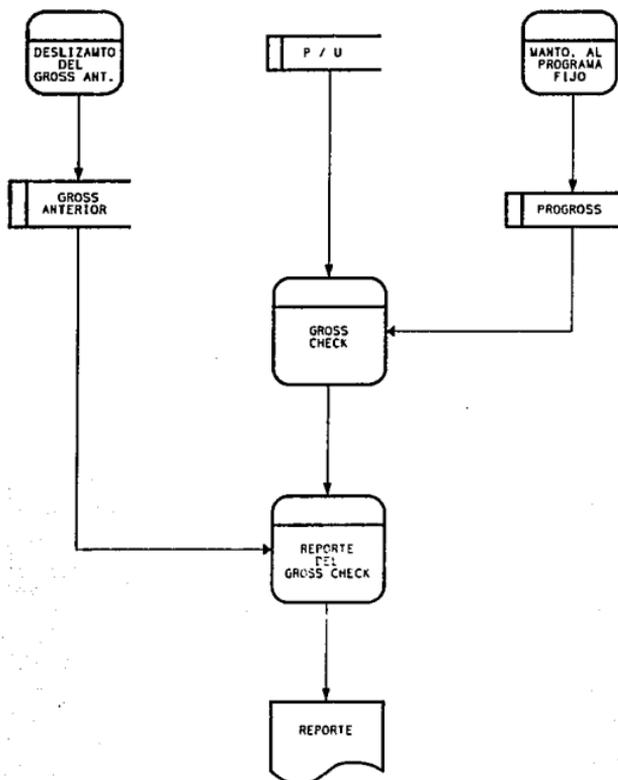


DIAGRAMA DE FLUJO DE DATOS



CONDICIONES DE PROCESO

1.- Selección de registros del archivo de Partes a Usar (P/U).

a) Deslizar tal y como se encuentre el último archivo de Gross (mes N-1 -mes firme pasado-) hacia un archivo diferente.

b) Cargar el programa de producción fijo a memoria, e identificar el mes firme (N) de requerimiento, así como calcular la fecha del último día del mes tentativo 2 (mes N + 2).

c) Procesar únicamente los registros con origen de compra = 'C' o a 'F' y con fecha de adopción mayor o igual a la fecha del mes firme y con fecha de abolición menor o igual al último mes tentativo 2.

2.- Procesamiento de registros:

a) Comparar cada elemento del programa de producción fijo con los usages del registro seleccionado de la siguiente manera.

Checar si la planta, código modelo y año modelo del programa de producción corresponden a los de la parte.

Si corresponden entonces :

Revisar si la parte leída se utiliza en el usage apuntado. Si esto cumple revisar si la parte tiene color interior y si existe entonces revisar si este checa con el programa de producción, si lo anterior se cumple continuar con el proceso de lo contrario checar el siguiente usage.

b) Por medio de las fechas de abolición y adopción ajustar las cantidades para los meses firme y tentativos, si la parte es pedida en alguno de los meses o en todos grabar en el archivo de GROSS la parte.

4.2.2.2 PROCESO DE EXPLOSION DE PARTES

OBJETIVO:

Realizar la explosión de un programa de producción real en base en la especificación de uso de partes, para calcular la cantidad de partes que serán utilizadas en la fabricación de unidades automotrices.

ESPECIFICACION DEL PROCESO :

1.- Cargar el programa de producción real en memoria.

Leer en forma secuencial el programa de producción real y cargar la información del programa en un arreglo. La llave manejada para el programa de producción es el año modelo, el código de modelo, el color interior, el color exterior y el número de usage.

2.- Procesar cada una de las partes del archivo de partes a usar (P/U).

Leer secuencialmente el archivo de partes a usar de partes (PU) y revisar que el código de origen de compra de la parte sea igual a 'C' o a 'F', que la fecha de adopción sea diferente a la fecha de abolición y que la primera fecha sea mayor o igual a la fecha del mes de requerimiento y la segunda fecha sea menor igual al mes de requerimiento mas dos meses, con esto delimitaremos el requerimiento para el mes firme y los dos meses tentativos.

Si el registro cumple con la condición anterior leer el programa de producción que se encuentra en memoria para identificar si los usages coinciden con las especificaciones de la parte. Para cada uno estos registros que coinciden, se acumulará su programa diario en un arreglo de trabajo de tal manera que al terminar el barrido del programa de

producción, se tenga en dicho arreglo la cantidad requerida de la parte por día.

Ajustar la cantidad requerida de la parte a nivel diario de acuerdo al número de piezas por carro que se utilicen por modelo, es decir, multiplicar el número de piezas por carro por la cantidad requerida por día, de esta manera obtendremos la cantidad por día necesaria para cumplir con el programa de producción real, que es el que se utiliza en planta.

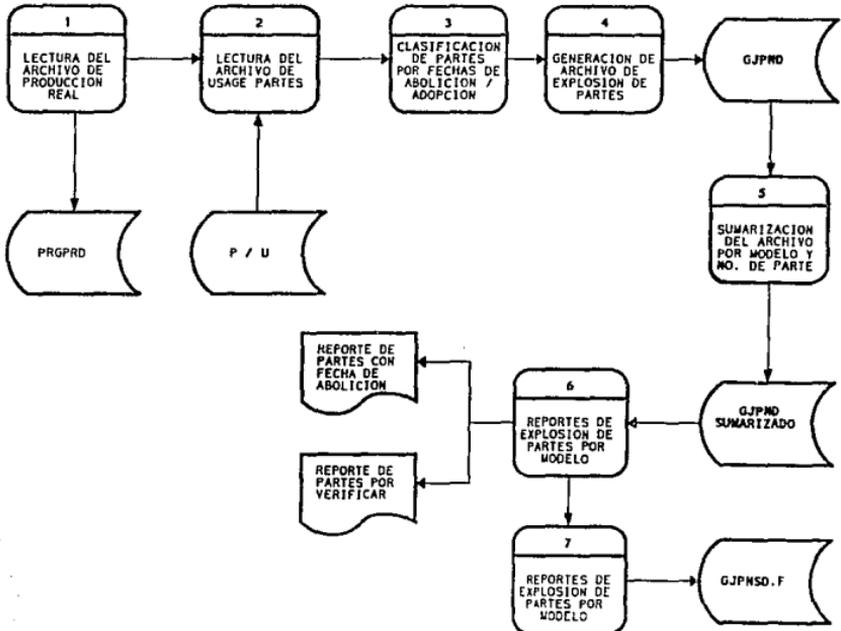
El siguiente paso consiste en ajustar también la cantidad requerida por día en base a la fecha de abolición. Si ésta se encuentra dentro del periodo de requerimiento (tres meses), es necesario volver a cero las cantidades requeridas para los días posteriores a la fecha de abolición, es decir, el arreglo de trabajo contendrá la información para todos los días del mes, pero como existe una fecha de abolición, hasta esa fecha tendrá que tomarse en cuenta la producción.

Otro ajuste similar al anterior debe hacerse en caso de que la parte tenga fecha de adopción, de tal manera que si la fecha de adopción se encuentra dentro del periodo de requerimiento, las cantidades requeridas para los días menores a la fecha de adopción deberán volverse cero.

Por último se revisa si el arreglo de trabajo en que se guardan las cantidades requeridas es mayor que cero, si es así esto indica que la parte es requerida ya sea en el mes firme o en alguno de sus meses tentativos, de lo contrario la parte no es tomada en cuenta para el requerimiento del periodo, almacenar el arreglo en el archivo secuencial de salida GJPND (Gross Japon diario).

Este archivo es tomado en cuenta para realizar el proceso de cálculo de ordenes. La información de este archivo son las partes con la cantidad por día necesarias para la producción del mes de producción en cuestión. Cabe recalcar que dicho mes de producción esta adelantado para cumplir con los periodos de flota establecidos por NISSAN MEXICANA Y POR NISSAN MOTORS LANDAU en Japón.

FUNCIONES PRINCIPALES:



CONDICIONES NECESARIAS:

A) El archivo de producción real debe ser obtenido previamente mediante el proceso de deslizamiento del programa de producción (Interface con el sistema de PROGRAMACION).

b) Se debe totar el archivo de partes a usar (P/U) más actualizado que se tenga, por lo que es necesario procesar la conversión del BOM.

TIEMPO DE REALIZACION:

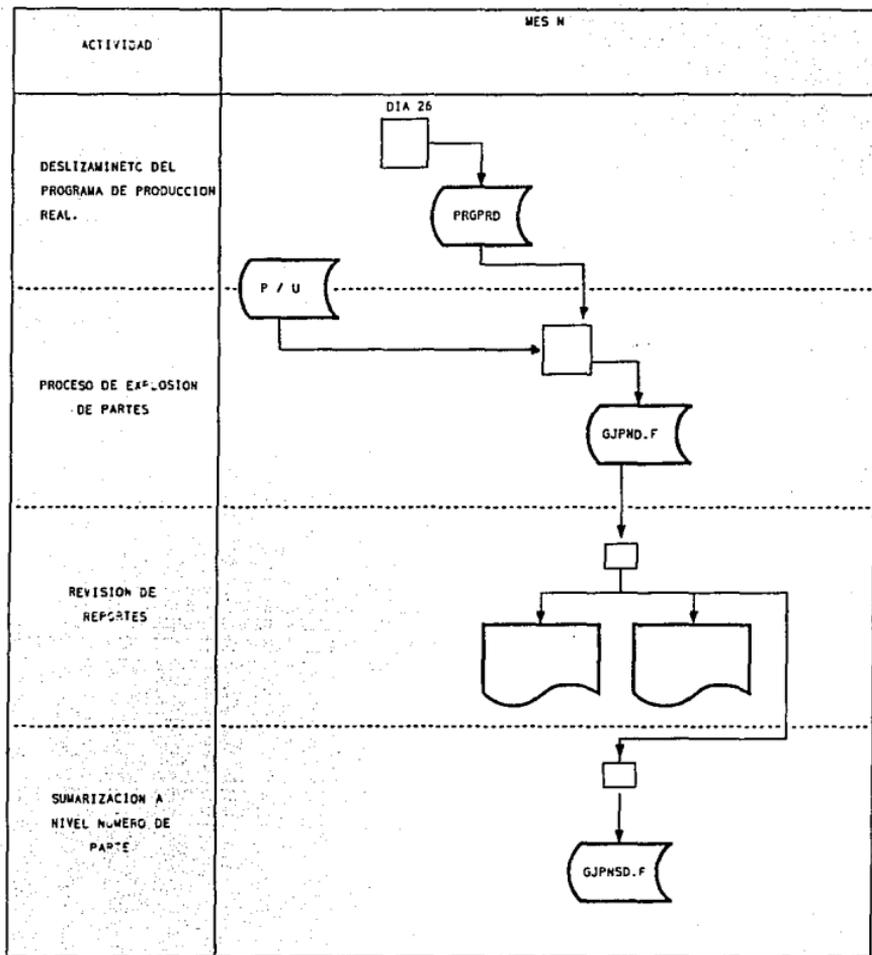
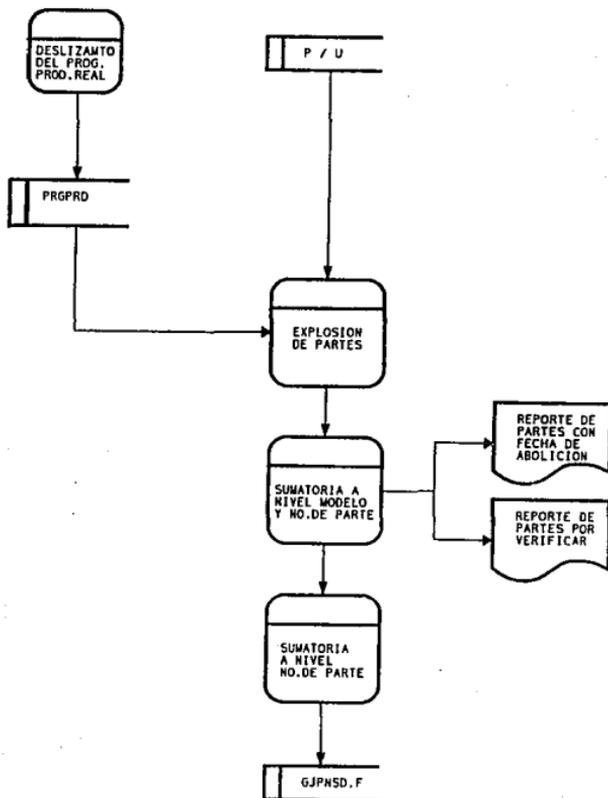


DIAGRAMA DE FLUJO DE DATOS



CONDICIONES DE PROCESO

1. Selección de los registros del archivo de Partes de Uso (P/U).

a) Deslizar tal y como se encuentre el último archivo de Gross hacia un archivo de trabajo llamado Gross Anterior.

b) Cargar el programa de producción fijo a memoria e identificar el mes firme, así como obtener la fecha del último mes tentativo (Tentativo 2).

c) Procesar únicamente los registros con origen de compra igual a "C" o "F" y con fecha de adopción mayor o igual a la fecha del mes firme (mes "N"), y con fecha de abolición menor o igual a la última fecha del mes tentativo 2.

2.- Procesamiento de Registros.

a) Comparar cada elemento o registro del programa de producción fijo con los usages del registro seleccionado de la manera siguiente:

 Checar si la planta, modelo y año modelo del programa de producción corresponde con los datos de la parte en cuestión.

 Si corresponde entonces revisar si la parte leída utiliza el usage apuntado. Si esto cumple, revisar si la parte tiene control por color interior y exterior, si es así continuar con el proceso, de lo contrario checar el siguiente registro del programa de producción.

b) Por medio de las fechas de adopción y abolición ajustar las cantidades para los meses en que son requeridas, que pueden ser firme y/o tentativos. Si las parte es pedida en algunos de estos meses guardar en el archivo de Gross.

4.2.3 PROCESO DE ALTAS A LOS ARCHIVOS MAESTRO DE PARTES (P/F) Y AL MAESTRO DE CONTROL DE INVENTARIOS (MCI)

OBJETIVO :

Realizar las altas de aquellas partes que son adoptadas en el proceso de GROSS CHECK, la especificación de cada número de parte esta dada por el archivo de usage de partes (P/U). Las partes se darán de alta en el archivo maestro de partes (PART FILE -PF-) y en el maestro de control de inventarios (MCI).

ESPECIFICACION DEL PROCESO:

Este proceso es dividido en dos partes, la razón de esto es que como el usuario de planta es el que lleva todo el registro y control de las partes, necesita conocer de antemano que partes serán dadas de alta dentro del archivo maestro de partes; por lo que existe un reporte previo a las altas al archivo maestro de partes. Los procesos son esencialmente los mismos únicamente con la diferencia de que en lugar de escribir sobre el archivo se imprime un reporte conocido como "Reporte de altas al Part File y MCI".

Descripción del proceso:

1.- Ordenar el archivo de GROSS CHECK actual (GJPNB) por medio de los campos que conforman la llave del archivo maestro de partes (P/F), estos son : el identificador de planta, el año modelo, el número de parte, la secuencia de trabajo, la diferencia de proveedor y la diferencia de parte.

2.- Leer el archivo GROSS ordenado y para cada registro acceder el archivo maestro de partes (P/F), por medio de la llave conformada.

2.1- Buscar el archivo de partes de uso (P/U) la descripción de la parte cuando se cumpla la condición de que el número de parte, el año modelo y la secuencia de trabajo del GROSS sea igual a la de P/U.

Una vez encontrada la descripción, se procede a dar de alta o imprimir según sea el caso los campos de interés al usuario.

3.- Durante el proceso se tiene que incrementar una variable en una unidad, que será utilizada como un contador de registros procesados, para conocer al final de este la cantidad de partes que se dieron de alta en el maestro de partes.

4.- Es necesario que el reporte de altas al maestro de partes sea ejecutado antes, de esta manera se asegura que las partes a adicionar dentro del archivo sean las correctas.

5.- Accesar el archivo maestro de control, de inventarios (MCI), por medio de los campos: año cuota, secuencia de trabajo, y número de parte. Si el registro no es encontrado dar de alta el número de parte dentro del archivo, de lo contrario continuar con el proceso. En este caso se también se tienen que contar los registros dados de alta, por lo que es necesario incrementar en una unidad otro contador.

6.- Al final del proceso se deben imprimir tanto en el reporte como en el proceso, las variables de control, que en este caso son los contadores e indicarán cuantas partes fueron dadas de alta en el maestro de partes y cuantas en el maestro de control de inventario.

FUNCIONES PRINCIPALES

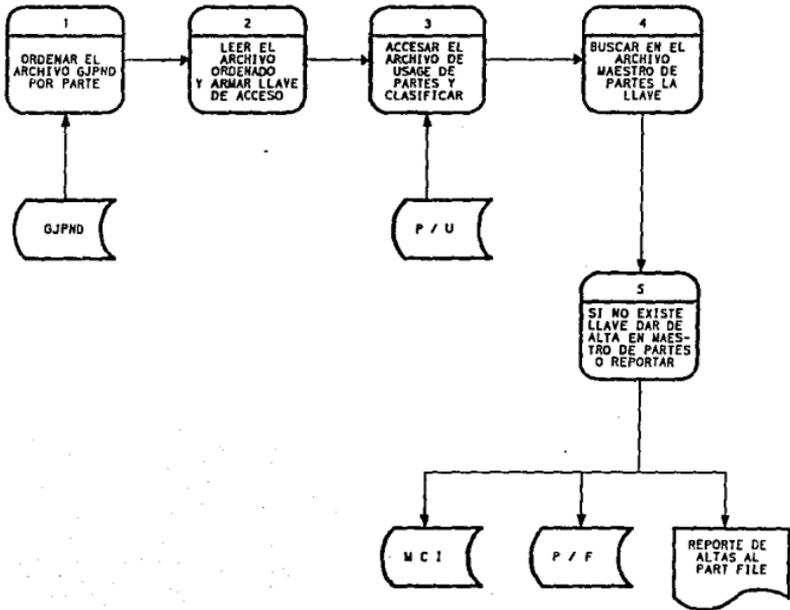
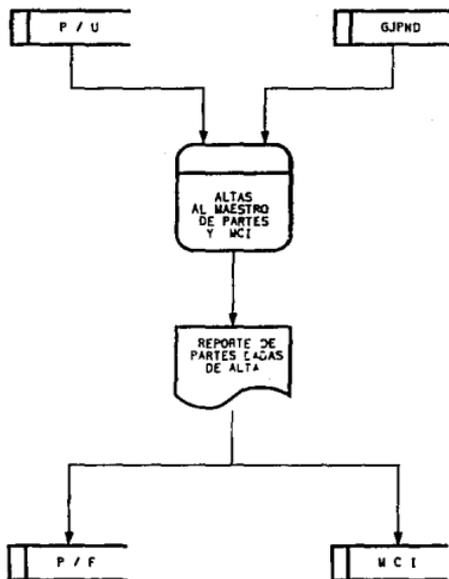


DIAGRAMA DE FLUJO DE DATOS



4.2.4 MODULO DE MANTENIMIENTO A RANES.

OBJETIVO.

Proporcionar un módulo en el sistemas de Requerimiento de Partes a Japón, que permita registrar ordenes adicionales y/o cancelar ordenes ya emitidas en el proceso de "Cálculo de Ordenes" original.

DESCRIPCION GENERAL.

Como resultado de la revisión de las ordenes (ranes) que Japón realiza, eventualmente se encuentran discrepancias en cuanto a la última especificación de la parte (última liberación o especificación) o tamaño del lote con el que fueron emitidas (SNP).

Estas discrepancias son informadas y ajustadas con Japón por medio de un documento donde se reportan aquellas partes con discrepancia ("Unmatch Order List" - Lista de Partes con Discrepancia). En este documento se detallan todas aquellas partes con sus números de ranes que tienen problemas, así como la causa de dicha discrepancia. Básicamente podemos tener tres tipo de discrepancias, los cuales son los siguientes:

a) Parte NO abolida por México (cancelación de número de parte no actualizada por especificaciones México).

Clave de error -> E1

b) Parte con cambio de ingeniería (cambio en número de parte no actualizado por Nissan Mexicana).

Clave de error -> E2

c) Cambio de SNP (cambio de lotificación).

Clave de error -> E3

Esta información "Relación de Partes" es analizada por los departamentos de Especificaciones y Material Importado de plantas y contestada a Control de Producción Japón con el fin de confirmar los ranes a cancelar y ajustar los nuevos números de ranes a adicionar en el pedido original.

ESTRUCTURA DEL MODULO DE MANTENIMIENTO A RANES.

El módulo de mantenimiento a ranes se divide en dos submódulos que son "Cancelación de Ranos" y "Adición de Ranos". Las dos funciones o submódulos se dividen a su vez en tres actividades, las cuales son:

- a) Mantenimiento de ranes a "cancelar" o "adicionar".
- b) Reporte de ranes a "cancelar" o adicionar".
- c) Aplicación de ranes a "cancelar" o "adicionar".

La primera actividad "Mantenimiento a ranes" permite registrar en un archivo de paso (archivo de trabajo) todos aquellos ranes que se cancelarán o en su defecto se adicionarán a la orden original. El objetivo de trabajar con archivo de paso es el de no estar adicionando o borrando registros del archivo maestro de ranes. Con esto el mantenimiento es más eficiente y eficaz. Eficiente por que sólo se hace un acceso al archivo maestro por esta necesidad y con ello ganamos consistencia y seguridad en nuestra información. Eficaz porque el acceso durante el mantenimiento, ya sea captura o actualización, es más fácil y más rápido en un archivo pequeño.

En la segunda actividad se genera un reporte con toda la relación de los ranes que hasta el momento se han capturado. Esto con el objetivo de facilitar y agilizar la revisión de los mismos.

En la tercera y última actividad confirmamos los ranes que se encuentran en nuestro archivo de trabajo, esto es borrar los ranes a cancelar o dar

de alta los ranes a adicionar en nuestro archivo maestro de ranes así como inicializar nuestro archivo de trabajo.

La opción de Mantenimiento a Ranes activa el menú principal del módulo que consta de 6 opciones:

1) REGISTRO DE RANES A CANCELAR

Esta opción permite altas, bajas, cambios y consultas de la información de los ranes a cancelar que se encuentran en el área de trabajo. Este mantenimiento se lleva a cabo a través del programa que permite realizar las actualizaciones a un número de RAN.

2) REPORTE DE RANES A CANCELAR

El proceso lee el archivo "Mantenimiento a Ranes" (donde son almacenados los números de ran temporalmente), solo acepta los registros con estatus igual a "P", es decir los registro de ranes a cancelar no aplicados (Pendientes de aplicar).

Imprime la información de cada registro aceptado en un reporte. El reporte se encuentra ordenado por número de parte y número de ran a cancelar, y hace cortes por número de parte.

3) CANCELACION DE RANES

Esta opción confirma los ranes del área de trabajo al archivo maestro de ranes (RAN-ASN-RECIBO) así mismo actualiza la cantidad acumulada requerida en el archivo Maestro de Control de Inventarios (MCI) y el Inventario Teórico en el archivo Maestro de Partes (P/F).

Proceso:

3.1) Lee todos los registros del archivo mantenimiento ranes que correspondan a ranes a

cancelar cuyo estatus sea igual a "P", es decir, pendientes de aplicación en el RAN-ASN-RECIBO.

3.2) cada uno de los registros aceptados, se procesan de la siguiente manera:

a) Localiza en el archivo de ranes el registro correspondiente usando como llave el número de parte y número de ran del registro aceptado del paramétrico. Al localizarlo, desliza los campos : número de ran, 'C', clave de discrepancia y número de ran sucesor hacia los campos : no-ran, status-ran, causa-cancelación y ran-sucesor del archivo RAN-ASN-RECIBO.

Actualiza el archivo maestro de ranes.

b) Mueve la llave del PART-FILE del registro localizado y del archivo RAN-ASN-RECIBO a variables de trabajo.

c) Con los datos de la llave del P/F, localiza la parte correspondiente en el archivo Maestro de Partes (P/F).

d) Debido a que la cancelación de un ran implica una reducción en el requerimiento del mes firme (N+2), se deberán hacer los ajustes correspondientes a la cantidad requerida. Esta reducción se hace descontando la cantidad del RAN cancelado.

e) Una vez descontada la cantidad requerida del ran cancelado, debe hacerse un ajuste al registro de Inventario Teórico (I/T) que se guarda también en el P/F; el algoritmo de cálculo del I/T es:

INV.TEORICO DIA J = INV.TEORICO DIA J-1
+ REQUERIMIENTO DIA J - GROSS DIA J

tanto el inventario teórico como el gross de la parte tienen un registro igual al del requerimiento (5 meses de histórico, 10 días del mes N+1, 31 días del mes firme (N+2), 10 días del mes tentativo 1 (N+3), el concentrado del mes tentativo 1 y el concentrado del mes tentativo 2 (N+4).

f) Una vez ajustado el inventario teórico de la parte se le mueve una "A" al campo de "status" del

registro del archivo Mantenimiento a Raner indicando con esto que ya fue aplicada la cancelación.

g) La actualización de todas las transacciones se hace al final para asegurar la integridad de las mismas.

3.3) Se lleva un conteo de los registros de raner pendientes de cancelación leídos, así como de los raner cancelados y al final se presenta esta información en una ventana al final del proceso.

4) REGISTRO DE RANER ADICIONALES

Esta opción permite altas bajas, cambios y consultas de los raner adicionales y se lleva a cabo a través del programa P32CN130.

5) REPORTE DE RANER ADICIONALES

El proceso es idéntico al de la generación del reporte de raner a cancelar, con la diferencia que la selección de raner a adicionar esta identificada con estatus "P".

6) ADICION DE RANER

Esta opción actualiza los raner adicionales que se encontraban en el archivo de trabajo al archivo maestro de raner (RAN-ASN-RECIBO), así mismo actualiza el acumulado requerido en el archivo maestro de control de inventarios (MCI) y el inventario teórico en el maestro de partes (P/F).

4.2.5 AJUSTES - PARTES DE SERVICIO.

Debido a que en tiempo real de producción las partes pueden ser dañadas, robadas o simplemente llegan a tener otros tipos de salida de planta como lo son las partes para Servicio (Refacciones) se diseñó un módulo en el sistema de Control de Inventarios donde se registran cada uno de los casos anteriormente mencionados.

En el sistema de Control de Inventarios se tienen las herramientas como lo son pantallas de consulta o mantenimiento, reportes y otras para tener un perfecto control del estatus de las partes en planta. Esto es, se tiene registrado por ejemplo el número de partes que son dañadas durante su traslado de puerto Japonés a Puerto Mexicano, si las partes han sido dañadas en almacén de planta o en la misma línea de ensamble, partes robadas o extraviadas, así como las partes que tienen salida como partes de servicio.

Debido a que el control de las partes que por alguna de las razones anteriormente mencionadas no se encuentran disponibles para ser ensambladas es necesario desarrollar una interfase entre dicho sistema y el sistema de Requerimiento de Partes, esto con el objetivo de satisfacer el nivel de inventarios para cada una de estas partes. Es así como fué desarrollada esta interfase, la cual consiste en un acceso al maestro de Ajustes, archivo donde se encuentran cada una de las partes con el volumen a requerir por este medio (ajuste).

Por medio del sistema de Control de Inventarios se dan de alta, baja, cambios y consultas a las partes que serán requeridas como ajustes. Las partes que son almacenadas en el archivo maestro de Ajustes son accedidas por el sistema de Requerimiento de Partes en el cálculo de ordenes y este volumen es adicionado al gross o cantidad a utilizar en el primer día de producción sin afectar al inventario teórico de la parte para efectos del cálculo del volumen a requerir en los subsecuentes días del mes de producción firme.

El volumen de ajustes es requerido el primer día ya que es material crítico (en el caso de

material dañado) o es material que es enviado para partes de servicio por lo que no debe afectar al volumen a requerir para satisfacer la producción tanto del primer día como la de los restantes días de producción.

Una vez hecha la transmisión del requerimiento a Japón toadas las partes en el maestro de Ajustes que tienen estatus "P" (pendiente de requerir) son actualizadas con estatus "2" el cual indica que dicho ajuste ya fué requerido (estatus de ajuste confirmado) y con esto evitar que sea requerido nuevamente en futuros requerimientos. Al actualizar dicho estatus después de la transmisión a Japón garantizamos que en el caso de que sean necesarios reprocesos sean siempre considerados los ajustes que desde un inicio ya han sido capturados.

4.2.6 PROCESO DE CALCULO DE ORDENES

OBJETIVO :

Realizar el cálculo de órdenes y la cantidad requerida diaria para cada parte para cumplir el programa de producción de la planta ensambladora de CIVAC en función al cálculo de explosión de partes.

ESPECIFICACION DEL PROCESO:

Leer secuencialmente el archivo de explosión de partes (GJPND), y armar la llave de acceso al maestro de partes por medio de los campos AÑO CUOTA, SECUENCIA DE TRABAJO, NUMERO DE PARTE, DIFERENCIA DE PARTE Y DIFERENCIA DE PROVEEDOR.

Si existe la parte buscada realizar un corrimiento en los campos de "gross", "inventario teórico" y "requerimiento", para efectuar el control histórico del comportamiento de dicha parte. Los corrimientos consisten en mover la información que se tiene a una ocurrencia anterior de donde se encuentran, por lo consiguiente el último mes N-3 se pierde. Mover los cálculos del mes firme y tentativos de la explosión en los correspondientes ocurrencias del campo llamado "gross". Con la misma llave acceder el archivo de ajustes y si existen ajustes para esta parte (se puede identificar si el campo de estatus de ajuste contiene un "1"). Mover la cantidad de ajuste hacia el campo PFWK-AJUSTE en la ocurrencia 6 y cambiar el estatus del ajuste por una letra "P".

Guardar toda la información en el archivo de partes de trabajo (PFWK - Part File Work), es decir moveremos la información encontrada hacia un archivo de trabajo (esto se hace con la finalidad de no interrumpir las transacciones que se realicen en él por el sistema de recibo de partes).

16 - 46 = mes firme
47 - 56 = 10 días del mes tentativo
57 = primer mes tentativo
58 = segundo mes tentativo.

Realizar el cálculo del requerimiento de la parte para las ocurrencias 6 a la 46 que es el mes firme según las fórmulas siguientes:

DE IND1 = 1 HASTA 5:
 $INV\text{-}TEORICO = INV\text{-}TEORICO (IND1\text{-}1) + CANT\text{-}REQ,$
 $(IND1) - CANT\text{-}GROSS$

DE IND2 = 6 HASTA 46:
 $CANT\text{-}REQ = CANT\text{-}GROSS (IND) - INV\text{-}TEOR (IND\text{-}1)$
 $+ AJUSTES$

después ajustar la cantidades a SNP que es el lote mínimo de embarque, es decir, por ejemplo si consideramos un SNP = 70 y se tiene una cantidad requerida de 50, la cantidad que se pedirá será 70 y el inventario teórico sera de 20, en cambio si la cantidad requerida es 100 la cantidad se ajustará a 140 y el inventario teórico sera de 40.

Hacer lo mismo para las ocurrencias 47 a la 58 que son las de los meses tentativos.

Generar los archivos detalle 1 y detalle 3, así como un archivo de ranes que es donde se reparten lo lotes por entregas de palets o contenedores dependiendo de la cantidad mínima y máxima.

FUNCIONES PRINCIPALES

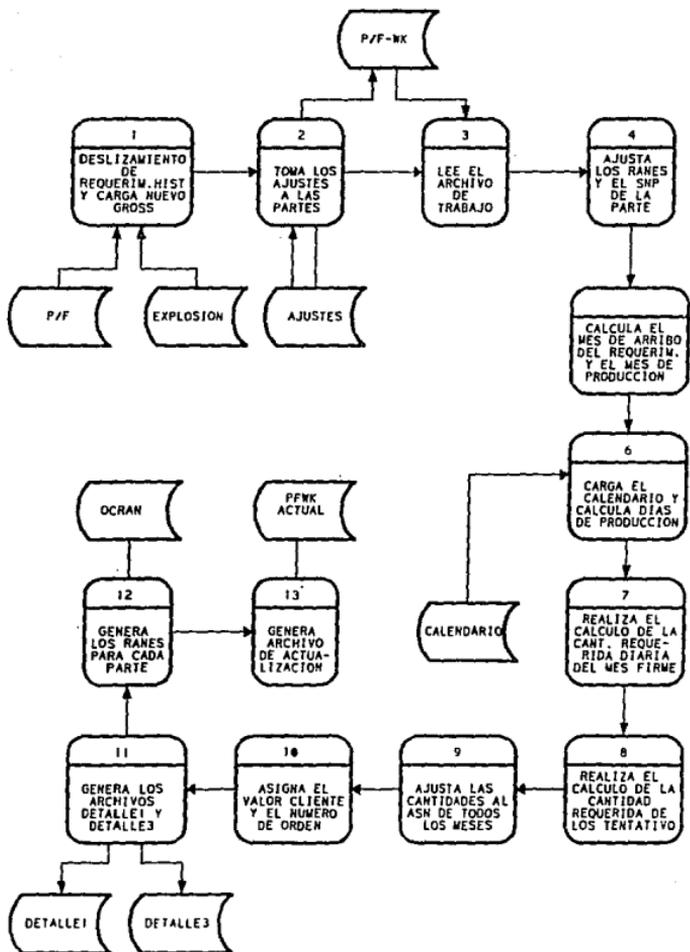
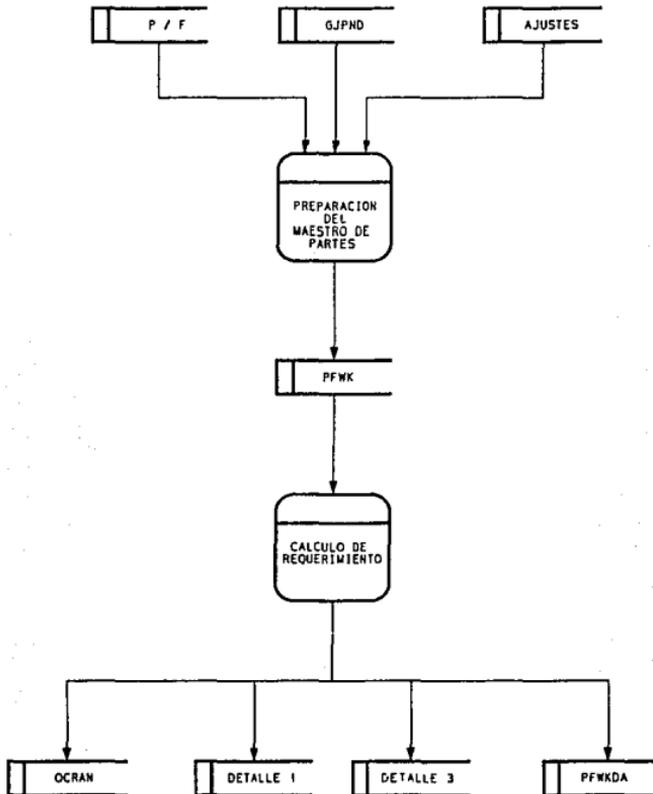


DIAGRAMA DE FLUJO DE DATOS



4.2.7 TRANSMISION DE ARCHIVOS

OBJETIVO :

Enviar o transmitir los archivos de DETALLES 1 y 3 a Japón que fueron generados durante el proceso de Cálculo de Ordenes utilizado la red de valor agregado VAN que es un paquete de comunicaciones con el que cuenta la empresa.

ESPECIFICACION DEL PROCESO:

Llamar por medio del menú el proceso VAN y direccionar los archivos DETALLE 1 y DETALLE 3 los cuales serán formateados por el paquete en un solo archivo llamado detalles con las siguientes características:

Longitud del archivo = 80 bytes

Bloqueo de archivo = 800 bytes

Tamaño del archivo = hasta 3 cilindros

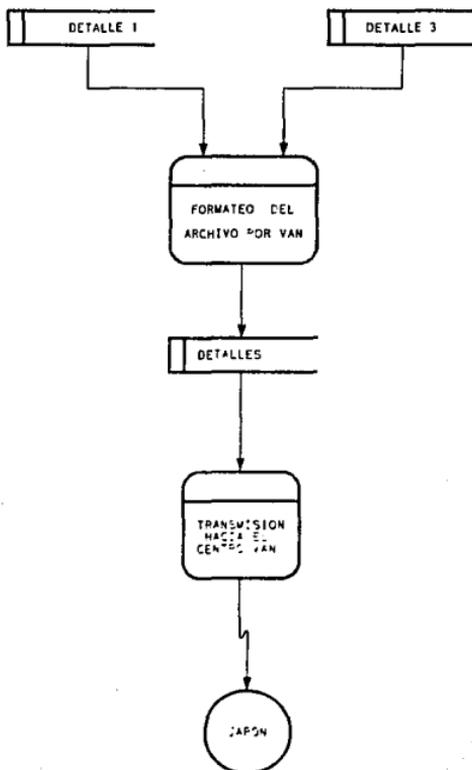
Este archivo contiene la información necesaria tal como: número de parte cantidad lotificada en ranes, así como los días que deberán ser enviados hacia México por parte de Japón, que día deberá ser recibidos en México, también el número de pedido y que planta de Japón es la encargada de proporcionar las partes automotrices (Actualmente proveen las plantas de Kioshu y Opama).

Previamente Japón indica a NISSAN MEXICANA mediante su área de sistemas cual es el destino y código transmisión o de "apartado postal" al cual deberá ser enviado. y estos deberán ser referenciados dentro del JCL por medio de una tarjeta de control

La utilería VAN se encarga de codificar todos los datos y envía por secciones de bloques la transmisión

vía satélite hacia Los Angeles en E.U. y después se transmiten hacia Japón vía cable submarino (Para mayor información vea el apéndice B).

DIAGRAMA DE FLUJO DEL PROCESO DE TRANSMISION DE ARCHIVOS



4.2.8 RESPALDO Y ACTUALIZACION DEL SISTEMA

OBJETIVO :

Actualizar los archivos maestros de partes y de recibo de ranes para tener la información actualizada de las partes que se han requerido así como también respaldar los archivos que se utilizaron durante el proceso del mes para que en caso de errores se vuelva a reprocesar.

ESPECIFICACION DEL PROCESO:

Leer el archivo PFWKDA secuencialmente y por medio de los campos: PFWKAD.AÑO-CUOTA, PFWKAD.NO-PARTE, PFWKAD.SWS, PFWKAD.DIFS-PTE, PFWKAD.DIFS-PROV, armar una llave de acceso por medio del superdescriptor del archivo maestro de partes (P/F) y cuando esta sea encontrada mover los campos del primer archivo hacia el segundo archivo :

mover PFWKAD.GROSS	a PF.B1-GSS-MES
mover PFWKAD.INV-TEORICO	a PF.B1-INV-TEORICO
mover PFWKAD.REQUERIDO	a PF.B1-REQ-MES
mover PFWKAD.AJST-MES	a PF.B1-AJST-MESN

para las ocurrencias 6 a la 58 para cada una de las partes leídas en el primer archivo y actualizar el segundo archivo.

Para la actualización del archivo de ranes únicamente se pasa la información generada y depositada en el archivo OCRAN y al archivo de ranes

(RAN-ASN-RECIBO). Esto implica que el proceso deberá ejecutarse una sola vez puesto que si se hiciera más de una se generarían ranes duplicados que como consecuencia acarrearía una incompatibilidad en el área de recibo cuando lleguen los contenedores que portan ranes de Japón hacia las plantas en México.

Una vez terminado el proceso de actualización se dispara automáticamente el proceso de respaldo del sistema que es programado utilizando una utilería de tareas en JCL, en este paquete se declaran todos los archivos necesarios para realizar un reproceso en caso de ser necesario.

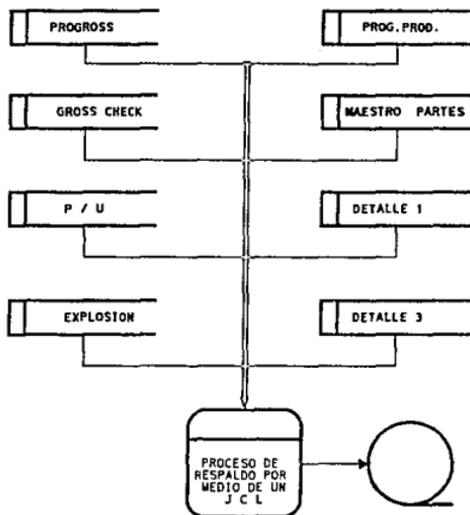
Actualizar en el archivo paramétrico los parámetros que corresponden a las banderas de control de ejecución de procesos, todas aquellas que tengan un status '1' deberán cambiarse a status '0' para dejar listo nuevamente el sistema para el siguiente mes de requerimiento.

La condición necesaria para ejecutar este proceso es que los archivos hayan sido transmitidos, actualizados y respaldados.

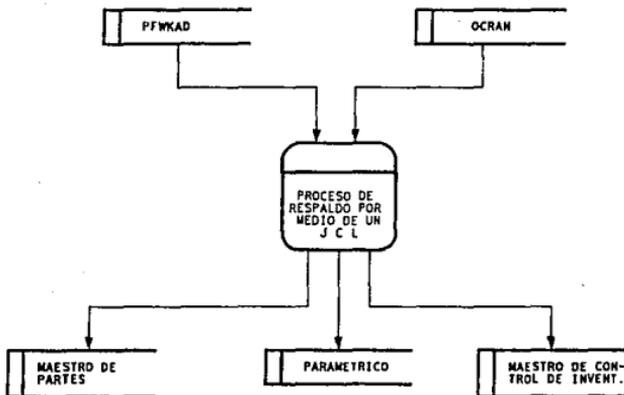
Este es el último proceso del sistema. Es importante recalcar que el sistema ha sido diseñado pensando en que pueden existir muchos usuarios que pueden utilizarlo al mismo tiempo y como de alguna manera un proceso no puede ir antes o después de otro, se han declarado parámetros de control interno para que los procesos no se puedan enviar más de una vez a menos que sea necesario, o que dos usuarios manden un proceso al mismo tiempo o que quisieran adelantar procesos que rompen la continuidad. El parámetro que controla todos los procesos es el 3250, este parámetro prende y apaga el status según comience o termine un proceso, mientras este en '1' no se podrá ejecutar ningún otro proceso y conforme los procesos vayan ejecutándose el estatus del proceso cambiará de '0' a '1'. Los parámetros y seriación para cada uno de estos procesos se muestran en la siguiente tabla:

PARAMETRO	NOMBRE DEL PROCESO
3210	BOM CONVERSION
3215	DESLIZAMIENTO DE PROGRAMA DE PRODUCCION
3220	REPORTE DE DESLIZAMIENTO
3225	PREPARACION DEL GROSS CHECK
3230	GROSS CHECK
3235	REPORTE DE ALTAS AL MAESTRO DE PARTES
3240	ALTAS AL MAESTRO DE PARTES
3250	CONTROL DE JOBS
3255	CREACION DE RANES ADICIONALES
3256	CREACION DE RANES CANCELADOS
3260	EXPLOSION DE PARTES
3265	CALCULO DE ORDENES
3270	PREPARACION DE COMPARATIVO
3275	REPORTE COMPARATIVO DE DETALLES
3280	TRANSMISION DE ARCHIVOS
3285	RESPLADO DE LA BASE DE DATOS

PROCESO DE RESPALDO DE LA BASE DE DATOS



PROCESO DE ACTUALIZACION DEL SISTEMA



5 CONCLUSIONES

Tanto el analista de datos como el usuario final necesitan de un amplio rango de herramientas que cuenten con características y facilidades tales como una sintáxis compatible, de clasificación, correo electrónico, manejo de oficina, calendarios, hojas de cálculo inteligentes, apoyo en decisiones, gráficas, y facilidades para la creación de aplicaciones.

En un futuro, más usuarios tendrán estaciones de trabajo en lugar de terminales ciegas. Las estaciones de trabajo darán respuestas en décimas de segundo a operaciones locales no complejas, ya sea por una computadora personal o porque estén conectadas a una computadora local. También darán tiempos de respuestas en milisegundos a operaciones que involucren computadoras remotas.

Por otro lado, el desarrollo de prototipos es algo vital en el proceso de desarrollo de los sistemas, y sobre todo ahorran recursos en mantenimiento. Actualmente todavía no se generaliza el uso de prototipos, debido a la renuencia por parte de los encargados del procesamiento de datos dentro de las organizaciones a cambiar sus métodos, ya que carecen de educación acerca de las herramientas ahora disponibles.

En nuestro punto de vista, no debería de haber un sistema de procesamiento de datos que no se desarrolle en base a un prototipo, ó construido con un lenguaje de cuarta generación ya sea parcial ó totalmente, debido a los grandes costos que involucra el desarrollo de los sistemas en base a los lenguajes de tercera generación.

En lo que se refiere a nuestro sistema desarrollado en NATURAL, se puede observar con el todas las ventajas que ofrece un lenguaje de cuarta generación, ya que con la implementación del sistema se obtuvieron las siguientes ventajas:

Se tiene un mayor control de inventarios en planta, teniendo ahora una existencia óptima de material.

Una reducción considerable en los costos del Control de Inventarios.

El "Lead-Time" o tiempo que transcurre desde el requerimiento hasta la llegada del material a planta disminuyó de 4 a 3 meses

Se presenta una mayor flexibilidad en los requerimientos, la cual permite ahora soportar los cambios de producción durante la cuota o tiempo de vida de los modelos.

Hay una disminución en el volumen de partes dañadas y una disminución considerable de material obsoleto.

Se redujeron las áreas para almacenes en planta.

Optimización en áreas de recibo y clasificación de partes en almacenes de planta.

Hay un mejor conocimiento del material físico, esto es que se tiene bien identificado el material en lo que respecta a cantidad y ubicación en almacén.

Finalmente se optimizó el tiempo necesario para realizar el requerimiento de partes a Japón, pudiendo ahora cambiar de un requerimiento mensual a un requerimiento quincenal e incluso a un requerimiento semanal.

A P E N D I C E A

DICCIONARIO DE TERMINOS DEL SISTEMA

Diccionario de términos Técnicos del Sistema
Requerimiento de Partes

AÑO CUOTA :

Es el año de modelo de producción al cual pertenece la parte.

SECUENCIA DE TRABAJO (SUBJECT WORK SEQUENCE (SWS)) :

Es la Secuencia de Trabajo del Subensamble e indica código de planta general, planta interna y línea de ensamble donde se emplea la parte. Se compone de cinco campos donde la primera posición indica la el código de la planta externa, el segundo el código e la planta interna, la tercera y cuarta posición indica el código de manufactura o el lugar de ensamble de la parte, la última posición indica el lado de la línea (izquierda o derecha) de ensamble.

NO-PARTE (Número de la parte) :

Indica el número de la parte o equipo original establecido por diseño (sistema ANPICS). Esta compuesta de quince posiciones donde a las cinco primeras se le conoce como prefijo que indica la función básica de la parte, los siguientes cinco campos son el subfijo que indica una especificación detallada de la parte y los restantes son para uso futuro.

DIFERENCIACION DEL PROVEEDOR (DIFS-PROV):

Es el código nemotécnico que diferencia a la parte en caso de presentarse proveedores múltiples.

DIFERENCIACION DE LA PARTE (DIFS-PTE) :

Es la diferencia de partes CKD y partes requeridas por el sistema de control por partes.

CODIGO DE UNIDAD :

Es el código de proceso de ensamble general.

ORIGEN DE FABRICACION :

Indica forma de fabricación de la parte (Interna o Compra externa), origen de la parte (Nacional o importado de Japón u otros países), y tipo de la parte (estandard o nes (tornillería)).

COLOR INTERIOR :

Es el código de color de los interiores.

COLOR EXTERIOR :

Es el código de color de las partes externas del automóvil tales como salpicaderas, defensas, etcétera.

UNIDAD DE MEDIDA :

Unidad de medida de la parte o material procesivo. Por ejemplo LT = litros, KG = kilogramos.

MODELO BASICO :

Es el código de modelo básico donde es ensamblada la parte.

FECHA DE ADOPCION :

Es la fecha en que se adopta la parte (AAMMDD).

FECHA DE ABOLICION :

Es la fecha en que se abole la parte (AAMMDD).

PUNTO DE EFECTIVIDAD :

Cantidad de partes antes de efectuar la cancelación de la parte, esta información es utilizada para efectuar el punto de quiebre entre la parte actual y la nueva que la sustituye.

GRUPO SEGUIDOR :

Indica el grupo al que pertenece el seguidor y el código del seguidor de la parte. Los seguidores pueden ser agrupados por el tipo de proveedor, por ejemplo plásticos, metálicos, etc. o bien por el proceso de ensamble en la planta.

CLASIFICACION DE LA PARTE :

Indica la clasificación en costo al cual pertenece la parte.

FRECUENCIA DE EMBARQUE :

Indica la frecuencia con la cual se debe embarcar la parte.

FACTORES DE FLOTA :

a) Días de tránsito: Número de días requeridos para transportar el material de salida desde proveedor hasta la llegada a planta.

b) Días ciclo : Número de días entre embarques.

c) Días Proceso : Número de días requeridos entre la llegada de la parte y el punto de ensamble en la línea.

d) Reserva Operacional : Se consideran factores de daño e inestabilidad del proveedor.

CODIGO DEL PROVEEDOR :

Indica la clave del proveedor, en caso de partes abastecidas por NISMEX se le pone el código de proveedor de la planta. La primera posición nos indica si es parte de equipo original nacional o si es material de equipo y herramienta nacional o de importación E.U.

PORCENTAJE DE COMPRA :

Indica el porcentaje de compra en caso de presentarse proveedores múltiples.

NUMERO DE PEDIDO :

Indica el número de pedido más reciente de la parte.

RAN MINIMO :

Indica la cantidad mínima de piezas por cada embarque, esta información se emplea para la generación del requerimiento de partes. La definición de este concepto esta en función del costo de transportación de las partes del proveedor NISSAN.

RAN MAXIMO :

Indica la cantidad máxima de piezas por cada embarque o RAN, esta información se emplea en la generación del requerimiento de partes. La definición de este concepto esta en función de la capacidad de transportación de las partes de proveedor NISSAN.

USO DE PROMEDIO DIARIO :

Indica la cantidad utilizada promedio por día del mes de producción y cinco meses por adelantado.

S.N.P. EXTERNO :

Es la cantidad estandard del empaque exterior (Pallet, rack, caja o bolsa)

S.N.P. INTERNO :

Es la cantidad estandard del empaque interno (caja de cartón o bolsa)

RAN

Número autorizado de recibo, un número de recibo es asignado para cada lote (SNP) que se requiere.

MES FIRME :

Es el mes de proceso para el cual se requerirán las partes, para el caso de Japón se piden con una anticipación de tres meses.

MESES TENTATIVOS :

Son los dos meses que le siguen el mes firme, estos son una proyección del comportamiento de los meses posteriores al requerimiento, y se calculan con el fin de que el proveedor pueda observar el comportamiento de los pedidos de partes.

A P E N D I C E B

TRANSMISION VIA VAN

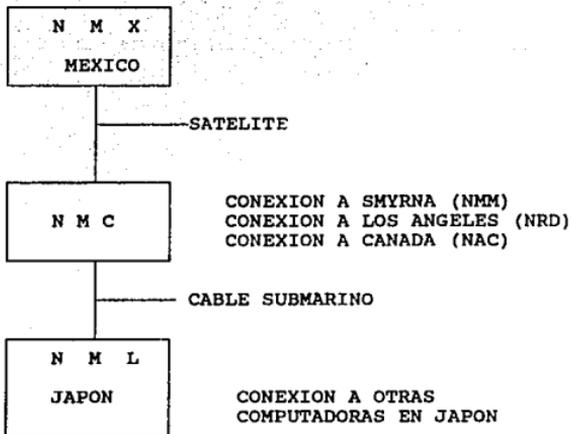
(Red Local de Valor Agregado)

El sistema V A N (Value Added Network) es un conjunto de programas diseñados para establecer una interfase entre computadores con el objeto de facilitar la transmisión de archivos y su control entre las subsidiarias de NISSAN y/o proveedores diseminados en diferentes zonas geográficas del mundo.

El sistema VAN se encuentra instalado en la computadora HITACHI de Japón, la cual esta conectada a otros computadores IBM en Japón.

Para el caso de Norteamérica, existe una conexión entre la maquina VAN de Japón y el computador IBM instalado en Los Angeles, Estados Unidos (NMC). NMC Los Angeles tiene definida a la máquina VAN como una terminal remota manejada por el JES2.

La comunicación de México a Japón, utilizando el sistema VAN se puede describir con el siguiente esquema:



SISTEMA VAN

Este sistema provee las siguientes funciones:

A) Actua como interfase entre computadoras de subsidiarias NISSAN, proveedores y Vendedores de NISSAN.

B) Correo electrónico dentro del territorio de Japón.

C) Facilidades de "apartado postal" para transferencia de archivos.

D) Servicios de conversión de protocolos de comunicación entre máquinas y software de diferente marca.

E) Consulta interactiva del estado del sistema VAN y archivos.

DIAGRAMA DE FLUJO :

La función principal del sistema es la transferencia de archivos de cualquier longitud de registro desde un computador hacia otro donde este instalado el sistema VAN.

DESCRIPCION DE FUNCIONES

Los procesos principales del sistema VAN son :

PROCESO DE TRANSMISION HACIA EL CENTRO VAN

En este proceso se determina el nombre del archivo a transmitir y se codifica el registro de control por medio de una constante (@@HDR), el identificador de estación (STATION-ID) de la localidad destino, la clave del archivo registrado en VAN, la descripción de la información, el formato del registro su longitud y el tamaño del bloque del registro.

Compactación del archivo a transmitir con una longitud de 80 bytes y antepone 2 registros de control. El archivo comprimido y sus registros de control son leídos por un programa intermedio el cual genera como salida un archivo en SPOOL de JES2 con destino a el centro VAN.

Transmisión del archivo en SPOOL vía satélite a Los Angeles, Estados Unidos y de ahí es retransmitido a la computadora del centro VAN en Japón por medio de cable submarino.

Los datos enviados son recibidos y almacenados en la computadora del centro VAN, donde permanecen en un "apartado postal" o spool hasta que son

recuperados o requeridos desde el computador en localidad de destino final.

PROCESO DE RECEPCION DE ARCHIVOS DESDE EL CENTRO VAN

Por medio de este proceso se puede realizar lo siguiente:

- * Consulta de status del sistema NNET (Sistema de Red)
- * Requerimiento de recepción
- * Transmisión y almacenamiento
- * Descompresión y recuperación
- * Borrado del miembro temporal
- * Liberación de espacio

SISTEMA DE CONSULTA

El sistema de consulta "NNET" de VAN es una aplicación en línea (Iterativa) la cual brinda facilidades a usuarios del sistema VAN.

La consulta se hace a través de terminales predefinidas para ello y solo a través de ellas se puede acceder el sistema NNET para consultar el status de los archivos transmitidos o recibidos por medio del sistema VAN.

IDENTIFICACION DE ARCHIVOS

El sistema VAN solo se encuentra instalado en el centro de cómputo del edificio corporativo, desde donde se retransmiten hacia las diferentes plantas de México, aquellos archivos que son recibidos desde el centro de operaciones VAN de Japón. Así mismo, en las oficinas corporativas de México se concentran los archivos de las plantas del país que requieran ser transmitidos a Japón.

Para poder transmitir o recibir archivos a través del sistema Van, es necesario que dichos archivos estén registrados con un identificador de ocho caracteres en el centro VAN.

Esta identificación es almacenada en el sistema VAN y deberá ser usado en todas las referencias que se haga al archivo que se este procesando.

OPERACION Y AUTOMATIZACION

DESCRIPCION DE REGISTROS DE CONTROL

Los siguientes cuadros muestran la descripción de algunos registros de control, los cuales son usados en diferentes funciones o procesos, por lo que es importante conocer los campos que deberán ser codificados.

@@DEL - Control Record Layout

COLS	BYTES	TYPE	NAME	REMARKS
01-02	2	AN	CONTROL SYMBOLS	@@CONSTE.
03-05	3	AN	CONTROL CODE	DEL CONSTE
06-13	8	AN	DATA ID	FILEID
14-80	67	AN	FILLER	ESPACIOS

@@HDR - Control Record Layout

COLS	BYTES	TYPE	NAME	REMARKS
01-02	2	AN	CONTROL SYMBOLS	@@CONSTE.
03-05	3	AN	CONTROL CODE	HDRCONSTE.
06-06	1	AN	PROCESS CODE	'T' TX AUTOMAT
				'P' "
				'N' "
				'M' MAIL BOX
07-12	6	AN	DEST TERMINAL ID	'ZZGAOC'
13/20	8	AN	DATA ID	EJ. NMX34A781
21-24	4	N	TRANSMISSION DATE	MMDD
25-28	4	N	TRANSMISSION TIME	HHMM
29/34	12	AN	SENDER TERMINAL ID	
35-46	12	AN	DATA NAME	
47-47	1	AN	FORMATO DE REG	
48-51	4	N	LONGITUD DE REG	
52-56	5	N	BLOQUEAJE DE REG	
57-61	5	N	CONTADOR DE REG	
62-63	2	N	DAYS KEEP	DEFAUL TRES
				DIAS
64-80	17	AN	FILLER	ESPACION EN
				BLANCO

A P E N D I C E C

LENGUAJE CONTROLADOR DE TRABAJOS (JCL)

Existen programas que controlan y coordinan la administración de los recursos del computador, este programa de control hace posible procesar los trabajos mas rápidamente y de manera mas eficiente, reduciendo la posibilidad de error humano.

Para el sistema operativo un "JOB" (trabajo) es la unidad de trabajo que un usuario desea procesar. La unidad de trabajo asociada al proceso de un solo programa se denomina "JOB STEP" (paso).

Para definir que es un JCL (Jobs Control Language) es necesario mencionar las propiedades de un sistema operativo con memoria virtual. Los sistemas virtuales simulan que la memoria de la CPU es mayor que la memoria real. A la memoria simulada se le llama memoria virtual. El tamaño de la memoria virtual esta limitado por el numero de bits de direccionamiento de las instrucciones de máquina (24 bits para un sistema MVS y 32 bits para un sistema MVS/XA). Por lo tanto un direccionamiento de 24 bits permite una memoria virtual de un tamaño de 16 megabytes.

A un programa que corre en un sistema se le asigna espacio en la memoria virtual, como si fuera memoria real. Las instrucciones del programa tendrán direcciones virtuales, de tal manera que cuando la memoria real, de la máquina es de 1 mega y en el programa hay una instrucción que tiene la dirección 14,512,003 antes de que se pueda ejecutar esta instrucción, la dirección se convierte a una localización en la memoria real. Esta traducción es automática y completamente transparente para el programador.

Una región es el área de memoria virtual que se le asigna a un programa. Cuando se carga el programa, sus direcciones son relocalizadas para ajustarlas a la región asignada.

Para establecer una comunicación entre el programador y la computadora, es necesario usar algún lenguaje comprensible para ambos, esto se consigue a través de un JCL. Por medio del JCL, se le dice al sistema operativo el trabajo que se quiere realizar, esto es, los programas que se van a ejecutar y los recursos que estos necesitan para llevar a cabo dicho trabajo.

En el JCL existen tres tipos de postulados:

- JOB : que es la parte que identifica un trabajo

- EXEC : Identifica un programa a ejecutar.

- DD : Identifican los archivos que usará el programa.

Los postulados de JCL se codifican y se colocan en un dispositivo de entrada para ser leídos por el sistema. A este JCL se le llama "job stream". El JOB STREAM esta formado por uno o mas jobs. cada job es una colección de pasos relacionados. El JOB MANAGEMENT dirige y controla el flujo de jobs.

El JOB MANAGEMENT tiene dos componentes :

El "Master Schedule" que es el medio por el cual el sistema se comunica con el operador. También es usado por el operador para programar e iniciar el trabajo del job scheduler.

El "Job Scheduler" que se encarga de leer, interpretar, programar, iniciar, registrar la salida y terminar los pasos de una serie de

trabajos (jobs) definidos y cargados por los usuarios.

CAMPOS DENTRO DE POSTULADOS DE JCL.

Los postulados de JCL se dividen en cuatro campos -nombre, operación, operando y comentarios-, pero no todo postulado incluye los cuatro campos.

El campo nombre identifica al postulado de control.

El campo de operación especifica el tipo de postulado de control (JOB, EXEC, DD, etc.). El campo de operación debe ir precedido y seguido al menos de un blanco.

El campo de operandos contiene parámetros separados por comas. El campo de operandos va después del campo de operando y debe ir seguido y precedido por al menos un blanco. Los parámetros del campo operandos son posicionales o palabras llave. Los parámetros posicionales se deben decodificar al principio del campo de operandos y deben seguir un orden. Los parámetros de palabras llave se codifican en cualquier orden. La ausencia de un parámetro posicional se indica con una coma, la ausencia de una palabra llave no requiere de codificación alguna.

El campo comentarios contiene información que pueda ser útil para la persona que codifica o usa el jcl. Los comentarios deben seguir al campo de operandos y deben ser precedidos por lo menos por un espacio en blanco.

A P E N D I C E D

INFORMACION TECNICA DEL SISTEMA 9121 / IBM

Este sistema es de la familia 9000 de IBM, como algunas características básicas podemos mencionar:

- Sistema operativo M V S (Multiple Virtual Storage)
- Uniprosesador enfriado por aire
- Puede operar en siete particiones lógicas
- Característica de proceso de vector
- Manejo de Criptografía avanzada

Como característica adicional cuenta con canales de comunicaciones de fibra óptica con los que se logran velocidades de transferencia de datos arriba de 10 mega bytes por segundo.

La capacidad de memoria puede tener de entre 16 y 512 megabytes configurada como almacenamiento central y arriba de 2 gigabytes configurada como memoria expandida en tiempo IML (Initial Machine Load).

Ventajas de diseño :

Un procesador independiente de E/S incorporado en todos los modelos manipula todas las actividades de los canales relacionados; este libera al procesador central para otras tareas y mejora todo el funcionamiento del sistema.

Facilidad de disponibilidad del procesador.

Esta facilidad minimiza el impacto de muchos errores no recuperables de procesador central (PC) en sistemas con dos o mas procesadores centrales. Esto es realizado moviendo el programa en ejecución del PC de falla a otro PC operacional.

Administración de recuperación dinámica.

Una capacidad en los modelos enfriados por aire trabajando junto con el nivel apropiado del sistema operativo MVS, permite la reconfiguración dinámica del sistema de E/S (Entrada/Salida). Los canales, unidades de control y dispositivos pueden ser adicionados o removidos sin la necesidad de apagar el sistema, ni tampoco de ejecutar el IPL (Initial Program Load - Programa Inicial de Carga); esto hace una contribución importante para mejorar la disponibilidad del sistema.

Almacenamiento en Mega bytes:

MODELO	PROCESADOR		CENTRAL		EXPANDIDO	
	MIN	MAX	MIN	MAX	MIN	MAX
320	64	1024	16	256	0	1008

Canales :

MODELO	TOTAL DE CANALES		CANALES EN PARALELO	
	MIN	MAX	MIN	MAX
320	12	48	0	48

Especificaciones:

Requerimiento de energía	MIN	MAX
50/60hz, KVA	6.0	7.9
Calor de salida para aire en KBTU/hr	19.1	25.2
Area en pies (ft) Area en mts.	14.7 1.4	24.7 2.3
Peso aproximado libras (lb) kilogramos (Kg)	1 750 794	2 800 1 270

A P E N D I C E E

MANUAL DEL USUARIO

SISTEMA DE REQUERIMIENTO POR NUMERO DE PARTE

C O N T E N I D O

	pág
1. Antecedentes.....	138
1.1. Panorama actual.....	138
1.2. Análisis de la situación actual....	140
2. Objetivos.....	148
3. Funciones.....	150
3.1 Diagrama funcional.....	150
3.2 Mantenimiento de base de información	151
3.2.1. Conversión del BOM.....	152
a) Flujo de trabajo.....	152
b) Calendario del proceso.....	153
3.2.2. Mantenimiento del archivo Maestro	
Partes (P/F).....	154
a) Flujo de trabajo.....	154
b) Calendario del proceso.....	156
3.3 Cálculo y emisión de órdenes de	
partes (normal-adicional).....	157
3.3.1. Información básica para la explosión	157
3.3.2. Ajustes.....	158
3.3.3. Lotificación de requerimientos...	159
3.3.4. Cálculo de órdenes.....	161
3.3.5. Transmisión de requerimientos....	161
3.3.6. Ejemplo de cálculo.....	161
3.4 Daily Due Date.....	167
3.4.1. Asignación de fecha de recibo....	167
3.4.2. Proceso de cálculo de Gross.....	169
3.4.3. Part file.....	169
3.4.4. Cálculo de órdenes.....	170
3.4.5. Información a Japón.....	173
4. Descripción de actividades.....	174
4.1. Actualización y mantenimiento del BOM	174
4.2. Parámetros de conversión	174
4.3. Mantenimiento nuevas partes Part/File.	175
4.4. Altas al Part File	176
4.5. Explosión de partes en base al programa	
de producción real	177

4.6. Detalle 1	177
4.7 Detalle 3	178
4.8. Control y actualización por RAN	179
4.9. Actualización del Part File con información de requerimiento	180
4.10. Respaldo del requerimiento enviado a Japón.....	180
4.11. Actualización del archivo de base de datos RAN-ASN-RECIBO.....	180

1. Antecedentes.

1.1 Panorama actual

El sistema actual de requerimiento de partes para planta Nissan en CIVAC esta constituido por un sistema híbrido de requerimiento por número de parte y por paquete CKD (Complete Knock Down)

La distribución del requerimiento de partes entre un sistema y otro se muestra en la siguiente tabla.

TIPO DE REQUERIMIENTO	VOL. PARTES	DESTINO	FRECUENCIA	LEAD-TIME	RESPONS.
* CKD	- 70 %	JPN/ESP	MENSUAL	- 90 DIAS	Y110
* NUMERO DE PARTE - A/E - VENDOR RELEASE - COMPLEMENTARIO	- 30 %	LOCAL U.S.A. JPN/ESP	MENSUAL MENSUAL EVENTUAL	- 15 DIAS - 30 DIAS ***	P420/Y110 P430 P430

*** DEPENDE DEL MEDIO DE TRANSPORTE

nota: Lead-Time es el tiempo que transcurre desde el envío del requerimiento hasta la llegada de partes a planta.

Bajo el sistema de requerimiento CKD no se puede manejar un nivel de existencia óptimo para cada parte (basado en costo y/o uso) ya que no son requeridas individualmente, lo cual impacta en el costo del mantenimiento de inventario en planta. éste problema se vuelve crítico a medida que los volúmenes de producción se incrementan.

Otro problema que produce el trabajar con esta clase de sistema se refleja en el manejo de materiales en planta, ya que para abastecer partes a línea es necesario realizar operaciones de desempaque y modulación, lo que se traduce en daños de partes.

Este problema se evita si para cada parte se establecen normas de empaque más apropiadas a fin de abastecer a línea en el mismo empaque de proveedor, pero con el sistema CKD esta solución no es

realizable ya que las todas las partes son manejadas con la misma norma de empaque.

El abastecimiento y costo del inventario requieren que las partes sean clasificadas por factores tales como:

- uso
- costo
- frecuencia de embarques

Con el fin de mantener en planta los niveles de existencia más apropiados para cada parte.

Otro problema que acarrea el abasto de partes por requerimiento CKD es el de la falta de control físico de las partes en planta , ya que el desempaque de cajas no es inmediato, lo cual provoca que no se conozca la existencia y ubicación de las partes en los almacenes.

Otro factor en contra del sistema de requerimiento CKD es el de que las áreas de almacenamiento que requieren van creciendo a la par que los incrementos de producción. Esta situación no es apropiada ya que en un momento determinado el área de almacenes puede ser mayor al área productiva.

En al medida que los almacenes sigan creciendo la inversión para lograr el control de los mismos representara gastos muy fuertes y el beneficio obtenido no se reflejara en la producción.

Es por ello que el sistema de requerimiento CKD debe ser sustituido por un sistema de requerimiento por número de parte bajo el cual pueda ser manejada el requerimiento planeado de partes, a fin de abatir los problemas anteriormente señalados.

Con la sustitución del requerimiento CKD se lograría una reducción en el lead-time de 90 días a 75 días. esta reducción de tiempo beneficia en la respuesta que la planta pueda dar a distribuidores respecto a los requerimientos que ellos hacen.

Junto con el sistema de requerimiento por número de partes deben ser planeados los sistemas de control de inventario y de programación, y establecer la comunicación necesaria con el sistema de control vehículo (opics).

1.2 Análisis de la Situación Actual

Conforme al origen de compra de las partes, se ha establecido la siguiente agrupación para las partes manejadas en planta:

a) partes importadas. Aquí son agrupadas las partes que son compradas a proveedores localizados en Japón, USA y España.

b) partes locales. Son abastecidas a planta por proveedores ubicados en México.

Esta división es importante ya que el método de requerimiento es diferente y además los tiempos de reacción varían dependiendo de la localización de proveedores.

De hecho para el manejo y control de las partes en planta, según su origen, se han creado áreas separadas tanto en control de producción como en manejo de materiales (recibo y almacenes).

En cuanto al requerimiento, éste es efectuado básicamente por las siguientes áreas:

a) material de Japón (CKD): staff de control de producción central.

b) material de USA, España y complementarios Japón: control de partes importadas.

c) material nacional: control de partes nacionales.

Como se mencionó anteriormente el tiempo de requerimiento es muy particular y esta en función de la ubicación de proveedores. En el siguiente diagrama se muestra la calendarización de actividades para efectuar el requerimiento.

De los diagramas anteriores podemos observar que los tiempos desde el requerimiento de partes hasta la llegada a planta, varían de 90 días a 16 días. en particular el tiempo de 90 días debe ser reducido con el propósito de que el requerimiento de partes sea lo más cercano a la producción real.

A continuación se muestra detallada la situación actual para los sistemas de requerimientos de partes.

1.3 Requerimiento de Partes

1.3.1 Partes Importadas (Japón y U.S.A.)

Para requerir esta clase de partes se manejan actualmente los siguientes tipos de pedido:

- a) c.k.d. (complete knock down)
- b) complementario establecido
- c) vendedor release
- d) órdenes de emergencia.

a) Pedido C.K.D.

El requerimiento CKD se utiliza para solicitar a Japón las partes para llevar a cabo un plan de producción. éste requerimiento no se hace a nivel de número de parte sino que se envía el plan de producción a nivel end/item con especificación de color interior, el cual Japón explota a fin de calcular la cantidad de partes que deben ser embarcadas para México.

Esta situación requiere que Japón mantenga un nivel exacto de especificación vehicular para las unidades que México produce, lo que exige que exista un canal de información oportuna entre las áreas encargadas de mantener la especificación en México y Japón (diseño y control de especificaciones del área de control de producción)

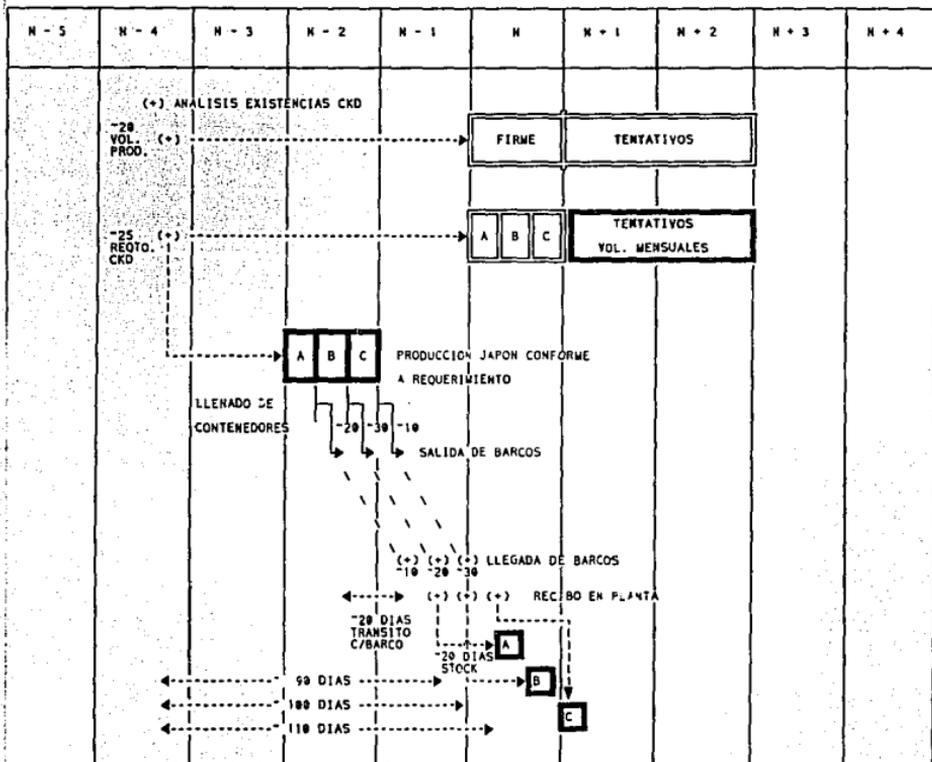
El problema que implica no requerir a nivel de número de parte es el de no poder controlar de manera individual los stocks de cada una de las partes, ya que estas llegan empacadas bajo una norma de empaque común (s.n.p.) finalmente esto se reduce en un costo alto de mantenimiento de inventarios.

Otro problema que resulta de éste tipo de requerimiento es en lo relacionado con el daño de partes por un exceso de operaciones en el manejo de partes en la planta, ya que es necesario efectuar operaciones de desempaque y modulación con el fin de

abastecer la línea de ensamble. Esto es provocado porque la norma de empaque no es la más apropiada para el manejo de partes en la planta.

Además los ajustes que se deben efectuar por causas de material dañado o inventario físico bajo, no pueden ser reflejados en el requerimiento CKD. Esto es solucionado a través de pedidos complementarios o pedidos de emergencia.

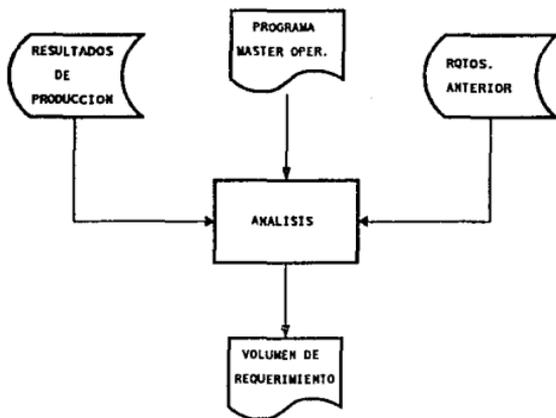
A continuación se muestra un procedimiento para efectuar el requerimiento CKD.



El requerimiento CKD para producción de unidades domésticas y de exportación se maneja por el staff de Control Producción Central (CPC).

La responsabilidad de (CPC) respecto al requerimiento CKD inicia desde el análisis de existencias hasta la confirmación de embarques.

Para efectuar el análisis de existencias CKD se consideran los resultados de producción, los recibos en planta, el material en tránsito y el programa maestro de producción. La combinación de estos factores da como resultado el volumen del requerimiento.



Este volumen es enviado a Japón, obteniendo como respuesta una propuesta para la distribución del volumen requerido en 3, o en ocasiones 4, embarques. ésta propuesta es analizada por el staff de planeación de materiales y si no afecta los niveles de stock que se manejan como objetivos se establece el requerimiento a nivel E/I (End/Item - Tipificación de Unidad).

En caso de que la propuesta de Japón para la distribución de los embarques no sea satisfactoria para la planta, es necesario negociar una distribución que no afecte los objetivos de stock.

El sistema de requerimiento CKD es completamente manual y para soporte del mismo se tienen dos analistas, uno para requerimiento doméstico y otro para exportación.

La comunicación con Japón es mediante el envío y recepción de fax.

Pero el problema de mayor impacto en éste tipo de requerimiento es la diferencia de tiempo entre el requerimiento y la producción de planta México, ya que esta diferencia en tiempo provoca que los cambios por adelantos o atrasos de la producción no puedan reflejarse inmediatamente en los requerimientos.

La posibilidad de acortar la diferencia de tiempo esta en el cambio del sistema CKD al de requerimiento por número de parte.

b) Pedido complementario

Este tipo de pedido es el manejo para requerir a Japón partes que:

- i) no son incluidas como material CKD, por ejemplo partes a consignación, algunas partes "NES" (tornillería), etc.
- ii) partes para refacciones.
- iii) por ajustes de inventario. daños o pérdidas.
- iv) para muestras.

El requerimiento de estas partes es a nivel individual. El tiempo desde el requerimiento hasta la llegada de partes a planta depende del tipo de pedido empleado. En general los tipos empleados y el tiempo de cada uno se mencionan a continuación:

TIPO PEDIDO	LEAD-TIME	USO
1M	30 DIAS	REPOSICION PARTES DANADAS
2M		
3M	90 DIAS	PARTES NES, CPNSIGNACION
4M		
5M		
6M		

La frecuencia de esta clase de pedidos es mensual.

El control de éste tipo de requerimientos queda a cargo del área de control de partes importadas, en donde los analistas se encargan de calcular manualmente el volumen de partes a requerir.

Este cálculo esta en función de la explosión de un programa de producción dentro de la especificación vehicular.

Los volúmenes obtenidos son lotificados por el factor de standard de empaque (SNP) definido por cada parte y del nivel stock deseado.

El programa utilizado para explosionar es el que sirvió como base del requerimiento CKD, aunque al volumen obtenido como resultado se le adicionan los requerimientos para refacciones o los ajustes por perdidas o daños.

La normatividad para la elaboración de éste tipo de pedidos es la establecida por el sistema "Neptune", manejado por Japón para la atención de requerimientos por número de partes.

El sistema de requerimiento por pedido complementario es manual y la comunicación de órdenes de partes es a través de fax por lo que requiere personal dedicado a esta función.

2.- Objetivos

- Sustituir el sistema de requerimiento CKD por un sistema de requerimiento por número de parte.

- Planeación y programación de requerimiento de partes para producción, a fin de que estos se encuentren disponibles en la cantidad exacta y en tiempo requeridos.

- Abatir el costo de mantenimiento de inventarios al controlar el nivel de existencia de cada parte en función de clasificación, factores de flota (Stock de Seguridad) y de frecuencias de embarque.

- Eliminar operaciones de desempaque y modulación de partes para abastecer la línea por medio de la adaptación de normas de empaque (S.N.P.) apropiadas para abastecimiento a producción.

- Reducir la generación de partes obsoletas al considerar puntos de efectividad.

- Eliminar la emisión de órdenes de partes por pedidos complementarios establecidos.

- Incluir bajo el control del sistema de requerimiento, la emisión de órdenes de partes adicionales.

- Eliminar el trabajo manual para el cálculo y ajuste de volúmenes de partes a requerir por medio de la comunicación de información entre el sistema de control de inventarios y de requerimiento de partes.

- Controlar cada requerimiento desde su emisión hasta la llegada planta por medio de la adopción del número de autorización de recibo (R.A.N.).

- Reducir el lead-time de requerimientos con Japón, actualmente se tiene un tiempo de 90 días desde el requerimiento hasta la llegada de partes a planta.

El logro de los objetivos anteriores esta sujeto al cumplimiento de las siguientes condiciones:

- La fuente de información para la especificación debe ser el B.O.M. que mantiene control de

producción. La exactitud de la especificación debe ser garantizada para el 100% de las partes.

Esta condición es necesaria ya que a partir del cambio del sistema CKD la responsabilidad de la especificación es de la planta CIVAC y no de la planta de empaque en Japón.

- Eliminar desviaciones de uso de partes (IACP's Instrucciones Autorizadas por Control de Producción) a fin de no generar excesos de partes en almacenes.

- Que los programas de producción sean cumplidos en la cantidad y tiempos con que fueron planeados. esto permite que la planeación de materiales pueda ser manejada adecuadamente con los proveedores.

- Tener un sistema para el de control material dañado que produzca la información oportunamente de los ajustes de partes para que se incluyan como requerimientos.

- Clasificar las partes por su importancia en el costo del inventario y a través de esta clasificación establecer factores de flota (niveles de existencia), frecuencias de embarque, frecuencias de recuentos. estos factores son los parámetros que nos permitirán el control de las partes.

- Que el área de manejo de materiales establezca las normas de empaque más apropiadas para el abastecimiento a línea.

- Planear la distribución de partes en los almacenes para facilitar el control dentro de planta de las mismas.

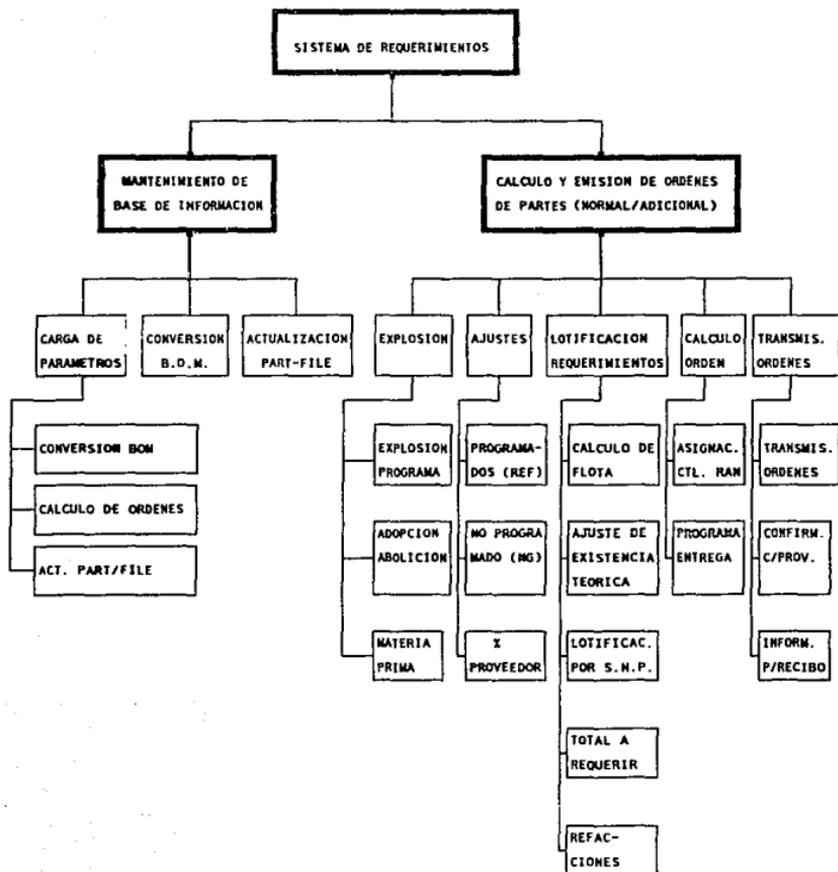
- Adoptar un sistema de control de órdenes de partes que considere cada requerimiento individualmente. esto es con la idea de que el proveedor cumpla con los embarques en la fecha y cantidad requeridos (R.A.N.)

- Ajustar con las áreas que están relacionadas con el sistema de requerimientos de partes, la forma de trabajo bajo el nuevo sistema: finanzas, depósito industrial, control de partes, abastecimientos, ventas, manejo de materiales, control de especificaciones, programación y sistemas.

- Adoptar sistemas de comunicación con proveedores más eficientes con el propósito de que la información fluya oportunamente.

3. Funciones.

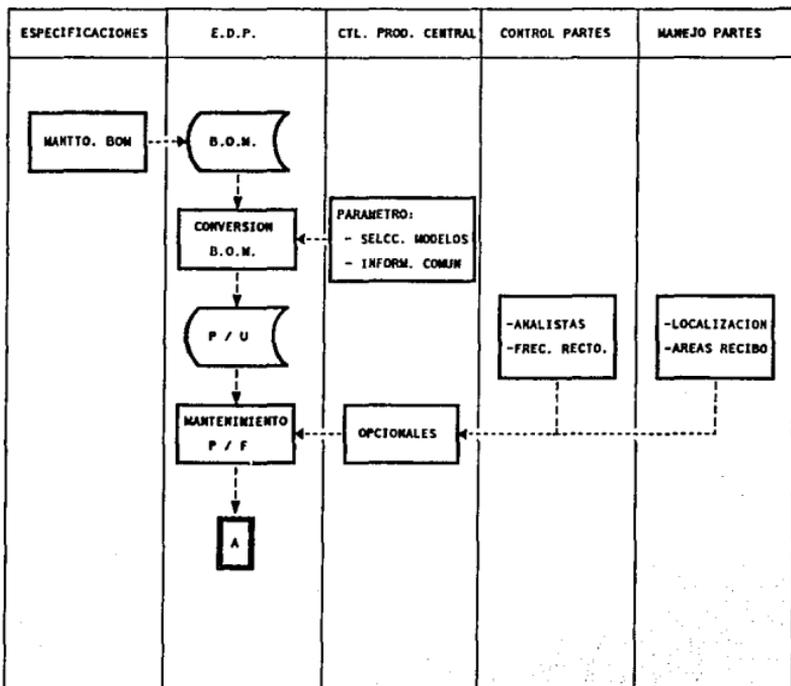
3.1. Diagrama Funcional



3.2. Mantenimiento de Base de Información.

La garantía de la exactitud de la información que el sistema de requerimiento maneja depende del mantenimiento oportuno que tiene. La principal fuente de información para el mantenimiento es el B.O.M., ya que a partir de esta información se generarán los archivos básicos del sistema, archivo de uso de partes (P/U) y el archivo de partes (P/F) mas archivo refacciones.

Adicionalmente se requiere de un conjunto de parámetros que servirán para orientar los cálculos que el sistema de requerimientos efectúa para emitir órdenes de partes.



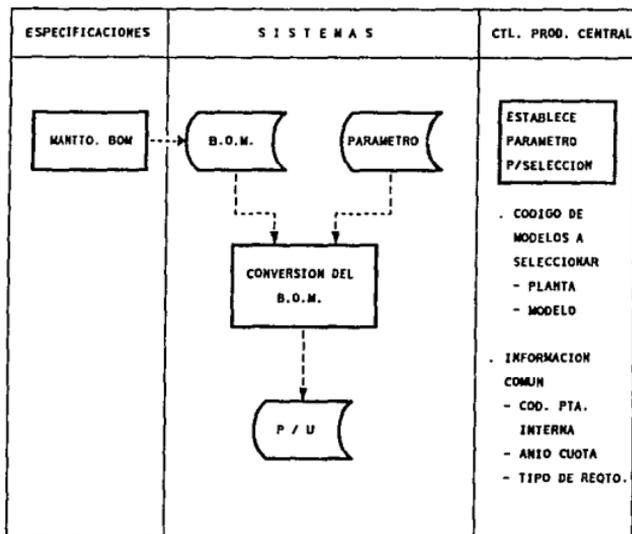
3.2.1. Conversión del B.O.M.

Los registros necesarios para el cálculo de órdenes y para el del inventario son seleccionados del maestro de especificaciones B.O.M. y deslizados al archivo de usos de partes de algún modelo (código de planta interna, año cuota, código de origen de partes y código de tipo de requerimiento).

La frecuencia de ejecución de éste proceso será semanal para garantizar que la exactitud de la información en el P/U es la requerida para el cálculo de Órdenes.

La selección de registros del B.O.M. hacia el P/U será orientada por un parámetro, en el que se indicaran los modelos a deslizar así como inflación común para las partes.

a) Flujo de Trabajo.



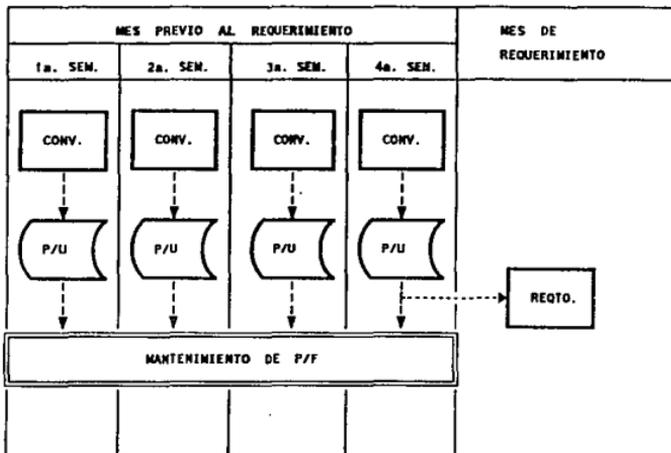
El archivo P/U es generado cada vez que el proceso de conversión es ejecutado.

La especificación del parámetro es anual o cada vez que haya cambios de modelo.

b) Calendario de Proceso.

Con el fin de garantizar la exactitud de la especificación que ha de ser usada por el sistema de requerimiento de partes, el proceso de conversión del BOM será efectuado semanalmente.

B) CALENDARIO DE PROCESO



3.2.2. Mantenimiento del Archivo Maestro de Partes (P/F)

El archivo P/F contiene la información básica para el cálculo de órdenes y para el control de partes en planta.

En el proceso masivo de mantenimiento al P/F el archivo P/U es usado como la fuente de información para crear nuevos registros, a establecer información común conforme a un atributo.

Una vez creado el P/F éste archivo será actualizado a través de procesos masivos (a través del P/U y parámetros) y de procesos en línea (sistema de control de inventario).

El archivo contendrá datos de las áreas siguientes:

- especificaciones
- control de partes
- manejo de materiales
- inspección
- abastecimiento
- deposito industrial.

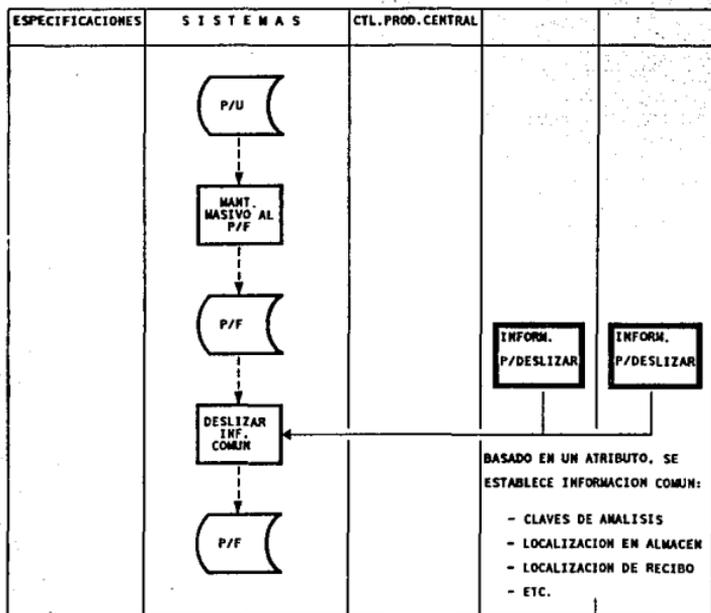
a) Flujo de Trabajo.

El proceso de actualización masiva del P/F será efectuado con la misma frecuencia que el proceso del P/U.

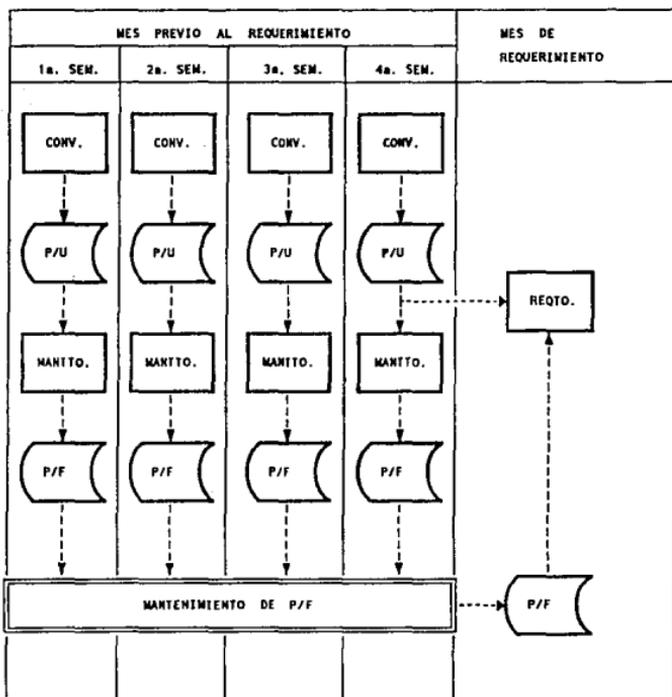
El objetivo de éste proceso es la de adicionar las partes que el área de especificaciones libera. al mismo tiempo se tendrá que emitir un listado para las áreas de control de partes y de manejo de materiales a fin de que establezcan la información necesaria para las partes que se agregen.

El deslizamiento de información común sólo será realizado por requerimiento de las áreas de control de partes y manejo de materiales. básicamente éste proceso es efectuado al inicio de año cuota a fin de establecer de manera automática información para el control de las partes.

A) FLUJO DE TRABAJO



B) CALENDARIO DE PROCESO



3.3 Cálculo y Emisión de Ordenes de Partes (normal/adicional)

Se describen a continuación las actividades que incluyen éste punto:

3.3.1. Información básica para la explosión.

- a) Part Usage
- b) Programa de Producción

El proceso de explosión consiste en calcular la cantidad de cada parte que se requiere para cubrir un programa de producción a nivel modelo.

EJEMPLO:

	0	0	0	0	0	0	
	0	0	0	0	0	0	
	1	2	3	4	5	6	
PROGRAMA	C	a	b	c	d	e	f
END/ITEM	K	A	B	C	D	E	F

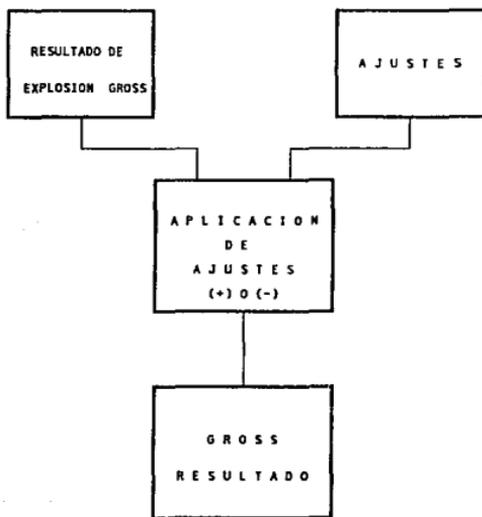
----) CANTIDADES

PARTE X	0	0	0	0	0	0		E/I
	0	0	0	0	0	0		
	1	2	3	4	5	6		
2 PARES/COLOR C	X		X	X	X	X		USAGE

$$\text{TOTAL GROSS} = 2Xa + 2Xc + 2Xd + \dots$$

3.3.2 Ajustes

El proceso de requerimiento considera un archivo de ajustes como resultado de diversas transacciones, tales como recuentos cíclicos. Estos ajustes pueden ser positivos o negativos.



El proceso marca los registros del archivo de ajustes como considerados para requerimiento con el objeto de no volver a considerarlos.

status 1..... status 2

3.3.3 Lotificación de Requerimientos

Para la lotificación se consideran los siguientes conceptos:

- 1.- Flota requerida para el siguiente periodo
- 2.- SNP de la parte
- 3.- Existencia teórica

1.- La flota para el siguiente periodo se calcula con:

programa / días laborales = uso promedio diario

flota (cantidad) = uso promedio diario x flota (días)

2.- El SNP tomado del part file, cuyo origen es negociado por manejo de materiales con Japón y mantenido en el mismo.

inventario teórico = total requerido - total Gross mas inventario teórico (n-1)

cantidad a requerir Gross = Gross mas ajuste mas flota - inv. teórico (n-1)

cantidad a requerir lote = (cantidad a requerir Gross) / SNP
(si hay fracción se agrega un SNP)

3.3.4 Cálculo de Ordenes

El paso final de IPO es la preparación de los archivos que serán enviados al proveedor (detalle 1 de detalle 3) los que incluyen información para requerimiento a nivel número. de RAN.

El total a requerir es seleccionado por número de RAN de acuerdo a :

- 1.- SNP
- 2.- RAN máximo y RAN mínimo

ejemplo:

P A R T E X	MES FIRME N			TENTATIVOS	
	D1	D2	D3	N+1	N+2
	500	500	700	1200	1500

SNP = 100

CANTIDAD A REQUERIR = 1700

RAN MAXIMO = 500

ENTONCES: $1700 / 500 = 3 \times 500 + 200$

EL REQUERIMIENTO ES EL SIGUIENTE:

R A N	CANTIDAD	FECHA RECIBO
RA00001	500	1205 PRIMER DEC
RA00002	500	1215 SEGUNDA DEC
RA00003	500	1225 TERCER DEC
RA00004	500	1225 TERCER DEC

DONDE FECHA RECIBO _____

.....

.) DIA DE RECIBO

.) MES DE RECIBO

3.3.5 Transmisión de Requerimientos

DESTINOS	OBJETIVO	ALTERNATIVA
J A P O N	VIA SATELITE SISTEMA VAN	CINTA MAGNETICA
U S A	* VIA SATELITE (E D I - A I A G)	FACSIMIL (DOCUMENTO)
N A C I O N A L	* CORREO ELECTRONICO Y TRANSFERENCIA DE ARCHIVOS (EDI-AIAG)	CORREO (DOCUMENTO)

3.3.6 Ejemplo de Cálculo

Se presenta a continuación un ejemplo del cálculo de órdenes de partes. en el se consideran el inventario teórico, la cantidad Gross y la cantidad a requerir.

CALCULO DE ORDENES

PASO 1. CALCULO DE GROSS

E/I SCHEDULE

usage	N + 2			N+3 N+4	
	1	2	3		
001	100	100	100	200	200
002	50	50	0	100	100
003	100	100	100	300	300
.					
.					

numero parte	pzas. carro	color int.	usages....			
			001	002	003	.
A	1		0	0		
B	1				0	
C	1		0	0	0	
.						
.						

CALCULO GROSS

usage	N + 2			N+3 N+4	
	1	2	3		
A	150	150	100	300	300
B	100	100	100	300	300
C	250	250	200	600	600
.					
.					

AJUSTES

REQUERIMIENTO DE
PARTES A JAPON

Cálculo de la cantidad a requerir del mes firme (N+2).

Consideraciones :

i.- Hay un programa de producción mensual en el cual se especifican el mes firme (N+2) decenalmente y los dos meses siguientes como tentativos

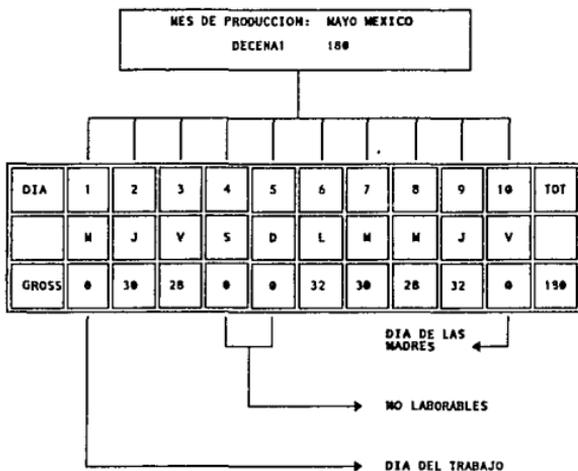
ii.- Para la explosión del programa a Daily Due Date (Requerimiento Diario) se toma solamente el mes firme (N+2) ya que los otros dos son tentativos.

iii.- Se explicará el proceso del cálculo de requerimiento con cantidades como ejemplo.

A) MES DE PRODUCCION MAR-JPN/MAY-MEX

	MES FIRME (N+2)			TENT	TENT
	D1	D2	D3	N+3	N+4
TOTALES PROG. DE PRODUCCION	180	205	170	462	464

B) EXPLOSIONADO DE LOS REQUERIMIENTOS DECENALES A DAILY DUE DATE



EL EXPLOSIONADO SE LLEVA A CABO DE LA MISMA MANERA EN CADA UNA DE LAS TRES DECENAS.

TOTAL DE DIAS HABILES PRIMERA DECENA : 6

TOTAL DE DIAS HABILES SEGUNDA DECENA : 6

TOTAL DE DIAS HABILES TERCERA DECENA : 9

EL GROSS DECENAL SE REPARTE EQUITATIVAMENTE ENTRE LOS DIAS HABILES DE CADA DECENA.

c) Requerimiento

Para el día 1 al día 30 o 31 del mes se llevan a cabo los siguientes cálculos:

requerimiento día n = (Gross día n - inv. teórico día n-1) ajustado al SNP

inv. teórico día n = inv. teórico día n-1 + req. día n - Gross día n

considerando un SNP = 40

Se presentan a continuación los cálculos de órdenes para el ejemplo anterior:

DIA	GROSS	REQUERIMIENTO	INV. TEORICO
1	0	0	20
2	30	REQ = GROSS DIA2 - INV. TEO. DIA 1 REQ = 30 - 20 = 10 AJUSTADO AL SNP -----) 40	INV. TEO = INV TEO DIA 1 + REQ DIA 2 - GROSS DIA 2 INV. TEO = 20+0-30 = 10
3	20	REQ = 20 - 30 = - 10 0	2
4	0	0	2
5	0	0	2
6	32	40	10
7	30	40	20
8	28	40	32
9	32	0	0
10	0	0	0

* SE CONSIDERA QUE HABIA UN INVENTARIO TEORICO DE 20 A INICIO DE MES

** CUANDO RESULTA NEGATIVO GROSS DE DIA (N) - INV TEO DE DIA (N-1).

INDICA QUE HAY MATERIAL PARA ESE DIA, POR LO QUE NO ORDENA MATERIAL.

Los cálculos son similares para las 2 decenas restantes

RESUMEN

DECENA	GROSS	INV. TEO. AL INICIO DE LA DECENA	REQUERIDO	INV. TEO. AL FINAL DE LA DECENA
DECENA1	180	20	160	0
DECENA2	205	0	240	35
DECENA3	170	35	160	25
TOTALES MES DE MAYO	555		560	INV. TEO. A FIN DE MES 25

D) PARA EL CASO PARTICULAR DE LOS AJUSTES

- EL AJUSTE TOTAL SE CARGA AL PRIMER DIA DE GROSS DEL MES, ASI TENEMOS QUE PARA EL EJEMPLO ANTERIOR:

	MES FIRME			TENT	TENT
	DEC1	DEC2	DEC3	N + 3	N + 4
SIN AJUSTES	180	205	170	462	464
SIN AJUSTES	220	205	170	462	464

EXPLOSIONADO DAILY DUE DATE DE DECENA1

CONSIDERANDO UN AJUSTE DE 40

D I A	1	2	3	4	5	6	7	8	9	10
GROSS SIN AJUSTES	0	30	28	0	0	32	30	30	28	32
GROSS CON AJUSTES	0	70	28	0	0	32	30	30	28	32

Considerando un ajuste de - 40

D I A	1	2	3	4	5	6	7	8	9	10
GROSS SIN AJUSTES	0	30	28	0	0	32	30	30	28	32
GROSS CON AJUSTES	0	0	18	0	0	32	30	30	28	32

NOTA: EL AJUSTE POSITIVO SE APLICA AL PRIMER DIA DE GROSS DEL MES, PORQUE SI SE TIENEN CRITICOS SE DESEA QUE LAS PARTES LLEGEN CUANTO ANTES . O CUANDO HAY AJUSTE NEGATIVO POR SOBRESTOCK PARA EVITAR QUE SIGAN ACUMULANDOSE.

3.4 Daily Due Date (Requerimiento Diario)

Especificación de los cambios más significativos para el cálculo de órdenes a Japón, considerando la asignación de fecha de recibo de partes en planta.

Se presenta a continuación los cambios en la lógica de requerimiento de partes a Japón, entre ellos:

- asignación de fecha de recibo de RANES
- programa de producción para requerimiento
- cambios en la estructura del part file
- proceso del cálculo del Gross
- cálculo de órdenes de requerimiento (a nivel diario)

nota: Los cambios se refieren básicamente a la forma de pedido de partes, la cual se hacía en forma decenal. es decir en los reportes del detalle 3 se indicaban los RANES con sus fechas de arribo a planta decenalmente, ahora se indicaran sus fechas de arribo por día.

3.4.1 Asignación de Fecha de Recibo

Objetivo.

En función del programa de producción diario y de la flota definida a cada parte, se asignará la fecha requerida de recibo de partes en planta.

Beneficio Esperado.

- Soportar la función de llenado de contenedores en función de la fecha de arribo especificado a cada parte, en adición a la restricción de mezcla de partes.

- Facilitar el control y seguimiento de materiales en tránsito.

- Distribuir la actividad de recibo de materiales en planta conforme a la fecha requerida de recibo.

Condiciones.

A fin de llevar a cabo la implantación de esta nueva función es necesario complementar los siguientes elementos:

- El programa de producción tendrá que mostrar los volúmenes de producción del mes firme a nivel diario. Los primeros 10 días del primer mes tentativo también tendrán que ser especificados a nivel diario y el volumen de los días restantes se presentara sumariado.

- El factor de flota tendrá que ser especificado para cada parte. La función de éste factor será la de indicar cuantos días antes del uso de las partes en producción, tendrán estas que ser recibidas.

- Especificación de calendario de trabajo en el cual se indiquen días laborables. el mantenimiento al calendario garantiza el cálculo exacto de la fecha requerida de recibo de partes.

- Los procesos actuales de cálculo de Gross y órdenes así como el detalle de información que producen deberán ser rediseñados.

- Las casos especiales de vacaciones de Japón serán considerados como un adelanto en el requerimiento de los primeros diez días del mes tentativo.

3.4.2 Proceso de Cálculo de Gross.

En general la mecánica del cálculo de Gross será la misma que actualmente se usa (en función del programa de producción, modelo y usages por modelo).

Reporte para verificación de cálculo de Gross.

El diseño del reporte mostrando el detalle del mes firme dividido en tres decenas, y en los meses tentativos el detalle es mensual.

3.4.3 Part File.

La estructura del Part File es modificada a fin de manejar adecuadamente el detalle de información en el mes firme y en el primer mes tentativo, en los conceptos de cantidades de Gross, inventario teórico y cantidad requerida.

En general el resto de la información contenida en el Part File no cambia.

El detalle del cambio efectuado se muestra a continuación:

DETALLE P/F (SIN CAMBIOS)	DETALLE DE CANTIDADES INV. TEORICO / GROSS /																
	N-3	N-2	N-1	N	N+1	N + 2							30	31	1		
						1	2	3	4	5	6	7				

3.4.4 Cálculo de Ordenes

La generación de órdenes (RANES) cambia de una base decimal a una diaria, y por lo tanto la asignación de fecha de recibo depende del día programado de producción y de la flota asignada.

La cantidad de partes a requerir se calcula bajo la siguiente base:

1) Més firme (n+2).

(1) Cálculo cantidad a requerir día i-esimo (i=1,..31)

$\text{cant-tot} = \text{Gross}(i) + \text{ajuste} - \text{inv-teo}(i-1)$

si $\text{cant-tot} > 0$

$\text{cant-lot} = \text{int}((\text{cant-tot} + \text{snp-ext} - 1) / \text{snp-ext}) * \text{snp-ext}$

$\text{cant-req} = \text{int}((\text{cant-tot} + \text{ran-max} - 1) / \text{ran-max}) * \text{ran-max}$

si $\text{cant-req} > \text{cant-lot}$

$\text{cant-req} = \text{ran-min} + \text{cant-req} - \text{ran-max}$

de otra forma

$\text{cant-req} = 0$

$\text{req}(i) = \text{cant-req}$

$\text{inv-teo}(i) = \text{inv-teo}(i-1) + \text{req}(i) - (\text{Gross}(i) + \text{ajuste})$

notas:

a) cuando $i = 1$ tomar inv-teo de mes $n+1$

b) cuando $i = 1$ tomar los ajustes.

(2) Asignación de número de RAN y asignación de fecha de recibo.

Una vez calculada la cantidad a requerir el siguiente paso será asignar un número de RAN a cada embarque y al mismo tiempo calcular la fecha de arribo a planta.

(d) graba RAN en archivo

2) mes tentativo (n+3).

(cálculo cantidad a requerir para día i-esimo (i=1,...10)

cant-tot = Gross(i) - inv-teo(i=1)

si cant-tot > 0

cant-lot = int((cant-tot + snp-ext - 1) / snp-ext) * snp-ext

cant-req = int((cant-lot + ran-max - 1) / ran-max) * ran-max

si cant-req > cant-lot

cant-req = ran-min + cant-req - ran-max

de otra forma

cant-req = 0

req(i) = cant-req

inv-teo(i)=inv-teo(i-1) + req(i) - Gross(i)

nota: a) cuando i=1 tomar inv-teo de día 31 de mes n+2

(2) asignación de número de RAN y asignación de fecha de recibo

Igual a cálculos del mes n+2.

(3) cálculo cantidad a requerir días restantes

cant-tot = Gross(n+3) - inv-teo(día 10 de n+3)

si cant-tot > 0

cant-lot = int((cant-tot + snp-ext - 1) / snp-ext) * snp-ext

cant-req = int((cant-lot + ran-max - 1) / ran-max) * ran-max

si cant-req > cant-lot

```

cant-req = ran-min + cant-req - ran-max
de otra forma
cant-req = 0
req(n+3) = cant-req
inv-teo(n+3) = inv-teo(día 10 de n+3) + req(n+4)
- Gross(n+4)

3) mes tentativo (n+4).
(1) cálculo cantidad a requerir
cant-tot = Gross(n+4) - inv-teo(n+3)
si cant-tot > 0
cant-lot = int((cant-tot + snp-ext - 1) / snp-
ext) * snp-ext
cant-req = int((cant-lot + ran-max - 1) / ran-
max) * ran-max
si cant-req > cant-lot
cant-req = ran-min + cant-req - ran-max
de otra forma
cant-req = 0
req(n+4) = cant-req
inv-teo(n+4) = inv-teo(día 10 de n+3) + req(n+4)
- Gross(n+4)

```

3.4.5 Información a Japón (detalle 1 y 3)

Estos archivos contienen las órdenes de compra a nivel número de parte. La transmisión es realizada vía satélite a Los Angeles mediante el sistema VAN, y de ahí es retransmitido a Japón vía cable submarino.

4. Descripción de Actividades

4.1 Actividad: Actualización y Mantenimiento del B.O.M.

Departamentos: Especificaciones y Diseño

Descripción: El mantenimiento y actualización del B.O.M. consiste en lo siguiente:

a) Dar fecha de adopción a las nuevas partes registradas en el B.O.M. por diseño.

Diseño se encarga de registrar la parte en el B.O.M. y Especificaciones de darle su fecha de adopción y liberarla.

La liberación de una parte consiste en llenar los siguientes campos:

- manufacturing process code
- fecha de adopción
- processing classification
- unite code
- sourcing classification

b) Dar fecha de abolición a las partes abolidas por acuerdo Nisme-Japón, las cuales han sido registradas por diseño en el B.O.M..

c) Registrar los cambios efectuados en alguna parte.

Tanto las adopciones como las aboliciones y cambios en partes son notificadas por Diseño a Especificaciones a través de una forma de control.

4.2 Actividad: Parámetros de Conversión

Departamento: Especificaciones

Descripción:

Es el deslizamiento de los registros de partes del maestro B.O.M. hacia el archivo de utilización de partes (part usage P/U) orientado por varios parámetros definidos por especificaciones.

Los registros necesarios para el cálculo de órdenes y para el del inventario son seleccionados del maestro de especificaciones B.O.M. y deslizados al archivo de utilización de partes P/U.

Adicionalmente es establecida información común para las partes de algún modelo, como es la siguiente:

* Código de modelo a seleccionar:

- planta
- modelo

* Información común.

- código de planta interna
- año de cuota
- tipo de requerimiento

* Información para control de partes en el maestro de partes

La información anterior es definida al sistema IPO por el departamento de Especificaciones, de acuerdo a las necesidades de la planta. Posteriormente se llevará acabo la conversión del B.O.M..

Esta actividad la solicitará Control de Producción al sistema solo con la autorización de planta mediante un formato de control específico.

4.3 Actividad: Mantenimiento de Nuevas Partes al Maestro de Partes (P/F)

Departamentos: Manejo de Materiales y Especificaciones.

Descripción:

El mantenimiento al P/F consiste en registrar en éste archivo todos los cambios sufridos por cualquiera de las partes en uso.

Para esto, Especificaciones informa inmediatamente a Manejo de Materiales cuando hay que llevar a cabo esta actividad.

Algunos de los cambios pueden caer en los siguientes campos:

- asignación de proveedor analista.
- datos de proveedor
- rans (múltiplos de SNP)

- SNP (interno y externo)
- recuentos cíclicos
- nivel de flota
- tolerancias (bajo embarques, sobreembarques, etc.)
- localizaciones (recibo, almacén, línea, etc)
- abasto a línea
- norma de empaque

4.4 Actividad : Altas al Maestro de Partes (P/F)

Departamento: Especificaciones

Descripción:

Básicamente, las altas al maestro de partes (P/F) consiste en registrar en éste archivo las nuevas partes (partes adoptadas).

Cada una de las partes en éste archivo tiene una llave de identificación, la cual esta compuesta por varios campos. Cuenta también con campos que se encuentran vacíos en el momento posterior al registro de la parte, pero que serán llenados cuando se haga el requerimiento de esta parte.

El procedimiento llevado a cabo es el siguiente:

- a) Obtener el archivo de usage de partes del P/U
- b) Se genera un reporte de todas las altas al P/F con el fin de verificarlas, tomando en cuenta el archivo de Gross Check.
- c) Mediante un procedimiento similar se dan las partes de alta, es decir, todas aquellas que sean adoptadas en el maestro de partes P/F.

Los campos vacíos de los registros de las partes del P/F son los correspondientes al requerimiento de la parte, por lo que después de la explosión de partes se cargan con su cantidad correspondiente (cantidad a pedir de esa parte).

4.5 Actividad: Explosión de partes en base al programa de producción real.

Departamento: Control de Producción Central

Descripción:

El explosionado de partes se lleva a cabo de igual manera que con el del programa de producción constante, sólo que ahora se realiza con el programa de producción real. Este maneja un mes firme y dos tentativos.

El explosionado se lleva a cabo realizando una comparación del archivo de partes en uso P/U vs programa de producción real, seleccionando los registros correspondientes con:

- planta
- año
- modelo
- usage
- etc.

El resultado de éste proceso es el total teórico de partes a pedir a Japón.

4.6 Actividad: Detalle 1

Departamento: Especificaciones

Descripción:

El detalle 1 es el resultado del cálculo del requerimiento de las partes a pedir a Japón, con algunas ajustes aplicados a algunas partes. Es decir, después de la sumarización se aplican ajustes a ésta, lo que trae como resultado la generación de P/F WORK, de donde se obtiene el archivo de Detalle 1, que es el total de las partes a pedir a Japón.

El detalle 1 contiene los siguientes encabezados:

- a) número de parte.
- b) descripción de la parte
- c) cantidad a requerir en el mes n+2 repartida en tres decenas.
- d) cantidad a requerir en los meses (n+3 y n+4)

De tal forma que en el detalle 1 se pide un mes firme n+2 repartido en tres decenas y dos tentativos (n+3 y n+4).

Así pues, el detalle 1 es un reflejo de las operaciones llevadas a cabo para obtener el total a requerir de una parte. En el no se especifican los RANES requeridos para una de las partes, pero si el total a requerir de las mismas.

4.7 Actividad: Detalle 3

Departamento: Especificaciones

Descripción:

El detalle 3 es el requerimiento de cada una de las partes contenidas en el detalle 1, pero ajustado al SNP de cada una de ellas.

Es decir, el total de cada parte a requerir se ajusta a su SNP y se le asigna un número de RAN.

la forma de detalle 3 tiene los siguientes encabezados:

- a) número de parte.
- b) localización de bloque (dock location)
- c) número de RAN
- d) fecha promedio en que se requiere en planta (mes día) = mmdd
- e) cantidad pedida de la parte.

Tanto el archivo de detalle 1 como el archivo de detalle 3 son enviados a Japón por medio de la red de valor agregado (VAN). Esta operación se lleva a cabo una vez que Especificaciones ha realizado una revisión exhaustiva de todas las partes, para confirmar que efectivamente se esta pidiendo la cantidad correcta de la parte correcta.

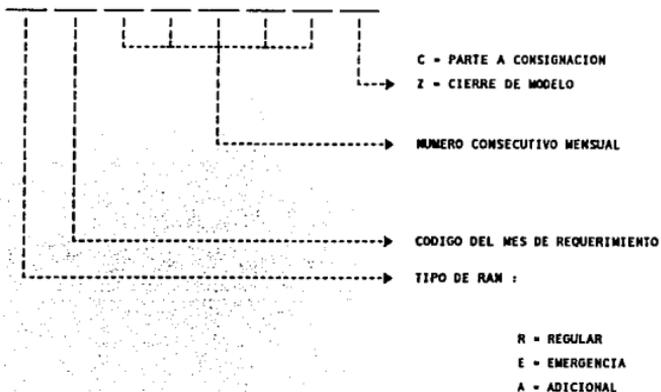
Aún después de la revisión pueden surgir diferencias con Japón por lo que existen formas para pedir RANES adicionales o para cancelar RANES solicitados.

4.8 Actividad: Control y Actualización por RAN

Departamentos: Sistemas

Descripción:

Esta es una forma de control de partes por número de RAN. A continuación se presentan los campos contenidos en el :



4.9 Actividad: Actualización del Part File con información de requerimiento.

Departamentos: Control de Producción Central

Descripción:

Durante el cálculo de las órdenes de trabajo, se genera el P/F de trabajo, porque se necesita que la información de las partes permanezca constante. así que una vez obtenidos los detalles 1 y 3 se procederá a actualizar el Part File, con los últimos cambios de las partes modificadas.

4.10 Actividad: Respaldo del requerimiento enviado a Japón.

Departamento: Control de Producción Central

Descripción:

Se realiza el respaldo de los archivos básicos del sistema en cinta tales como GROSS-CHECK, P/U, PROGRESS, EXPLOSION, P-F DE TRABAJO, DETALLE1 Y 3.

4.11 Actividad : Actualización del archivo de Bases de Datos RAN-ASN-RECIBO

Departamento: Control de Producción central

Descripción:

Este proceso consiste en la actualización del archivo de RANES que será verificado con el ASN de Japón, esta actividad será realizada por el departamento de Operaciones posterior a la recepción de las partes en las plantas ensambladoras.

A P E N D I C E F

BIBLIOGRAFIA

- Fourth-Generation Languages
Vol. I
Martin, James.
Prentice Hall, USA 1985.
- Ingeniería de Software
Richard Fairley.
Mc Graw Hill, México 1987.
- Software Maintenance: The Problem and Its
Solutions
C. McClure y Martin, James.
Englewood Cliffs NJ. Prentice Hall, Inc.1983
- Application Development Without Programmers
Martin, James.
Englewood Cliffs NJ. Prentice Hall, Inc.1982
- Programming Languages
J.E.Sammet.
Englewood Cliffs NJ. Prentice Hall, Inc.1969
- Information Engineering
C. Finkelstein y Martin, James.
Technical Report 20, Savant Institute, 1981.
- IMS/VS Users Talk About Productivity and
Control
Publicación No. G520-3511-0
White Plains, NY, IBM Corporation, 1980.

- Databases: Improving Usability and Responsiveness
B. Shneiderman
Academic Press, NY, 1978.

- The Psychology of Human Computer Interaction
Hillsdale, NJ
Lawrence Erlbaum Associates, 1983

- Information Engineering
C. Finkelstein y Martin, James.
Technical Report 22, Savant Institute, 1981.

- Computer-Based Medical Consultations
E.H. Shortliffe.
MYCIN, New York
American Elsevier, 1976.

- Processor Models IBM ES/9000 9121
IBM Product Information.
No. G511-1491-02