

PROYECTO INFORMÁTICO

Una Metodología Simplificada

Marcelo Claudio Perissé

©Marcelo Claudio Perissé

Doblas 515, piso 2, Buenos Aires.

Teléfono: (011) 4923-3958

e-mail: marceloperisse@hotmail.com

<http://www.geocities.com/marceloperisse>

Impreso en Argentina

febrero de 2001

ISBN: 987-43-2947-5

INDICE

PROYECTO INFORMÁTICO	7
INTRODUCCIÓN	7
1 PROYECTO	9
1.1. QUÉ ES UN PROYECTO INFORMÁTICO	11
1.2. INICIO DE UN PROYECTO NFORMÁTICO	12
2. PLANEAMIENTO DE UN PROYECTO INFORMÁTICO	14
2.1. MÉTODOS DE PLANIFICACIÓN TEMPORAL DE TAREAS	14
2.2. MÉTODOS PARA PLANIFICACIÓN DE RECURSOS	16
<i>PLANIFICACIÓN FINANCIERA</i>	17
2.3. CONSIDERACIONES EN UN PLAN ESTRATÉGICO INFORMÁTICO	18
3. BASE DE DATOS	19
3.1. ESTRUCTURA DE UNA BASE DE DATOS	20
3.1.1. TIPOS DE ARCHIVO	22
ARCHIVO MAESTRO	22
ARCHIVO DE TRANSACCIONES	22
ARCHIVOS DE CONTROL	22
ARCHIVO DE PLANEAMIENTO	22
3.1.2. LLAVE PRIMARIA O IDENTIFICADORA	23
3.1.3. INDICES DE ACCESO	23
4. HERRAMIENTAS DE ANÁLISIS	25
4.1. MODELOS CONCEPTUALES	25
4.2. LA MODELIZACIÓN DE LAS FUNCIONES DEL SISTEMA	26
4.2.1. LISTA DE EVENTOS	26
TÉCNICA PARA EL DISEÑO DE UNA LISTA DE EVENTOS	26
4.2.2. EL DIAGRAMA DE FLUJO DE DATOS	28
TÉCNICA DE DISEÑO DEL DFD	29
RESTRICCIONES DEL DFD	30
RECOMENDACIONES PARA UN DFD	31
4.3. EL DICCIONARIO DE DATOS	31
CONTENIDO DEL DICCIONARIO DE DATOS	31
4.4. LA MODELIZACIÓN DE DATOS ALMACENADOS	34
EL MODELO RELACIONAL DE DATOS (RDM)	34
TIPOS DE RELACIONES	35
BENEFICIOS DEL RDM	35
TÉCNICA DE DISEÑO DEL RDM	36
Diseño de la Relación uno a uno	37
Diseño de la Relación uno a varios	40
Diseño de la Relación varios a varios	41
NORMALIZACIÓN	42
PRIMERA FORMA NORMAL (1FN)	43
SEGUNDA FORMA NORMAL (2FN)	45
TERCERA FORMA NORMAL (3FN)	47
4.5. FLUJOGRAMAS	48
4.6. TABLAS DE DECISIÓN	49
4.7. MODULOS DE UN SISTEMA	50
EL ÁRBOL DE UN SISTEMA	51
ESPECIFICACIÓN DE LOS MÓDULOS DEL SISTEMA	51

5. HERRAMIENTAS CASE	52
QUE ES UNA HERRAMIENTA	52
COMO OPERA UNA HERRAMIENTA CASE	52
6. DESARROLLO	66
CARACTERÍSTICAS EN TODA METODOLOGÍA DE PROCESAMIENTO DE DATOS	66
6.1. METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS	67
6.2. METODOLOGÍA ESTRUCTURADA SIMPLIFICADA	68
APÉNDICE A	70
EJERCICIONº:1	71
PROCEDIMIENTO: COMPRAS MAYORES Y RECEPCION	71
DESCRIPCIÓN DEL PROCEDIMIENTO:	71
LISTA DE EVENTOS.....	72
DIAGRAMA DE FLUJO DE DATOS	73
EJERCICIO N°:2	76
PROCEDIMIENTO : VENTAS Y GESTIÓN DE DESPACHO	76
DESCRIPCIÓN DEL PROCEDIMIENTO	76
LA LISTA DE EVENTOS.....	76
DIAGRAMA DE FLUJO DE DATOS	77
EJERCICIO N°:3	78
PROCEDIMIENTO: VENTAS	78
DESCRIPCIÓN DEL PROCEDIMIENTO:	78
LISTA DE EVENTOS.....	78
DIAGRAMA DE FLUJO DE DATOS	80
EJERCICIO N°:4	83
PROCEDIMIENTO: COMPRAS	83
LISTA DE EVENTOS.....	83
DIAGRAMA DE FLUJO DE DATOS	86
EJERCICIO N°:5	92
PROCEDIMIENTO DE: PAGOS	92
LISTA DE EVENTOS.....	92
DIAGRAMA DE FLUJO DE DATOS	94
EJERCICIO N°:6	98
PROCEDIMIENTO: VENTAS EN CUENTA CORRIENTE	98
LISTA DE EVIENTOS	98
DIAGRAMA DE FLUJO DE DATOS	102
APÉNDICE B	108
EJERCICIO N°:1	109
1 PROCEDIMIENTO	109

2. OPERACIONES	109
3 ALGORITMOS	109
4 DISEÑO DE LOS ARCHIVOS	109
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	109
MÓDULOS	110
ARBOL DEL SISTEMA	113
EJERCICIO N°:2	114
1 PROCEDIMIENTO	114
2. OPERACIONES	114
3 ALGORITMOS	114
4 DISEÑO DE LOS ARCHIVOS	114
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	114
MÓDULOS	115
ÁRBOL DEL SISTEMA	117
EJERCICIO N°:3	118
1 PROCEDIMIENTO	118
2. OPERACIONES	118
3 ALGORITMOS	118
4 DISEÑO DE LOS ARCHIVOS	118
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	118
MÓDULOS	119
ÁRBOL DEL SISTEMA	124
EJERCICIO N°:4	125
1 PROCEDIMIENTO	125
2. OPERACIONES	125
3 ALGORITMOS	125
4 DISEÑO DE LOS ARCHIVOS	125
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	125
MÓDULOS	126
ÁRBOL DEL SISTEMA	129
EJERCICIO N°:5	130
1 PROCEDIMIENTO	130
2. OPERACIONES	130
3 ALGORITMOS	130
4 DISEÑO DE LOS ARCHIVOS	130
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	130
MÓDULOS	131
ÁRBOL DEL SISTEMA	135
EJERCICIO N°:6	136
1 PROCEDIMIENTO	136
2. OPERACIONES	136
3 ALGORITMOS	136
4 DISEÑO DE LOS ARCHIVOS	136
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	136
MÓDULOS	137
ÁRBOL DEL SISTEMA	142
EJERCICIO N°:7	144
1 PROCEDIMIENTO:	144
2. OPERACIONES	144
3 ALGORITMOS	144
4 DISEÑO DE LOS ARCHIVOS	144
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	144
MÓDULOS	145
ÁRBOL DEL SISTEMA	150
EJERCICIO N°:8	151

1 PROCEDIMIENTO:	151
2. OPERACIONES	151
3 ALGORITMOS	151
4 DISEÑO DE LOS ARCHIVOS	151
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	151
MÓDULOS	152
ÁRBOL DEL SISTEMA	160
EJERCICIO N°:9	161
1 PROCEDIMIENTO:	161
2. OPERACIONES	161
3 ALGORITMOS	161
4 DISEÑO DE LOS ARCHIVOS	161
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	161
MÓDULOS	162
ÁRBOL DEL SISTEMA	166
EJERCICIO N°:10	167
1 PROCEDIMIENTO:	167
2. OPERACIONES	167
3 ALGORITMOS	167
4 DISEÑO DE LOS ARCHIVOS	167
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	167
MÓDULOS	168
ÁRBOL DEL SISTEMA	174
EJERCICIO N°:11	175
1 PROCEDIMIENTO:	175
2. OPERACIONES	175
3 ALGORITMOS	175
4 DISEÑO DE LOS ARCHIVOS	175
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	176
MÓDULOS	176
ÁRBOL DEL SISTEMA	187
EJERCICIO N°:12	188
1 PROCEDIMIENTO:	188
2. OPERACIONES	188
3 ALGORITMOS	188
4 DISEÑO DE LOS ARCHIVOS	188
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	188
MÓDULOS	189
ÁRBOL DEL SISTEMA	195
EJERCICIO N°:13	196
1 PROCEDIMIENTO	196
2 OPERACIONES	196
3 DISEÑO DE LOS ARCHIVOS.....	196
4 DISEÑO DEL MODELO RELACIONAL DE DATOS	196
MÓDULOS	197
ÁRBOL DEL SISTEMA	200
EJERCICIO N°:14	201
1 PROCEDIMIENTO	201
2 OPERACIONES	201
3 DISEÑO DE LOS REGISTROS.....	201
4 DISEÑO DEL MODELO RELACIONAL DE DATOS	201
MÓDULOS	202
ÁRBOL DEL SISTEMA	204
EJERCICIO N°:15	205
1 PROCEDIMIENTO	205

2. OPERACIONES	205
3 ALGORITMOS	205
4 DISEÑO DE LOS ARCHIVOS	205
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	206
MÓDULOS	206
ÁRBOL DEL SISTEMA	209
EJERCICIO N°:16	210
1 PROCEDIMIENTO	210
2. OPERACIONES	210
3 ALGORITMOS	210
4 DISEÑO DE LOS ARCHIVOS	210
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	210
MÓDULOS	211
ÁRBOL DEL SISTEMA	215
EJERCICIO N°:17	216
1 PROCEDIMIENTO	216
2. OPERACIONES	216
3 ALGORITMOS	216
4 DISEÑO DE LOS ARCHIVOS	216
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	216
MÓDULOS	217
ÁRBOL DEL SISTEMA	222
EJERCICIO N°:18	223
1 PROCEDIMIENTO	223
2. OPERACIONES	223
3 ALGORITMOS	223
4 DISEÑO DE LOS ARCHIVOS	223
5 DISEÑO DEL MODELO RELACIONAL DE DATOS	223
MÓDULOS	224
ÁRBOL DEL SISTEMA	228
BIBLIOGRAFÍA.....	229

PROYECTO INFORMÁTICO

INTRODUCCIÓN

Los cambios tecnológicos, producidos en los últimos 10 años, le han permitido a los profesionales que actúan en el área de administración tener acceso a una gran cantidad de recursos informáticos y en la actualidad estos recursos informáticos son manejados por los propios usuarios. Pues así como la tecnología fue perfeccionando al automóvil hasta hacer innecesario al conductor profesional, la misma tecnología va perfeccionando la interfaz entre el **usuario y el computador**; de modo tal que los administradores al utilizar aplicaciones informáticas, para mejorar sus procedimientos de trabajo, ya no precisan de un especialista en programación.

Sin embargo, en la actualidad, se puede percibir que para aplicaciones simples y sin sofisticaciones, a ser desarrolladas por ejemplo con procesadores de textos y planillas de cálculo, los usuarios pueden realizarlas sin mayores dificultades; pero estos mismos usuarios, comienzan a tener serias dificultades cuando precisan aplicar recursos que apenas usan eventualmente, o cuando interactúan con sistemas que exigen un conocimiento más **conceptual** y específico de computación.

Es por ello que en **proyectos administrativos** desarrollados por **profesionales de la administración en pequeñas y medianas empresas**, con la utilización de **microcomputación**, el profesional se encuentra con una gran dificultad en la utilización de **metodologías** para la automatización de procedimientos administrativos. Especialmente debido: a las exigencias y al esfuerzo adicional que requiere la elaboración de **modelos**, a la gran cantidad de **documentación** necesaria y a la falta de dominio de **métodos**.

Este trabajo busca presentar una **metodología**, basada en **técnicas estructuradas simplificadas**, para el desarrollo de proyectos informáticos de pequeño y mediano porte, y que tengan como objetivo la automatización de tareas administrativas. Esta metodología se encuentra apoyada en el uso de herramientas de Ingeniería de Software Asistida por el Computador (**Computer Aided Software Engineering -CASE-**).

Las herramientas y la metodología, que aquí veremos, apoyan las etapas de análisis del sistema, análisis de los requerimientos de los usuarios y diseño del software, siendo adecuadas para proyectos que **no requieran** más de dos o tres hombres por mes, coordinados por un proyectista.

Con referencia a las **técnicas simplificadas**, cabe aclarar que el término simplicidad de ninguna manera puede confundirse o asociarse al de sencillo, tosco o burdo; una técnica simplificada significa que cuando un proyecto informático se cumple satisfactoriamente y para ello se han empleado en él: menor cantidad de personal, menor cantidad de recursos técnicos y además se ha realizado en un tiempo menor, resulta pues un **perfeccionamiento** evidente del acto informático.

Puede decirse entonces que en informática aplicada a administración, simplificación y perfeccionamiento son la misma cosa, con la inteligencia necesaria para cumplimentar el requisito fundamental de todo proyecto que es la solución de un problema, o al menos el mejoramiento de una situación dada.

En otras palabras, llegar al final por el camino más corto y sencillo significa, indudablemente, un progreso; y cuando ante cualquier procedimiento se deja en el ánimo del observador la impresión de que el mismo ha sido fácil, es sin duda alguna, por que se ha efectuado con: pulcritud, sencillez, tiempos bien reglados y con prescindencia de procedimientos redundantes y recursos inútiles; vale decir, se ha efectuado con una buena técnica.

Partiendo de estos razonamientos y hechos de observación, los conceptos de simplificación y orden riguroso, son inversos a los equipos de trabajo numerosos y al instrumentismo exagerado. Sin llegar a la torpeza de desarrollar un proyecto absolutamente solo, puede decirse que en la mayoría de los proyectos informáticos, aplicados a administración, se pueden realizar muy bien con un solo ayudante. El secreto estará en saber disponer de las herramientas necesarias, seleccionarlas según sea el problema a ser abordado, y desarrollar un buen plan para su uso.

Destaco tanto el aspecto de las técnicas simplificadas, pues pienso en las distintas alternativas que vivenciará un profesional; pues si al profesional le tocara desarrollar su actividad en un medio mal provisto, sentirá un gran alivio de poder hacerlo sin mayores recursos de personal y herramientas. Si, por el contrario, tuviera que desempeñar su profesión en un medio mejor dotado, tendrá tiempo para rodearse de comodidades; que no siempre redundarán en beneficio del usuario.

1 PROYECTO

Un proyecto es esencialmente un conjunto de **actividades** interrelacionadas, con un inicio y una finalización definida, que utiliza **recursos** limitados para lograr un **objetivo** deseado¹.

Los dos elementos básicos que incluye esta definición son: las actividades y los recursos.

1. **LAS ACTIVIDADES** son las tareas que deben ejecutarse para llegar en **conjunto** a un fin preestablecido (objetivo deseado); por ejemplo: recopilar información; realizar diagnósticos; confeccionar un diseño global de un procedimiento, programar, escribir manuales de procedimiento, etc. ([ver 6.1. metodología para el desarrollo de sistemas](#))

Un aspecto fundamental en todo proyecto es el orden en el cual se realizan las actividades. Y para determinar la **secuencia lógica** de las actividades se debe establecer el **método**, el **tiempo** y el **costo** de cada operación.

2. **LOS RECURSOS** son los **elementos** utilizados para poder realizar la ejecución de cada una de las tareas; como por ejemplo: hardware, programas de base (sistemas operativos), programas de aplicación, discos de almacenamiento, energía, servicios, inversiones de capital, personal, información, dinero y tiempo ([ver 2.3 consideraciones en un plan estratégico informático](#)).

Entonces: El fin primario de desarrollar un proyecto debe ser producir un **programa calendario** en el cual los **recursos**, siempre limitados, se asignen a cada una de las **actividades** en forma económicamente óptima.

Estas limitaciones en cuyo contexto se resuelve planear un proyecto pueden ser **internas**, por ejemplo: computadoras disponibles, capacidad del personal, disposiciones presupuestarias, o bien **externas**, como ser: fechas de entrega de cualquier tipo de recursos, factores climáticos, aprobaciones de organismos oficiales. En ambos casos las limitaciones deben tenerse particularmente en cuenta al estimar los tiempos de cada actividad.

En cuanto al objetivo del proyecto, este puede ser sencillo y no demandar ni muchas tareas ni demasiados recursos; o por el contrario, puede ser complejo y exigir múltiples actividades y una gran cantidad de recursos para poder alcanzarlo.

Pero independientemente de su complejidad, característicamente todo proyecto reúne la mayoría de los siguientes criterios²:

1. Tener un principio y un fin
2. Tener un calendario definido de ejecución
3. Plantearse de una sola vez
4. Constar de una sucesión de actividades o de fases
5. Agrupar personas en función de las necesidades específicas de cada actividad
6. Contar con los recursos necesarios para desenvolver las actividades

Ahora piense por un instante en cada uno de los proyectos que se desarrollan en las empresa, y verá que todos ellos tienen cometidos que deben cumplirse en un cierto plazo de tiempo y que además requieren de la concurrencia de otras personas.

Y es aquí donde empieza a tener relevancia la figura del administrador, en los proyectos a realizarse en las organizaciones; incluidos los proyectos informáticos.

Creo importante traer el pensamiento de Deming, quien señala y remarca muy claramente, al presentar su teoría de Calidad Total, que el administrador de un proyecto al planificar las actividades, debe tener presente que los mejores esfuerzos constituyen un elemento esencial;

pero desgraciadamente, si estos esfuerzos se toman aisladamente sin una debida orientación basada en principios administrativos, éstos esfuerzos pueden causar profundos daños.

La necesidad de la consistencia en los esfuerzos supone que si cada uno sabe lo que tiene que hacer y, que si cada uno hiciese lo mejor que puede, el resultado sería la dispersión del conocimiento y de los esfuerzos; por lo tanto, no hay nada que substituya al trabajo en equipo y a los buenos líderes, para alanzar una consistencia entre los esfuerzos y el conocimiento necesario.³

Algo importante a tener siempre presente es que: si el administrador realiza un buen trabajo en la gestión del proyecto, su éxito podrá ser visto y verificado por los demás; en caso contrario, naturalmente, el fracaso también estará a la vista de todo el mundo. La responsabilidad es muy alta: alcanzar el objetivo o no. Pero la oportunidad de “demostrar la capacidad profesional”, es de las que no pueden dejarse pasa por alto.

Los administradores **eficaces** de proyectos, son los que logran que el trabajo se ejecute a **tiempo**, dentro del **presupuesto**, y conforme a las **normas de calidad** especificadas⁴.

1.1. QUÉ ES UN PROYECTO INFORMÁTICO

De la definición de proyectos, vista en el punto anterior, podemos aplicarla a los proyectos informáticos; y decir que: un proyecto informático es un sistema de cursos de acción simultáneos y/o secuenciales que incluye personas, equipamientos de hardware, software y comunicaciones, enfocados en obtener uno o más resultados deseables sobre un sistema de información.⁵

En el capítulo 6, presentaremos cuáles son las tareas y en qué secuencia deben realizarse ([Metodología estructurada](#)) para alcanzar el objetivo. Ahora vamos a describir los distintos objetivos que caracterizan a un proyecto informático.

El inicio de un proyecto informático generalmente está dado en la solicitud de requerimientos de los usuarios, y siendo que los diferentes sistemas de Información abordan los diferentes tipos de problemas organizacionales; podemos clasificar a los Sistemas de Información según sean las aplicaciones que necesite cada usuario en: Sistemas de Transacciones, Sistemas de Soporte para la toma de decisiones, y Sistemas Expertos.

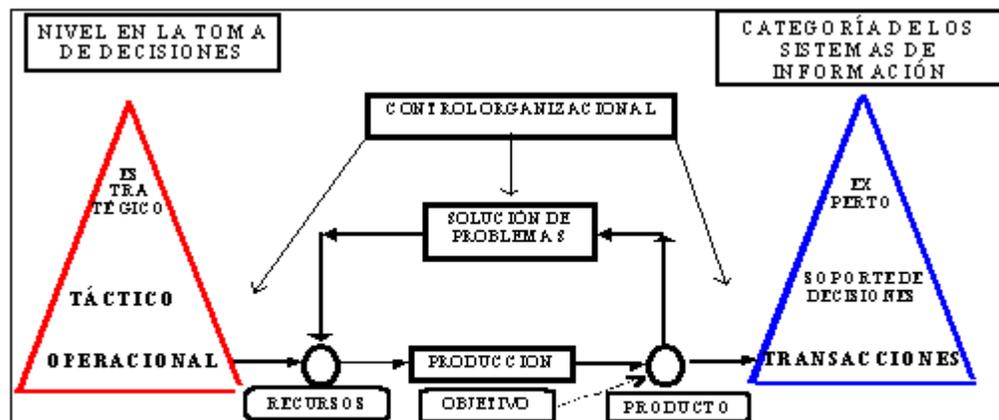


FIGURA 1.1 Clasificación de los sistemas de información

También una buena manera de abordar la estructura conceptual para los sistemas de información, es desde los trabajos de Gorry y Scott Morton⁶. En el que relacionan el trabajo fundamental de Herbert.A.Simon sobre la decisión estructurada y no estructurada, con el planeamiento estratégico, control gerencial y control operacional de Anthony.

Simon divide la toma de decisiones en tres fases: Inteligencia Diseño y Elección; en una decisión estructurada, las tres fases son totalmente inteligibles y computables por el decisor humano; por lo cual esa decisión es programable. En una decisión no estructurada no hay pleno entendimiento de una o algunas de esas fases.

Henry C. Lucas también toma a dichos trabajos, para poder diferenciar los distintos tipos de sistemas a fin de llegar a una implementación exitosa, de sistemas de información computarizados.⁷

La siguiente tabla muestra la estructura conceptual de Gorry y Scott Morton.⁸:

	<u>Control operacional</u>	<u>Control Gerencial</u>	<u>Planeamiento Estratégico</u>
<u>Estructurado</u>	Cuentas a Cobrar Ingreso de pedidos Control de Inventarios	Análisis presupuestario costos proyectados Presupuesto a corto plazo	Ubicación de depósitos y fabricas
<u>Semi-estructurado</u>	Programación de la Producción Administración de / fondos	Preparación del presupuesto Análisis de Varianza presupuesto total	Planeamiento de nuevos productos Fusiones y Absorciones
<u>No-estructurado</u>	Sistema PERT y - / de Costos	Ventas y Producción	Planeamiento de investigación y desarrollo

Los recursos mas frecuentemente utilizados que caracterizan a un sistema de información, son los componentes de la Tecnología de la Información (**TI**) como ser el uso de **Hardware, Software y Comunicaciones**.

En cuanto a estos elementos de la Tecnología de la Información, podemos considerar que ya han llegado a un desarrollo más que suficiente para la aplicación en una operación informática. Lo que nos lleva a que la gestión de un hecho informático como un **proyecto integral**, tanto sea en su entorno de diseño, como en su planificación y control, definen una nueva etapa; una mayoría de edad en el tratamiento informático.

Es así que hoy dada la evolución en la Tecnología de la Información, los proyectos de aplicación típicamente administrativos, desarrollados principalmente en pequeñas y medianas empresas, y que desarrollan su planeamiento informático basado en el uso de las microcomputadoras; puedan ser administrados por un único profesional⁹

Es por todo esto que, los conceptos de Proyecto y de Metodología de diseño que, hasta hace poco tiempo, eran solamente aplicados a grandes emprendimientos; hoy también deben ser aplicados a medianos y pequeños emprendimientos.

Considerando entonces, la importancia que la informática tiene en los planes estratégicos de cualquier empresa moderna; no solamente se debe tener en cuenta la evolución de los recursos de la tecnología de la información, sino también las distintas metodologías para el desarrollo de los sistemas de información.

Así es que, el solo hecho de considerar a un asunto informático como un proyecto al que se asocian técnicas y procedimientos de diseño, supone un paso importante.

1.2. INICIO DE UN PROYECTO NFORMÁTICO

Ya vimos una clasificación (ver figura 1), que nos permite clarificar el origen de un proyecto informático, pero ¿cómo podremos determinar la magnitud de un Proyecto informático?.

En un entorno informático estable, la decisión de iniciar un proyecto viene dada por las necesidades de: **mantenimiento, modificación, mejoramiento, reemplazo o capacidad**; encuadrándose así, el proyecto informático, dentro de una categoría de complejidad mostrada en la figura 1.2¹⁰:

El Mantenimiento del programa; es una consecuencia de una omisión realizada en la etapa del diseño del sistema ([ver 6.1. metodología para el desarrollo de sistemas](#)) e involucra

solucionar **fallas** menores del sistema, que obligará a la realización de **cambios en el programa**; como por ejemplo el descuido de no considerar que puedan ocurrir en el sistema, ciertas condiciones extraordinarias; como sería el caso de un aumento no previsto del 60 %, en la emisión de órdenes de compra. Las fallas también pueden provenir de otros factores, como ser en el caso de que existan cambios en las expectativas de los usuarios.

La Modificación del programa; involucra algo más que un simple cambio en el programa; involucra un cambio estructural de una entidad Por ejemplo, un cambio en el número de dígitos del código postal, o en el código de zona telefónica. La diferencia con el Mantenimiento es el grado de **importancia**

El Mejoramiento del sistema; es el agregado de **capacidades** que no formaron parte del sistema de información original; por ejemplo cuando en una división se implementó un sistema de inventarios, este sistema no incluía un módulo para calcular la futura demanda de bienes y partes. La inclusión de este sofisticado módulo de cálculo es considerado un mejoramiento del sistema.

El Reemplazo del sistema; ocurre cuando los sistemas de información se tornan físicamente, tecnológicamente o competitivamente **obsoletos**. Como es el caso de la utilización del láser, en el reconocimiento óptico de caracteres para la lectura del código de barras, reemplazando a la entrada por teclado.

La Nueva Capacidad del sistema; son sistemas de información para los cuales no es necesario el uso de la automatización. Están dados por la capacidad de poder **modelizar** la aplicabilidad de nuevos sistemas. Un ejemplo de ello, es la aplicación de los sistemas expertos.

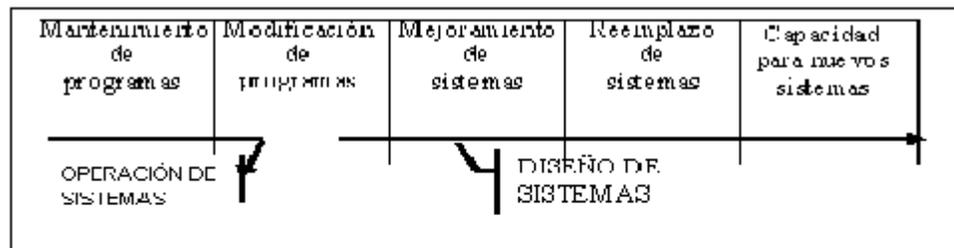


Figura 1.2 Categorías de los sistemas de información

2. PLANEAMIENTO DE UN PROYECTO INFORMÁTICO

Según Rusell Ackoff, la esencia de la sabiduría es la preocupación por el futuro; pero no es, la misma preocupación que tiene el adivino por el futuro, pues él solamente intenta preverlo; el sabio intenta controlarlo.¹¹

La planificación consiste en diseñar un futuro deseable y seleccionar o crear formas de lograrlo, hasta donde sea posible.

Por lo tanto, al planificar se construye la **secuencia** de tareas con la **lógica** necesaria, y la asignación de **recursos** necesarios para alcanzar el objetivo del proyecto en un **tiempo** óptimo.

La disponibilidad de recursos, hace que la secuencia de tareas pueda variar en el tiempo; dependiendo de los recursos con que se dispongan. Por lo tanto, al momento de planificar, hay que considerar, las tareas y los recursos; con el mismo grado de importancia. ([ver. 1.1 que es un proyecto informático](#)).

2.1. MÉTODOS DE PLANIFICACIÓN TEMPORAL DE TAREAS

La planificación temporal de un proyecto de software, no difiere mucho de la de cualquier otro esfuerzo de desarrollo multitarea. Además, se pueden utilizar las técnicas y herramientas generales de planificación temporal de proyectos para el desarrollo de software, con pequeñas modificaciones; entre ellas podemos citar a la técnica de Evaluación y Revisión de Programas, el método del Camino Crítico y al diagrama de Gantt¹².

La **Técnica de Evaluación y Revisión de Programas** (Program Evaluation and Review Technique-**PERT**) y el método del **Camino Crítico** (Critical Path Method-**CPM**) son dos **métodos de planificación temporal** de proyectos que pueden aplicarse al desarrollo de proyectos informático. Ambas técnicas desarrollan una **descripción de la red de tareas** del proyecto, es decir, una representación **gráfica o tabular** de las tareas que deben realizarse desde el principio hasta el final del proyecto.

En el método PERT/CPM se coordinan todos los elementos de un proyecto en un plan maestro, mediante la creación de un modelo lógico, para lograr el mejor tiempo y con el mínimo costo.

La red se define desarrollando una lista de todas las tareas asociadas con el proyecto específico, y una lista de secuenciamientos, que indica en qué orden deben realizarse las tareas.

Se estiman luego los tiempos correspondientes; y para ello se debe:

1. establecer, con la aplicación de modelos estadísticos, las estimaciones de tiempo, mas probables para cada una de las tareas;
2. luego se calculan los límites de tiempo que definen una amplitud temporal para cada tarea (teniendo en cuenta los recursos disponibles), y por último;
3. se halla el camino crítico, o sea el conjunto de actividades, que determina la duración total del proyecto y que sus atrasos o adelantos originarán atrasos o adelantos de iguales unidades de tiempo en la duración total del proyecto.

Una vez establecido el camino crítico, se lo utiliza para: considerar alternativas, elaborar la lógica del plan y precisar las estimaciones de tiempo de las actividades críticas, así como la influencia de limitaciones y las posibles soluciones de situaciones conflictivas

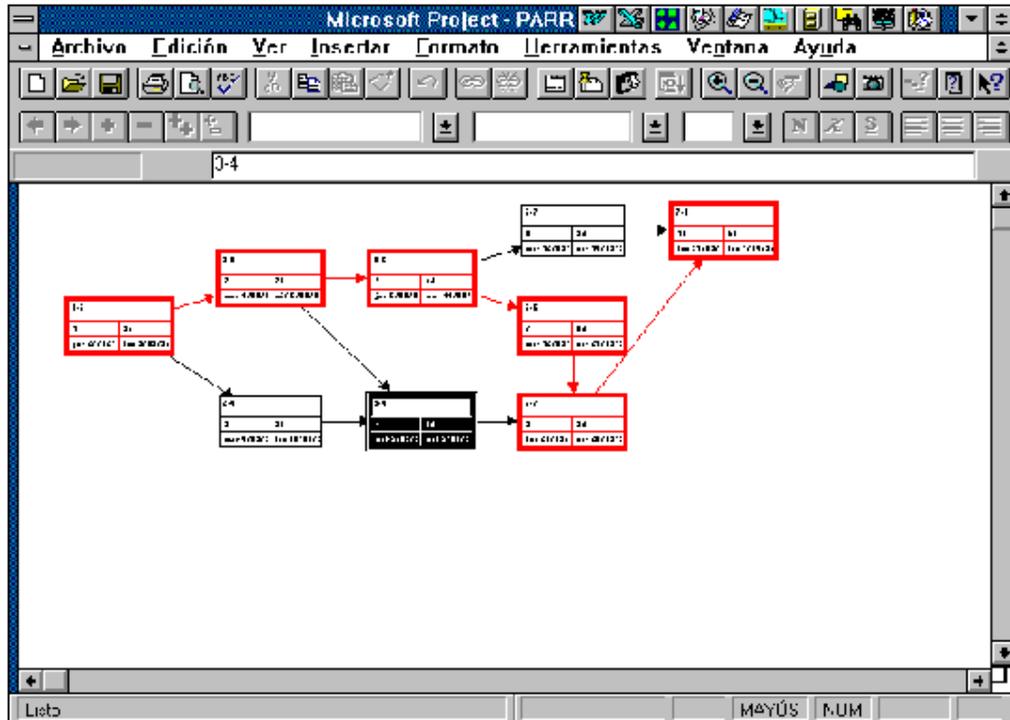


FIGURA 2.1. PERT Y CPM

Otra herramienta de diseño es el **Diagrama de Gantt**¹³; ésta es una representación gráfica cronológica, de las etapas componentes de un proyecto. Este gráfico se sustenta en una estructura de barras horizontales, en las cuales la longitud es directamente proporcional al tiempo requerido para su ejecución. El objetivo de este gráfico es el de planear un proyecto y verificar el cumplimiento.

A los efectos de su confección, se requiere determinar.

- a) Las tareas a desarrollar
- b) La relación o dependencia entre las tareas
- c) El tiempo Planeado para la ejecución de cada tarea

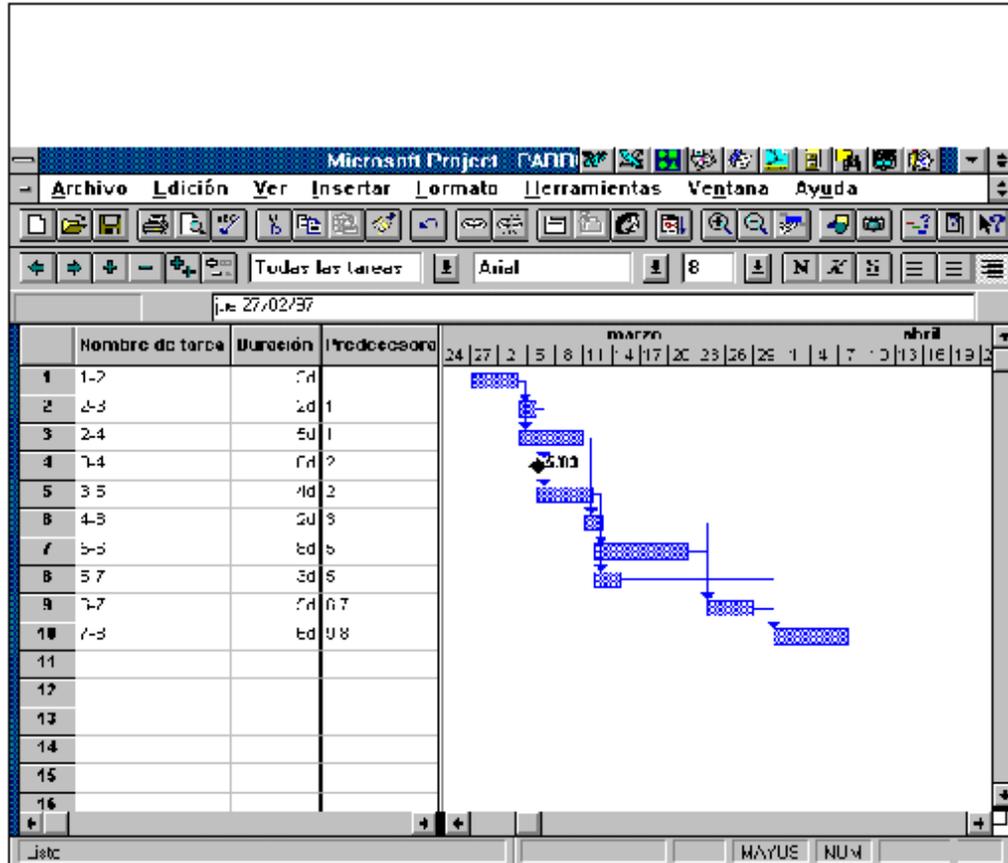


FIGURA2.2 Diagrama de GANTT.

La utilización de una herramienta automatizada de administración de proyectos, como es el caso de Microsoft Project, le otorgará una mayor eficacia en el control del proyecto; también le permitirá mantener una mejor comunicación entre los participantes del proyecto.

2.2. MÉTODOS PARA PLANIFICACIÓN DE RECURSOS

La planificación de recursos pretende determinar qué recursos serán necesarios, cuándo, cómo y dónde se obtendrán los que no están disponibles y en qué forma serán generados o adquiridos.

Se debe tener en cuenta cinco tipos de recursos:

1.  Los insumos (materiales, piezas, energía y servicios);
2.  Las instalaciones y equipo (inversiones de capital);
3.  El personal;
4.  La información;
5.  El dinero.

La herramienta principal para la planificación de recursos es el presupuesto; y éste se compone de la asignación de responsabilidades para generar y utilizar el dinero, y del calendario para hacerlo.

PLANIFICACIÓN FINANCIERA

Vimos que un proyecto involucra tareas y recursos; por lo tanto, en la planificación son tan importantes las tareas como los recursos disponibles.

Al momento de asignar los recursos, debe tener en cuenta algunas consideraciones como: la simultaneidad de tareas para un mismo recurso, la importancia de cada tarea, si es una actividad crítica o no.

Lo importante es que una vez que fueron identificados los recursos para cada tarea, se deben realizar los siguientes **análisis**:

1. De Costo;
2. De Beneficio;
3. De Riesgo;
4. De Sensibilidad.

Es importante considerar que la utilidad de los modelos financieros, aumenta cuando se los computariza. Esto facilitará una exploración financiera rápida, y de una gran cantidad de medios alternativos y/o supuestos sobre el ambiente. A través de los análisis de riesgo y sensibilidad.¹⁴ dichas exploraciones alcanzarán un gran valor en el proceso de planificación

Entre tantas condiciones comerciales, en la que se puede estimar la sensibilidad, podemos citar:

1. La tasa de interés bancaria;
2. El costo del dinero accionario;
3. El índice de inflación.

ALTERNATIVA A-1 COMPRAR EL EQUIPO						
CONCEPTO	0	1	2	3	4	5
COMPRA E INSTALACION DE LA MAQUINA	(1000)					
COMPRA E INSTALACION DE ACCESORIOS	(200)					
VALOR RECUPERACION					2400	
AMORTIZACION	(2400)	(2400)	(2400)	(2400)	(2400)	(2400)
GROS DE SERVICIOS DE MANTENIMIENTO		(1000)	(1000)	(1000)	(1000)	(1000)
MONTO IMPONIBLE	(2400)	(3400)	(3400)	(3400)	(3400)	(1000)
IMPUESTOS		1080	1530	1530	1530	450
FLUIR FONDOS NETOS	(12000)	80	630	630	2930	460
VAN =		2265				
TIR =		0.29				

FIGURA2.3. ANÁLISIS DE FLUJO DE FONDOS

2.3. CONSIDERACIONES EN UN PLAN ESTRATÉGICO INFORMÁTICO

Bien, nuevamente concentrando nuestra atención en los proyectos informáticos. Tenemos que en el proceso de planeamiento, de un sistema de información, se debe determinar:

1. La situación actual 📄
2. La situación deseada 🖨
3. Las distintas alternativas 🖨📄🔄

También se deben considerar, los **recursos** necesarios específicos de la **Tecnología de la Información**:

❖ **Físicos**

- Sistema Central (Microprocesador, Memoria principal)
- Periféricos (Unidades de entrada, Unidades de salida; Unidades de entrada/salida)
- Comunicaciones (Modem, Repetidores, Hub)

❖ **lógicos**

- Estructuras de almacenamiento (Base de datos relacional, orientada a objetos)
- Monitores de comunicaciones
- Lenguajes (Pascal, Cobol, C++, SQL)
- Métodos de desarrollo (Ciclo de Vida, Prototipo, Espiral)
- Control de seguridad y calidad

❖ **humanos**

- Selección
- Formación
- Incentivos

3. BASE DE DATOS

El conjunto unificado de información, resultante de nuestro proyecto informático y, que será compartida por los diferentes usuarios de la organización, va a conformar la denominada Base de Datos.

La función básica de una base de datos es permitir el almacenamiento y la recuperación de la información necesaria, para que las personas de la organización puedan tomar decisiones. Es así que las Bases de Datos se tornan esenciales para la supervivencia de cualquier organización; pues los datos estructurados constituyen un recurso básico para todas las organizaciones.

Dependiendo de la capacidad de almacenamiento y procesamiento del hardware, la organización puede contar con una única Base de Datos, o con múltiples Bases de Datos.

Es común que en las pequeñas y medianas empresas se cuente con microcomputadoras, y por ello tengan que distribuir su información en un conjunto de Bases de Datos; asignándole a cada una de ellas, información sobre cada área específica de la empresa. Un ejemplo sería el de contar con una base de datos para el almacenamiento de la información correspondiente al área financiera, otra para el área de personal, una más para el área de ventas o el área de producción.

Mientras tanto las Grandes organizaciones poseen computadoras de gran porte, y es así que pueden almacenar toda la información necesaria, integrada, consistente y consolidada, en una única base de datos.

Independientemente de la Base de Datos que será implementada, ésta necesita de un **Sistema de Gestión de Base de Datos (SGBD o DBMS)**. Los sistemas de Gestión de Base de datos, son programas de software para la administración de las Bases de Datos; y en particular, para: almacenar, manipular y recuperar datos en una computadora. El SGBD también se encargará de la comunicación entre el usuario y la base de datos, proporcionándole al usuario, los medios necesarios para poder obtener información, introducir nuevos datos y actualizar los ya existentes.

3.1. ESTRUCTURA DE UNA BASE DE DATOS.

Una Base de Datos está compuesta por un conjunto de **tablas** o **archivos**. Para una mayor comprensión podemos ejemplificar la siguiente **Base de Datos de compras**.

ARCHIVO DE PRODUCTOS

Código artículo	Descripción del material	Unidad	Cantidad
1.01.01	CD-ROM RW IDE	Unidad	10
1.01.02	Disco rígido ATA 66	Unidad	20
1.02.01	Disco Flexible de 3 1/2" 1,44 Mbytes	Caja de 10	20
2.01.01	Sonido de 16 bit	Unidad	5
3.01.01	Papel carta para impresora.	Resma 100 hojas	25
4.01.01	Pentium II 200Mhz	Unidad	7
4.01.02	Pentium III 500Mhz	Unidad	8
4.01.03	Pentium III 800Mhz	Unidad	9

ARCHIVO DE PROVEEDORES

Código proveedor	Nombre del proveedor	Teléfono del proveedor	Dirección del proveedor
001	Inca Tel	4923-4803	Av. La Plata 365
002	Infocad	4633-2520	Doblas 1578
003	Herrera Compusistem	4232-7711	Av. Rivadavia 3558

ARCHIVO DE ORIGEN DE LOS PRODUCTOS¹⁵

Código proveedor	Código del artículo	Precio
001	1.01.01	70,00
002	1.01.01	80,00
003	1.01.01	75,00
002	2.01.01	50
001	4.01.03	450

Esta Base de Datos contiene información de tres **Entidades**:

- Datos sobre productos (Entidad producto), almacenados en el archivo de **PRODUCTOS**;
- Datos sobre proveedores (Entidad proveedores), almacenados en el archivo **PROVEEDORES** y;

- Datos sobre el origen de los productos (Entidad origen del producto), o sea, los productos son provistos por cada proveedor y viceversa, almacenados en el archivo de **ORIGEN DEL PRODUCTO**.

La información almacenada en cada uno de estos archivos se conoce con el nombre de **Entidad**. Por lo tanto una entidad es cualquier persona, cosa o evento, real o imaginario, de interés para la organización y acerca del cual se capturan, almacenan o procesan datos.

Además, cada uno de estos archivos está formado por un conjunto de registros que describe, a través de los **atributos** o datos (columna), cada entidad en él almacenado. Un atributo es pues, cualquier detalle que sirve para identificar, clasificar, cuantificar o expresar el estado de una entidad.

Todos los **registros** de un archivo, identificados por las filas de cada tabla, poseen el mismo formato, o sea tienen el mismo conjunto de datos o atributos, identificados por las columnas, que describen a las entidades.

En otras palabras los registros están formados por un conjunto de datos almacenados en los campos de cada atributo; y cada registro debe contener el conjunto de atributos necesarios, para describir completamente cada entidad sobre la cual una organización necesita almacenar y obtener información.

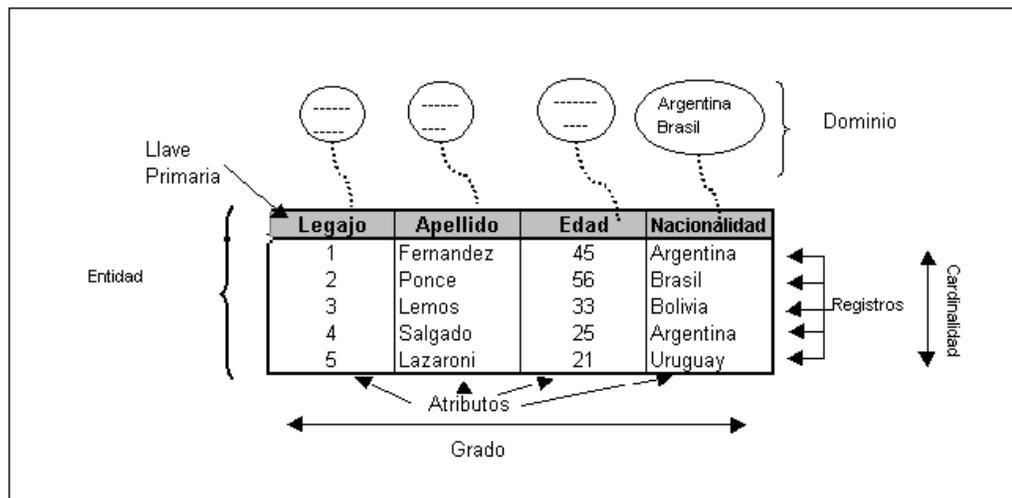


FIGURA 3.1 Modelo relacional de una tabla

3.1.1.TIPOS DE ARCHIVO

Los archivos pueden clasificarse en cuatro tipos básicos; que son: los **archivos maestros**, los **archivos de transacciones**, los **archivos de control** y los **archivos de planeamiento**. Esta clasificación dependerá de la relación lógica que tengan que tener los datos, para dar apoyo a la actividad de la organización.

ARCHIVO MAESTRO

Un archivo maestro es un conjunto de registros que se refieren a algún aspecto importante de las actividades de una organización, como por ejemplo el archivo de VENDEDORES. Un archivo maestro también puede reflejar la historia de los eventos que afectan a una entidad determinada, como es en el caso de un archivo HISTÓRICO DE VENTAS. Otros ejemplos son los archivos maestros de: PLAN DE CUENTAS; BANCOS, NÓMINA DEL PERSONAL, CLIENTES, VENDEDORES, PRODUCTOS, PROVEEDORES, COMPETIDORES.

ARCHIVO DE TRANSACCIONES.

Un archivo de transacciones es un archivo temporal que persigue básicamente dos propósitos; uno es el de acumular datos de eventos en el momento que ocurran, y el segundo propósito es el de actualizar los archivos maestros para reflejar los resultados de las transacciones actuales. En otras palabras, guardan información sobre los eventos que afectan a la organización y sobre los cuales se calculan datos; como es en el caso de los archivos de VENTAS, ORDENES DE PRODUCCIÓN o PAGO DE SALARIOS. Otros ejemplos de archivos de transacciones son los archivos de: REGISTROS CONTABLES, COSTOS, FACTURAS, PAGOS A RECIBIR, PROCESOS DE EXPORTACIÓN, CONSULTA DE CLIENTES, PEDIDOS DE CLIENTES Y PEDIDOS A PROVEEDORES.

ARCHIVOS DE CONTROL.

Los archivos de control contienen datos de los archivos maestros y de transacciones, para permitir el análisis del desempeño de la organización. Estos archivos generan medidas de control de los negocios, como ser el VOLUMEN DE VENTA POR PRODUCTO, VOLUMEN DE VENTA POR VENDEDOR, VOLUMEN DE VENTA POR CLIENTE, COMPRAS POR PROVEEDOR, COSTO DE REPOSICIÓN.

ARCHIVO DE PLANEAMIENTO.

Los archivos de planeamiento, contienen datos referentes a los niveles esperados de los datos existentes en los archivos maestros y de transacciones; como por ejemplo: PROGRAMA DE VENTAS, PROGRAMA DE COMPRAS, PROGRAMA DE PRODUCCIÓN; PRESUPUESTO FINANCIERO. Por lo tanto los datos existentes en un archivo de planeamiento provienen de los archivos maestros, de transacciones, y de control.

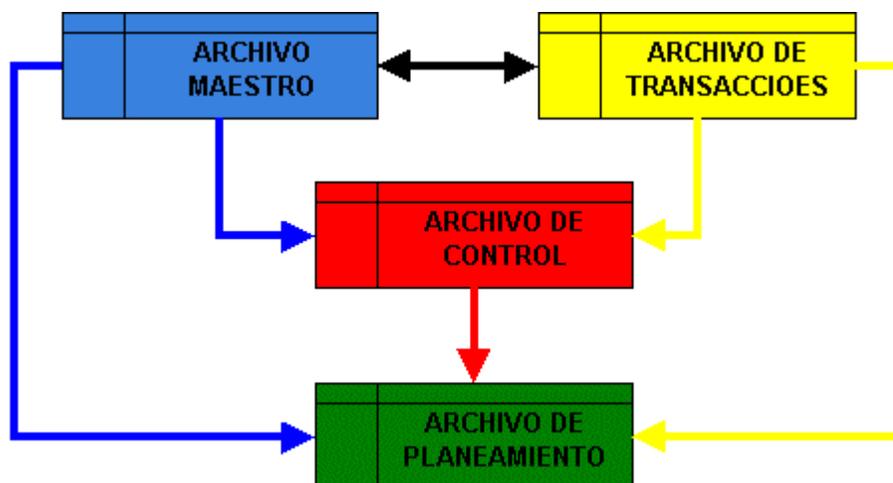


Figura 3.1.1. Flujo de información entre los distintos tipos de archivos

3.1.2.LLAVE PRIMARIA O IDENTIFICADORA.

Cada instancia de una entidad debe ser unívocamente identificable, de manera tal que cada registro de la entidad debe estar separado y ser unívocamente identificable del resto de los registros de esa misma entidad; y quien permite esta identificación es la **llave primaria**. La llave primaria, que generalmente se identificada por medio de la letra @, puede ser un atributo o una combinación de atributos.

En consecuencia en cada archivo solo podrá existir un único registro que posea un valor determinado para su llave primaria. En otras palabras no puede existir en un archivo un registro que cuente con el mismo valor de otro registro en el campo de la llave primaria; la llave primaria no puede tener valores repetidos para distintos registros.

La llave primaria debe permitirle a un Sistema de Gestión de Base de Datos (SGBD), correctamente proyectado, generar un error si un usuario intenta incluir un nuevo registro cuya llave primaria coincida con la de otro registro ya existente en el archivo.

En el caso de la Base de Datos de compras, descrita anteriormente ([ver 3.1.Estructura de una Base de datos](#)), las llaves primarias de cada archivo son:

- ARCHIVO DE PRODUCTOS: @ Código artículo
- ARCHIVO DE PROVEEDORES: @ Código proveedor
- ARCHIVO ORIGEN DE LOS PRODUCTOS: @(Código proveedor + Código producto).

3.1.3.INDICES DE ACCESO

Un índice de acceso es un archivo auxiliar utilizado internamente por el SGDB para acceder directamente a cada registro del archivo de datos. La operación de indexación, creada por el SGDB, ordena a los registros de un archivo de datos de acuerdo con los campos utilizados como llave primaria e, incrementa sensiblemente la velocidad de ejecución de algunas operaciones sobre el archivo de datos. Normalmente para cada archivo de datos debe existir un índice cuya llave de indexación sea idéntica a su llave primaria. Este índice es llamado **índice primario**.

También es posible crear índices para un archivo de datos utilizando atributos (campos), o conjunto de atributos, diferentes de los de la llave primaria. Este tipo de índice, **llamado índice secundario**, es utilizado para reducir el tiempo de localización de una determinada información

dentro de un archivo o para clasificar los registros del archivo de acuerdo con el orden necesario para la obtención de la información deseada.

4. HERRAMIENTAS DE ANÁLISIS

4.1. MODELOS CONCEPTUALES

Un modelo es una descripción capaz de ser comunicada y que busca:

- Comunicar un cierto aspecto (visión)
- De una parte de la realidad (sistema)
- Con cierto grado de detalle (abstracción)
- Conforme perseguido por alguien (autor del modelo)
- Con el objetivo de servir a los propósitos del usuario.¹⁶

Sowa¹⁷ Argumenta que el conocimiento sobre alguna cosa es la habilidad de formar un modelo mental que represente esta cosa, como así también las acciones que ella puede realizar o se pueden realizar sobre ella. Cuando el individuo verifica acciones sobre este modelo él puede predecir las implicaciones que estas acciones tendrán sobre el mundo real.

Según Sowa, al relacionar las cosas entre sí y al pensar de forma estructurada sobre ellas, podremos describir el funcionamiento de un sistema, y esto debería ser el propósito de todo modelo.

Los modelos pueden tener diferentes clases de estructuras; pero las clases más comunes son: la verbal, la simbólica y la matemática.

En los modelos verbales, las variables y sus relaciones se funden en forma de prosa. El manual de procedimientos, el manual de organización o la **Lista de eventos**, que describiremos próximamente ([ver 4.2.1. la modelización de las funciones del sistema](#)), son ejemplos de modelos verbales.

Los modelos simbólicos generalmente son más específicos que los verbales; Ellos representan un puente útil en el proceso de simbolizar un modelo verbal; por ejemplo, sería muy conveniente que en un manual de organización se incluya un organigrama (esquema para modelizar la estructura de la empresa). La mayoría de los modelos simbólicos se usan para aislar variables y sugerir las direcciones de las relaciones, como lo veremos mas adelante al describir los **Diagramas De Flujo de Datos** y el **Modelo Relacional de Datos**, pero pocos se diseñan para dar resultados numéricos específicos. El mayor beneficio de los modelos simbólicos está en la representación gráfica de los hechos a través de cuadros o nodos; y es así que el fenómeno se despoja de lo que no es esencial, permitiendo al investigador (observador) entender el conjunto y seleccionar las relaciones a examinar.¹⁸¹⁹

Algunos modelos pueden combinar componentes icónicos y análogos; como por ejemplo los **flujogramas** ([ver 4.5. flujogramas](#)), dichos diagramas por lo general tienen carácter cualitativo pero pueden convertirse en modelos simbólicos cuantitativos muy exactos.

Un punto muy importante de los modelos es el de saber como probarlos, a fin de determinar su validez; y estos tienen básicamente dos formas de ser probados, una es la forma prospectiva (contra el desempeño futuro), y la otra es de forma retrospectiva (contra el desempeño pasado); en éste último caso, o sea si un modelo se prueba retrospectivamente, es de vital importancia que los periodos utilizados cubran las situaciones que tal vez se encurte en el futuro.

Cuando un modelo no se puede probar en forma prospectiva ni en forma retrospectiva, el análisis de su sensibilidad al error puede servir de base para evaluarlo. Dicho análisis consiste en determinar cuánto tienen que bajar los valores de las variables del modelo para que los medios mejores especificados en dicho modelo tengan un desempeño inferior al de un medio

alternativo. Después, utilizando el juicio sobre la posibilidad de esta baja, se puede hacer una evaluación parcial del modelo.

Además de su utilidad para evaluar medios; los modelos se pueden utilizar heurísticamente, es decir, para facilitar el descubrimiento. Con frecuencia son un medio efectivo para explorar la estructura asumida de una situación determinada, y para descubrir posibles cursos de acción que de otra manera se pasarían por alto²⁰.

4.2. LA MODELIZACIÓN DE LAS FUNCIONES DEL SISTEMA

4.2.1. LISTA DE EVENTOS.

Las primeras actividades de diseño de los sistemas (ver cap1.1 [que es un PI](#) y 1.2 [inicio de un PI](#)), están especialmente influenciadas por la naturaleza de los requerimientos y éstos incluyen principalmente descripciones en lenguaje natural, que según lo visto en el tópico anterior (4.1); representan una realidad dada e interpretada de diferentes maneras según sea la visión y la capacidad de abstracción, de cada uno de los participantes del proyecto.

En el caso de que los requerimientos, fuesen realizados en forma oral o escrita en lenguaje natural, es indispensable realizar un análisis profundo del texto para poder entender en detalle el o los significados de todos los términos involucrados en el proyecto (libres de contradicciones e incongruencias). Luego esta lista estructurada, en el diseño inicial, será la base para la construcción de las entidades y sus relaciones; y que estarán representadas en los diagramas de flujo de datos y en el modelo relacional de datos²¹.

TÉCNICA PARA EL DISEÑO DE UNA LISTA DE EVENTOS

A continuación presentamos una lista de reglas empíricas que ayudarán a la construcción, en forma estructurada, de la lista de eventos.

- **Elegir el nivel apropiado de abstracción para los términos.**

Se debe preferir, la utilización de, palabras concretas a palabras abstractas. Las palabras concretas se refieren a objetos o sujetos tangibles; el lector las puede descifrar fácilmente, porque se hace una clara imagen de ellas asociándolas a la realidad.

En cambio, las palabras abstractas designan conceptos o cualidades más difusos, y suelen abarcar un número mayor de acepciones. El lector necesita más tiempo y esfuerzo para captar su sentido, pues no hay referentes reales.

Por lo tanto es muy importante el escoger la acepción más apropiada, entre las diversas alternativas posibles²².

Por ejemplo veamos los siguientes términos:

El gerente del área de finanzas, es quien autoriza las compras.

Es una oración demasiado ambigua. Su principal dificultad reside en el significado de *compras*. Al tratarse de una palabra bastante genérica, entran en juego muchas acepciones

Compras se refiere a:

Si se considera en función del tiempo, se refiere a: ¿compras programadas, no programadas o ambas?. Si se evalúa en función del volumen, se refiere a: ¿grandes pedidos, a pedidos pequeños o ambos?. En función de su origen, involucra a: ¿las importaciones o las de plaza local?. Y en función del bien: ¿en insumos y/o bienes de capital?.

¿Cuál de estos términos es el correcto?; si el resto del texto no ofrece la información necesaria para sobre la alternativa correcta; solo queda la alternativa de hacer una hipótesis

de significado genérica. Lo que significa asumir un riesgo, que obviamente no debería existir.

- **Evitar el uso de casos en lugar de conceptos generales.**

Es común observar que los usuarios de los sistemas de información, adoptan términos más específicos de los que verdaderamente son necesarios. Por ejemplo, el encargado de almacenes dice: "necesito conocer a diario la cantidad en existencia de pastillas de frenos", El término pastillas de frenos no describe un concepto, sino una instancia o componente del concepto correcto, esto es, un componente. Por lo tanto el término debería ser insumos.

- **Evitar las expresiones vagas o indirectas.**

Al usar rodeos, se incurre en el riesgo de expresar el significado de los conceptos en términos de referencias implícitas a otros conceptos, en lugar de referencias explícitas a los mismos conceptos. Por ejemplo cuando se dice: "mirá el repuesto en la cajonera", en vez de decir; "mirá las cajoneras". La segunda oración indica una clase específica de entidad (cajonera), mientras que la primera se refiere a la misma clase indicando una interrelación con otra clase de entidad (repuesto). Es así que la segunda oración, "mirá las cajoneras", permite una clara clasificación de los conceptos.

- **Elegir un estilo estandarizado de enunciado.**

Lo que se busca con un modelo sintáctico es lograr una comunicación buena y eficaz. Idealmente, se debe buscar elaborar enunciados que respondan a algún estilo estándar, en el caso de las descripciones de los datos, éstas deben ser frases afirmativas, compuestas por hasta cuatro elementos-llave, que son el <suje>, el <verbo>, el <objeto> y el <complemento>, que pueden ser el instrumento o el modificador. Estos elementos-llave pueden estar acompañados de otras palabras como artículos, adjetivos, etc.; por ejemplo²³²⁴:

El encargado del sector ALMACENES verifica el **PARTE DE RECEPCIÓN** con la **SOLICITUD DE COMPRA**

Generará la siguiente estructura-llave:

ALMACENES verifica **PARTE DE RECEPCIÓN** con **SOLICITUD DE COMPRA**

Donde ALMACENES es el sujeto, verifica es el verbo, **PARTE DE RECEPCIÓN** es el objeto y **SOLICITUD DE COMPRA** es el instrumento.

Considere que una frase puede estar incompleta; Por ejemplo:

ALMACENES emite **SOLICITUD DE COMPRA**

En ella no hay complemento.

También es importante que los enunciados que describen operaciones deben utilizar, tanto como les sea posible, estructuras sintácticas no ambiguas (PRODUCTOS, en LISTA DE PRODUCTOS o en STOCK), similares a las de los lenguajes de programación, como **si, condición, entonces, sino, cuando, hacer, acción**. Por ejemplo: Si el monto es menor a 100 aprueba el pedido, sino eleva el pedido a Gerencia Financiera.

- **Verificar los sinónimos y los homónimos.**

Distintas personas pueden dar el mismo significado a diferentes cosas (sinónimo) o diferentes significados con las mismas palabras (homónimos). En un procedimiento de ventas pueden encontrarse los siguientes términos: Cliente, comprador, usuario, parroquiano, y referirse al mismo concepto (sinónimos) En el caso de que el mismo término sea utilizado, en diferentes lugares, con significados diferentes es considerado pues un homónimo. Por ejemplo Para finanzas el cliente es quien compra un producto, mientras que para Marketing el cliente, o potencial cliente, es el usuario del producto.

- **Hacer explícitas las referencias entre términos.**

Se debe evitar cometer ambigüedades²⁵; es decir: frases que puedan interpretarse de dos o más maneras distintas. Algunas ambigüedades surgen al no especificar las referencias entre los términos. La ambigüedad puede provocar o un doble sentido o una incertidumbre.

En el caso de: Recepción firma remito.

Cuál remito firma, el original o alguna copia.

O por ejemplo: El jefe de compras se reúne con cada uno de los proveedores en su despacho.

En qué despacho se reúnen, en el de compras o en el de los proveedores.

O en el caso particular de nuestros archivos, si contamos con dos archivos PRODUCTO Y STOCK y ambos cuentan con los mismos atributos: Código del producto y Nombre del producto y, STOCK se diferencia por contar además con el atributo Saldo del producto; Lo que ocurre es que, probablemente no sean dos entidades distintas sino una sola entidad: PRODUCTOS EN STOCK y que debería contener a los atributos de ambas ([ver 4.4. diseño de relación uno a uno](#)).

- **Hacer un Diccionario de Datos.**

Como veremos más adelante ([ver 4.3. el diccionario de datos](#)), ir confeccionando el diccionario de datos, es una buena manera de entender el significado de los términos y de eliminar las ambigüedades de los requerimientos. Aunque, la confección del diccionario de datos, demande bastante tiempo es fundamental su elaboración y dejar de lado esta herramienta, no se justifica en ningún caso. Recuerde que puede utilizar cualquier herramienta de ingeniería de software para su construcción.

4.2.2. EL DIAGRAMA DE FLUJO DE DATOS

El **Diagrama de Flujo de Datos (DFD)** es una herramienta de modelización que permite describir, de un sistema, la transformación de entradas en salidas; el DFD también es conocido con el nombre de Modelo de Procesos de Negocios (**BPM, Business Process Model**)²⁶.

El objetivo del DFD es:

1. Describir el contexto del sistema, determinando lo que ocurrirá en cada una de las áreas de la empresa, denominadas Entidades externas, que participen de este sistema;
2. Detallar los procesos a ser realizados;
3. Enumerar los archivos de datos necesarios, en cada proceso;
4. Definir los flujos de datos, que participen en el procedimiento.

En otras palabras, el DFD permite representar de forma completa el sistema de información, al relacionar los datos almacenados en los archivos de datos del sistema, con los procesos que transforman a estos datos.

Una de las principales características de este modelo es su simplicidad, y se debe al hecho que son solamente cuatro los símbolos utilizados que representan a los elementos (entidades externas, archivos, procesos y flujos de información); con los cuales se puede producir un esquema, que alcance el nivel de detalle requerido por el proyectista; y éste pueda ser interpretado por todas las personas involucradas en el proyecto, sin el requerimiento de un conocimiento previo de informática.²⁷

TÉCNICA DE DISEÑO DEL DFD

En el diseño de un **DFD**, como ya lo dijimos anteriormente, son utilizados cuatro símbolos :

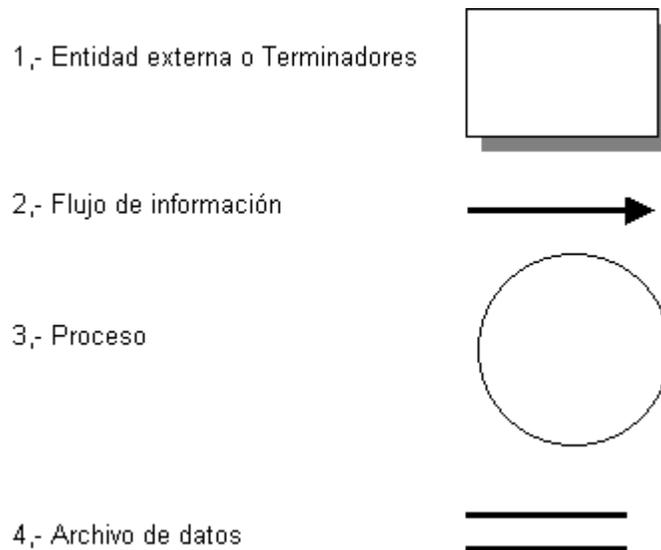


Figura 4.2.2. Simbología del DFD Metodo Yourdon

1. Las **Entidades externas**, que pueden representar a una persona, a un grupo de personas o, a un sistema; Un ejemplo respectivo para cada uno de ellos sería Gerente Financiero, Clientes y un sistema de liquidación de sueldos y jornales.

En sí, las entidades externas, muestran a las entidades con las cuales el sistema se comunica y por lo tanto no forman parte del sistema en estudios; pues lo que ocurre en estas entidades no es de interés para el proyecto. Si así lo fuera, esto está indicando que la frontera del sistema, es más amplia de lo que se determinó; y los procesos involucrados en esta entidad, deben pasar a ser parte del sistema en estudio.

Las entidades externas son consideradas también como **Terminadores**, pues representan el origen y el destino de los **Flujos de datos** para adentro y para fuera del sistema.

Son representadas por medio de un cuadrado, que puede tener un sombreado en dos de sus lados para otorgarle un relieve (ver figura 4.2.2). Y en el centro del cuadrado se escribe el nombre de la entidad externa que está siendo representada.

Cuando una entidad externa provee datos al sistema, debe existir un flujo de datos saliendo de la entidad y en dirección al sistema. Y cuando una entidad externa recibe datos del sistema, debe existir un flujo de datos que viene del sistema y termina en la entidad externa.

Las entidades externa pueden duplicarse, si fuese necesario darle claridad al diseño y evitar largos vectores, que representan a los flujos de datos, o bien evitar gran cantidad de entrecruzamientos de los mismos.

2.-Los **flujos de datos** son representados por vectores direccionados. Ellos son las **conexiones** entre los distintos elementos del sistema y los procesos; y representan a la información que los procesos exigen como entrada y/o las informaciones que ellos generan como salida. Los flujos pueden representar a una información compuesta por un solo elemento como por ejemplo: precio, cantidad, Apellido; o bien pueden representar a una información que contiene una estructura de elementos como por ejemplo: Orden de compra, Remito, Factura.

3.- Los **procesos** se pueden mostrar como burbujas, o como rectángulos con sus vértices redondeados; según sea la metodología para modelar los procesos de Yourdon o la de Gane & Sarson; en el diagrama ellos representan las diversas **funciones individuales** que el sistema ejecuta; Estas funciones son las que transforman a las entradas en salidas. El proceso es nominado en función de la acción que realiza sin especificar el algoritmo utilizado para la transformación. Este algoritmo debe ser detallado en el diccionario de datos ([ver 4.3. Diccionario de datos](#)) o esquematizado en un flujograma ([ver 4.5. flujograma](#))

4.- Los **archivos de datos** son mostrados por dos líneas paralelas según la metodología de Yourdon.; o como un rectángulo abierto por uno de sus lados en la metodología de Gane & Sarson. Ellos muestran la **colección de datos** que el sistema debe mantener en la memoria en un período de tiempo. Al terminar el diseño del sistema y la construcción del mismo, los archivos serán las tablas que compongan la base de datos.

RESTRICCIONES DEL DFD.

Como regla general, en un DFD, lo tratamiento de errores y de excepciones no deben ser representados; a menos que estos sean muy relevantes para los usuarios del sistema. El DFD debe ser visto como una herramienta de planeamiento del sistema, y no como una especificación detallada del sistema. Su finalidad es mostrar el flujo normal de datos entre los principales elementos, y no los detalles de implantación del sistema.²⁸

Lo que queremos decir es que, el diagrama de flujo de datos ofrece una visión general y práctica de los principales componentes funcionales del sistema, pero **no provee detalles sobre esos componentes**. Para mostrar los detalles de qué información es procesada y cómo es transformada, precisamos de una herramienta de soporte de modelización textual y una de ellas es el **diccionario de datos** ([ver 4.3.el diccionario de datos](#)).

El **DFD** Tampoco provee ninguna indicación explícita de la *secuencia* del procesamiento. El procesamiento o la secuencia puede estar implícitamente en el diagrama, pero la representación procedimental, de cuando inicia y finaliza cada proceso quedará explícita en el flujograma. ([ver 4.5. flujograma](#))

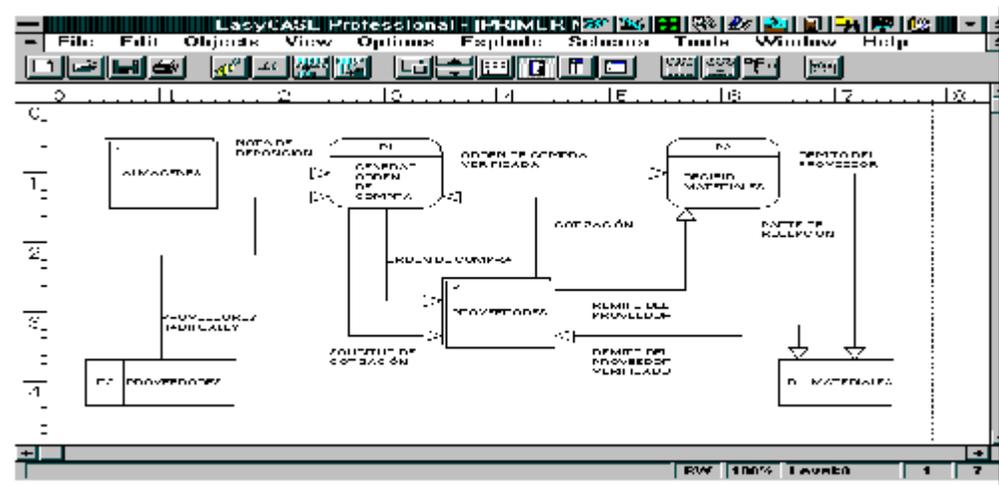


FIGURA 4.1. Diagrama de Flujo de Datos.

RECOMENDACIONES PARA UN DFD.

1. Los DFD son más legibles, si las entidades externas son diseñadas sobre los bordes del diagrama; de tal forma, que la frontera del sistema (o contexto) se sitúe dentro del contorno de las entidades externas
2. Si los flujos de datos principales van del lado izquierdo hacia el lado derecho del diagrama, la lectura se hará más fácil y más rápida.
3. Las duplicaciones de símbolos deben ser mantenidas al mínimo, pero cuidando de mantener un número aceptable de líneas de flujo de datos cruzándose unas con otras.
4. Inicie la construcción del DFD por las entidades externas, a continuación siga con las salidas que de ellas son originadas, juntamente con las entradas que irán para ellas.

Al diseñar el primer borrador del DFD, piense en como el sistema funciona realmente, cuál es la entrada o proceso que inicia, y por ahí comience el diseño.

Los primeros diseños de un DFD siempre tendrán la finalidad de borrador. El objetivo es la identificación de todos las entidades externas, procesos y archivos de datos que formarán parte del sistema, además de incluir los flujos de datos entre ellos. Próximas versiones mejorarán las definiciones y el diseño.

El orden más lógico para diseñar un DFD es definir la entidad externa o proceso que genera una entrada de datos, después el proceso que trata esa entrada, y a continuación los archivos de datos que son utilizados para almacenarla y para garantizar el funcionamiento de ese proceso y por último definir las salidas que son generadas por dicho proceso.

El primer borrador puede ser realizado en papel, pero los posteriores deben ser realizados utilizando alguna herramienta de software automatizada (CASE) específicamente diseñada para la modelización del sistema de información; estas herramientas cuentan con un diccionario de datos, que almacenan los detalles del modelo lógico del sistema²⁹.

4.3.EL DICCIONARIO DE DATOS

Un análisis del ámbito de información estaría incompleto si solo se considera el flujo de la información. Cada flecha del diagrama de **flujo de datos** representa uno o varios elementos de información ([ver 4.2. la modelización de las funciones del sistema](#)); cada **archivo de datos** es una colección de elementos de datos individuales; incluso puede que el contenido de una **entidad externa** requiera ser expandido antes de que su significado pueda ser definido explícitamente. Por lo tanto, el analista debe disponer de algún método para representar el contenido de cada componente del modelo de flujo de datos.

Se ha propuesto el **Diccionario de Datos** como gramática casi formal para describir el contenido de los objetos definidos durante el análisis estructurado.

Esta importante notación ha sido definida de la siguiente marea:

El Diccionario de Datos es un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que le permite al usuario y al proyectista del sistema tener una misma comprensión de las entradas, de las salidas, de los componentes de los repositorios, y también de cálculos intermedios.

CONTENIDO DEL DICCIONARIO DE DATOS

El Diccionario de datos debe contener la siguiente información:

Nombre: el nombre principal del elemento; del flujo de datos, del repositorio de datos o de una entidad externa.

Alias: otros nombres usados para la entrada, dado que un mismo elemento puede ser conocido por diferentes nombres.

Definición: Exposición clara y precisa de las características genéricas y diferenciales del objeto.

Descripción: Explicar las diversas partes o circunstancias, que componen la definición, de los objetos.

Dónde se usa/cómo se usa: Un listado de los procesos que usan un elemento de datos, o del control de cómo lo usan.

Descripción del contenido: El contenido es representado mediante una anotación que se describe en la siguiente tabla.

Existen muchos esquemas de anotación usados por los analistas de sistemas el que sigue es uno de los mas usados

Símbolo	Descripción
=	Está compuesto de
+	Y
()	Opcional (puede estar presente o ausente)
{ }	Interacción entre componentes
[]	Elección de una de las opciones
* *	Comentario
	Separa opciones de alternativas en la construcción []
@	Identificador campo llave

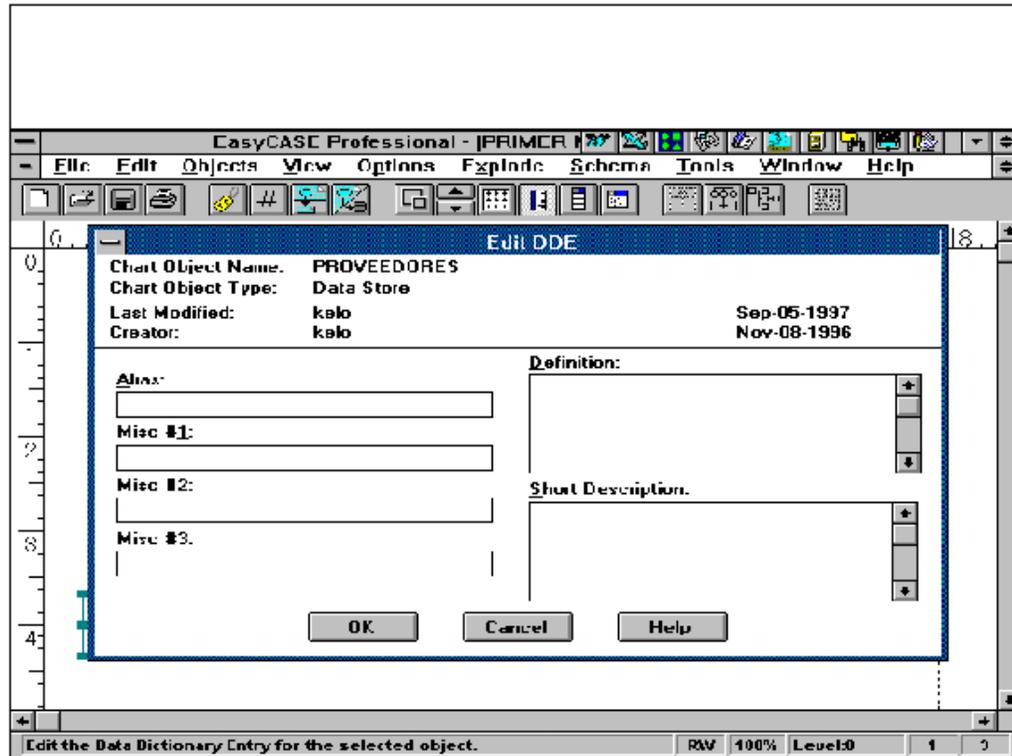


FIGURA 4.2 Diccionario de Datos - Descripción

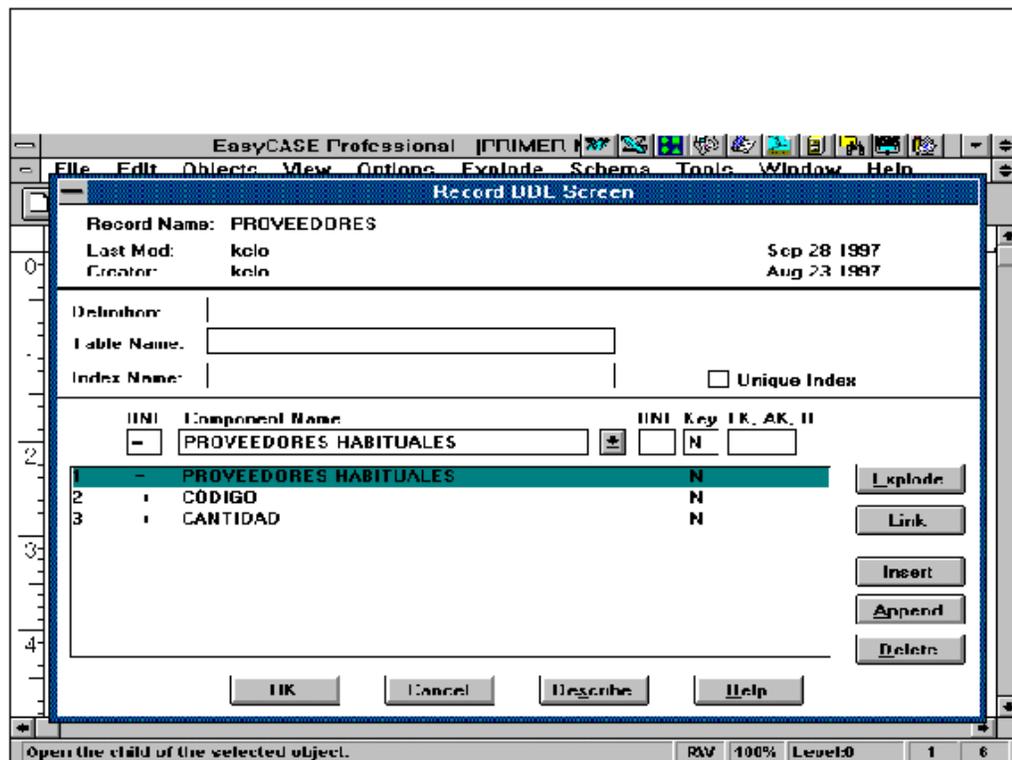


FIGURA 4.3 Diccionario de Datos - Estructura

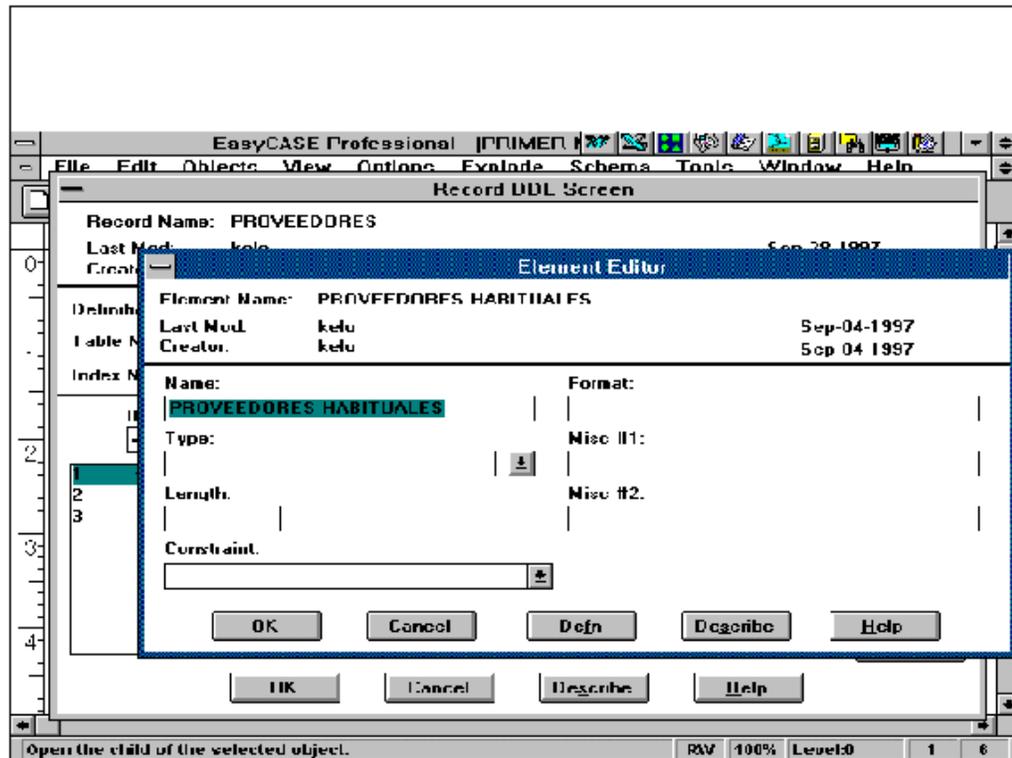


FIGURA 4.4 Diccionario de Datos - Definición de un elemento

4.4. LA MODELIZACIÓN DE DATOS ALMACENADOS

EL MODELO RELACIONAL DE DATOS (RDM).

Todos los sistemas almacenan y usan información sobre el ambiente con el cual interactúan, algunas veces la información es mínima, pero en la mayoría de los sistemas, es bastante compleja. No solamente queremos saber, en detalle, qué información está contenida en cada archivo de datos, sino también que **relaciones existen entre los archivos de datos**. Este aspecto del sistema no está representado por el diagrama de flujo de datos; pero sí está activamente representado por el **Modelo Relacional de Datos (Relational Data Model)**.

Como la anotación de los repositorios de datos en el DFD dice muy poco acerca de los **detalles de los datos**, es necesario que a partir de este modelo, se requiera una clara definición de las entidades (archivos de datos) y de sus relaciones, que conforman parte del proyecto y que por lo tanto son de especial interés para el usuario. Estos datos y relaciones deben ser almacenados a través de archivos que posteriormente formarán la base de datos del sistema.

Por lo tanto, el objetivo de un **RDM** es el de ilustrar la estructura de los datos del sistema, a través de la identificación de las entidades detectadas en el sistema y el diseño de sus relaciones.

El RDM posee dos importantes componentes, que son las Entidades y las Relaciones:

1. **Entidades o Tipos de objetos:** Son representadas por un cuadrado en el RDM. Una Entidad representa a una colección o conjunto de objetos (cosas) del mundo real, cuyos

miembros diseñan un papel en el sistema que se está desarrollando. Las Entidades pueden ser identificadas de forma única y, ser descritas a través de uno o mas hechos (Atributos). Como regla general, tomamos que, en cada archivo de datos definido por el DFD, se almacenan los datos que describen a las Entidades del sistema de información, o sea, a cada archivo de datos del DFD le corresponde una Entidad al RDM.

2. **Relaciones:** Una relación representa un conjunto de conexiones o asociaciones entre las **Entidades**, interligadas por vectores al relacionamiento. Normalmente, cada entidad que compone la base de datos de un sistema podrá estar relacionada con otras; por ejemplo, un cliente podrá estar relacionado con varias ventas, una venta con varios productos, un vendedor con varias ventas, y así sucesivamente en cada uno de los procedimientos.

Por lo tanto, considerando que las entidades de una base de datos están relacionadas, y que a través de esa relación son generados informes, como por ejemplo: todos los productos vendidos a un cliente, es importante definir todas las relaciones entre las entidades y su correspondiente **tipo de relación** y que veremos a continuación.

TIPOS DE RELACIONES

El RDM muestra los tres tipos de relaciones posibles entre los archivos de datos y los procesos de un DFD: uno – a – uno; uno – a – varios y varios – a – varios. Pero veamos cómo son cada una de estas relaciones:

Relación uno a varios.

Es el tipo de relación más común; y en este tipo de relación, un registro de la Tabla A puede tener muchos registros coincidentes en la Tabla B, pero un registro de la Tabla B sólo tiene un registro coincidente en la Tabla A.

Relación varios a varios.

En una relación varios a varios, un registro de la Tabla A puede tener muchos registros coincidentes en la Tabla B y viceversa. Este tipo de relación sólo es posible si se define una tercera tabla (denominada tabla de unión), cuya clave principal consta de al menos dos campos; y que además, estos campos, correspondan a las claves externas de las Tablas A y B.

Relación uno a uno.

En una relación uno a uno, cada registro de la Tabla A sólo puede tener un registro coincidente en la Tabla B y viceversa. Este tipo de relación no es habitual, debido a que la mayoría de la información relacionada de esta forma estaría en una sola tabla. Puede utilizar la relación uno a uno para dividir una tabla con muchos campos, para aislar parte de una tabla por razones de seguridad o para almacenar información que sólo se aplica a un subconjunto de la tabla principal.

BENEFICIOS DEL RDM

Los principales beneficios en la utilización del RDM son:

1. Da una visión de alto nivel de los archivos de datos involucrados en el sistema.
2. Ayuda a descubrir los elementos o las entidades que no fueron detectadas, al momento de diseñar y analizar el DFD.
3. Simplifica la estructuración de los datos.
4. Facilita la definición y el análisis de las Llaves primarias de cada archivo de datos; como así también sus llaves foráneas, que son necesarias para establecer la relación entre las

entidades, y que a través de las cuales podrán ser procesados y consultados los registros ([ver 3.1.2.llave primaria o identificadora](#)).

5. Facilita la definición y el análisis del **tipo de relación** existente entre las entidades u objetos, que conformarán la base de datos:
 - **uno – a – uno**, en este caso se debe verificar que cada entidad sea única o puede ser formada por un conjunto de entidades de menor nivel.
 - **uno – a – varios**,
 - **varios – a – varios**; en este caso se debe subdividir en dos relaciones del tipo uno – a – varios. ([ver diseño de la relación uno a uno](#))

Todos estos beneficios hacen que el RDM sea fundamental para poder proyectar una base de datos.

Después de la construcción del RDM, también es necesario que sean incorporados al Diccionario de Datos todos los datos que fueron definidos en este modelo y que serán almacenados en cada archivo, y que posteriormente formarán la base de datos del sistema proyectado.

TECNICA DE DISEÑO DEL RDM.

Cada entidad es representada por un rectángulo,

La relación entre las entidades es representada por una línea uniendo a los rectángulos a relacionar,

El tipo de relación es representada por un par de números en la extremidad de la línea de relación: **1** identifica una relación con un único registro y **N** identifica una relación con muchos registros y **0** identifica la relación con ningún registro,

La descripción de la relación debe ser hecha a lo largo de las líneas que ligan las entidades relacionadas.³⁰

En la Fig. 4.4.1. se representa la relación entre dos entidades; la entidad PERSONA y la entidad DEPARTAMENTO. El par de números (**1** , **1**) indica que como **mínimo** una (**1**) PERSONA trabaja en un DEPARTAMENTO y como **máximo** una (**1**) PERSONA trabaja en un DEPARTAMENTO. Por otro lado, el par de números (**0** , **N**) indica que en un DEPARTAMENTO pueden trabajar como **mínimo** ninguna (**0**) PERSONA y como **máximo** varias (**N**) PERSONAS.

Por lo tanto, una PERSONA está relacionada a un DEPARTAMENTO (1,1) y un DEPARTAMENTO está relacionado a ninguna o varias PERSONAS (0,N)

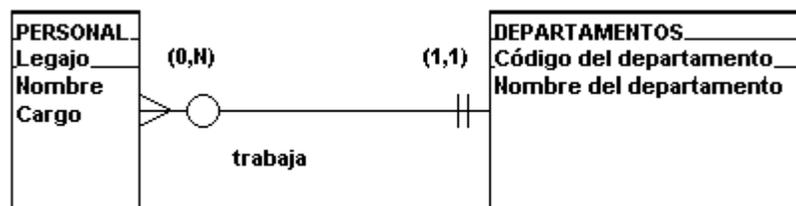


FIGURA 4.4.1. Relación entre entidades

En el ejemplo de la Fig. 4.4.2. cada VENTA involucra uno o mas (1,N) productos vendidos; pero un PRODUCTO es parte de solamente una VENTA (1,1).

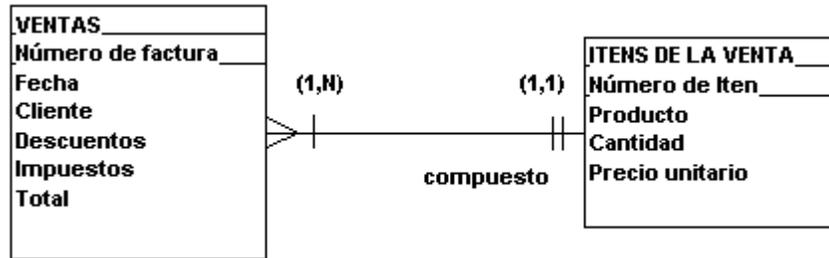


FIGURA 4.4.2. Propiedades de las entidades y las relaciones

En el ejemplo de la Fig. 4.4.3. cada PROVEEDOR puede suministrar uno o mas (1,N) PRODUCTOS y cada PRODUCTO puede ser provisto por uno o mas (1,N) PROVEEDORES o viceversa pues una relación entre dos entidades puede ser leída en cualquiera de las dos direcciones.

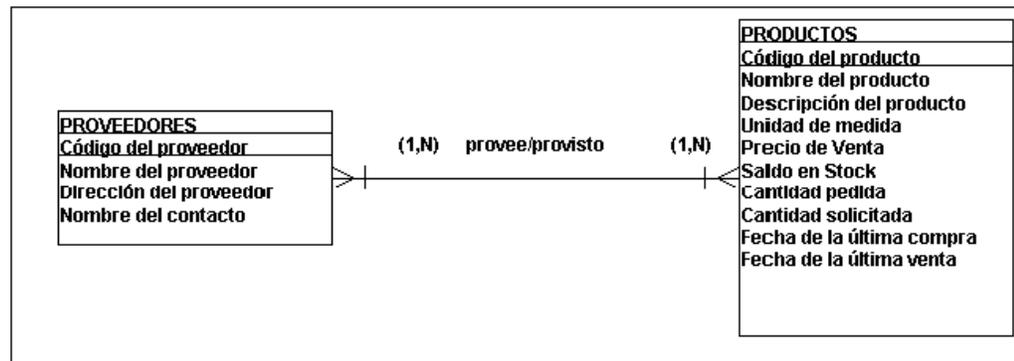


FIGURA 4.4.3. Direccionalidad de las relaciones

Diseño de la Relación uno a uno.

Al ser identificada una relación uno a uno (1,1), se debe inicialmente verificar si los dos objetos relacionados son realmente distintos o pueden ser unidos en un único elemento.

Si cada elemento fue identificado con la misma **llave primaria** y si ambos se complementan, hay una fuerte razón para unir a los dos elementos en uno solo. Por ejemplo tenemos a las entidades PRODUCTO Y STOCK.

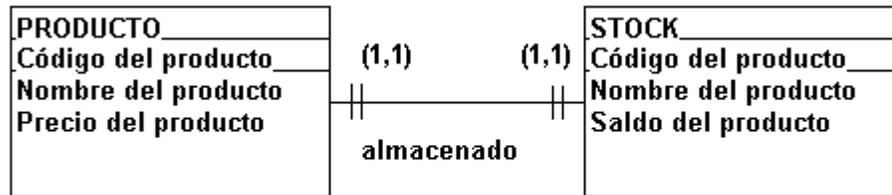


FIGURA 4.4.4. Relación uno a uno

Como cada PRODUCTO es almacenado en STOCK, podemos considerar una única entidad de PRODUCTOS EN STOCK, representada en la figura 4.4.5.

En este caso, las entidades PRODUCTO Y STOCK no son realmente distintas y por ese motivo, debemos almacenarlas en un único archivo de datos, pues el Saldo es apenas un atributo de cada PRODUCTO ([ver 4.2.1 Normalización](#)).



FIGURA 4.4.5 Unión de dos entidades relacionadas uno a uno

Si los dos elementos fuesen realmente distintos, cada uno debería ser identificado por una llave primaria que lo distinga de forma inequívoca de los demás. ([ver 3.1.2 llave primaria o identificadora](#)).

La relación entre los dos objetos deberá ser realizada a través de una llave relación, denominada llave foránea <FK> La llave foránea deberá estar indicada en el objeto relacionado, como se ilustra en la figura 4.4.6. La llave foránea recibe este nombre porque, necesariamente ella, no es un atributo del elemento relacionado, pero sí es la llave primaria del elemento al cual está se relaciona.

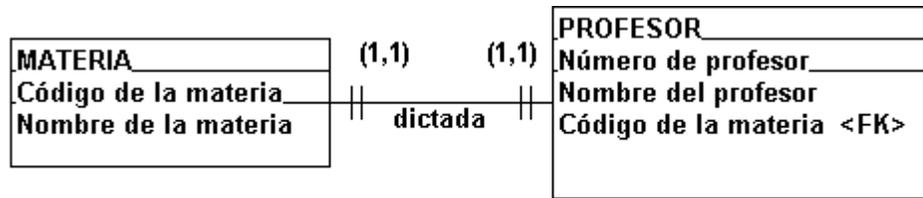


FIGURA 4.4.6.Llave foránea <FK>

En el caso de la relación (1,1), representada en la figura 4.4.6, entre una MATERIA y un PROFESOR que dicta una MATERIA; vemos al Código de la materia como la llave primaria de la entidad MATERIA; y la llave primaria Número de profesor de la entidad PROFESOR.

Si determinamos que un PROFESOR está relacionado a una MATERIA, precisamos pues de una llave que haga la relación entre las dos entidades; esta llave que como ya vimos se denomina llave foránea y es identificada con la sigla <FK>; y en nuestro caso quien cumple esta función es el Código de la materia y debe ser archivada en la entidad que describe al PROFESOR, y apunta a la MATERIA que él dicta, como se ilustra en la figura 4.4.8.

Por lo tanto, en el archivo PROFESOR, el dato “Código de la materia” es un campo llave foránea (FK), significando que se trata de un dato del archivo MATERIA, pero que precisa existir en el archivo PROFESOR para permitir la **RELACIÓN** entre ambos. Note que en esta relación, un PROFESOR puede dictar solamente una MATERIA, tal cual se observa en la figura 4.4.7.

Otra alternativa de relacionar a los archivos PROFESOR y MATERIA sería si admitimos que una materia solamente puede ser dictada por un profesor, esto significa que debemos incluir la llave foránea “Número del profesor” en el archivo MATERIA.

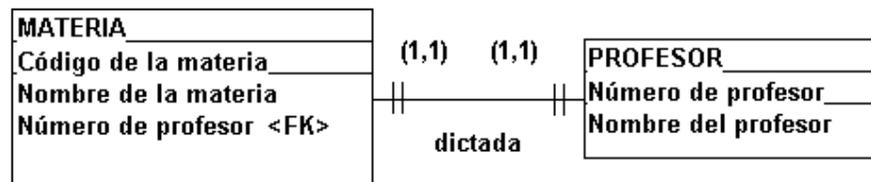


FIGURA 4.4.7 Llave foránea

Aunque estas dos soluciones sean posibles para la relación entre PROFESOR y MATERIA, ninguna de ellas está totalmente correcta. Una mejor solución debe permitir que un profesor pueda dictar varias materias o que una materia pueda ser dictada por varios profesores. O sea, la relación entre PROFESSOR y MATERIA no es uno a uno, sino por lo menos uno a varios (que se trata en el punto siguiente)

A continuación se presentan cuatro preguntas, que sirven como ejemplo, para presentar el análisis que debe ser hecho al proyectarse una relación uno a uno:

- ¿ La relación siempre será uno a uno?
- ¿Hay alguna posibilidad de que en el futuro ella pase a ser uno a varios?
- ¿De que forma se podrá adaptar ante un posible cambio del sistema?
- ¿En qué archivo deberá ser incluida la llave foránea para ser utilizada como apuntadora de la relación?

Diseño de la Relación uno a varios.

La relación uno a varios ocurre cuando una única instancia de una entidad está relacionado con otras instancias de otra entidad. Como cada entidad posee un archivo de datos conteniendo sus atributos, la llave primaria de la "entidad uno" debe ser una "llave foránea" en el archivo que describe a la "entidad muchos", pudiendo ser parte de su llave primaria o no.

En el ejemplo ilustrado por la Fig. 4.4.8., mostrando la relación entre una MATERIA y varios PROFESORES, el atributo "Código de la materia" es la llave foránea de PROFESOR.

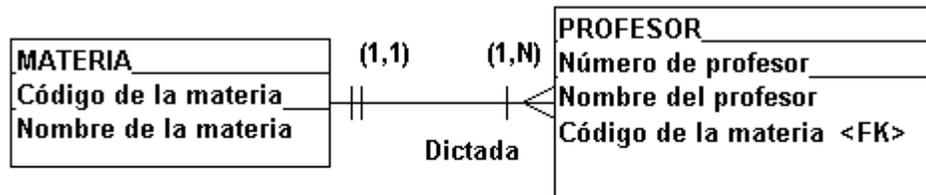


FIGURA 4.4.8. Relación uno varios cuando una materia es dictada por uno o varios profesores

En este caso, una materia puede ser dictada por uno o varios profesores (1,N), pero un profesor solamente puede dictar una única materia (1,1).

En el ejemplo ilustrado por la Fig. 4.4.9., muestra la relación entre un PROFESOR y varias MATERIAS, el atributo "Número del profesor" es la llave foránea de MATERIA.

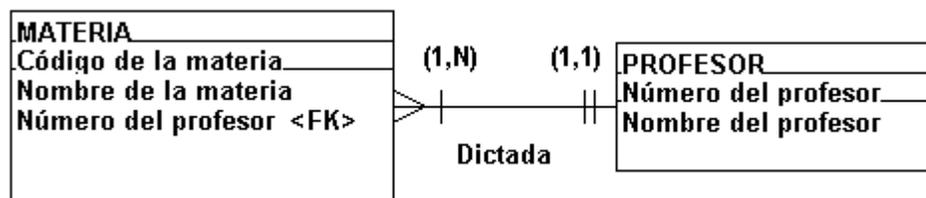


FIGURA 4.4.9. Relación uno a varios, una materia es dictada únicamente por un profesor.

En este caso un profesor puede dictar una o varias materias (1,N) pero una materia puede ser dictada solamente por un profesor (1,1).

Diseño de la Relación varios a varios.

Si analizamos los ejemplos anteriores, percibimos que la relación más correcta entre PROFESOR Y MATERIA no es ni uno a uno ni tampoco uno a varios, pero sí lo es varios a varios, o sea, un profesor puede dictar muchas materias y una materia puede ser dictada por muchos profesores.

Una relación (N,N) siempre debe ser resuelta por dos relaciones (1,N), pues no es posible que tanto PROFESOR como MATERIA reciban llaves foráneas. En este caso, únicamente las llaves primarias de ambos objetos relacionados (N,N) deberán ser identificadas y, a continuación, un “objeto de intersección” deberá ser creado. La llave primaria del objeto de intersección será la combinación o concatenación de las llaves primarias de los dos objetos de origen.

En el ejemplo ilustrado por la figura 4.4.10, en que un PROFESOR dicta varias materias(1,N) y una MATERIA puede ser dictada por varios profesores(1,N). La única línea de relación (N,N) puede ser considerada como una combinación de dos relaciones (1,N), ambas con un objeto de intersección.

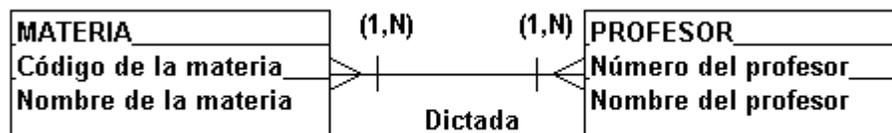


FIGURA 4.4.10 Relación varios a varios

Para determinar los datos que deberán estar contenidos en los objetos de intersección a ser creados debemos analizar la relación (N,N) entre MATERIA Y PROFESOR haciendo las siguientes preguntas.

¿Cuál debe ser el objeto que posea una llave primaria que corresponda a la concatenación de un determinado “Código de la materia” y de un determinado “Número de profesor”?

¿Qué datos o atributos dependen exclusivamente de esta combinación?

¿Qué datos pueden ser obtenidos si sabemos que estamos tratando con una determinada MATERIA dictada por un determinado PROFESOR?.

Al tratar de responder estas preguntas verificamos que diferentes materias pueden ser dictadas por diferentes profesores en diferentes horarios y aulas y, diferentes profesores dictan diferentes materias en determinadas aulas y en determinados horarios.

Por lo tanto; como una determinada materia puede ser dictada por diferentes profesores en diferentes aulas y en diferentes horarios, podemos crear un objeto de intersección denominado COMISIÓN. De esta forma, un determinado profesor podrá dictar varias materias, cada una en su respectiva aula y horario; así como cada materia podrá ser dictada por varios profesores, y para cada profesor habrá una determinada aula y horario.

La figura 4.4.11. ilustra la relación (N,N) entre MATERIA Y PROFESOR resuelta por una relación (1,N) entre MATERIA Y COMISIÓN y una relación (1,N) entre PROFESOR Y COMISIÓN.

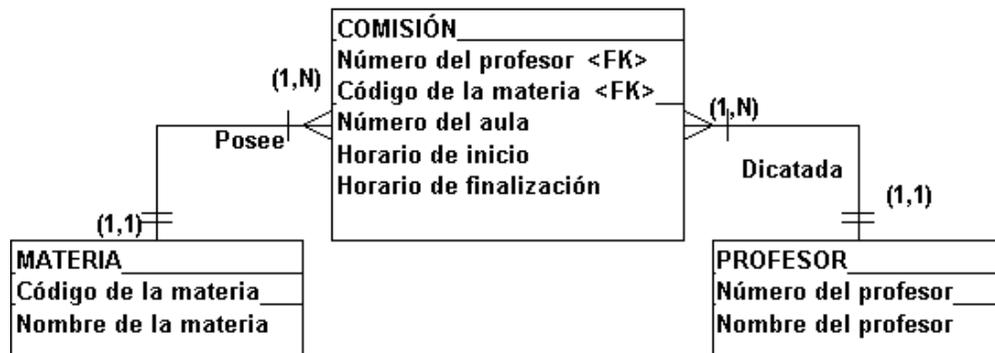


FIGURA 4.4.11 Relación varios a varios solucionada

En este caso, la llave primaria de COMISIÓN es compuesta por dos llaves foráneas. O sea, para que una COMISIÓN sea identificada es preciso saber cual es la materia y cual es el profesor. Como el “Código de la materia” pertenece a la MATERIA y el “Número de profesor” pertenece a PROFESOR ambos son llaves foráneas en COMISIÓN y concatenadas forman su llave primaria, pues la identifican.

NORMALIZACIÓN.

El proceso de la construcción del Modelo Relacional de Datos (RDM), tiene como objetivo:

- Percibir las cosas de significación sobre lo que se necesita saber y mantener la información. Esto es definir a las entidades y diseñarlas como un recuadro.
- Añadir las relaciones de gestión, las cuales se han nombrado como asociaciones significativas entre entidades. Esto es definir al conjunto de conexiones que ligan a las entidades u objetos y son representadas por medio de vectores.
- En cada entidad se listan los tipos de información que se podrían mantener o conocer. Esto es la definición de cada uno de los atributos por los cuales una entidad es conocida.
- Se determina la forma en que cada aparición de una entidad puede ser identificada de forma única. Esto es la definición de uno o más campos identificadores o llave.

Por lo tanto la modelización (RDM) permite:

- Minimizar la duplicación de datos;
- Proporcionar la flexibilidad necesaria para soportar requisitos funcionales y

- Que el modelo se estructure sobre una amplia variedad de diseños alternativos de bases de datos.

La mayor dificultad en este proceso es que se depende de la buena comprensión del analista acerca de lo que realmente es una Entidad, un Atributo y una Relación. Para evitar tal circunstancia es que se aplica el proceso de **NORMALIZACIÓN**.

Entonces denominamos **NORMALIZACIÓN** al proceso de simplificación de archivos de datos que componen una base de datos relacional (diseño eficaz de tablas); y que persigue como objetivo principal minimizar la duplicidad de información, prevenir inconsistencias, evitar redundancias, garantizar que no existan pérdidas de información. En resumen son las técnicas y algoritmos que ayudan, al proyectista de una base de datos relacional, a construir relaciones normalizadas, según sea el significado y el contenido del universo a ser modelado, evitando, anomalías en el manejo de estos datos³¹

El proceso de normalización consiste, básicamente, en la aplicación de un conjunto de reglas para definir adecuadamente los datos o campos que compondrán los archivos de datos. Esas reglas buscan:

- Minimizar redundancias;
- Eliminar anomalías de actualización;
- Proveer el mejor camino de acceso a cualquier dato;
- Asegurar resistencia a la mantención del modelo de datos;
- Evitar datos no identificables a través de una definición rigurosa de identificadores y relaciones.

Fueron establecidos cinco tipos de archivos normalizados, denominados, en orden creciente de simplicidad: primera forma normal (1FN), segunda forma normal (2FN), tercera forma normal (3FN), cuarta forma normal (4FN) y quinta forma normal (5FN).

En general, las tres primeras reglas básicas de normalización son suficientes para resolver la gran mayoría de casos. Es por ello que definiremos a continuación las tres primeras formas normales y discutiremos la manera de simplificar los archivos de datos hasta la tercera forma normal. Se podría resumir a estas tres formas normales más utilizadas, de la siguiente manera:

- Eliminar campos repetitivos;
- Eliminar datos redundantes;
- Eliminar atributos no dependientes.

Además la 1FN, 2FN y la 3FN son mecanismos para identificar entidades y relaciones perdidas.

PRIMERA FORMA NORMAL (1FN).

Asegurar que todas las entidades son identificadas de forma única por una combinación de atributos y/o relaciones.

Se refiere a cualquier archivo que posea un valor por campo; la relación entre la llave primaria de un archivo y cada uno de los otros campos debe ser de uno a uno.

De una manera práctica, debemos eliminar grupos repetidos de datos, hasta que cada dato tenga una llave primaria para cada ocurrencia.

El archivo de datos ejemplificado a continuación no está normalizado; entre otras cosas, hay más de un valor o supermercado en cada campo de Negocio.

Producto	Negocio
Arroz	Coto, Disco, Carrefour, Jumbo
Poroto	Coto, Macro, Carrefour, Jumbo
Harina	Coto, Macro, Carrefour
Azúcar	Tía, Disco, Carrefour

Como puede percibirse, en el campo Negocio existen varios valores de datos (grupos repetidos). A través de este archivo podemos obtener la información de que existe, por ejemplo, arroz en los supermercados Coto, Tía, Disco, Carrefour, Jumbo. Mientras tanto ¿cómo podríamos llegar a saber la cantidad existente de cada uno de los productos, en cada uno de los negocio?.

De acuerdo con la primera forma normal este archivo debe ser revisado para que sean eliminados los grupos repetidos, o sea, en el campo Negocio debe existir el nombre de apenas un supermercado. Esto implicará, la creación de un número mayor de filas o registros en el archivo. Pues deberá haber una fila para cada producto en cada negocio. A partir de esto, podremos fácilmente registrar la cantidad existente de cada producto en cada negocio.

Después de la aplicación de la primera regla de normalización, el archivo de datos de los productos en Stock asume la siguiente estructura de datos:

Producto	Negocio	Teléfono	Cantidad	Precio	Total
ARROZ	Coto	670-1158	200	10	2000
ARROZ	Disco	923-3951	500	9	4500
ARROZ	Carrefour	921-4802	700	11	7700
ARROZ	Jumbo	342-6400	1000	8	8000
POROTO	Coto	670-1158	300	13	3900
POROTO	Macro	923-4377	500	12	6000
POROTO	Carrefour	921-4802	200	14	2800
POROTO	Jumbo	342-6400	400	8	3200
HARINA	Coto	670-1158	400	8	3200
HARINA	Macro	923-4377	600	9	5400
HARINA	Carrefour	921-4802	100	7	700
AZUCAR	Disco	923-3951	1100	4	4400
AZUCAR	Carrefour	921-4802	900	5	4500
AZUCAR	Tía	449-7448	1200	3	3600

SEGUNDA FORMA NORMAL (2FN).

Eliminar atributos que dependen solamente de una parte del identificador único

Si una entidad tiene un identificador único compuesto de más de un atributo y/o relación, y si otro atributo depende sólo de una de las partes de este identificador compuesto, entonces el atributo, y la parte del identificador del que depende, deberán formar la base de una nueva entidad. La entidad nueva, se identifica por la parte emigrada del identificador único de la entidad original, y tiene una relación de uno a varios unida con la entidad original.

Para testear si un archivo de datos está en la segunda forma normal debemos hacer inicialmente las siguientes preguntas:

- ¿Cuál es el campo o conjunto de campos que constituye la llave primaria del archivo?
Si la llave primaria fuese concatenada, esto es, formada por mas de un campo, preguntamos también:
- ¿Hay algún campo no-llave que dependa de apenas, de una parte de la llave primaria?
En el archivo del ejemplo anterior, el producto, por sí solo no es suficiente para identificar inequívocamente un determinado registro, pues varios registros poseen el mismo producto. Para obtener una llave primaria exclusiva debemos concatenar producto con negocio, pues no hay ninguna llave "Producto + Negocio" duplicado. En este caso, como la llave es concatenada, debemos además hacer la segunda pregunta para cada campo no-llave:
- ¿La cantidad depende apenas de una parte de la llave?
La respuesta es no; pues es preciso conocer tanto el producto como el negocio para obtener la Cantidad.
- ¿El Precio depende apenas de una parte de la llave?

La respuesta es también no; pues es Preciso conocer tanto el Producto como el Negocio para obtener el Precio.

- ¿El Teléfono depende apenas de una parte de la llave?

En este caso la respuesta es sí; pues si usted conoce el Negocio también podrá saber cual es su Teléfono, independientemente del Producto; por lo tanto, el archivo ejemplificado anteriormente no está en la segunda forma normal, pues él no pasó por el test.

Cuando un archivo de datos no está en la segunda forma normal, la base de datos no estará correcta por las siguientes razones:

- El archivo de datos ocupará mas espacio en el disco del que será necesario, pues el número de Teléfonos se repite para cada Producto almacenado en el mismo archivo;
- Si un negocio cambia el número de Teléfono, todos los registros de Productos para aquel Negocio deberá tener el campo Teléfono modificado;
- Si ocurre algún problema con el proceso de actualización de datos, un mismo Negocio podrá aparecer con números de Teléfonos diferentes, dependiendo de cual registro sea por el que se accede, o sea, la integridad de la base de datos estará perdida;
- Cuando un negocio posee un único Producto y su registro fuese eliminado (por inexistencia en stock), también será eliminado el Teléfono del Negocio, pues podrá no existir otro lugar en la base de datos que lo almacene.

Para evitar estos problemas, el archivo anterior deberá ser dividido en dos, como se ilustra a continuación:

Producto	Negocio	Cantidad	Precio	Total
ARROZ	Coto	200	10	2000
ARROZ	Disco	500	9	4500
ARROZ	Carrefour	700	11	7700
ARROZ	Jumbo	1000	8	8000
POROTO	Coto	300	13	3900
POROTO	Macro	500	12	6000
POROTO	Carrefour	200	14	2800
POROTO	Jumbo	400	8	3200
HARINA	Coto	400	8	3200
HARINA	Macro	600	9	5400
HARINA	Carrefour	100	7	700
AZUCAR	Disco	1100	4	4400
AZUCAR	Carrefour	900	5	4500
AZUCAR	Tía	1200	3	3600

Negocio	Dirección	Teléfono
Coto	Av. Del trabajo 1176	670-1158
Disco	Emilio Mitre 515	923-3951
Carrefour	Av. La Plata 2222	921-4802
Jumbo	Av. Cruz 4897	342-6400
Macro	Av. Rivadavia 4735	923-4377
Tía	Av. Rivadavia 7788	449-7448

Ahora los dos archivos están en la segunda forma normal. El archivo de PRODUCTOS EN STOCK está en la segunda forma normal porque los campos no-llave(Cantidad, Precio y Total) son dependientes de toda llave primaria concatenada Producto + Negocio y de nada más.

El segundo archivo, NEGOCIOS, también está en la segunda forma normal porque él no posee una llave concatenada y, por lo tanto, una columna no-llave como Dirección o Teléfono naturalmente será dependiente del único campo llave, que es Negocio.

Analizando desde otra perspectiva, es fácil percibir que el archivo anterior, a pesar de estar en la primera forma normal, contiene datos que describen dos cosas distintas y que son por un lado PRODUCTOS y por el otro NEGOCIOS.

Como regla general es importante, que un archivo de datos en una base de datos debe almacenar datos que describan apenas una entidad o evento. Por lo tanto, un archivo de datos para estar en la segunda forma normal debe contener datos apenas sobre un único objeto de información o una única clase de objetos. En nuestro ejemplo, el primer archivo ahora contiene apenas datos sobre productos en stock y el segundo sobre negocios.

TERCERA FORMA NORMAL (3FN).

Eliminar los atributos dependientes de atributos que no son parte del identificador único.

Un archivo en la segunda forma normal también estará en la tercera forma normal si un campo no-llave depende de otro campo no-llave.

Para verificar si un archivo en la segunda forma normal también está en la tercera forma normal debemos preguntar: ¿Algún campo no-llave es dependiente de cualquier otro campo no-llave?

El archivo de los PRODUCTOS EN STOCK posee tres campos (o columnas) no-llave: Cantidad, Precio y Total. Si sabemos la Cantidad y el Precio, sabremos el Total. Por lo tanto, el campo "Total" es dependiente de dos campos no-llave, pues puede ser obtenido a partir de la Cantidad multiplicada por el Precio.

Concluimos entonces, que el archivo de PRODUCTOS EN STOCK no está en la tercera forma normal.

Si el campo "Total" fuese eliminado, el archivo de PRODUCTOS EN STOCK pasa a estar en la tercera forma normal, ocupando menos espacio en el disco, y sin pérdida de información.

Producto	Negocio	Cantidad	Precio
ARROZ	Coto	200	10
ARROZ	Disco	500	9
ARROZ	Carrefour	700	11
ARROZ	Jumbo	1000	8
POROTO	Coto	300	13
POROTO	Macro	500	12
POROTO	Carrefour	200	14
POROTO	Jumbo	400	8
HARINA	Coto	400	8
HARINA	Macro	600	9
HARINA	Carrefour	100	7
AZUCAR	Disco	1100	4
AZUCAR	Carrefour	900	5
AZUCAR	Tía	1200	3

4.5.FLUJOGRAMAS

Como se señaló anteriormente, el DFD es una herramienta muy adecuada para modelizar una red de **procesos comunicantes asincrónicos**. Es por eso que precisamos de otra herramienta para representar la **lógica y la secuencia** de un procedimiento.

El flujograma es la representación gráfica que muestra: el comienzo y el fin de un proceso de tratamiento de datos, y las operaciones de decisiones necesarias para cumplirlo, en el orden secuencial correspondiente.

No hay duda de que de las herramientas tales como los flujogramas, son una excelente forma gráfica de describir fácilmente los detalles procedimentales.

El flujograma es la representación gráfica más ampliamente usada para el diseño procedimental. Desgraciadamente, es también el método del que más se ha abusado.

Un flujograma es un gráfico muy sencillo. Las tres construcciones de la programación estructurada se representan como en la figura 5.5. La secuencia se representa como dos cuadros de procesamiento conectados por una línea de control. La condición, también denominada IF-THEM-ELSE (si- entonces - sino), se dibujo como un rombo de decisión que, si es verdad, hace que se realice el procesamiento de la parte them y, si es falso, pasa al procesamiento e la parte else.

Los flujogramas son usados principalmente para la documentación física o las interfaces del hardware dentro de un sistema.

Un flujograma contiene dos tipos e elementos: Los **bloques** y las **líneas**.

1. **Los bloques**, Los bloques pueden representar **acción o decisión**.

Un bloque de acción representa una actividad: efectuar una operación aritmética entre dos números, convertir un valor en cero, etc. Su descripción implica siempre aplicar un verbo (hacer algo): sumar, transferir, borrar, etc.

Un bloque de decisión: es una forma de expresar una consulta acerca del cumplimiento o no de una determinada condición o alternativa. Según sea la respuesta que se dé a dicha consulta (verdadero o falso) se seguirán diferentes caminos.³²

2. **Las líneas** de dirección o flechas que comunican los bloques y determinan el orden secuencial en que deben ser considerados.

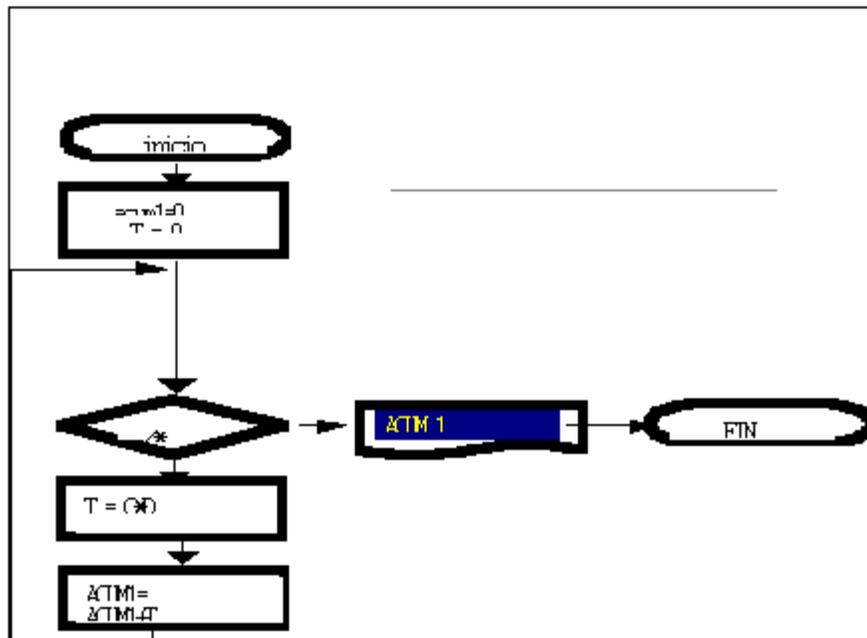


FIGURA 5.5 FLUJOGRAMA

4.6. TABLAS DE DECISIÓN

Es una forma particular de matriz mediante la cual se representan las **acciones** a tomar cuando se dan determinadas **condiciones (variables relevantes)**.

Es una técnica de aplicación en el análisis y diseño de sistema y procedimientos: presenta un modelo lógico de alternativas o conjunto de alternativas de forma completa y fácil de captar y visualizar.

En su documentación de los sistemas brinda la ventaja de evitar descripciones literarias de compleja comprensión. Y también como un medio de comunicación e instrumento de programación elimina todas las ambigüedades o falta de precisión que pueden surgir de las descripciones literarias facilitando al programador la conversión de las condiciones y decisiones a instrucciones aplicables a un computador.

Si hubiera N variables con valores binarios (verdadero / falso), entonces, habrá 2^N reglas distintas; si hubiera 3 condiciones habrá 8 normas.

Las tablas decisión están divididas en cuatro cuadrantes que conforman el siguiente esquema:

	REGLAS
DESCRIPCIÓN DE CONDICIONES	VALORES DE CONDICIONES
DESCRIPCIÓN DE ACCIONES	VALORES DE ACCIONES

Una metodología para la creación de las tablas es la siguiente

- 1 Definir e interpretar el problema (cuidado con las obviedades);
- 2 Poner por escrito en lenguaje narrativo el planteo del problema a fin de su corroboración
- 3 Distinguir y separar las condiciones de las acciones y agruparlas respectivamente
- 4 Crear la tabla de decisiones vacía, relacionando todas las condiciones y acciones en la columna izquierda y enumerando las combinaciones de condiciones en lo alto de la tabla (reglas)
- 5 Registrar los valores de las condiciones y de las acciones.
- 6 Analizar los resultados obtenidos (detección de omisiones redundancias contradicciones o ambigüedades)
- 7 Discutir los resultados con los usuarios

4.7. MODULOS DE UN SISTEMA

Un DFD precisa ser subdividido en diferentes partes, que llamaremos módulos, conteniendo cada una de ellos procedimientos manuales y/o automatizadas, a fin de que el sistema pueda ser desarrollado y ejecutado en unidades menores, más fáciles de ser implementadas controladas y manejadas. Estos módulos pueden ser: un programa, un procedimiento manual o automatizado, una relación de operaciones o comandos, o una combinación de estas tres.

Un módulo siempre será invocado como una unidad, y generalmente será desde una opción del menú; y constituye una operación o un procedimiento completo que el sistema debe ejecutar.

Lo normal es que los módulos estén relacionados con las entradas y salida de los datos, actualización de archivos, procedimiento de cálculo y otras operaciones específicas que el sistema deba efectuar. Como ejemplo de módulos presentamos los siguientes:

- Confección de una **NOTA DE PEDIDO**
- Modificación del los datos del **CLIENTE**
- Dar de baja a un **PROVEEDOR**
- Grabar el Archivo **HISTÓRICO DE VENAS**.
- Cálculo del **SALARIO**.
- Grabar una copia de seguridad de los archivos.

Como la división de un sistema en módulos, se debe realizar en función de las relaciones existentes entre los procedimientos y su contexto; La misma, debe tener su origen en los procesos del DFD, y en las entidades y sus relaciones definidas en el RDM.

Si fuese decidido que determinado proceso tendrá apoyo automatizado, se debe analizar la posibilidad y la conveniencia de su implementación por software.

Una regla práctica :

Un proceso es candidato a ser totalmente informatizado, si todo flujo de datos que en él entra o sale, se encuentra en uno de estos tres casos.

- a) se conecta a un repositorio o proceso ya definido para ser implementado por software,
- b) tiene su origen en una entidad externa y puede ser transferido directamente por procesamiento por software sin ningún procesamiento adicional no informatizado de sus datos
- c) tiene como destino una entidad externa y puede ser a él enviado directamente de la salida de software, sin ningún procesamiento adicional informatizado de sus datos.

En caso de no ser posible implementar el proceso totalmente por software, el debe ser explotado y revalidado continuamente, hasta que sean completamente separados los procesos manuales de los procesos a ser implementados por software.

Por último, luego de la definición de los módulos, se debe asignar un nombre a cada módulo (que se corresponda con el proceso definido en el DFD) y diseñar la relación entre los módulos.

EL ÁRBOL DE UN SISTEMA

Los módulos ya definidos, guardan una relación jerárquica entre sí, o sea, existen niveles de procesos y operaciones que serán desempeñados por el sistema, desde los mas amplios hasta los mas específicos. Y ésta jerarquía de módulos es la que da origen al **árbol del sistema**.

El árbol de sistema es un organigrama, que identifica a cada uno de los módulos y la jerarquía existente entre ellos. Una de las funciones principales del árbol es la de determinar la estructura de los menús de operaciones del sistema, pues cada módulo, según su nivel, dará acceso o ejecutará una determinada operación.

ESPECIFICACIÓN DE LOS MÓDULOS DEL SISTEMA

Habiendo ya definido los principales módulos y también elaborado el árbol del sistema y como cada uno de ellos está relacionado con el DFD y con el MRD, el desarrollo y prueba de los mismos debe ser planificado.

Normalmente, se debe producir y revisar una especificación escrita para cada módulo. Esta especificación, debe contener toda la información necesaria para que se pueda producir los códigos o programas necesarios para cada uno de los módulos.

La especificación de los módulos se realizará hasta el punto en que se tenga un modelo claro de los formatos de entradas y de salidas de datos; pues la lógica del sistema, los archivos a ser accedidos ya fueron definidos en el DFD y el MRD.

Si los formularios e informes del sistema fuesen generados por un generador automático (Asistente automático), quien programe debe saber qué campos o datos aparecerán en cada formulario e informe, y además podrá utilizar el mismo generador de formularios para definir la posición exacta de cada campo.

5. HERRAMIENTAS CASE

QUE ES UNA HERRAMIENTA

La sigla **CASE** significa **COMPUTER AIDED SOFTWARE ENGINEERING** (Ingeniería de Software asistida por computador) y es la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, procedimientos y su documentación.³³

Representa una forma que permite modelar las actividades de las empresas y desarrollar los sistemas de información gerenciales.

Algunos de los componentes de las herramientas CASE permiten:

- Confeccionar la definición de requerimientos de los usuarios
- Mejorar el diseño de los sistemas
- Mayor eficiencia en la programación (por su generación automática de códigos).
- Otorgar a la administración un mejor soporte en la documentación.

Para ello, y sin importar la arquitectura de la herramienta CASE, en general tales herramientas deben abarcar las siguientes propiedades:

- Tener una interfaz gráfica y textual, que le permita al usuario manejar los objetos de diseño.
- Contar con un Diccionario de Datos, a fin de rastrear y controlar los objetos diseñados.
- Disponer de un conjunto de herramientas que permitan chequear las reglas y analizar la lógica del diseño.

No es el interés de esta obra realizar una detallada descripción conceptual de estas herramientas; pero es sí el de hacer notar que las herramientas CASE serán un elemento muy importante, para el administrador de un proyecto informático, que le permitirá llevar adelante un proyecto informático de forma eficaz y eficiente.

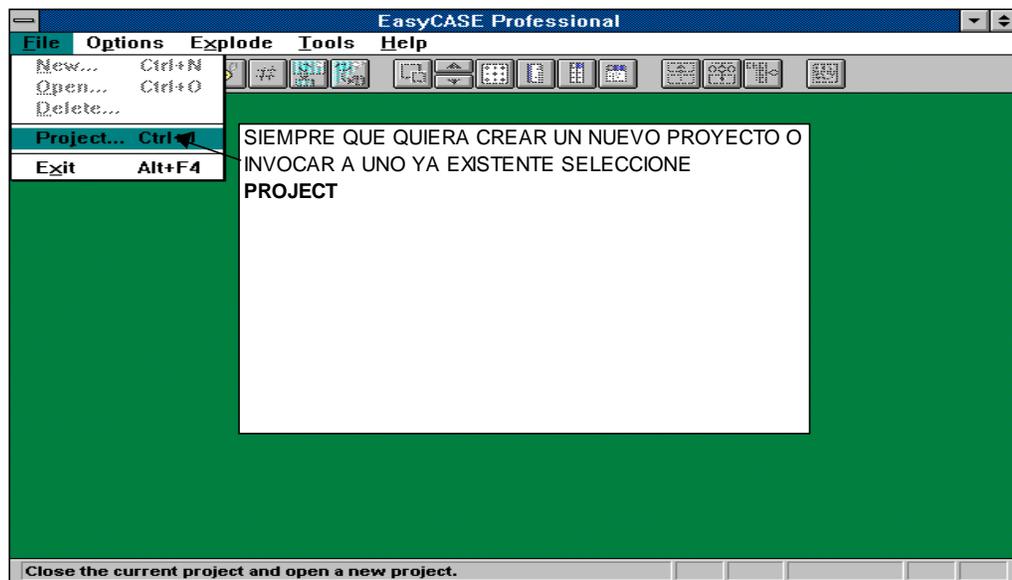
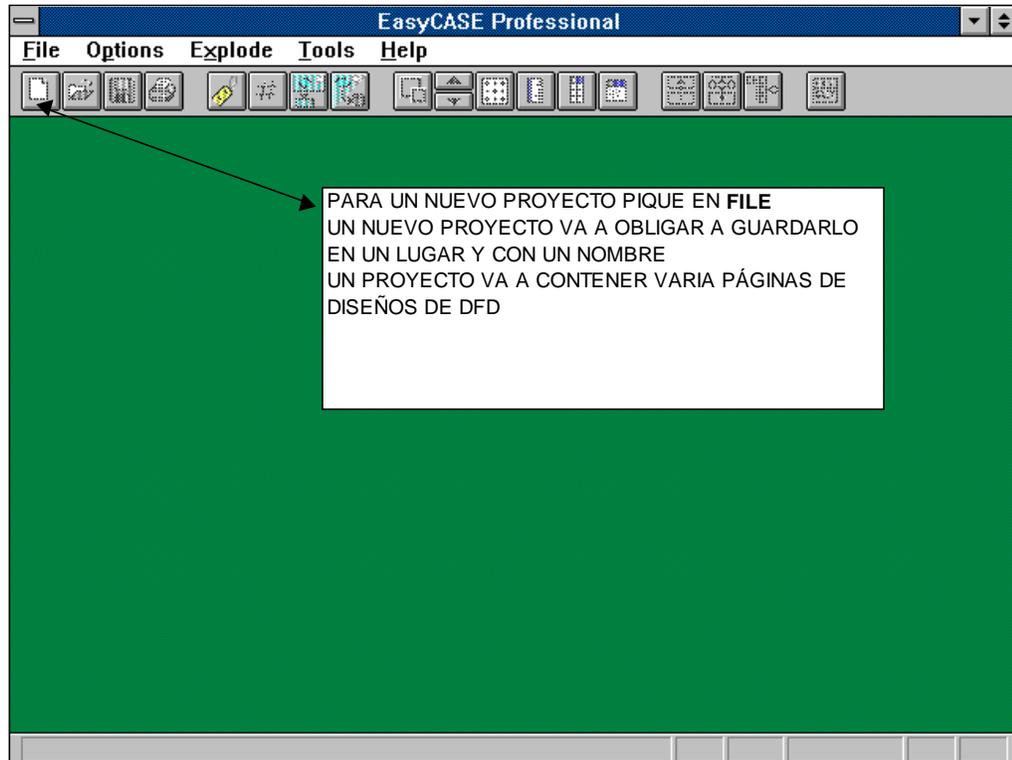
También es un hecho que estas mismas herramientas, como toda tecnología de la información se encuentran en continua evolución y existe además una gran variedad de proveedores y productos y cada uno de ellos con sus diferentes aplicaciones y especificaciones.

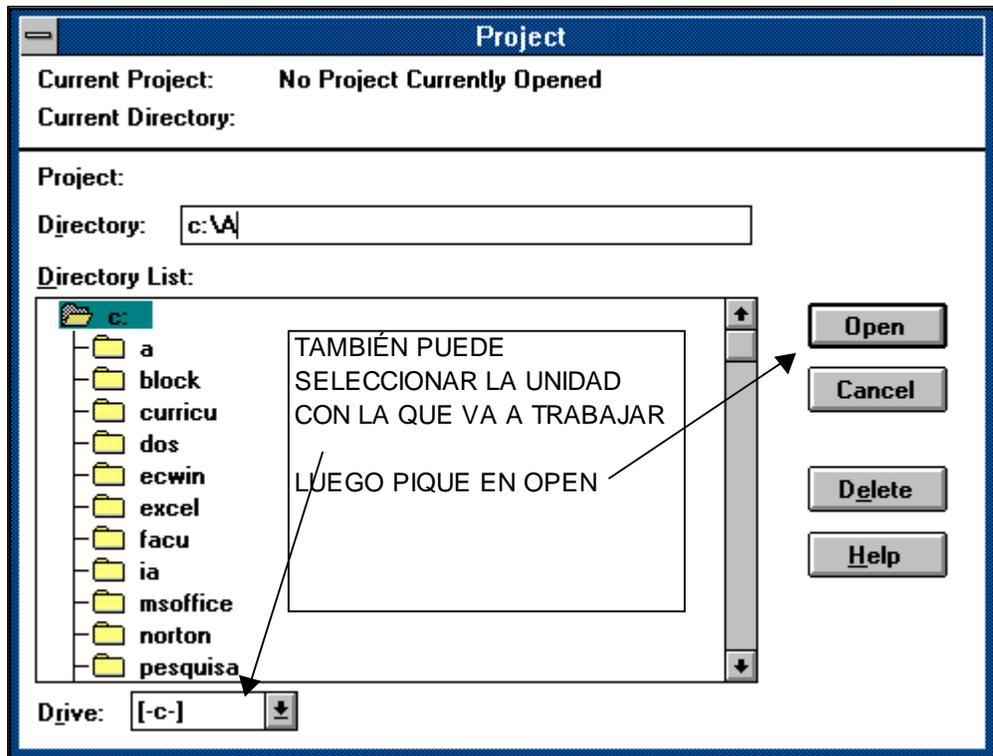
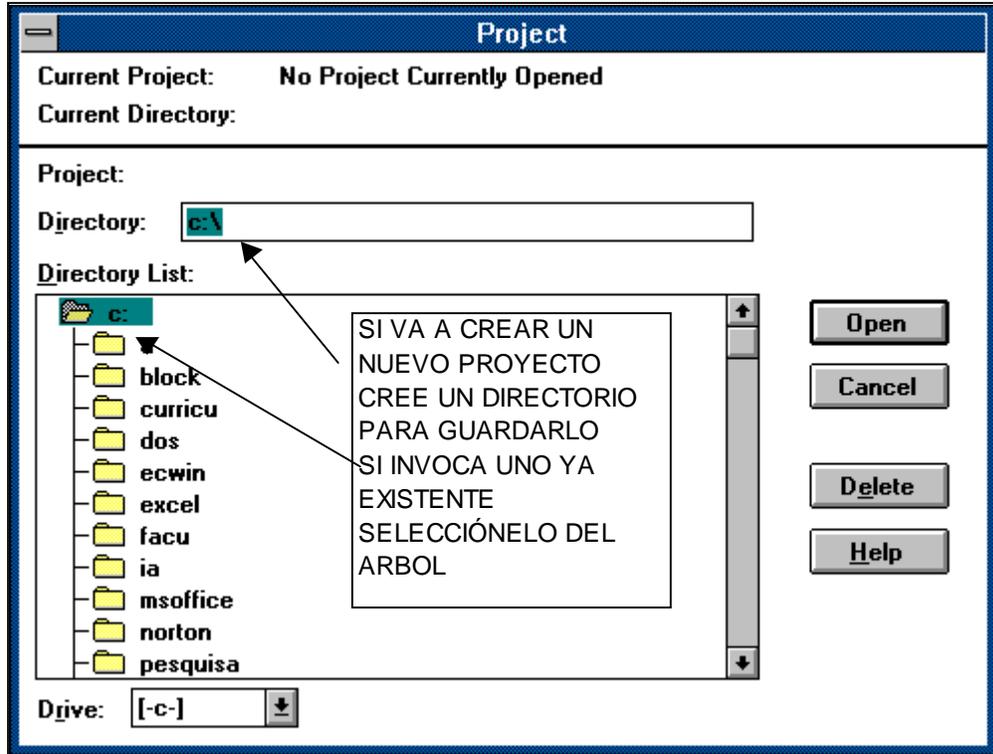
Es por ello que preferí presentar a las herramientas CASE, desde un producto determinado y al estilo de un manual de uso para el usuario.

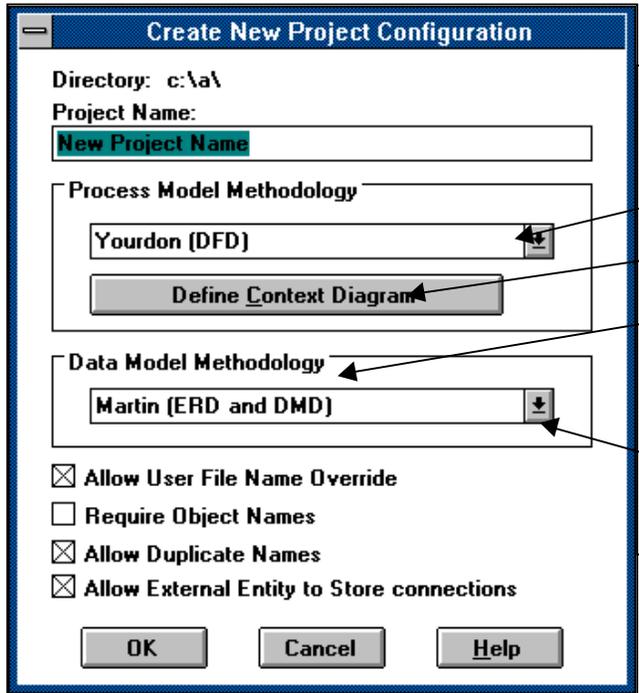
Otro elemento importante que creo conveniente destacar es que la herramientas CASE, son eso **HERRAMIENTAS**, que como tales permiten aumentar la productividad en el desarrollo de un proyecto y como tales deben ser aplicadas a una metodología determinada; nunca piense que ellas le solucionarán todos sus problemas o peor que eso, que ellas en sí mismas son una metodología; su uso está restringido a la metodología elegida para llevar adelante el análisis y diseño del proyecto.

Bien he aquí una manera de cómo opera una herramienta CASE de porte medio.

COMO OPERA UNA HERRAMIENTA CASE





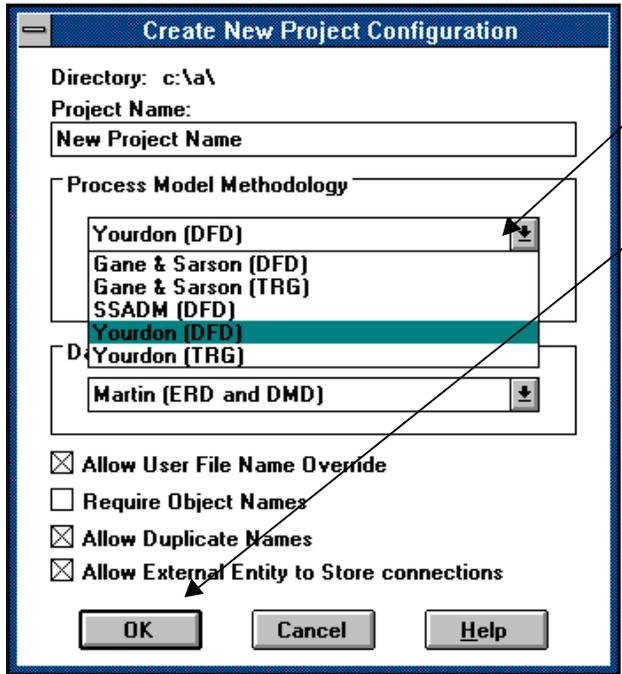


EN UN NUEVO PROYECTO DEBE SELECCIONAR LA METODOLOGÍAS A UTILIZAR

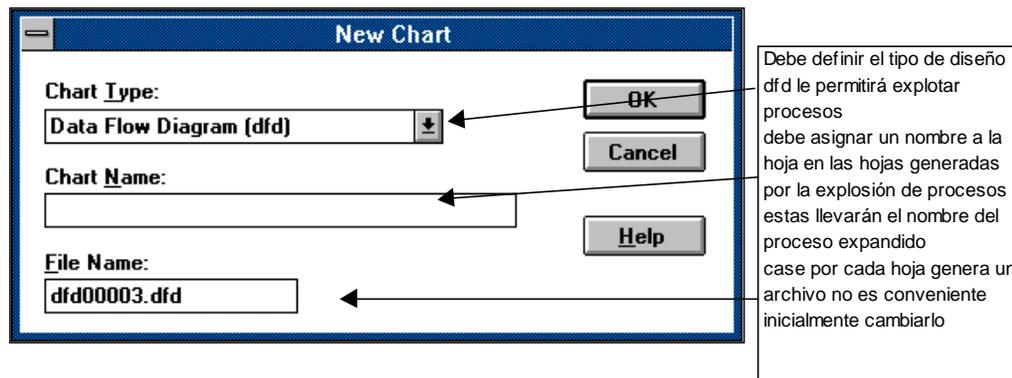
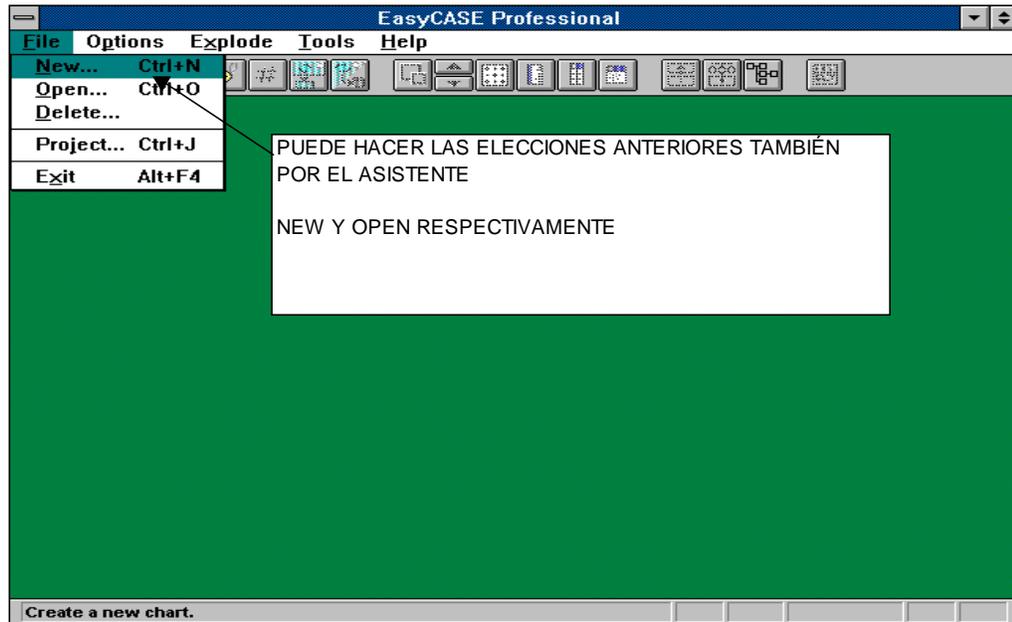
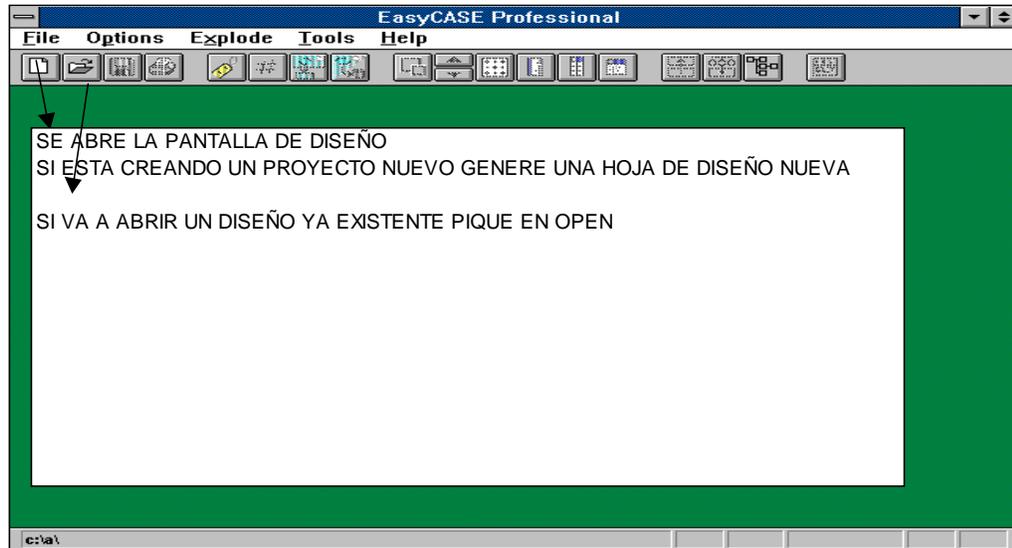
MÉTODO DE MODELIZACIÓN

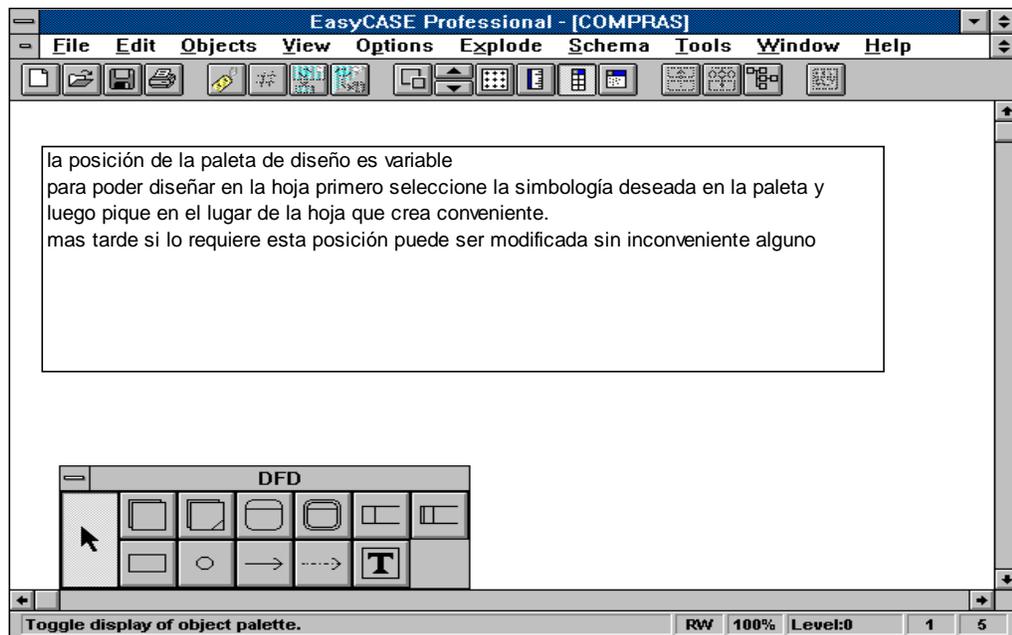
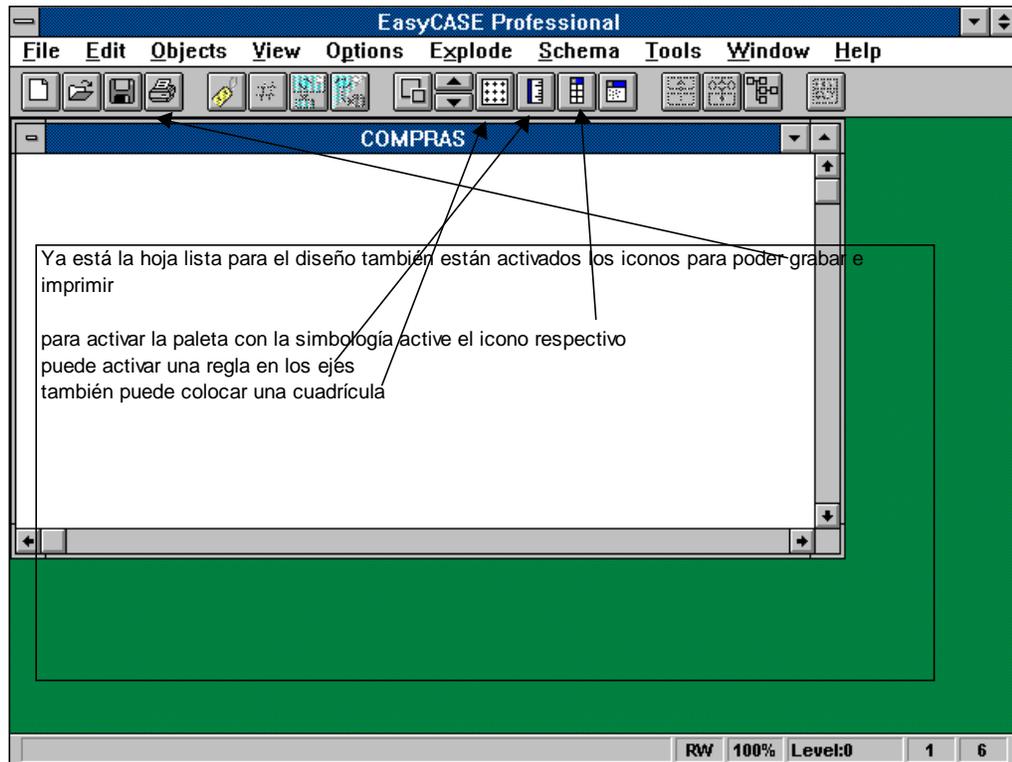
SI SE VÁ A DEFINIR UN DIAGRAMA DE CONTEXTO PIQUE SI NO QUIERE INICIAR CON UN DIAGRAMA DE CONTEXTO NO PIQUE

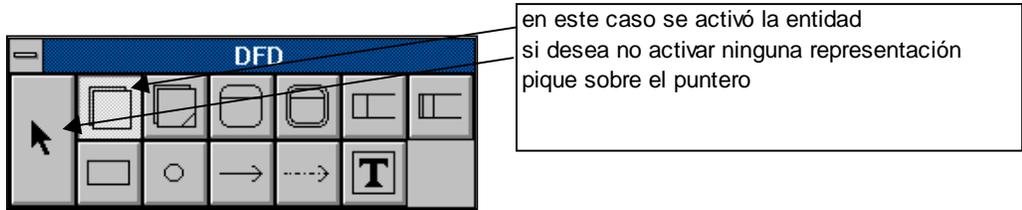
Y PUEDE SELECCIONAR EL MÉTODO DE MODELIZACIÓN DE LOS DATOS



SELECCIONADO DE LAS LISTAS RESPECTIVAS LAS METODOLOGÍAS DE MODELIZACIÓN DE EL OK







Name Object

Chart Object Type: External Entity

Chart Object Name:

OK Cancel Help

Name Object

Chart Object Type: External Entity

Chart Object Name:

OK Cancel Help

1

Define Child Object

Chart Object Name: ALMACENES
Chart Object Type: External Entity

Child Type:

Child Name:

Child File Name:

OK Cancel Break Link Help

2

Define Child Object

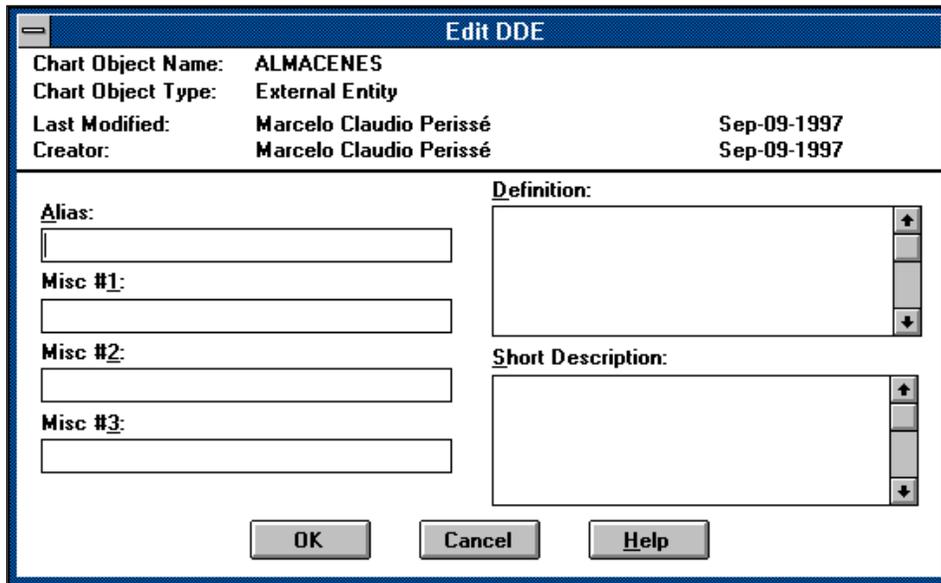
Chart Object Name: ALMACENES
Chart Object Type: External Entity

Child Type:

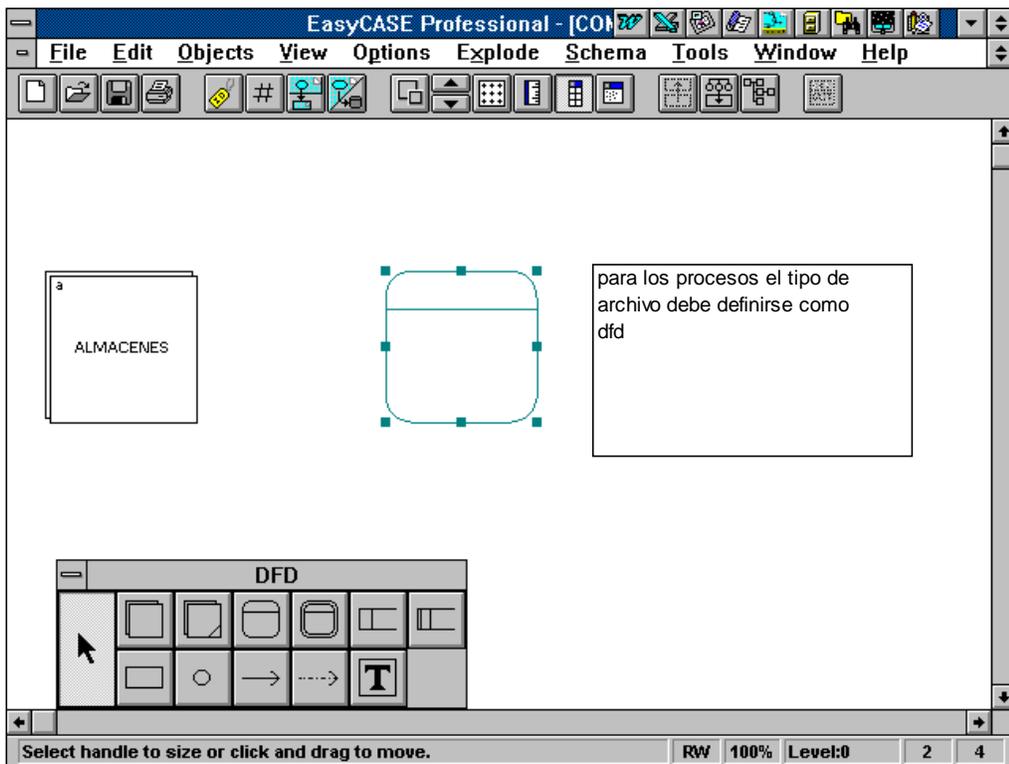
Child Name:

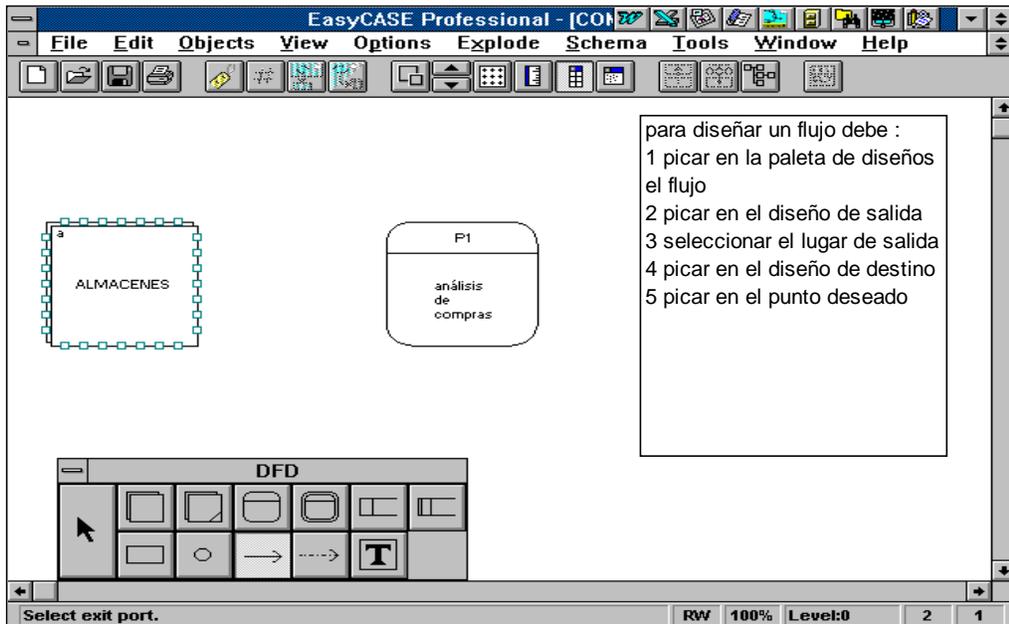
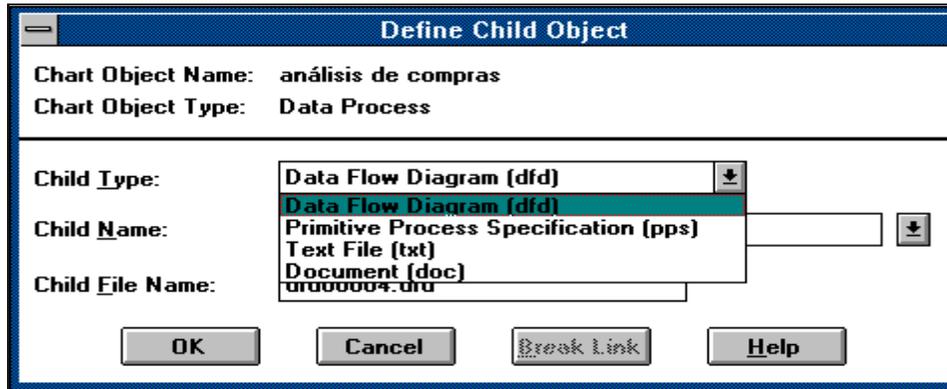
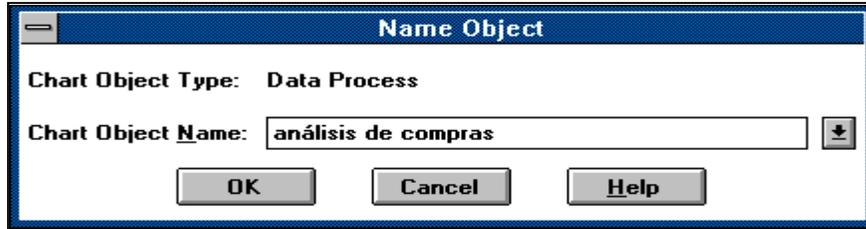
Child File Name:

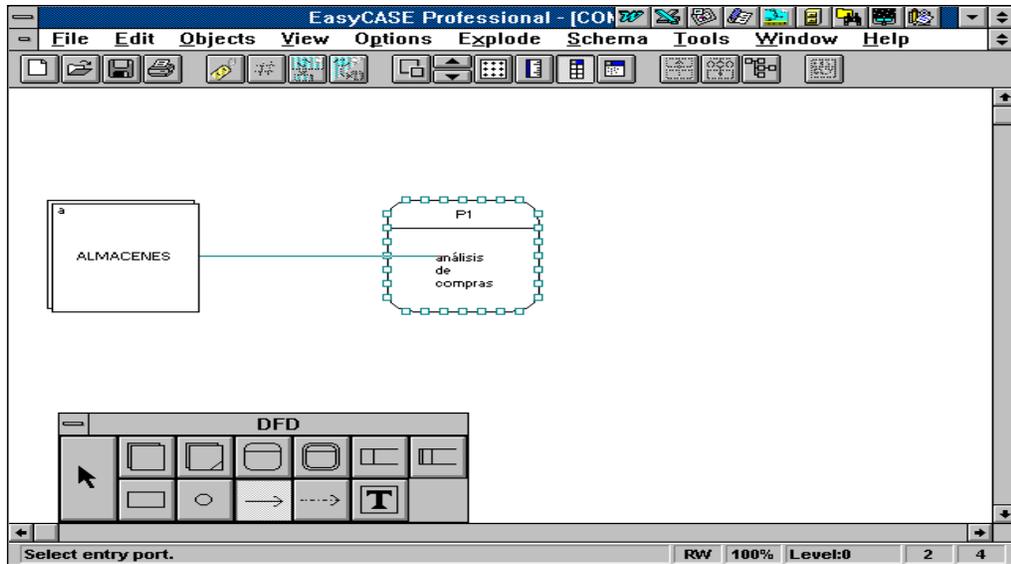
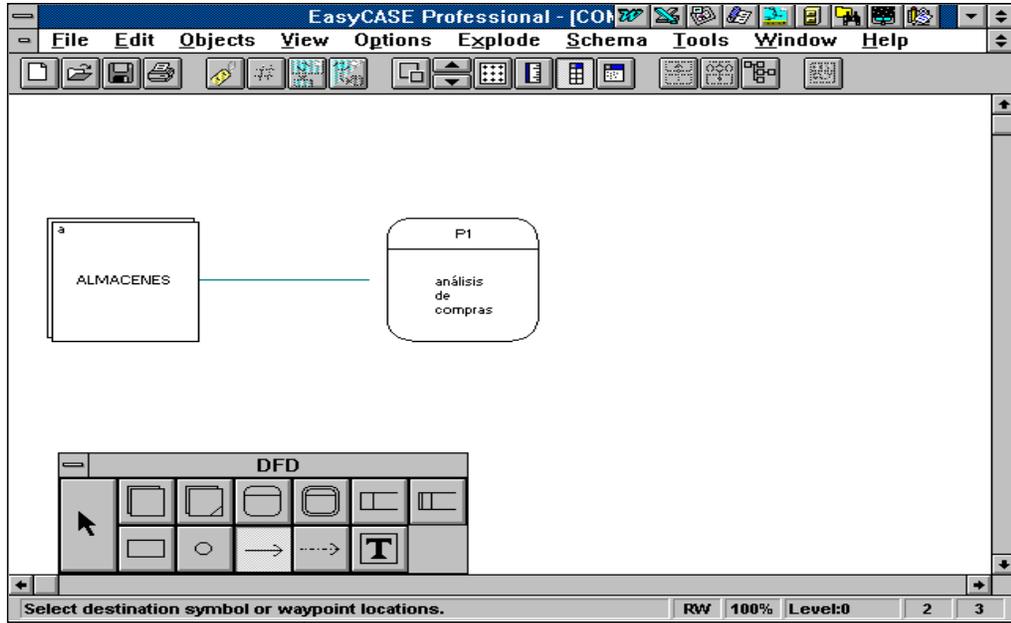
OK Cancel Break Link Help

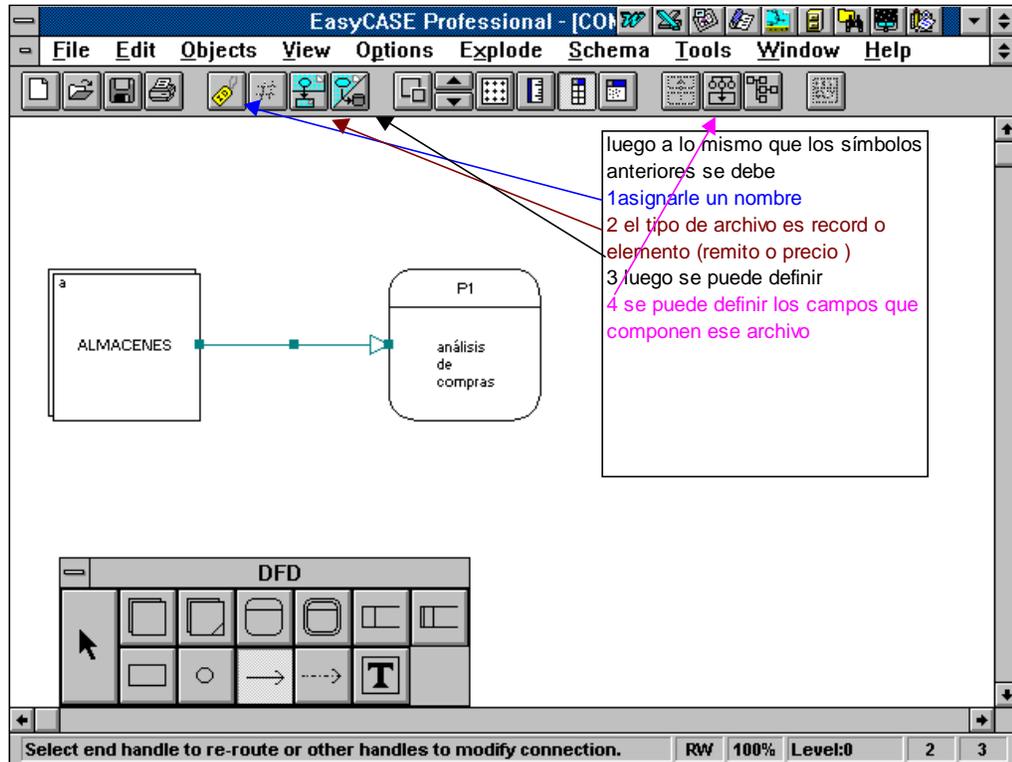


3









Name Object

Chart Object Type: Data Flow

Chart Object Name:

OK Cancel Help

Define Child Object

Chart Object Name: pedido

Chart Object Type: Data Flow

Child Type:

Child Name:

Child File Name:

Record
Record
Element
Text File (txt)
Document (doc)

OK Cancel Break Link Help

Record DDE Screen

Record Name: pedido
 Last Mod:
 Creator: Marcelo Claudio Perissé Sep-09-1997

Definition:
 Table Name:
 Index Name: Unique Index

BNF	Component Name	BNF Key	FK, AK, IE
=	<input type="text"/>	<input type="checkbox"/>	N

1 = N

Record DDE Screen

Record Name: pedido
 Last Mod:
 Creator: Marcelo Claudio Perissé Sep-09-1997

Definition:
 Table Name:
 Index Name: Unique Index

BNF	Component Name	BNF Key	FK, AK, IE
+	<input type="text"/>	<input type="checkbox"/>	N

1	= descripción		N
2	+ cantidad		N
3	+		N

se puede definir el tipo de campo (numérico, alfa data, etc.) como la extensión, etc.

Element Editor

Element Name: cantidad
 Last Mod:
 Creator: Marcelo Claudio Perissé Sep-09-1997

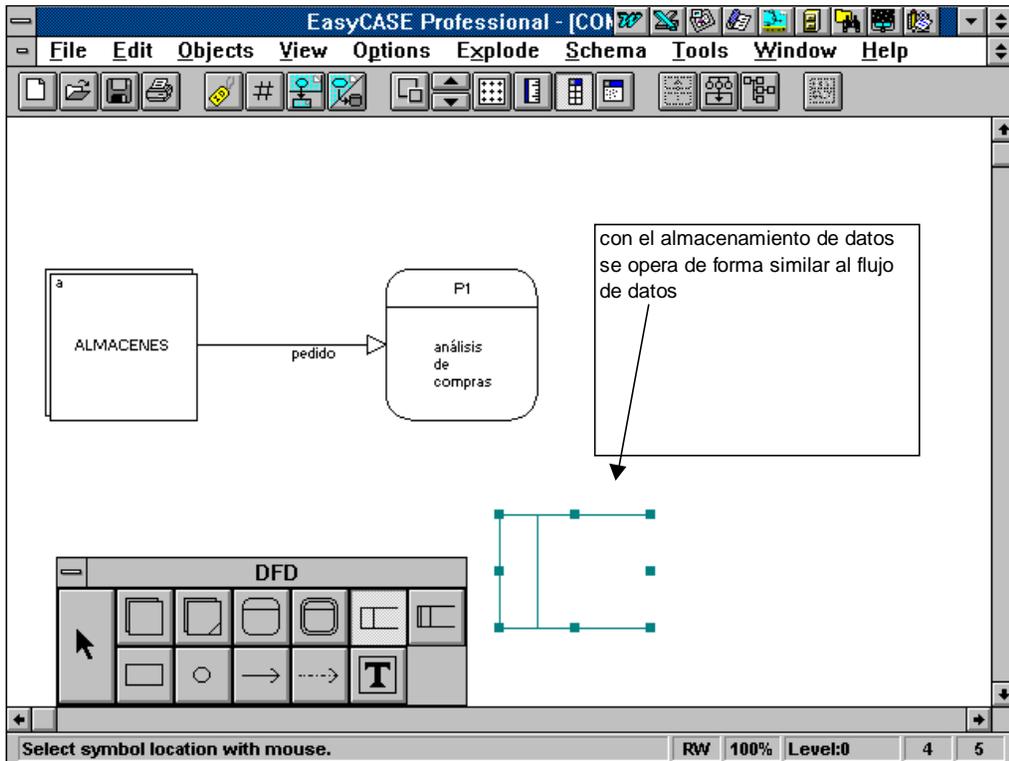
Name: Format:

Type: Misc #1:

Length: Misc #2:

Constraint:

OK Cancel Defn Describe Help



Name Object

Chart Object Type: Data Store

Chart Object Name: 

6. DESARROLLO

El administrador de un proyecto informático debe buscar la máxima automatización de las tareas que realizarán cada uno de los profesionales involucrados en un proyecto informático. Es importante destacar que lo que buscamos no es solamente que en todo proyecto informático se esté dispuesto a automatizar tareas requeridas por los usuarios; sino también la de automatizar las propias tareas del proyecto.

CARACTERÍSTICAS EN TODA METODOLOGÍA DE PROCESAMIENTO DE DATOS³⁴

A continuación presentamos una lista de atributos, que se consideran mínima en todo procesamiento de datos:

- **Automatización:** Como venimos diciendo, se debe buscar la máxima automatización posible de todas las tareas desarrolladas por los profesionales involucrados en un proyecto informático. Se debe evitar la programación manual; pues ésta es lenta y propensa a errores, por lo tanto es ineficaz e ineficiente.
- **Velocidad:** Tal lo visto en el primer capítulo ([ver 1. Proyecto informático, tareas y recursos](#)) otro de los problemas principales, en el desarrollo de todo proyecto informático, es el tiempo que involucra al mismo. Persiga altos niveles de productividad, aplicando técnicas y metodologías que le permitan alcanzar resultados rápidamente.
- **Cambiabilidad.** Cuando vimos las causas que dan inicio a un proyecto informático ([ver 1.2. inicio de un proyecto informático](#)), describimos que existirán cambios en el contexto o en los procedimientos requeridos por los usuarios o bien pueden producirse cambios en la tecnología; que implicarán cambios en los programas y en los sistemas. Es por eso que se deben aplicar técnicas y metodologías que permitan realizar dichos cambios, sin que esto involucre un incremento significativo tanto de los costos y como en el tiempo de implementación de estos cambios.
- **Verificación de condición correcta.** Confeccione y utilice herramientas de análisis, como el diccionario de datos ([ver 4.3 el diccionario de datos](#)), las tablas de decisión ([ver 4.6. tablas de decisión](#)), la diagramación lógica ([ver 4.5. flujogramas](#)), la lista de eventos ([ver 4.2.1. lista de eventos](#)); para poder detectar automáticamente todos los errores de sintaxis y de semántica interna. Si existen ambigüedades, contradicciones, incongruencias, la calidad del sistema se verá afectada, con todo lo que ello implica. Los errores provocan ineficiencia ineficacia y baja productividad
- **Técnicas que faciliten la comunicación con los usuarios finales.** Los usuarios deben desarrollar el conocimiento necesario para verificar cada etapa de evolución del proyecto. El usuario es quien más sabe del sistema involucrado en el proyecto. Además los usuarios deben estar en condiciones de utilizar sus propios lenguajes de consulta de actualización y de generadores de información; como: el Standard Query Language (SQL), el Query - By - Example (QBE), el Query - by - Diagram (QBD) o el Graphics Language for Database, entre otros. Por lo tanto se deben adoptar lenguajes que permitan que la gerencia extraiga nueva información de las bases de datos, con la máxima prontitud posible.
- **Diseño estable de base de datos.** La base de datos es el elemento principal de toda automatización de tareas. Tal cual lo visto en el tópico de la modelización de datos almacenados ([ver 4.4. el modelo RDM](#)) cuida las técnicas y los métodos para la construcción de las tablas.
- **Modularidad.** Los sistemas deben dividirse en módulos fácilmente identificables ([ver 4.7. módulos de un sistema](#)). Debe ser factible efectuar cambios en forma local dentro del módulo. Todo efecto de cambio exterior al módulo debe ser rigurosamente rastreable.

- **Control de operabilidad mutua.** Se necesita una técnica formal y rigurosa, para tener la seguridad de que el sistema y los módulos desarrollados separadamente operan correctamente en conjunto ([ver 4.7. el árbol de un sistema](#)).
- **Dialectos alternativos.** Se debe disponer de herramientas de ingeniería de software([ver. 5 herramientas CASE](#)) para conceptualizar, dibujar y diseñar sistemas, conectados en forma automática con la representación básica. Estas herramientas deben funcionar en forma integrada, evitando puentes manuales que introducen errores. Deben utilizar, en la medida posible, sintaxis y gráficos comunes.

Una propuesta interesante de destacar es la que propone Lucas H.C. Jr.. con el diseño creativo de sistemas³⁵, este modelo tiene básicamente tres componentes:

1. diseño controlado por el usuario
2. atención especial a las interacciones con el usuario
3. evaluación de la calidad de los sistemas según el criterio del usuario

El diseño controlado por el usuario significa que el usuario está a cargo del esfuerzo de diseñar

Esto crea un compromiso del usuario con el sistema aumentando la posibilidad de ser utilizado

El usuario participa activamente durante el diseño y por lo tanto está mejor preparado para usar el sistema, en razón de su familiaridad con él.

El usuario está a cargo del diseño lógico o conceptual del sistema incluyendo las salidas, las entradas y la lógica del procesamiento. El usuario en escribe ni controla programas estos pueden ser desarrollados con lenguajes de 4 generación y ser controlados con herramientas CASE.

El usuario creativo se basa en el control del diseño por parte del usuario, atención especial a las interacciones de éste con el sistema y evaluación de su calidad de acuerdo con el criterio del mismo usuario.

6.1. METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS

A lo largo de este texto, buscamos mostrar que toda actividad debe estar basada en una metodología y en principio, cualquier metodología es mejor que ninguna; Cualquier centro de desarrollo puede montar su metodología, aunque esta alternativa implica disponer del tiempo necesario para el desarrollo de la nueva metodología; por lo tanto, lo más práctico es seguir los métodos que ya han demostrado su validez y son de aplicación universal; sepa utilizar el conocimiento científico, que involucra tanto esfuerzo y sacrificio.

Todas las metodologías; MERISE, YOURDON Y SSADM (structured System Analysis Design Method) y tantas otras, consideran el hecho informático dividido en fases, cuyo conjunto forma el ciclo de vida de un sistema informático.

Todas tienen en común la idea de descomposición del hecho informático en cuatro grandes grupos

❖ **Análisis**

- definición del problema
- estudio de la situación actual
- requisitos a considerar
- estudio de factibilidad

❖ **Diseño lógico**

- análisis funcional

- definición de datos y procesos
- modelización
- ❖ **Diseño físico**
 - creación de ficheros y tablas
 - elaboración de programas
- ❖ **Implementación y control**
 - Formación del usuario
 - implantación del sistema
 - explotación del sistema
 - Mantenimiento

Esta metodología la podrá encontrar en un amplio universo bibliográfico, nosotros nos concentraremos, como lo describimos en la introducción de la obra en las metodologías simplificadas.

6.2. METODOLOGÍA ESTRUCTURADA SIMPLIFICADA.

Todo proceso de desenvolvimiento de software usando metodología Estructurada simplificada está basado en la identificación de los eventos a los que el sistema debe responder.

La secuencia metodológica es al siguiente:

1. Definir la lista de eventos

Desarrollar una lista de requerimientos en lenguaje natural según lo descrito en el punto 4.2.1.

2. Producir un diagrama de contexto

Modelizar la relación del sistema con el contexto, determinando cuales son las áreas de la empresa que participarán del sistema como fuentes de información ([ver 4.2.2. El diagrama de flujo de datos, objetivos](#)).

3. Definir el modelo comportamental

Utilizamos el DFD como herramienta modeladora de la transformación de las entradas en salidas ([ver 4.2.2. el diagrama de flujo de datos](#)).

4. Definir el modelo de datos

Modelizar la relación de los repositorios de datos con la técnica del Modelo Relacional de Datos. -RDM ([ver 4.4. Modelización de datos almacenados](#)).

5. Crear el modelo de implementación del usuario

Definir los módulos del sistema. En esta etapa son decididos los procesos a ser automatizados; se somete a la evaluación del usuario cada proceso del modelo comportamental ([ver 4.7. módulos del sistema](#)).

6. Definir los requisitos de implementación

Mientras son definidos los procesos a ser informatizados, se debe discutir y documentar los requisitos de implementación de esos procesos y del sistema de software como un todo: Desempeño, restricciones de costos, restricciones operacionales, consideraciones sobre seguridad y auditoría, tecnología a ser empleada, modificaciones en procedimientos manuales y en otros sistemas informatizadas ya existentes.

7. Elaborar diagramas de estructura. ([ver 4.7. el árbol de un sistema](#))

Para cada proceso a ser automatizado, será creado un diagrama de estructura. Las funciones de los diagramas son derivadas de los flujos de datos que entran y que salen de los procesos, y de las transformaciones que generan los datos de salida a partir de los datos de entrada.

8. Integrar los diagramas de Estructura.

Los diagramas de estructura deben ser integrados en programas, el agrupamiento de funciones puede ser hecho por proximidad temporal de utilización, rutinas On-Line, mensual, anual, etc., o por cualquier otro tipo de afinidad, como por ejemplo, en el caso de sistemas distribuido, el agrupamiento es hecho conforme al procesador en que serán ejecutadas las funciones. La estructura del software es completada, incorporándose a él módulos de apoyo operacional, como: módulos de implementación de backups, módulos de control, módulos para la creación y restauración de índices, módulos para alteración de parámetros de operaciones, etc. estos módulos serán incorporados al Diagrama de estructura, donde el acceso a ellos fuese mas conveniente

9. Proyectar la interfaz con el usuario

La parte mas importante y mas compleja de la interfaz con el usuario será desarrollada a partir de los flujos de datos de entrada y de salida de los procesos a ser automatizados. Una única interfaz puede ser generada para atender varios flujos simultáneamente. Las interfaces necesarias a los módulos que implementan menús de selección y a los módulos de apoyo operacional complementaran el proyecto de la interfaz con el usuario.

10. Proyectar la base de datos física

Definir las características físicas de cada dato, como el tipo el dominio; la organización de cada archivo, como la definición de las llaves principales, índices, etc. (ver 3. [Base de datos](#), [3.1.2. llave primaria](#), [3.1.3 índices de acceso](#)).

11. Especificar los módulos.

La especificación de los módulos, a través de pseudo código flujogramas u otros ([ver.4.5. Flujogramas](#)).

APÉNDICE A

EJERCITACIÓN PRÁCTICA DE:

- ❖ LITA DE EVENTOS
- ❖ DIAGRAMA DE FLUJO DE DATOS

En el presente apéndice se pretende practicar la construcción de la lista de eventos y los diagramas de flujo de datos.

EJERCICIONº:1

PROCEDIMIENTO: COMPRAS MAYORES Y RECEPCION

DESCRIPCIÓN DEL PROCEDIMIENTO:

Sector Almacenes. :Los días lunes de todas las semanas (el día hábil inmediato siguiente si aquel fuera no laborable) el encargado del sector recorre las **FICHAS DE EXISTENCIAS** que se llevan en el mismo y, en caso de considerar que el remanente de un ítem se encuentra por debajo de la cantidad que habitualmente se mantiene como aceptable (criterio personal del encargado, basado en su experiencia), ordena telefónicamente al **Sector Compras Locales** la iniciación del *trámite de adquisición* de la cantidad que él estima necesario reponer.

SECTOR COMPRAS LOCALES. : Si la compra no reviste carácter de urgente, en el sector se consulta la *guía de proveedores* y se invita por *nota* a cotizar (se envía una *solicitud de cotización*) a un cierto número de posibles **proveedores**, cuya cantidad depende del tipo de artículo a adquirir, quedando esta última decisión a cargo del jefe del sector.

Si la compra es urgente, porque así la requirió **almacenes**, en el sector **Compras** se solicita el *pedido* telefónicamente al **proveedor** que se estima responde la entrega con mayor celeridad.

La *solicitud de cotización* (caso de compras no urgentes) se emite en original y una copia. el original se envía a los **proveedores** y el duplicado se *archiva* por fecha de emisión.

Cuando se reciben las *cotizaciones*, los sobres se van abriendo a medida de su legada.

El jefe del sector **compras locales** efectúa la evaluación del contenido de las *cotizaciones* y decide la asignación.

Emite la correspondiente *Orden de compra* en original y tres copias.

Distribuye los ejemplares con el siguiente destino:

original envía por correo al **proveedor**;1 copia *archiva* en orden cronológico; 2 copia a **administración**;3 copia a **Contabilidad de Almacenes**

- *archiva* definitivamente la *cotización recibida* , por proveedor, en orden alfabético.

SECTOR RECEPCIÓN : Al llegar la mercadería proveniente de un **proveedor** junto con el *remito* de este (que entrega en original y duplicado) consulta telefónicamente al sector **compras** a efectos de comprobar la existencia real de un pedido, coincidente con el material que se recibe.

Si hay coincidencia firma de conformidad el *remito original* del **proveedor**, que devuelve a éste, aún cuando la recepción sea provisional, y retiene una *copia del remito* (duplicado).

Emite un *informe de recepción* en original y tres copias. Distribuye estos ejemplares de la siguiente manera.

original. al sector **compras** ;*Duplicado* y *Triplicado* al **sector almacenes**; Cuadruplicado. *archiva* en el sector por orden cronológico.

envía el duplicado del *remito* del proveedor, conformado, a **contaduría**.

SECTOR COMPRAS LOCALES :Recibe el original del *informe de recepción* lo *archiva* por orden cronológico.

SECTOR ALMACENES : Al recibir el duplicado y triplicado del *informe de recepción* actualiza la ficha de existencias y *archiva* ambos ejemplares por orden cronológico

LISTA DE EVENTOS

1. ALMACENES recorre **FICHAS DE EXISTENCIA**
2. ALMACENES determina si la compra es urgente
3. ALMACENES ordena adquisición al sector **COMPRAS LOCALES**

si la compra es no urgente

4. **COMPRAS LOCALES** inicia **TRAMITE DE ADQUISICIÓN**
5. **COMPRAS LOCALES** consulta **GUÍA DE PROVEEDORES**
6. **COMPRAS LOCALES** emite **SOLICITUD DE COTIZACIÓN**
7. **COMPRAS LOCALES** envía **SOLICITUD DE COTIZACIÓN** a los **PROVEEDORES**
8. **COMPRAS LOCALES** archiva **SOLICITUD DE COTIZACIÓN**_(por fecha de emisión)

si la compra es urgente

9. **COMPRAS LOCALES** solicita **PEDIDO TELEFÓNICO** a un **PROVEEDOR**

10. **COMPRAS LOCALES** recibe **COTIZACIONES** de los **PROVEEDORES**
11. **COMPRAS LOCALES** abre los sobres de **COTIZACIONES** (secuencialmente a su llegada)
12. **COMPRAS LOCALES** evalúa **COTIZACIONES**
13. **COMPRAS LOCALES** decide asignación

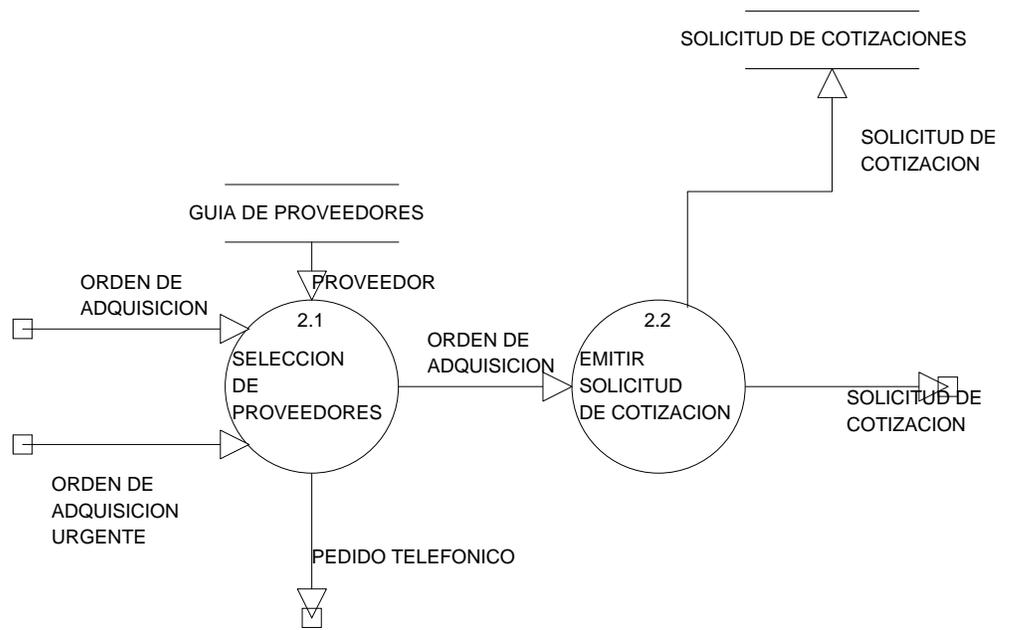
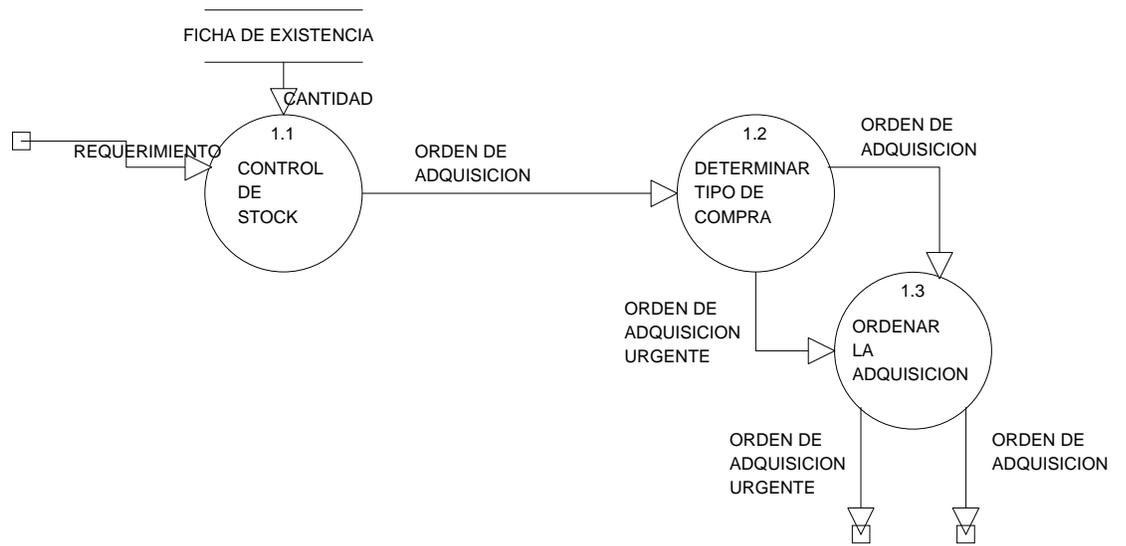
14. **COMPRAS LOCALES** emite **ORDEN DE COMPRA**
15. **COMPRAS LOCALES** envía **ORDEN DE COMPRA** por correo al **PROVEEDOR**(original)
16. **COMPRAS LOCALES** archiva **ORDEN DE COMPRA**(copia)
17. **COMPRAS LOCALES** envía **ORDEN DE COMPRA** a **CONTABILIDAD DE ALMACENES**
18. **COMPRAS LOCALES** archiva **COTIZACIONES RECIBIDAS**(definitivamente por proveedor)

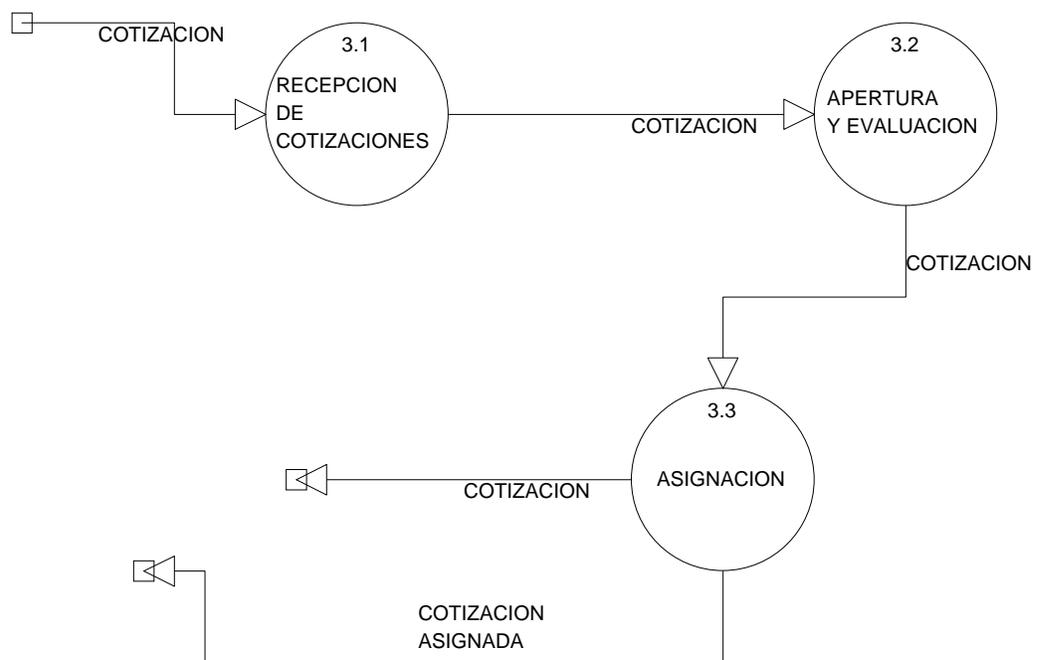
19. **RECEPCION** recibe **REMITO**(original y copia)
20. **RECEPCION** consulta **ORDEN DE COMPRA** (telefónicamente)a **COMPRAS LOCALES** por el pedido
21. **RECEPCION** controla el pedido
22. si hay coincidencia
23. **RECEPCION** firma **REMITO**(original)
24. **RECEPCION** devuelve **REMITO** al **PROVEEDOR**

25. **RECEPCIÓN** emite **INFORME DE RECEPCIÓN** (original y tres copias)
26. **RECEPCIÓN** envía **INFORME DE RECEPCIÓN** al sector **COMPRAS LOCALES** (duplicado)
27. **RECEPCIÓN** envía **INFORME DE RECEPCIÓN** al sector **ALMACENES**(triplicado)
28. **RECEPCIÓN** archiva **INFORME DE RECEPCIÓN** (cronológicamente)

29. **COMPRAS LOCALES** recibe **INFORME DE RECEPCIÓN** (original)
30. **COMPRAS LOCALES** archiva **INFORME DE RECEPCIÓN**(cronológicamente)

31. **SECTOR ALMACENES** recibe **INFORME DE RECEPCIÓN** (duplicado y triplicado)
32. **SECTOR ALMACENES** actualiza **FICHA DE EXISTENCIAS**.
33. **SECTOR ALMACENES** archiva **INFORME DE RECEPCIÓN** (por orden cronológico)





EJERCICIO Nº:2

PROCEDIMIENTO : VENTAS Y GESTIÓN DE DESPACHO

DESCRIPCIÓN DEL PROCEDIMIENTO

1 Los **vendedores** emiten una NOTA DE PEDIDO por cada venta con fecha de entrega a determinar. Las NOTAS DE PEDIDO se emiten en original y tres copias. Se distribuyen las copias de la siguiente forma:

- Original: se entrega al **cliente**
 - Duplicado: se remite a **Créditos**
 - Triplicado: lo archiva definitivamente el **vendedor** por número de pedido
- 2 **Créditos**, con los pedidos recibidos procede a realizar las siguientes tareas en el orden que se indica a continuación:

1. Valoriza cada PEDIDO, o sea determina el valor total neto.
2. Para el control de créditos resta del límite crediticio asignado el saldo de deuda vigente a esa fecha y el valor del PEDIDO.
3. Envía a **Auditoría** los PEDIDOS rechazados, para su verificación
4. Actualiza la cuenta corriente contable en base a los PEDIDOS aprobados
5. Envía los PEDIDOS aprobados a **Marketing**

3 **Stocks**, en base al PEDIDO recibido del **vendedor** controla el Pedido contra le stock proyectado y procede a:

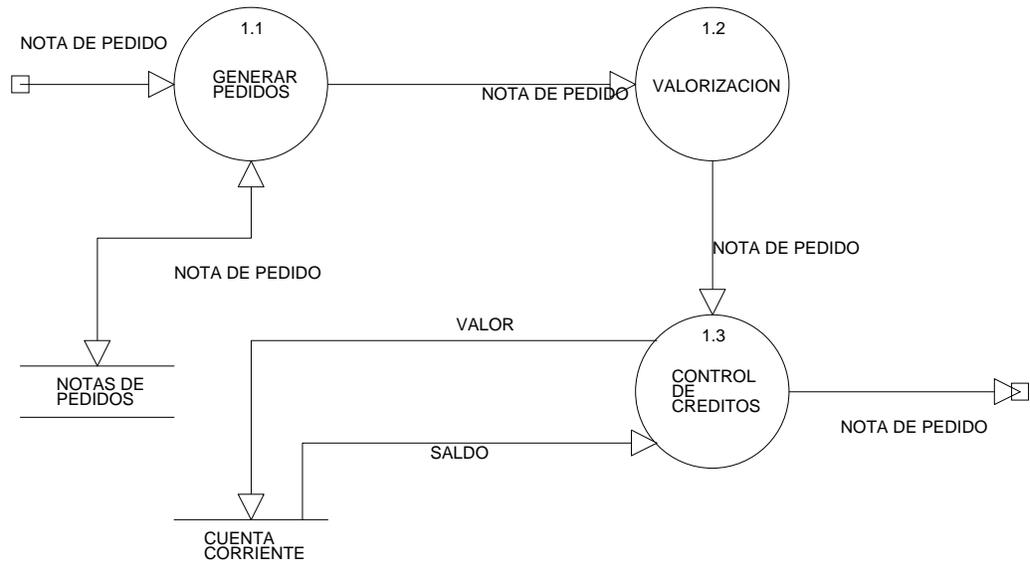
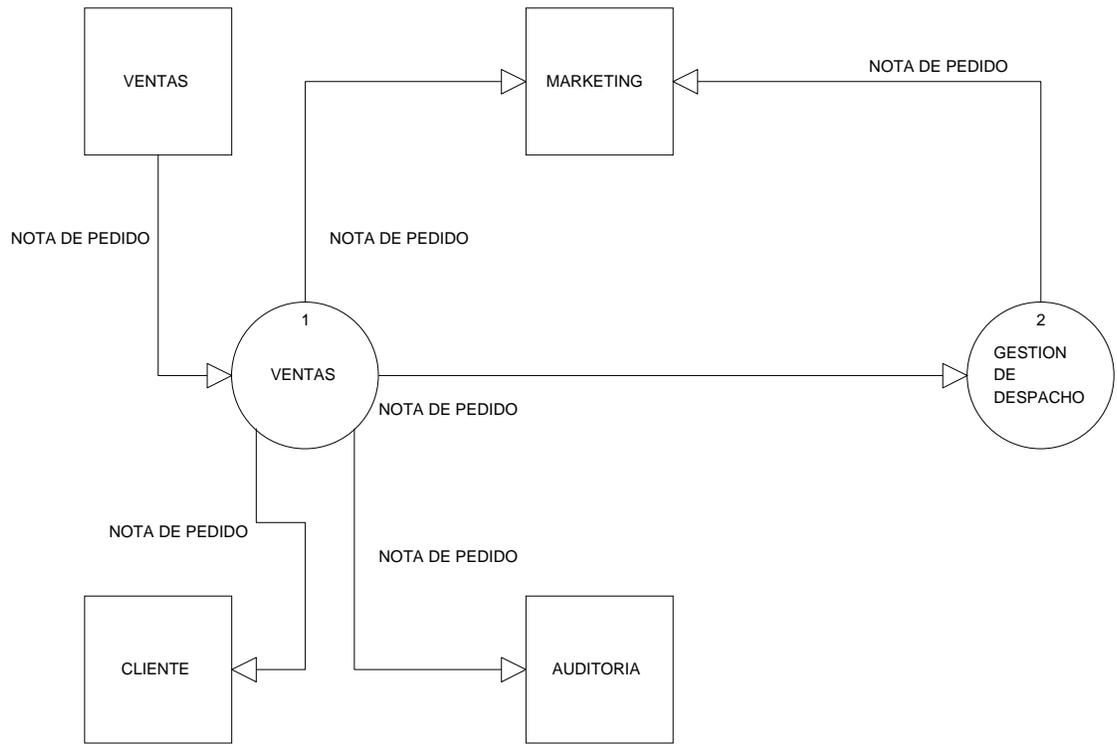
- Actualizar el stock proyectado por los Pedidos aprobados
- Cursar todos los Pedidos (aprobados y rechazados) a **Marketing**.

4 **Marketing** procede a prefacturar los Pedidos aprobados.

LA LISTA DE EVENTOS

1 VENDEDORES emiten NOTA DE PEDIDO con fecha a determinar
VENDEDOR envía NOTA DE PEDIDO (original) al CLIENTE VENDEDOR envía NOTA DE PEDIDO (duplicado) a CRÉDITOS VENDEDOR archiva NOTA DE PEDIDO (triplicado) por número de pedido
2 CRÉDITOS recibe y aprueba PEDIDOS
CRÉDITOS valoriza cada pedido (determina valor total neto) CRÉDITOS controla el CRÉDITO resta del límite crediticio el saldo de deuda vigente a esa fecha resta el valor del pedido
CRÉDITOS envía a AUDITORÍA los PEDIDOS RECHAZADOS (para su verificación) CRÉDITOS con los PEDIDOS APROBADOS actualiza CUENTA CORRIENTE CONTABLE CRÉDITOS envía PEDIDOS APROBADOS a MÁRKETING
3 STOCK controla el PEDIDO con el STOCK PROYECTADO
STOCK actualiza el STOCK PROYECTADO por los PEDIDOS APROBADOS STOCK envía los PEDIDOS (aprobados y rechazados) a MARKETING
4 MARKETING prefactura los PEDIDOS APROBADOS

DIAGRAMA DE FLUJO DE DATOS



EJERCICIO Nº:3

PROCEDIMIENTO: VENTAS

DESCRIPCIÓN DEL PROCEDIMIENTO:

El **vendedor**, por cada venta, emite una nota de pedido en original y dos copias. Procede a la siguiente distribución, previa firma por parte del **cliente** y también por la suya:

- original: al **cliente**.
- duplicado: a **Administración de Ventas**.
- triplicado: a **Auditoría**.
- **Auditoría** controla los precios y los cursa a **Contaduría**, donde se archiva a provisoriamente
- **Administración de Ventas** confecciona una estadística de pedidos ingresados y los cursa a **Créditos y Cobranzas**.
- **Créditos y Cobranzas** valoriza los pedidos y procede a efectuar el control de créditos contra la deuda vigente de los clientes. En caso de ser pedidos a cobrar contra entrega, no efectúa el control de créditos y los envía directamente a **Almacén**. Con los pedidos aprobados actualiza la deuda vigente y con los rechazados procede a su archivo definitivo
- . **Almacén** consulta el stock y procede a emitir remito/factura por las entregas del día y a archivar por cliente los pedidos no entregados.
- **Almacén** carga la mercadería en el camión y procede a su despacho.

LISTA DE EVENTOS

VENDEDOR emite NOTA DE PEDIDO (original y dos copias) VENDEDOR firma con el cliente la NOTA DE PEDIDO VENDEDOR distribuye NOTAS DE PEDIDO Cliente (original) Administración de Ventas (duplicado) Auditoría (triplicado)
--

AUDITORÍA controla los precios de la NOTA DE PEDIDO AUDITORÍA cursa NOTA DE PEDIDO a CONTADURÍA
--

CONTADURÍA archiva NOTA DE PEDIDO (provisoriamente)
--

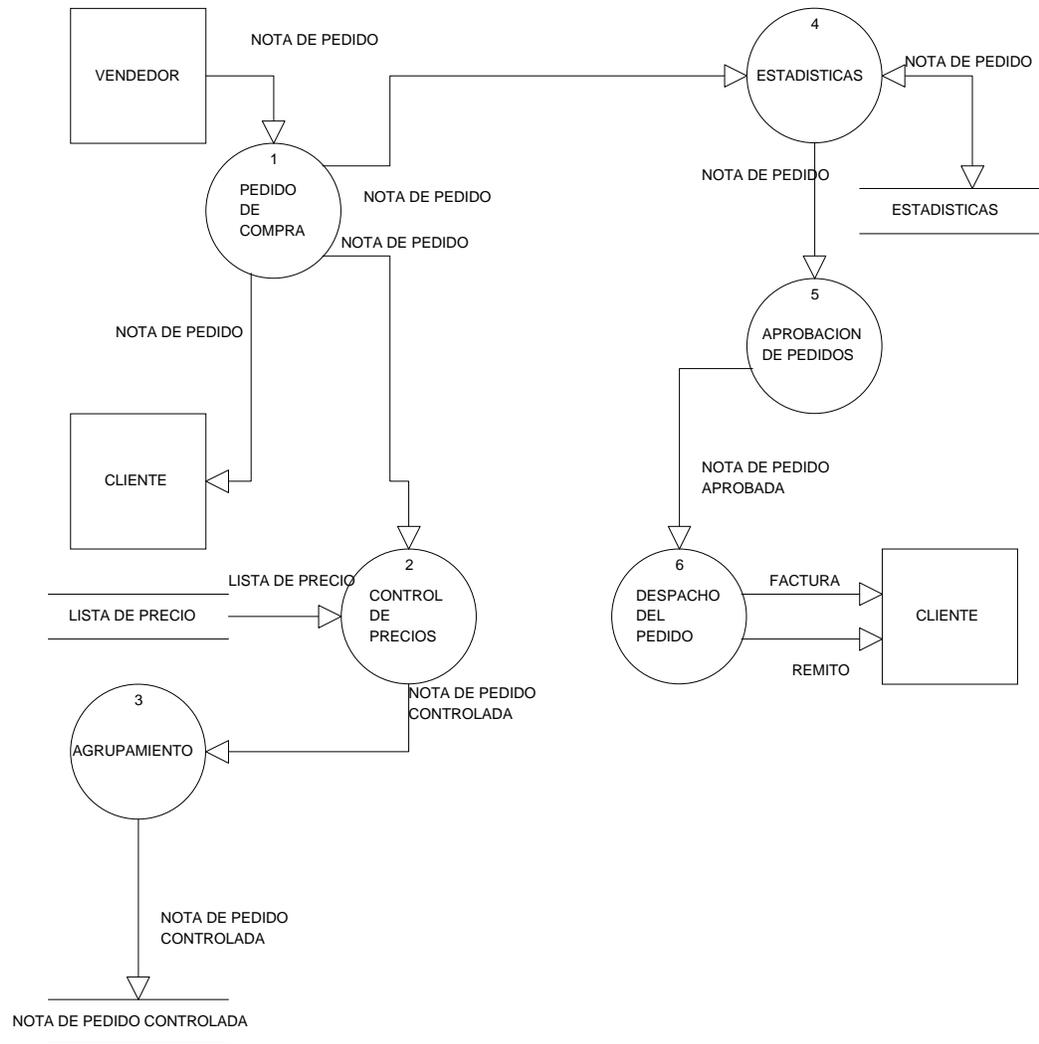
ADMINISTRACIÓN DE VENTAS confecciona ESTADÍSTICA DE PEDIDOS INGRESADOS ADMINISTRACIÓN DE VENTAS cursa NOTA DE PEDIDO a CRÉDITOS Y COBRANZAS
--

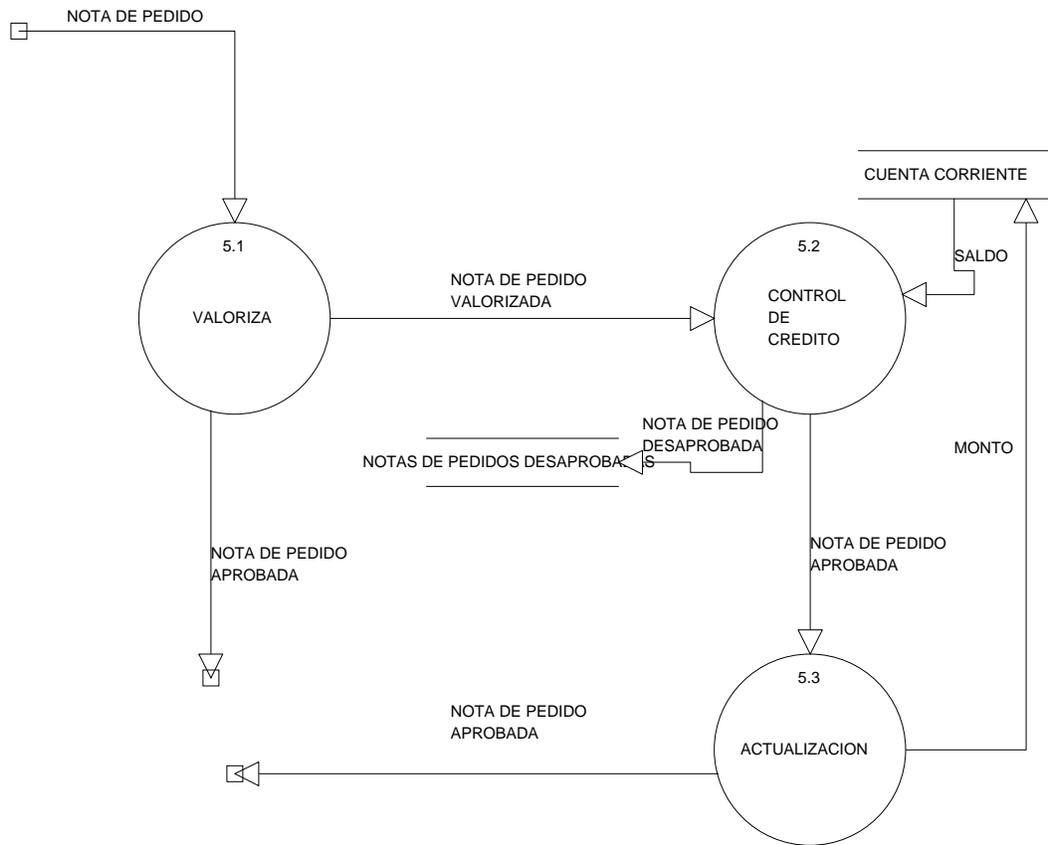
CRÉDITOS Y COBRANZAS valoriza NOTA DE PEDIDO CRÉDITOS Y COBRANZAS verifica si es pago contra entrega CRÉDITOS Y COBRANZAS si es el pago contra entrega envía a ALMACENES CRÉDITOS Y COBRANZAS si no es pago contra entrega controla CRÉDITOS con la deuda vigente del cliente CRÉDITOS Y COBRANZAS con los pedidos aprobados actualiza deuda vigente y los envía a ALMACÉN
--

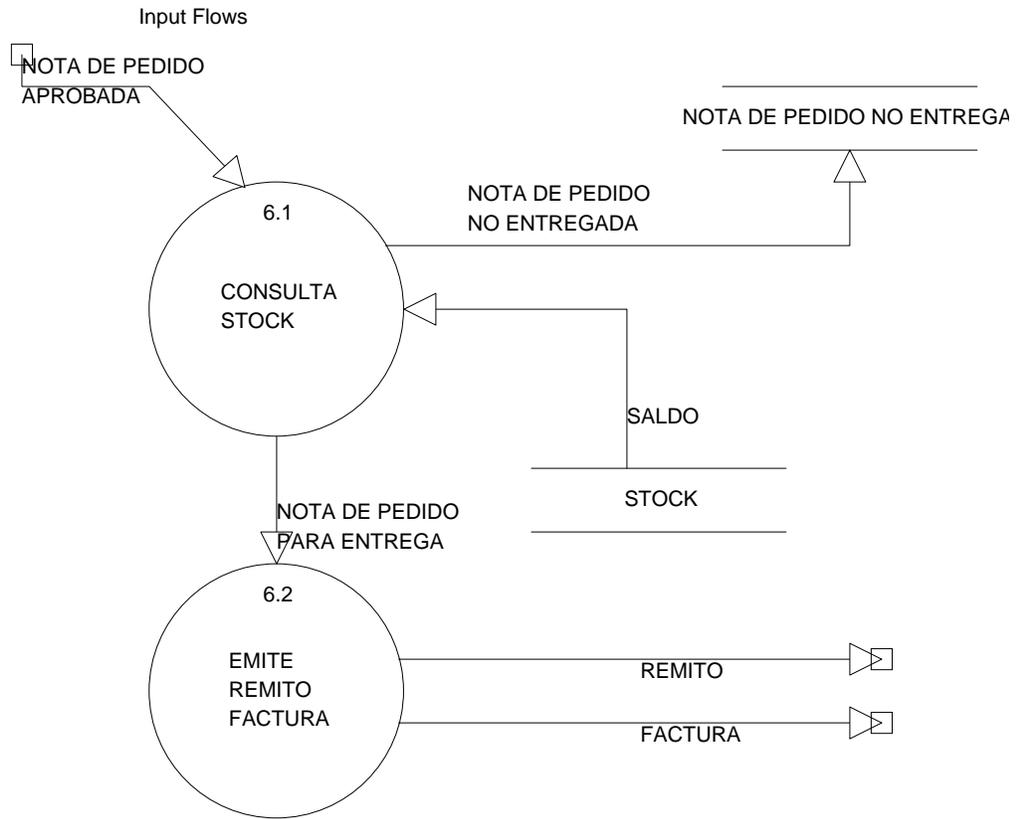
CRÉDITOS Y COBRANZAS con los **PEDIDOS RECHAZADOS** archiva
(definitivamente)

ALMACENES consulta **STOCK**
ALMACENES emite **REMITO FACTURA**
ALMACENES archiva (por cliente) **NOTAS DE PEDIDO NO ENTREGADAS**

DIAGRAMA DE FLUJO DE DATOS







EJERCICIO Nº:4

Con la letra @ se han identificado a posibles llaves candidatas; y con ◇ se han identificado las alternativas.

PROCEDIMIENTO: COMPRAS

LISTA DE EVENTOS

1. SOLICITUD DE COMPRA

- 1.1. ALMACENES controla el **SOTCK** y detecta necesidad.
 - @ Punto de pedido
 - Ingreso pendiente (Nota de pedido solicitada y no entregada)
 - Entregas parciales.
- 1.2. ALMACENES emite **SOLICITUD DE COMPRA** (original y duplicado).
 - Código del producto
 - Nombre del producto
 - Cantidad
 - Calidad requerida.
- 1.3. ALMACENES envía **SOLICITUD DE COMPRA** (original) a COMPRAS.
- 1.4. ALMACENES archiva (transitoriamente) **SOLICITUD DE COMPRA** (duplicado),
 - Orden @ **Cronológico** (a la espera de la llegada del material 5.1.).
- 1.5. ALMACENES registra en STOCK.
 - Ingreso pendiente
- 1.6. ALMACENES controla archivo (transitorio) **SOLICITUDES DE COMPRAS** (duplicado)
 - @ Fecha de entrega (para controlar los retrasos).
- 1.6.1. ◇ Si existe retraso en la entrega ALMACENES avisa a COMPRAS.

2. SELECCIÓN DEL PROVEEDOR

- 2.1. COMPRAS consulta **ORIGEN DEL PRODUCTO**.
 - @ Código del producto
 - @ Código del proveedor
- 2.2. COMPRAS consulta **PROVEEDORES**.
- 2.3. COMPRAS emite **PEDIDO DE COTIZACIÓN** (duplicado).
- 2.4. COMPRAS archiva (transitoriamente) **SOLICITUD DE COMPRA y PEDIDO DE COTIZACIÓN** (duplicado) adjuntadas
 - Orden @ **Cronológico** (de PEDIDO DE COTIZACIÓN).
- 2.5. COMPRAS envía **PEDIDO DE COTIZACIÓN** (original) al PROVEEDOR.

3. DECISIÓN DE COMPRA

- 3.1. COMPRAS recibe las **COTIZACIONES** de los **PROVEEDORES**.
- 3.2. COMPRAS controla las **COTIZACIONES** con **MANUAL DE ANTECEDENTES DE COMPRAS**
 - Reputación global del proveedor
 - Términos financieros
 - Flexibilidad del proveedor para ajustarse a las necesidades de su empresa
 - Experiencia con el proveedor en situaciones análogas
 - Servicio técnico ofrecido
 - Confiabilidad en el vendedor
 - Conveniencia en colocar la orden
 - Datos sobre la confiabilidad del producto
 - Precio
 - Especificaciones técnicas

Facilidad de operación o de uso
Preferencias del usuario principal del producto
Entrenamiento ofrecido por el proveedor
Tiempo de entrenamiento requerido
Confiabilidad en los datos de envío prometidos
Facilidad en el mantenimiento.
Servicio de ventas esperado después de la fecha de compra.

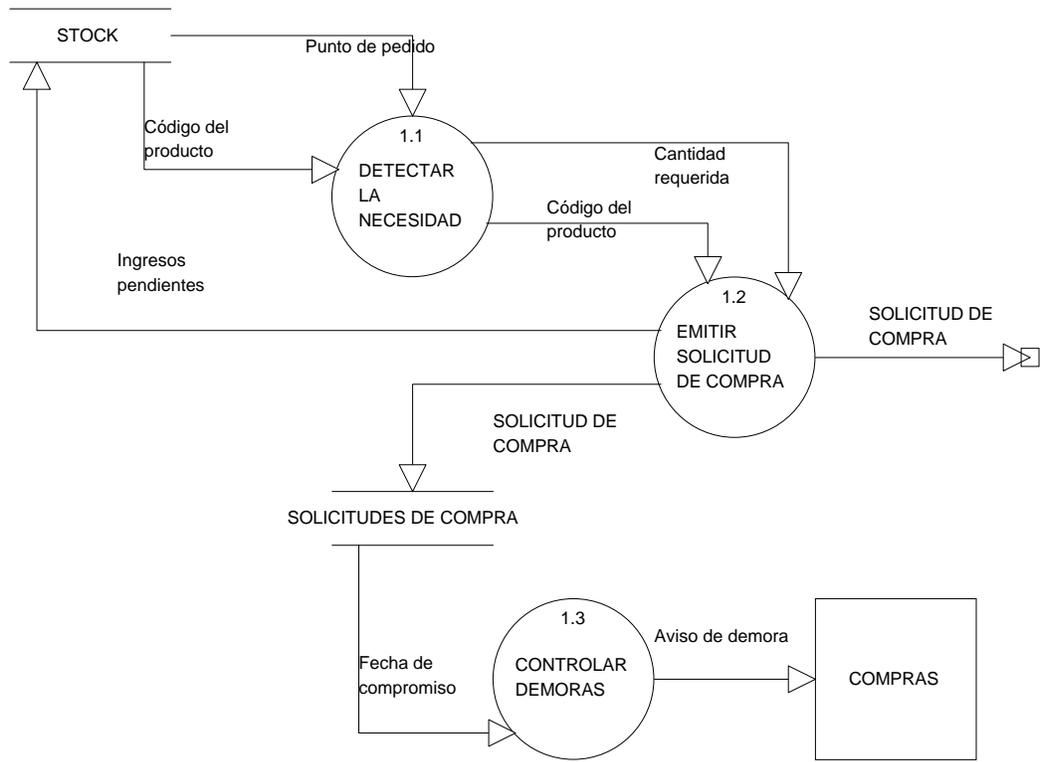
- 3.3. COMPRAS emite **ORDEN DE COMPRA** (por triplicado).
- 3.4. COMPRAS envía **ORDEN DE COMPRA** (original) al PROVEEDOR.
- 3.5. COMPRAS adjunta y constituye **LEGAJO DE TRAMITACIÓN DE COMPRA** :
ORDEN DE COMPRA (duplicado),
SOLICITUD DE COMPRA (original),
PEDIDO DE COTIZACIÓN (duplicado)y
COTIZACIÓN (original);
- 3.6. COMPRAS archiva (provisoriamente) **LEGAJO DE TRAMITACIÓN DE COMPRA**.
Se archiva definitivamente cuando se @ cumple la entrega con copia del **REMITO** o **PARTE DE RECEPCIÓN** (ver 4.12).
- 3.7. COMPRAS envía **ORDEN DE COMPRA** (triplicado) a **RECEPCIÓN**,
Sin precio.

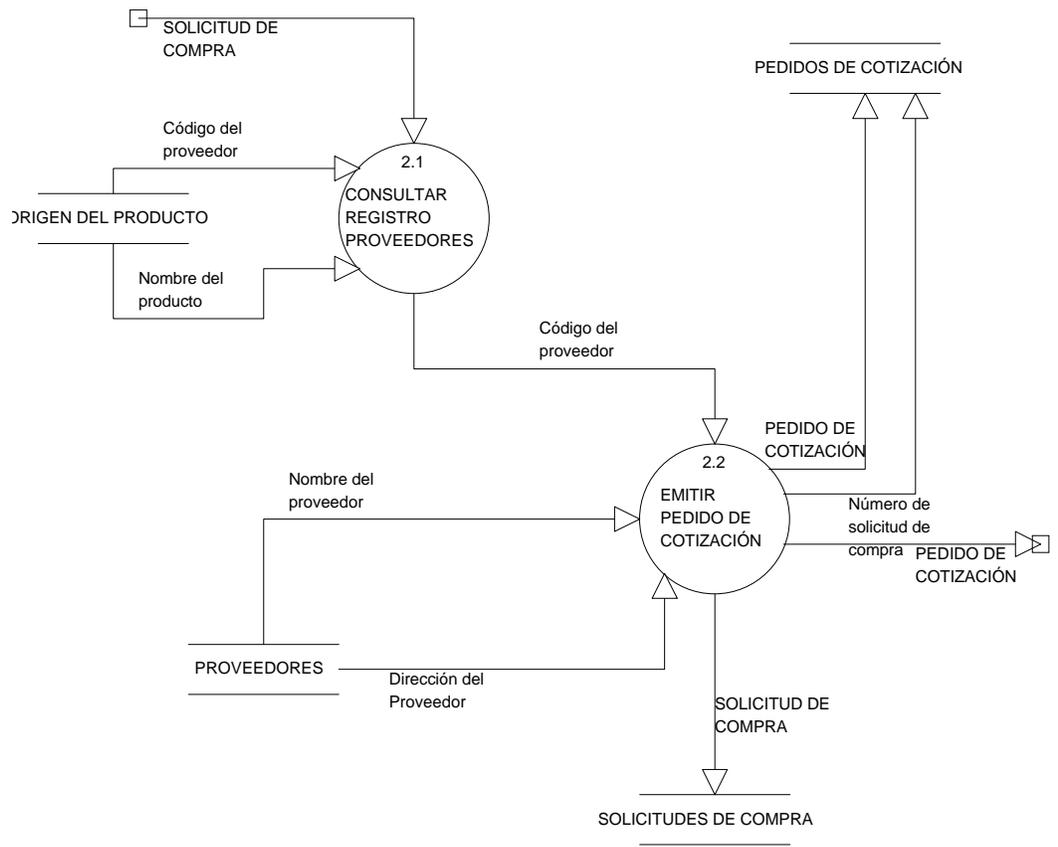
4. RECEPCIÓN DE LA MERCADERÍA

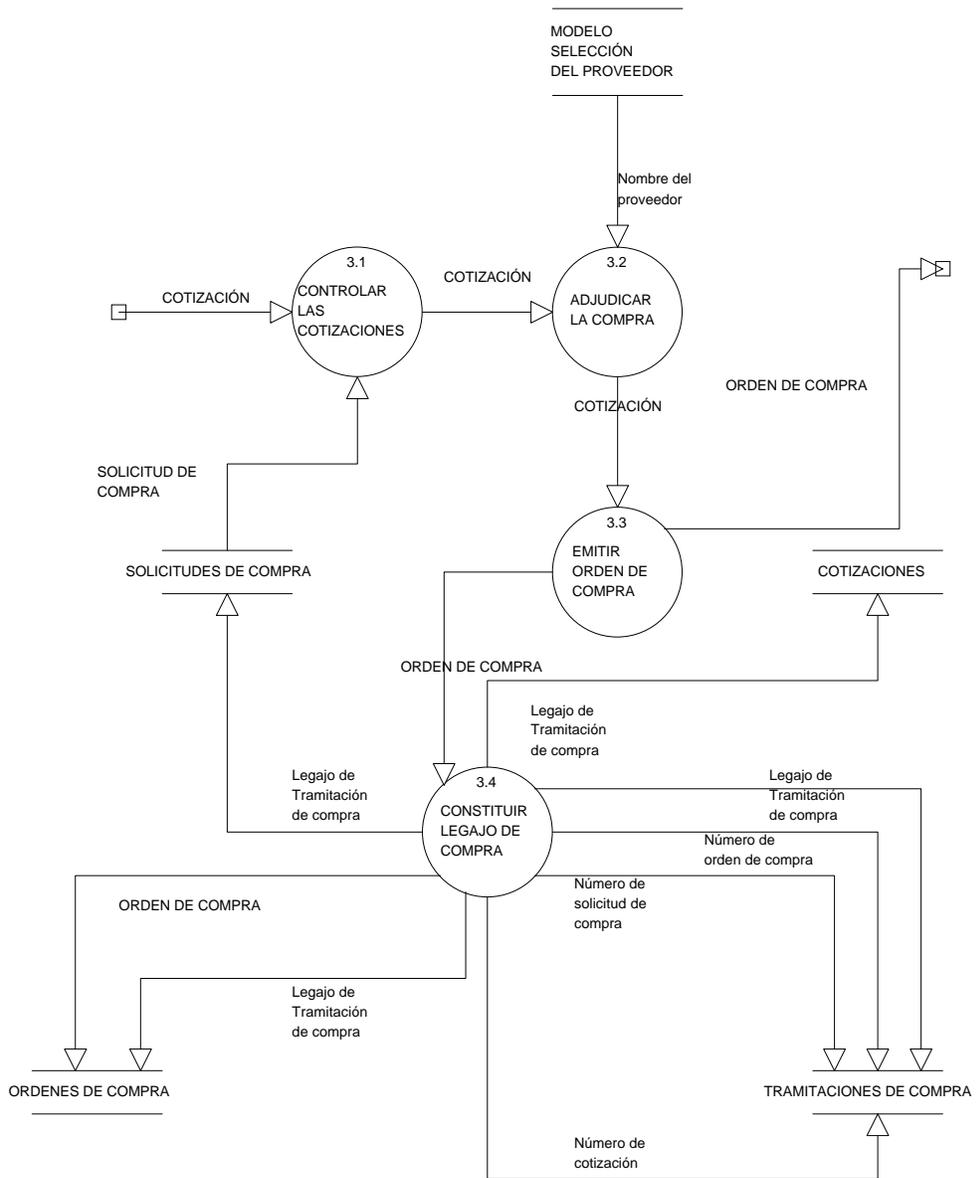
- 4.1. **RECEPCIÓN** controla (la mercadería recibida) **REMITO** (duplicado) con **ORDEN DE COMPRA** (triplicado)
- 4.2. **RECEPCIÓN** firma **REMITO** (original).
- 4.3. **RECEPCIÓN** devuelve **REMITO** (original) al PROVEEDOR.
- 4.4. **RECEPCIÓN** emite **PARTE DE RECEPCIÓN** (por triplicado)
Número de parte de recepción
Número de orden de compra
@ Nombre del proveedor
Dirección del proveedor
Fecha de recepción
@ Mercadería recibida
Unidad de medida
Cantidad
Nombre del producto
Código del producto
Precio
Conformidad de recepción.
- 4.5. **RECEPCIÓN** envía **PARTE DE RECEPCIÓN**
a **COMPRAS** (original).
a **ALMACENES** (duplicado y triplicado).
- 4.6. **ALMACENES** controla (con mercadería).
- 4.7. **ALMACENES** conforma **PARTE DE RECEPCIÓN** (triplicado).
- 4.8. **ALMACENES** envía **PARTE DE RECEPCIÓN** (triplicado) a **RECEPCIÓN**.
- 4.9. **RECEPCIÓN** adjunta **PARTE DE RECEPCIÓN** (triplicado) a **ORDEN DE COMPRA** y archiva (definitivamente).
- 4.10. **RECEPCIÓN** Conformo **REMITO** (duplicado).
- 4.11. **RECEPCIÓN** envía **REMITO** (duplicado) a **CUENTAS A PAGAR**.
- 4.12. **COMPRAS** verifica **PARTE DE RECEPCIÓN** (original) con **LEGAJO DE TRAMITACIÓN DE COMPRA** (en archivo provisorio).
- 4.13. **COMPRAS** Conformo **LEGAJO DE TRAMITACIÓN DE COMPRA**.
- 4.14. **COMPRAS** archiva (definitivamente) **LEGAJO DE TRAMITACIÓN DE COMPRA**.
- 4.15. **COMPRAS** destruye **PARTE DE RECEPCIÓN** (original).

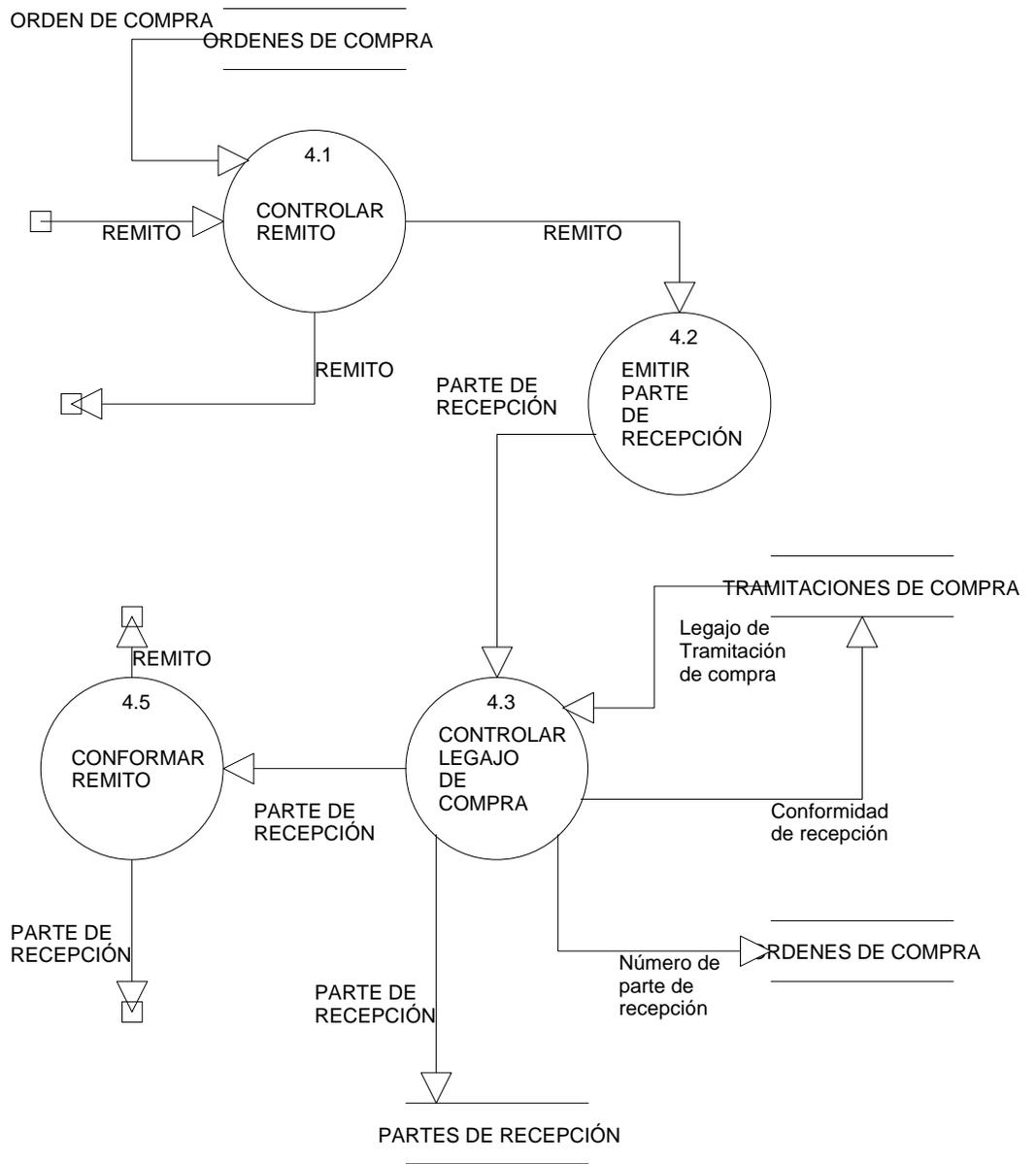
5. ALMACENAJE DE LA MERCADERÍA

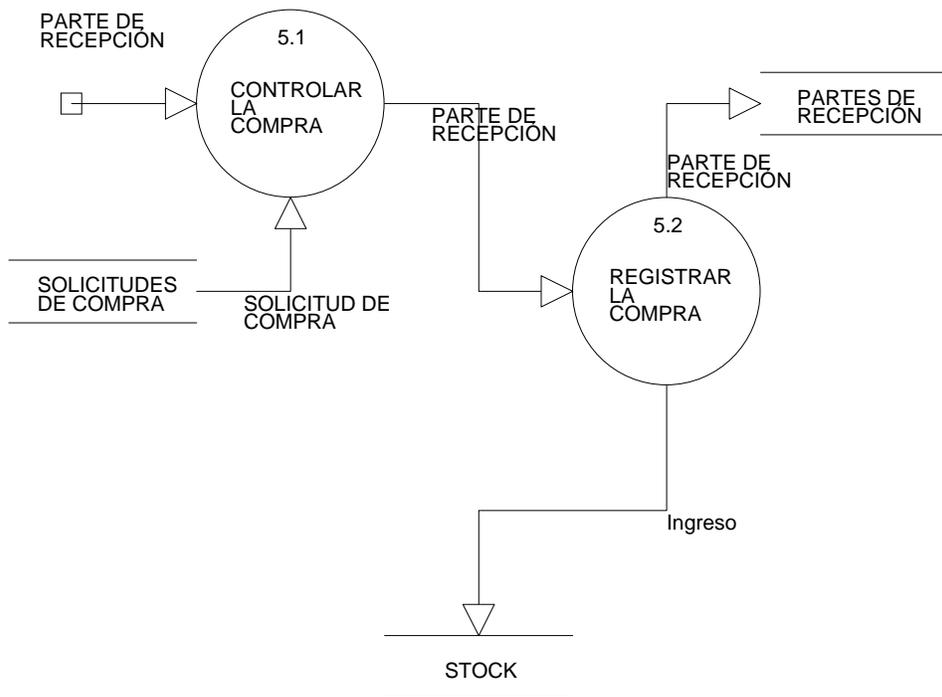
- 5.1. ALMACENES verifica **PARTE DE RECEPCIÓN** con **SOLICITUD DE COMPRA**
 - 5.1.1.◇ Si no hay coincidencia comunica a COMPRAS.
- 5.2. ALMACENES retira del archivo (transitorio) **SOLICITUD DE COMPRA**.
- 5.3. ALMACENES destruye **SOLICITUD DE COMPRA**
- 5.4. ALMACENES actualiza archivo de **STOCK** en base al **PARTE DE RECEPCIÓN**.
 - 5.4.1. Anula anotación En trámite de reposición;
Ingreso pendiente (ver 1.1 y 1.5)
 - 5.4.2. Da Ingreso
- 5.5. ALMACENES archiva (definitivamente) **PARTE DE RECEPCIÓN**











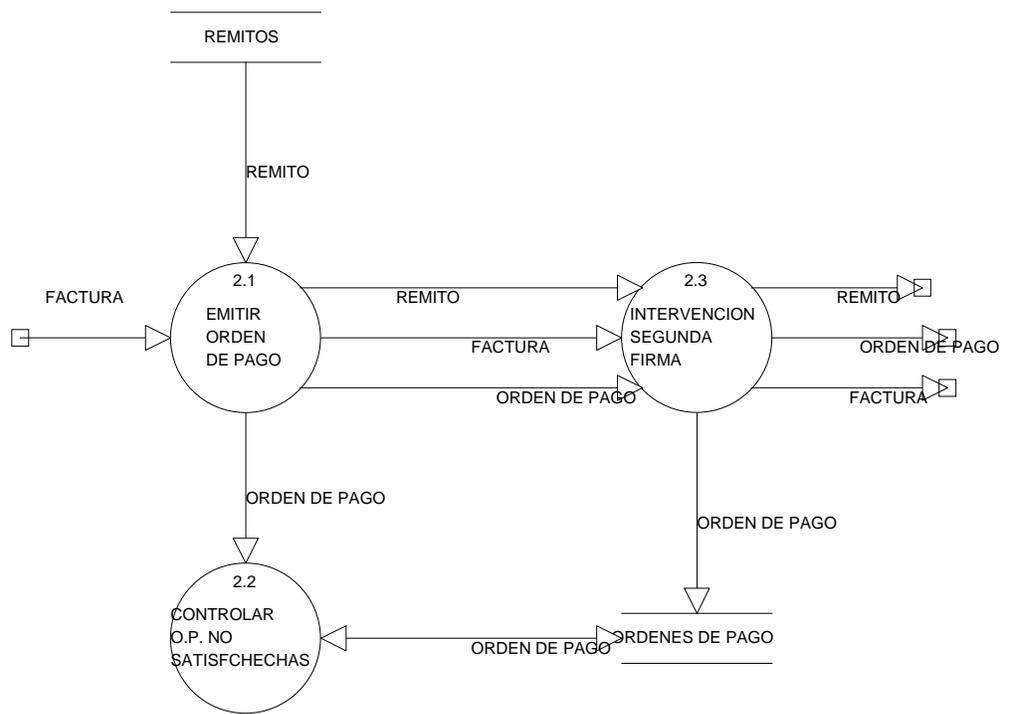
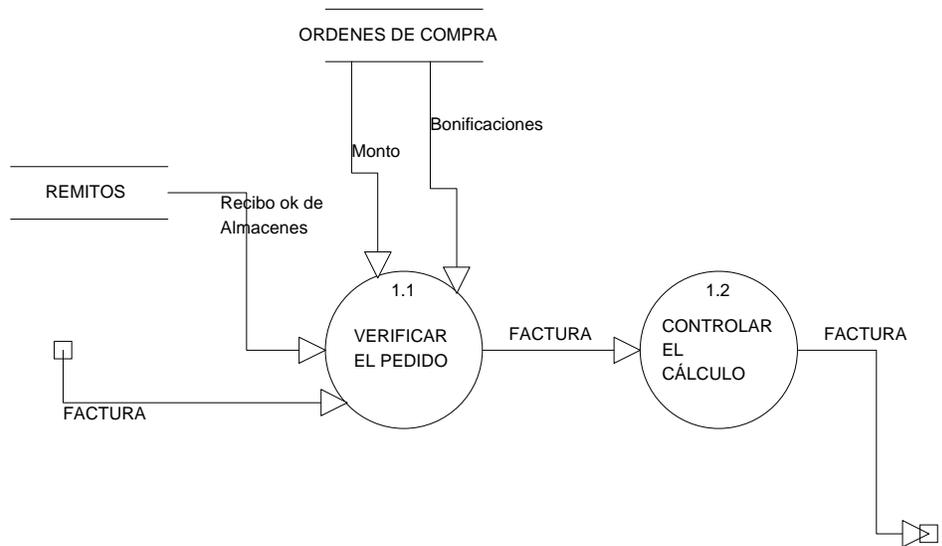
EJERCICIO Nº:5

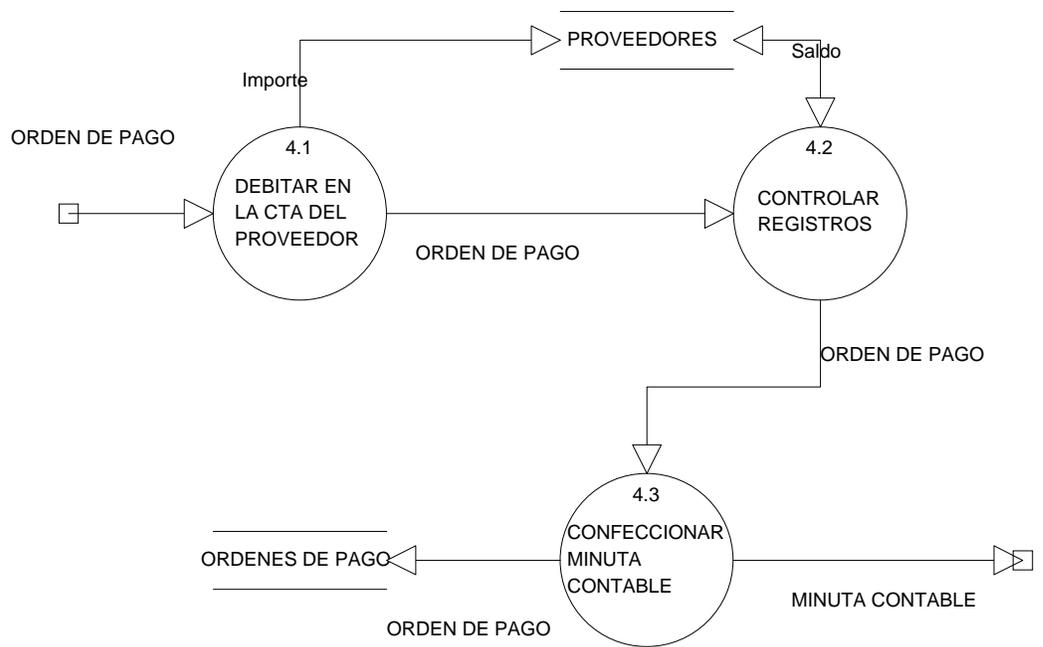
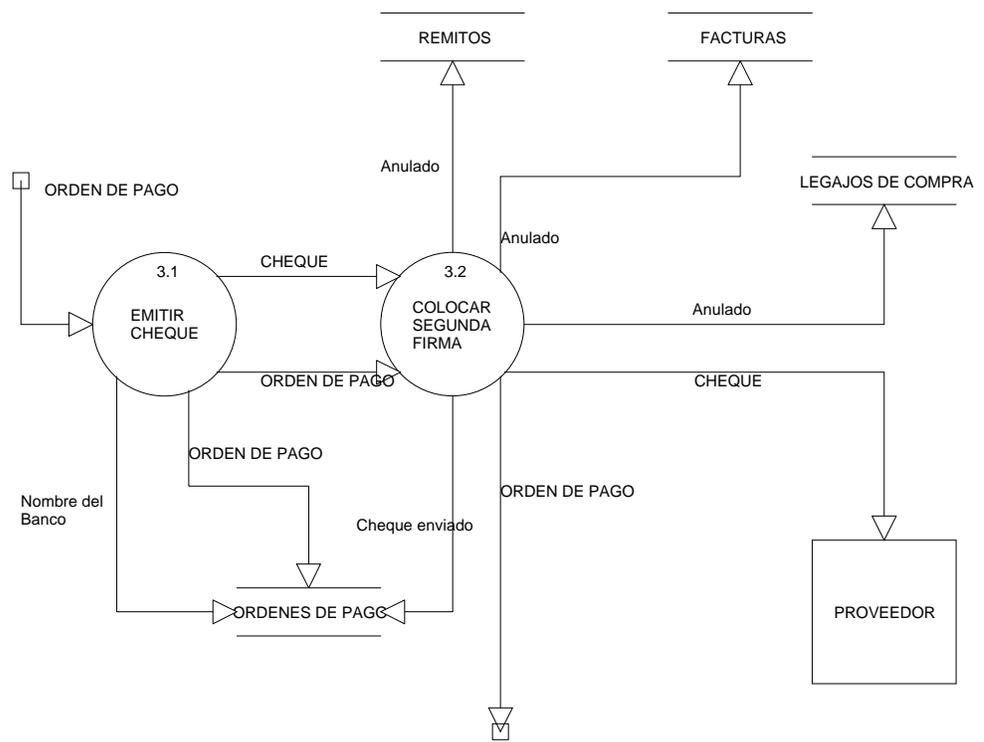
PROCEDIMIENTO DE: PAGOS

LISTA DE EVENTOS

1. CONTROL DE LA FACTURA DEL PROVEEDOR
 - 1.1. El PROVEEDOR envía la **FACTURA** a CUENTAS A PAGAR.
 - 1.2. CUENTAS A PAGAR CONTROLA
 - 1.2.1. Que fue un pedido realizado por la empresa; lo verifica con la **ORDEN DE COMPRA** que tiene en archivo transitorio.
 - 1.2.2. Los importes los controla con la **ORDEN DE COMPRA** o con la **COTIZACIÓN DEL PROVEEDOR**.
 - 1.2.3. Bonificaciones y Descuentos estipulados; con la **ORDEN DE COMPRA**.
 - 1.2.4. Que la mercadería fue recibida (En conformidad o con observaciones); lo verifica con **REMITO** firmado por RECEPCIÓN.
 - 1.2.5. Totales de la **FACTURA** con el cálculo aritmético.
2. AUTORIZACIÓN DE PAGO
 - 2.1. CUENTAS A PAGAR envía;
 - 2.1.1. **REMITO** (duplicado), **FACTURA** (original), **ORDEN DE PAGO** original); a la Segunda Firma.
 - 2.1.2. **ORDEN DE PAGO** (duplicado y triplicado) a TESORERÍA.
 - 2.1.3. **ORDEN DE PAGO** (cuadruplicado) al archivo transitorio.
 - 2.2. CUENTAS A PAGAR controla con **ORDEN DE PAGO** (cuadruplicado) **ORDENES DE PAGO** emitidas y no satisfechas.
3. PAGO
 - 3.1. TESORERÍA en base a la **ORDEN DE PAGO**, emite el **CHEQUE**.
 - 3.2. TESORERÍA, anota en **ORDEN DE PAGO** (triplicado).
@ Banco Girado.
 - 3.3. TESORERÍA firma el **CHEQUE**.
 - 3.4. TESORERÍA envía
 - 3.4.1. **CHEQUE** y **ORDEN DE PAGO** (duplicado) a la SEGUNDA FIRMA.
 - 3.4.2. **ORDEN DE PAGO** (triplicado) al archivo transitorio.
 - 3.5. SEGUNDA FIRMA verifica los componentes básicos de pago; **REMITO, FACTURA, ORDEN DE PAGO, LEGAJO DE COMPRA**.
 - 3.6. SEGUNDA FIRMA anula COMPONENTES BÁSICOS DE PAGO con:
Pagado
Fecha
Número de cheque
Iniciala.
 - 3.7. SEGUNDA FIRMA; firma EL **CHEQUE**.
 - 3.8. SEGUNDA FIRMA envía
ORDEN DE PAGO (duplicado) a TESORERÍA
FACTURA; REMITO a CONTADURÍA
CHEQUE; ORDEN DE PAGO (original) a SECRETARÍA DE GERENCIA.
 - 3.9. TESORERÍA archiva (definitivamente) **ORDEN DE PAGO** (duplicado).
 - 3.10. TESORERÍA envía a CUENTAS A PAGAR la **ORDEN DE PAGO** (triplicado).
 - 3.11. CUENTAS A PAGAR destruye **ORDEN DE PAGO** (cuadruplicado) y demora (pendiente) el **LEGAJO**.
 - 3.12. SECRETARÍA DE SEGUNDA FIRMA envía **CHEQUE** al PROVEEDOR.

- 3.13. SECRETARÍA DE SEGUNDA FIRMA anota en **ORDEN DE PAGO** (original), CHEQUE Enviado.
 - 3.14. SECRETARÍA DE SEGUNDA FIRMA envía **ORDEN DE PAGO** (original) a CUENTAS A PAGAR.
 - 4. CONTABILIZACIÓN
 - 4.1. CUENTAS A PAGAR en base a **ORDEN DE PAGO** (original), debita el Importe de la **CUENTA DEL PROVEEDOR**.
 - 4.2. CUENTAS A PAGAR al final del día, controla las registraciones.
 - 4.3. CUENTAS A PAGAR, confecciona **MINUTA CONTABLE** por los débitos.
 - 4.4. CUENTAS A PAGAR envía **MINUTA CONTABLE** a CONTADURÍA.
 - 4.5. CUENTAS A PAGAR archiva **ORDEN DE PAGO**(original).
 - 5. CONTROL DEL RECIBO
 - 5.1. CUENTAS A PAGAR obtiene **RECIBO DEL PROVEEDOR**.
 - 5.2. CUENTAS A PAGAR controla pago.
- CUENTAS A PAGAR archiva





EJERCICIO Nº:6

PROCEDIMIENTO: VENTAS EN CUENTA CORRIENTE

LISTA DE EVIENTOS

1. FORMULAR EL PEDIDO

- 1.1. El CLIENTE realiza una **SOLICITUD DE MERCADERÍA**.
- 1.2. El VENDEDOR confecciona **NOTA DE PEDIDO** (4 ejemplares)
 - Datos del **CLIENTE**
 - Nombre del Cliente
 - Número de Cuenta Corriente
 - Domicilio de entrega
 - Datos del **PRODUCTO**
 - Descripción del producto
 - Código del Producto
 - Cantidad solicitada
 - Datos del **VENDEDOR**
 - Nombre del Vendedor
 - Código del Vendedor (Código de empleado).
- 1.3. El VENDEDOR valoriza la **NOTA DE PEDIDO** (Forma de pago; la puede realizar CUENTA CORRIENTE)
 - LISTA DE PRECIO**
 - BONIFICACIONES**
 - CARGOS** (Por embalajes, transportes u otros)
 - Plazo de entrega.
- 1.4. El VENDEDOR envía **NOTA DE PEDIDO**
 - A VENTAS (original y triplicado)
 - AI CLIENTE (duplicado)
 - AI ARCHIVO, para control del VENDEDOR (cuadruplicado).
- 1.5. VENTAS controla la valorización (del paso 1.3.).
- 1.6. VENTAS iniciala **NOTA DE PEDIDO** (original y triplicado).
- 1.7. VENTAS envía **NOTA DE PEDIDO**
 - A CUENTA CORRIENTE (original)
 - AI ARCHIVO (triplicado).
- 1.8. VENTAS confecciona estadísticas en base a archivado **NOTA DE PEDIDO** (triplicado).

2. APROBACIÓN DEL PEDIDO

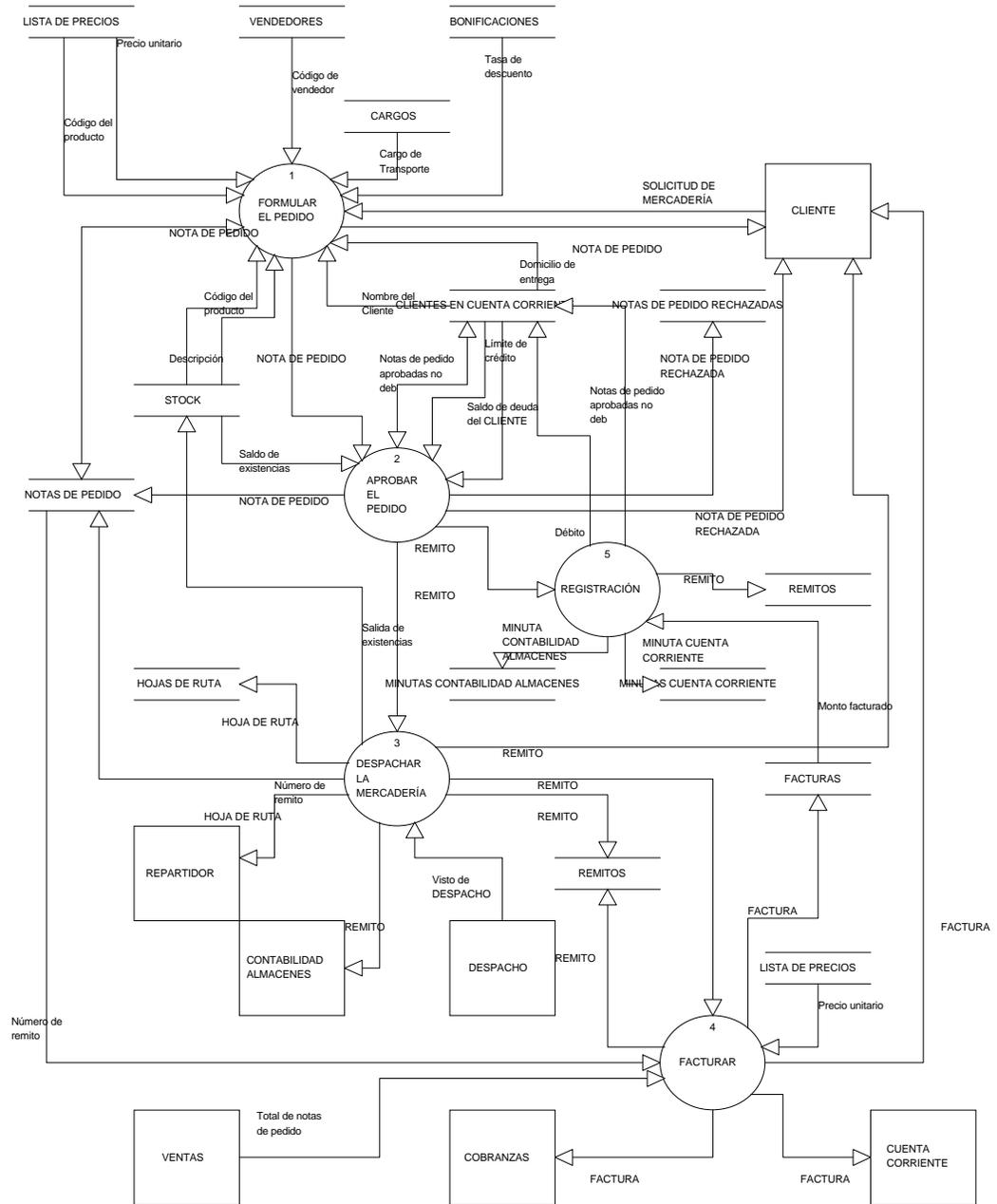
- 2.1. CUENTA CORRIENTE controla @ Límite de crédito.
 - 2.1.1.La suma del: Valor total + Saldo de deuda del CLIENTE + Notas de pedido aprobadas no debitadas; deben ser menor al Límite de crédito acordado.
- ◇ Si se cumple la ecuación anterior
 - 2.2. CUENTA CORRIENTE aprueba **NOTA DE PEDIDO**
 - 2.2.1.CUENTA CORRIENTE registra la aprobación en el Total de notas de pedido aprobadas y no debitadas de cada **CLIENTE**.
 - 2.2.2.Se inscribe en la **NOTA DE PEDIDO** la Aprobación.
 - 2.2.3.CUENTA CORRIENTE envía **NOTA DE PEDIDO** a ALMACENES.
- ◇ Si no se cumple la ecuación anterior
 - 2.3. CUENTA CORRIENTE no aprueba la **NOTA DE PEDIDO**
 - 2.3.1.CUENTA CORRIENTE inscribe en la **NOTA DE PEDIDO**
 - Importe de la Deuda actual
 - Importe de las Notas de pedido aprobadas y no contabilizadas.

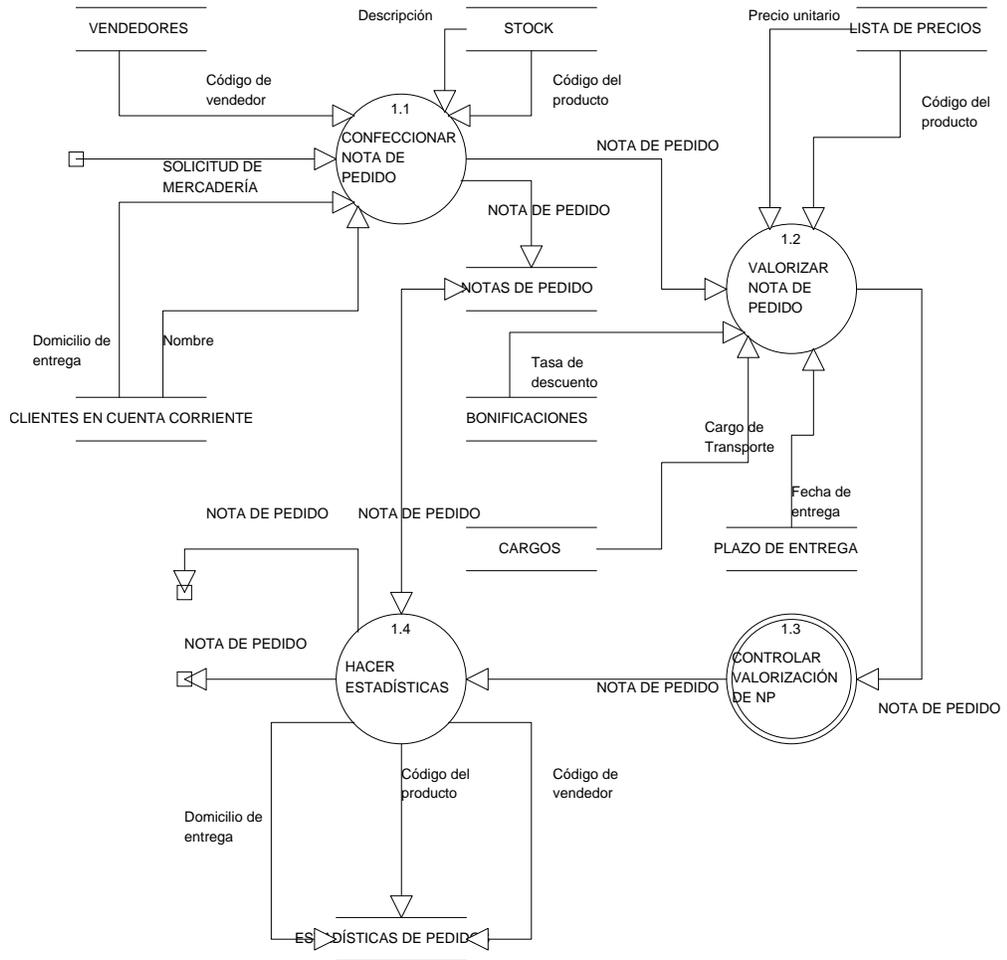
- 2.3.2. CUENTA CORRIENTE envía **NOTA DE PEDIDO** no aprobada a CRÉDITOS.
 - 2.3.3. CRÉDITOS consulta Límite de crédito del CLIENTE.
 - 2.3.4. CRÉDITOS decide si aprueba
 - 2.3.4.1. CRÉDITOS no aprueba la venta
 - 2.3.4.1.1. CRÉDITOS envía **NOTA DE PEDIDO** a VENTAS
 - 2.3.4.1.2. VENTAS registra el Rechazo
 - 2.3.4.1.3. VENTAS comunica al CLIENTE
 - 2.3.4.2. CRÉDITOS aprueba la venta
 - 2.3.4.2.1. CRÉDITOS Iniciala sobre la **NOTA DE PEDIDO**
 - 2.3.4.2.2. CRÉDITOS envía **NOTA DE PEDIDO** a CUENTA CORRIENTE
 - 2.3.4.2.3. CUENTA CORRIENTE registra el pedido aprobado en Total de notas de pedido aprobadas y no debitadas de cada CLIENTE (paso 2.2.1.).
 - 2.4. ALMACENES controla **NOTA DE PEDIDO** (original) con **EXISTENCIAS**.
 - ◇ Si hay **EXISTENCIAS** para cumplir el pedido
 - 2.4.1. ALMACENES emite **REMITO** (5 ejemplares)
 - 2.4.2. ALMACENES envía **REMITO** (5 ejemplares) a DESPACHO
 - 2.4.3. ALMACENES archiva (definitivamente) **NOTA DE PEDIDO**.
 - 2.4.4. ALMACENES descarga de **EXISTENCIAS**.
 - ◇ Si no hay **EXISTENCIAS** para cumplir el pedido.
 - 2.4.5. ALMACENES anota en la **NOTA DE PEDIDO** (original) tal circunstancia
 - 2.4.6. ALMACENES envía **NOTA DE PEDIDO** (original) a CUENTA CORRIENTE
 - 2.4.7. CUENTA CORRIENTE con la **NOTA DE PEDIDO** (original) descarga del archivo del CLIENTE, la inscripción Nota de pedido aprobada y no contabilizada.
 - 2.4.8. CUENTA CORRIENTE envía **NOTA DE PEDIDO** (original) a VENTAS
 - 2.4.9. VENTAS toma conocimiento
 - 2.4.10. VENTAS avisa al CLIENTE.
3. DESPACHO DE LA MERCADERÍA.
- 3.1. DESPACHO controla **REMITO**.
 - 3.2. DESPACHO conforma **REMITO** (cuadruplicado).
 - 3.3. DESPACHO envía **REMITO** (cuadruplicado) a ALMACENES.
 - 3.4. DESPACHO organiza la entrega
 - 3.4.1. DESPACHO emite HOJA DE RUTA (Original y 2 copias)
 - Datos del Cliente
 - Domicilio de entrega
 - Número de **REMITO**
 - Cantidad de bultos por **REMITO**
 - Nombre del repartidor (Ordenada por @ REPARTIDOR).
 - 3.5. DESPACHO entrega al REPARTIDOR
 - HOJA DE RUTA** (original y duplicado)
 - REMITO** (original y duplicado).
 - 3.6. REPARTIDOR controla
 - 3.6.1. Números de **REMITO** en **HOJA DE RUTA**
 - 3.6.2. Bultos o paquetes con **REMITO**.
 - 3.7. REPARTIDOR firma **HOJA DE RUTA** (triplicado)
 - 3.8. DESPACHO archiva (transitoriamente)
 - HOJA DE RUTA** (triplicado)
 - REMITO** (quintuplicado y triplicado).
 - 3.9. REPARTIDOR entrega mercadería al CLIENTE.
 - 3.10. CLIENTE firma **REMITO** (original).
 - 3.11. REPARTIDOR entrega **REMITO** (duplicado) al CLIENTE.

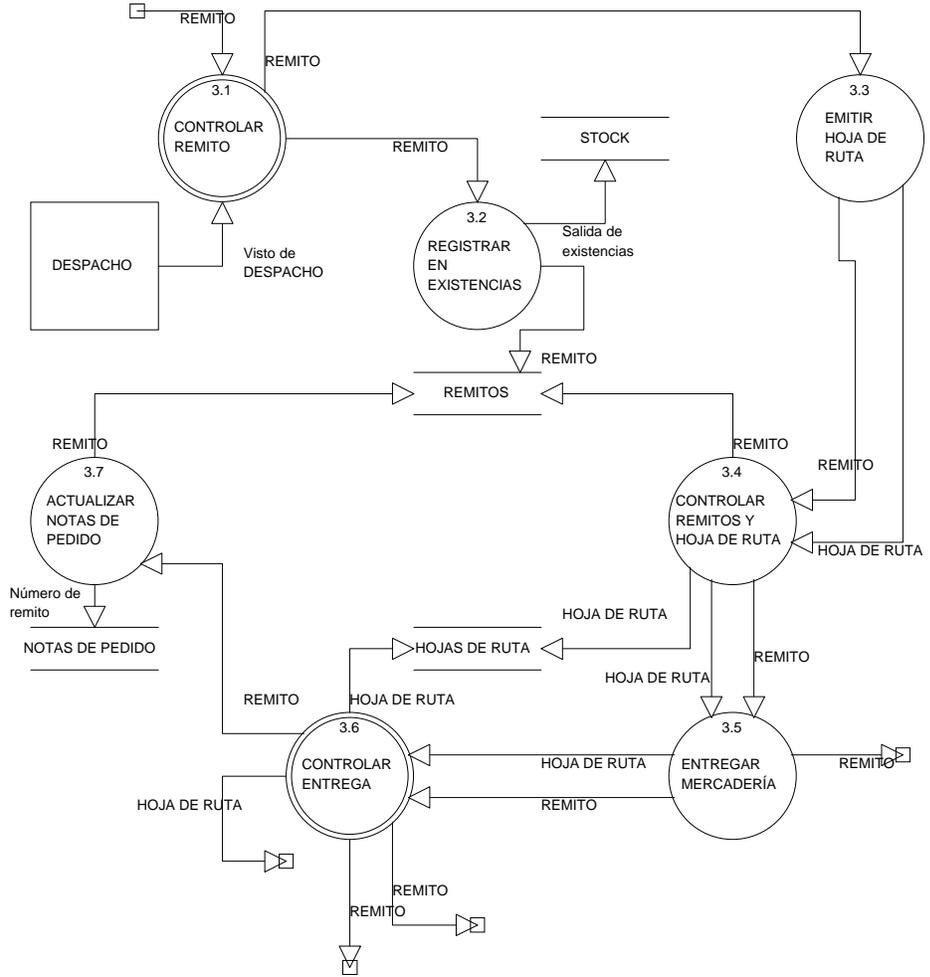
- 3.12. REPARTIDOR anota observaciones en HOJA DE RUTA (si las hay).
 - 3.13. REPARTIDOR entrega en DESPACHO
 - REMITO** (original)
 - HOJA DE RUTA** (original y duplicado)
 - 3.14. DESPACHO controla entrega
 - 3.14.1. Si fue todo entregado (todos los **REMITOS** firmados por los **CLIENTES**)
 - 3.14.1.1. DESPACHO devuelve **HOJA DE RUTA** (duplicado) al REPARTIDOR
 - 3.14.1.2. REPARTIDOR archiva **HOJA DE RUTA** (duplicado)
 - 3.14.2. Si una entrega no fue realizada
 - 3.14.2.1. Se excluye de la **HOJA DE RUTA**
 - 3.14.2.2. DESPACHO archiva transitoriamente el **REMITO**
 - 3.14.2.3. DESPACHO lo incluye en la **HOJA DE RUTA** del día siguiente.
 - 3.15. DESPACHO envía **REMITOS** de entregas efectuadas y **HOJAS DE RUTA** (original) a CONTABILIDAD ALMACENES.
 - 3.16. CONTABILIDAD ALMACENES controla **HOJA DE RUTA** con **REMITOS** entregados.
 - 3.17. CONTABILIDAD ALMACENES envía **HOJA DE RUTA** (original) a DESPACHO.
 - 3.18. DESPACHO archiva (definitivamente) **HOJA DE RUTA** (original).
 - 3.19. DESPACHO envía **REMITO** (triplicado) a **FACTURACIÓN**.
 - 3.20. DESPACHO envía **REMITO** (quintuplicado) a VENTAS.
 - 3.21. VENTAS adjunta **REMITO** (quintuplicado) a **NOTA DE PEDIDO** (archivada).
4. FACTURACIÓN
- 4.1. FACTURACIÓN valoriza **REMITO** (consulta LISTA DE PRECIO).
 - 4.2. FACTURACIÓN controla si existe alguna **BONIFICACIÓN ESPECIAL** (comunicada por documento) a ser incluida.
 - 4.3. FACTURACIÓN emite **FACTURA** (triplicado).
 - 4.4. FACTURACIÓN controla **REMITO** y **FACTURA**
 - Suma número de artículo
 - Suma cantidad.
 - 4.5. FACTURACIÓN, finalizada la emisión, calcula el Total de importes facturados.
 - 4.6. FACTURACIÓN comunica Total de importe facturado a CONTADURÍA.
 - 4.7. VENTAS emite un **TOTAL DE NOTAS DE PEDIDO**; de los **REMITOS** (quintuplicado) recibidos.
 - 4.8. VENTAS comunica **TOTAL NOTAS DE PEDIDO** a CONTADURÍA.
 - 4.9. CONTADURÍA controla Total de importe facturado y **TOTAL NOTAS DE PEDIDO**.
 - 4.10. CONTADURÍA archiva Total facturación y **TOTAL NOTAS DE PEDIDO**.
 - 4.11. CONTADURÍA comunica resultado de la verificación a **FACTURACIÓN**.
 - 4.12. FACTURACIÓN recibido resultado de verificación envía **FACTURA** (original) al **CLIENTE**.
 - 4.13. FACTURACIÓN envía **FACTURA** (duplicado) a **CUENTA CORRIENTE**.
 - 4.14. FACTURACIÓN envía **FACTURA** (triplicado) a **COBRANZAS**.
 - 4.15. **COBRANZAS** archiva (transitoriamente) **FACTURA** (triplicado). Para posterior cobro.
 - 4.16. FACTURACIÓN archiva **REMITO**.
5. REGISTRACIÓN
- 5.1. CONTABILIDAD ALMACENES valoriza (al costo) el **REMITO** (original).
 - 5.2. CONTABILIDAD ALMACENES acredita en las sub cuentas afectadas.
 - 5.3. CONTABILIDAD ALMACENES totaliza valores acreditadas.

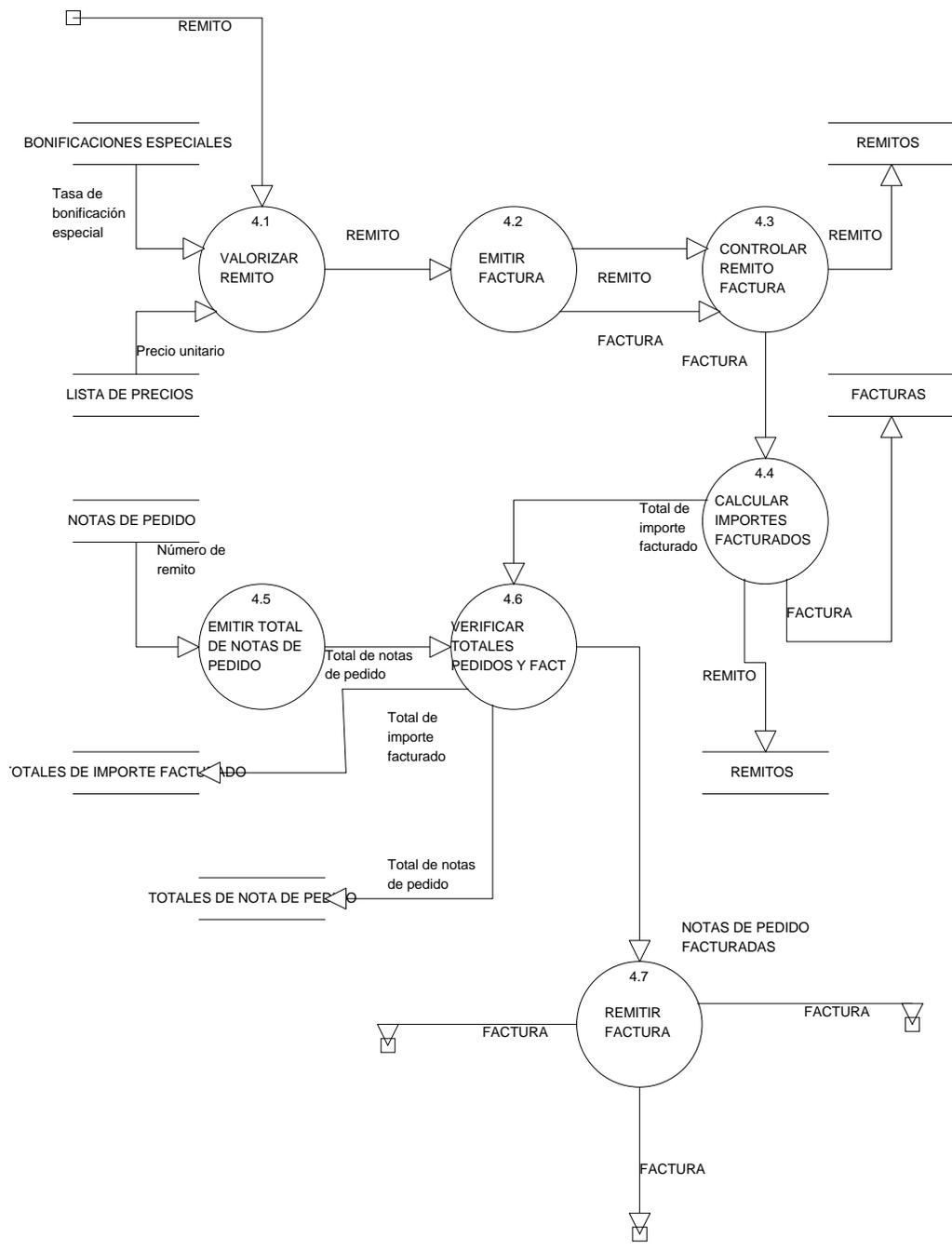
- 5.4. CONTABILIDAD ALMACENES verifica Total número de artículo y Total cantidad del REMITO con Total registrado.
- 5.5. CONTABILIDAD ALMACENES notifica DETALLE DE REGISTRACIONES (MINUTA CONTABLE) a CONTADURÍA.
- 5.6. CONTABILIDAD ALMACENES archiva **REMITO**.
- 5.7. CUENTA CORRIENTE, con la **FACTURACIÓN** (duplicado), debita en la subcuenta del CLIENTE.
- 5.8. CUENTA CORRIENTE, descarga de la subcuenta CLIENTE, importe de pedidos aprobados y no contabilizados.
- 5.9. CUENTA CORRIENTE, totaliza registros.
- 5.10. CUENTA CORRIENTE, informa detalles de los registros (MINUTA CONTABLE) a CONTADURÍA.
- 5.11. CUENTA CORRIENTE archiva **FACTURA** (duplicado).
- 5.12. CONTADURÍA con las MINUTAS CONTABLES recibidas de CONTABILIDAD ALMACENES controla la MINUTA CONTABLE de CUENTA CORRIENTE con el Importe facturado.
- 5.13. CONTADURÍA en base a las MINUTAS CONTABLES registra en CONTABILIDAD GENERAL.
- 5.14. CONTADURÍA archiva las MINUTAS CONTABLES.

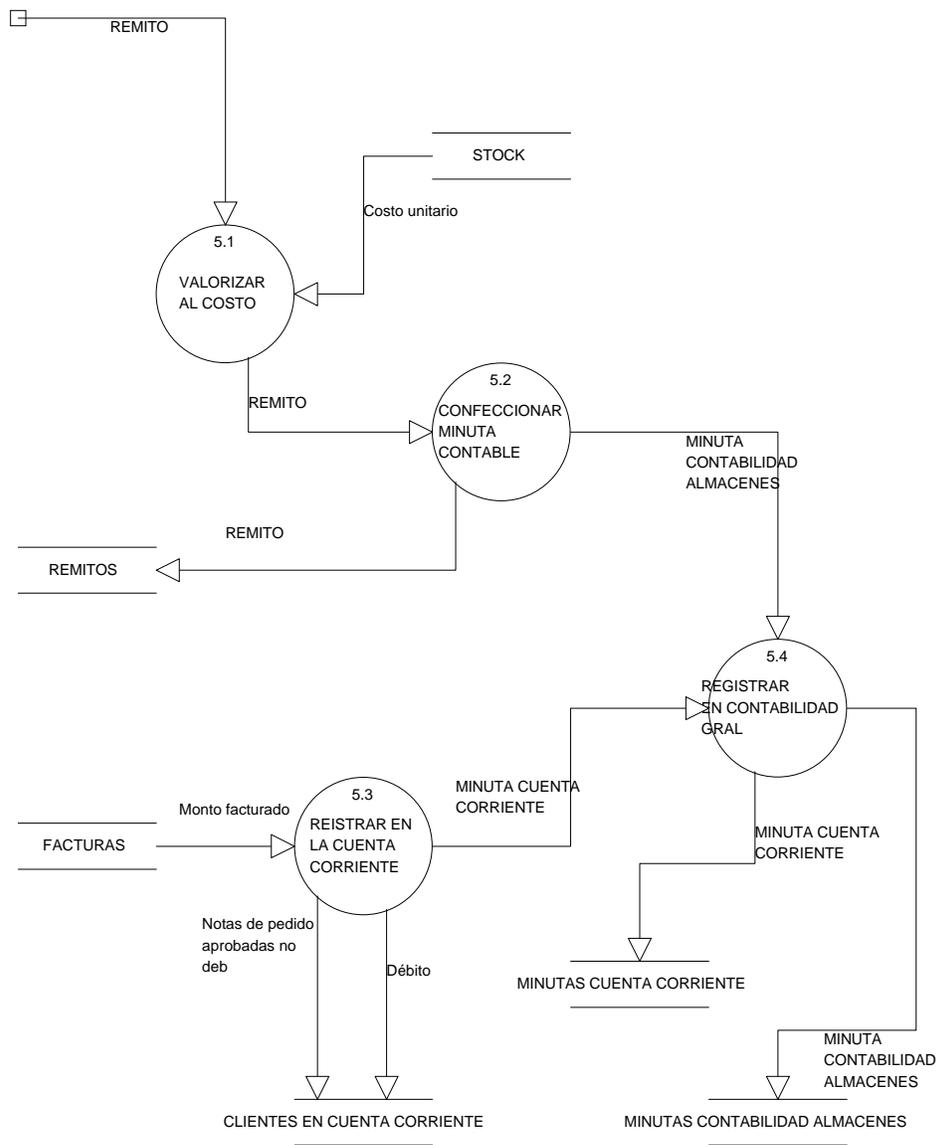
DIAGRAMA DE FLUJO DE DATOS











APÉNDICE B

EJERCITACIÓN PRÁCTICA DE:

- ❖ MODELO RELACIONAL DE DATOS
- ❖ MÓDULOS DE UN SISTEMA
- ❖ ÁRBOL DEL SISTEMA
- ❖ ESPECIFICACIÓN DE LOS MÓDULOS

EJERCICIO Nº:1

1 Procedimiento

Análisis de costo de la producción

2. Operaciones

1. Calcular:
 - 1.1. El Costo de cada pedido
 - 1.2. El Costo de todos los pedidos
 - 1.3. El Costo unitario promedio
2. Emitir los informes respectivos:
 - 2.1. El Costo de cada pedido
 - 2.2. El Costo de todos los pedidos
 - 2.3. El Costo unitario promedio

3 Algoritmos

COSTO DEL PEDIDO:

Agrupar por Número de pedido

Costo del pedido = Cantidad pedida x Costo unitario.

COSTO DE TODOS LOS PEDIDOS:

Costo de todos los pedidos = suma(Cantidad pedida x Costo unitario).

COSTO UNITARIO PROMEDIO:

Suma (Cantidad pedida).

Costo unitario promedio = Costo de todos los pedidos/ suma(Cantidad pedida)

4 Diseño de los Archivos

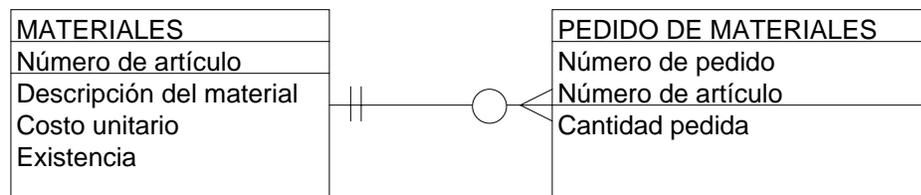
MAESTRO:

MATERIALES(Número de artículo, Descripción del material, Costo unitario, Existencia)

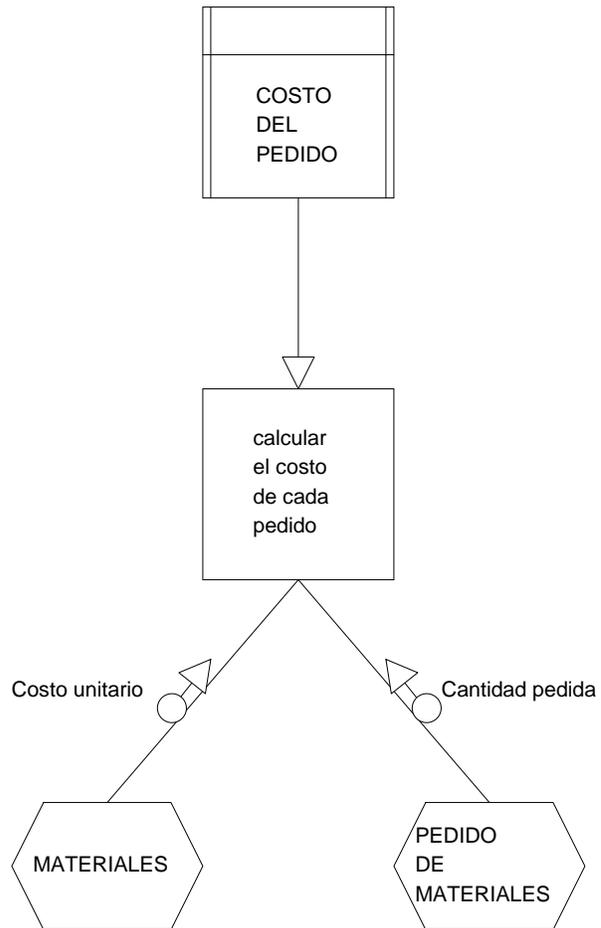
TRANSACCIONES:

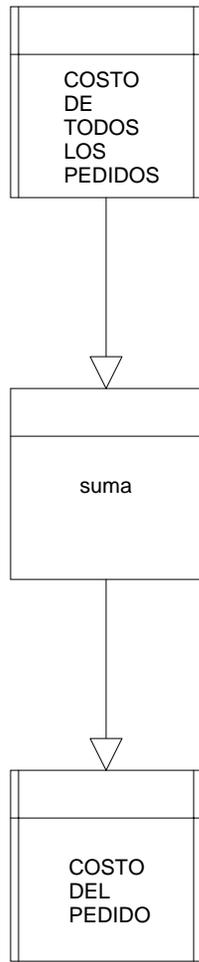
PEDIDO DE MATERIALES(Número de pedido, Número de artículo, Cantidad pedida)

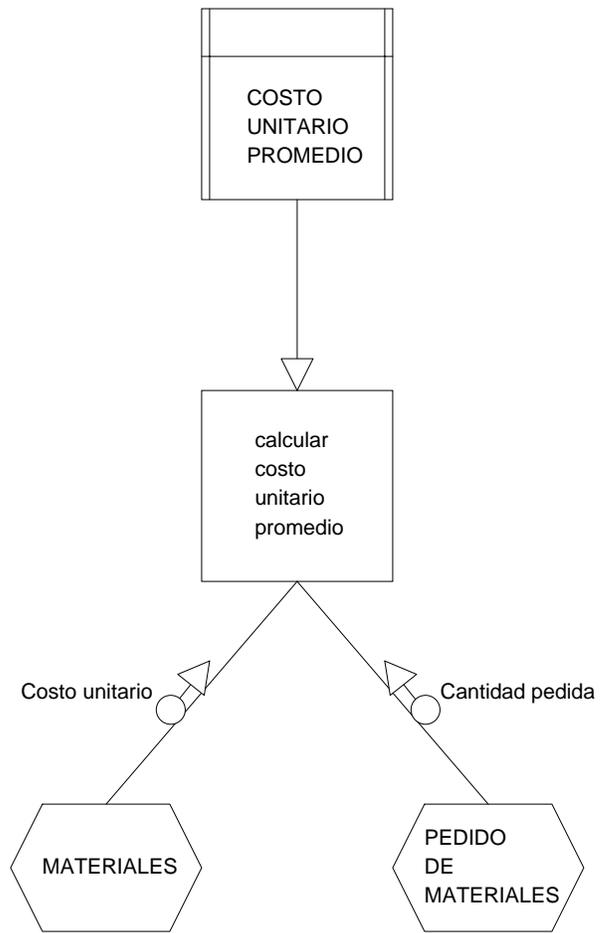
5 Diseño del modelo relacional de datos



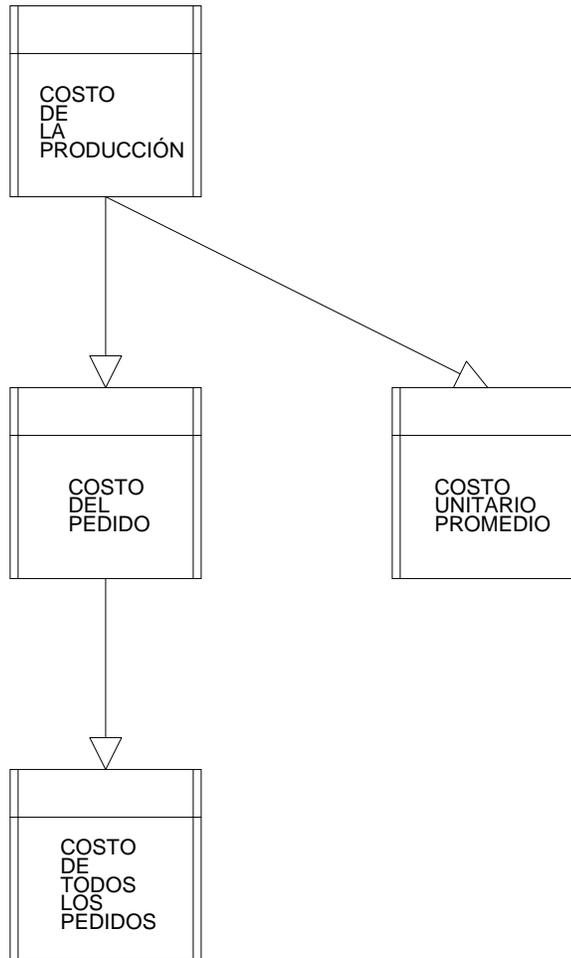
MÓDULOS







ARBOL DEL SISTEMA



EJERCICIO Nº:2

1 Procedimiento

Actualizar archivo **STOCK**

2. Operaciones

1. Actualizar el campo Cantidad en existencia del archivo **STOCK** con la información contenida en el archivo **RESUMEN DE VENTAS**.
2. Generar un informe con la totalidad de datos del archivo **STOCK** (ACTUALIZADO)
3. Generar un informe del Importe total de las ventas.

3 Algoritmos

Cantidad en existencia = Cantidad en existencia - Cantidad vendida

Importe total de ventas = suma(Precio unitario x Cantidad vendida)

4 Diseño de los Archivos

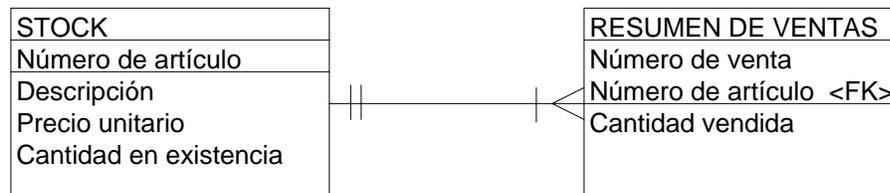
ARCHIVO MAESTRO:

STOCK(Número de artículo, Descripción, Precio unitario, Cantidad en existencia).

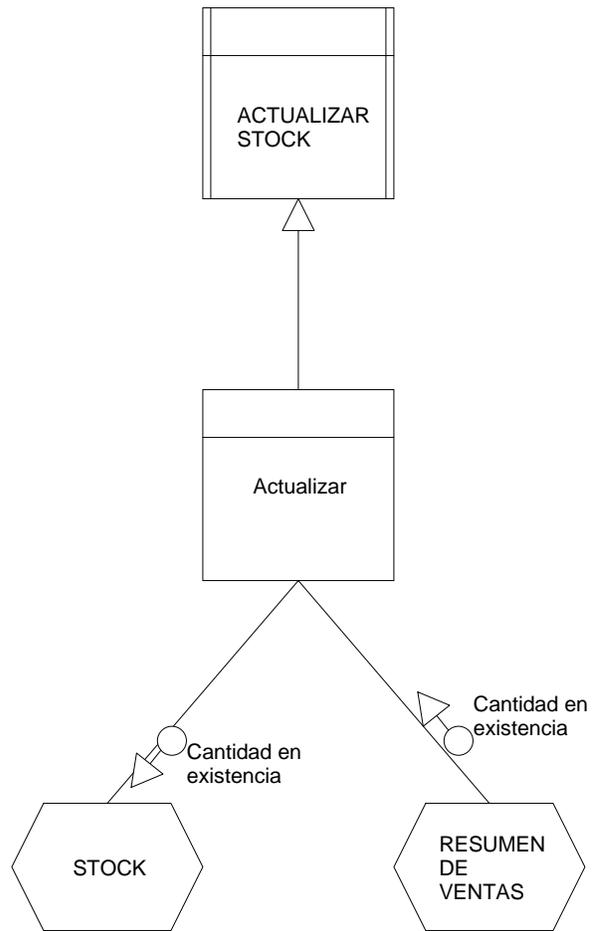
ARCHIVO DE TRANSACCIONES:

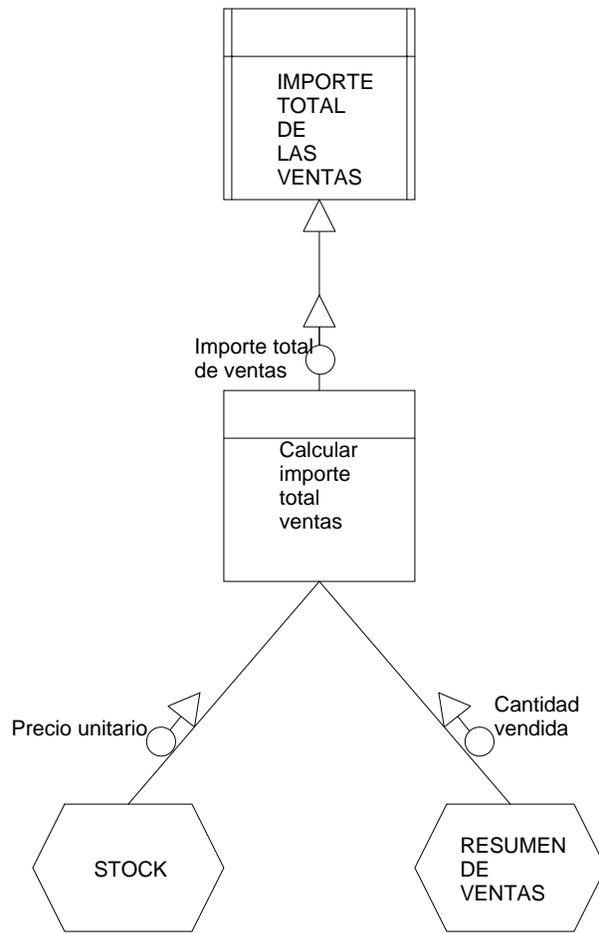
RESUMEN DE VENTAS(Número de venta, Número de artículo, Cantidad vendida).

5 Diseño del modelo relacional de datos

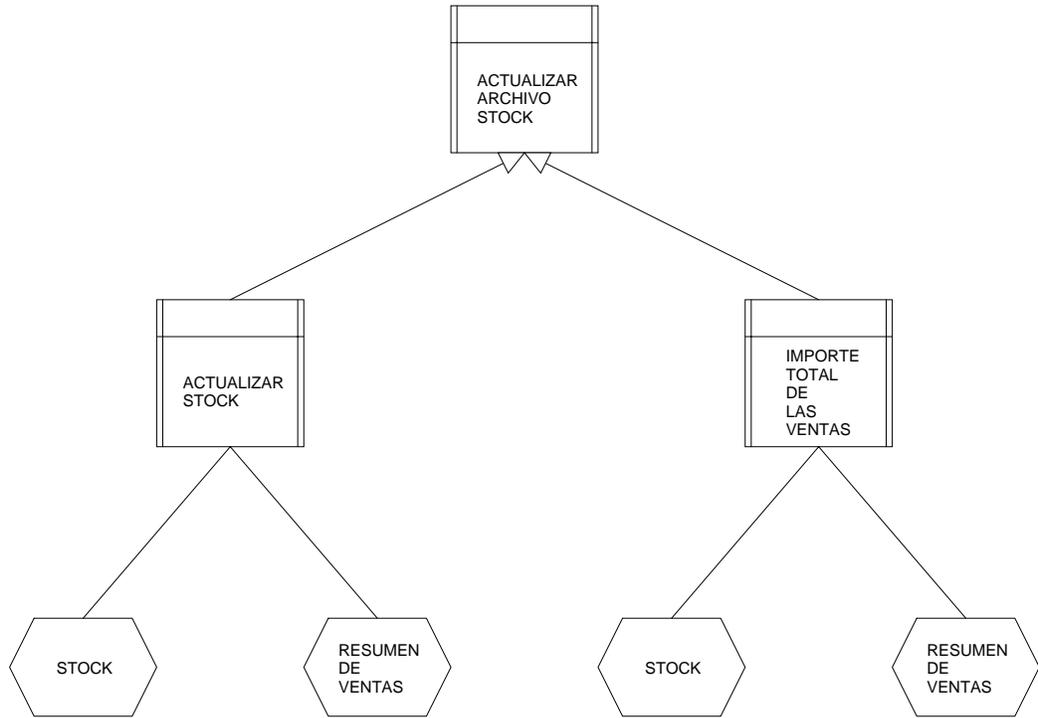


MÓDULOS





ÁRBOL DEL SISTEMA



EJERCICIO Nº:3

1 Procedimiento

Actualización y control de Existencias

2. Operaciones

1. Actualizar:
 - 1.1. Las Unidades en existencia del archivo **EXISTENCIAS** en base al archivo de **NOVEDADES**.
2. Seleccionar los registros de **EXISTENCIAS**:
 - 2.1. Con movimientos;
 - 2.2. que incrementaron su Cantidad;
 - 2.3. que disminuyeron su Cantidad.
3. Imprimir:
 - 3.1. El archivo **EXISTENCIAS** (actualizado);
 - 3.2. un informe del archivo **EXISTENCIAS** de los registros que tuvieron movimiento;
 - 3.3. un informe de Unidades en existencia que han aumentado su saldo;
 - 3.4. un informe de Unidades en existencia que han disminuido su saldo.

3 Algoritmos

Si Movimiento = Ingreso, entonces: Unidades en existencia = Unidades en existencia + Cantidad

Si Movimiento = Egreso, entonces: Unidades en existencia = Unidades en existencia - Cantidad

4 Diseño de los Archivos

ARCHIVO MAESTRO:

EXISTENCIAS(Código de repuesto, Descripción, Número de depósito, Número de estante, Costo unitario, Unidades en existencia)

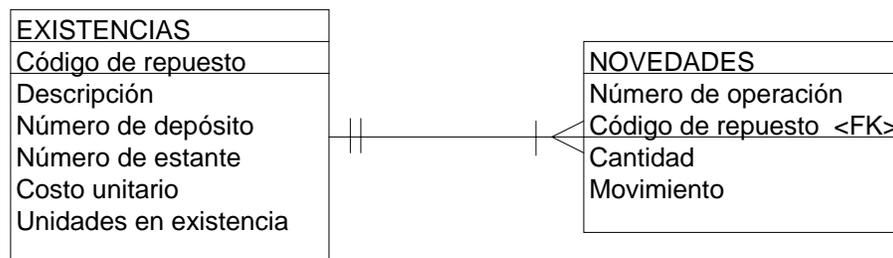
ARCHIVO DE TRANSACCIONES:

NOVEDADES(Número de operación, Código de repuesto, Cantidad, Movimiento).

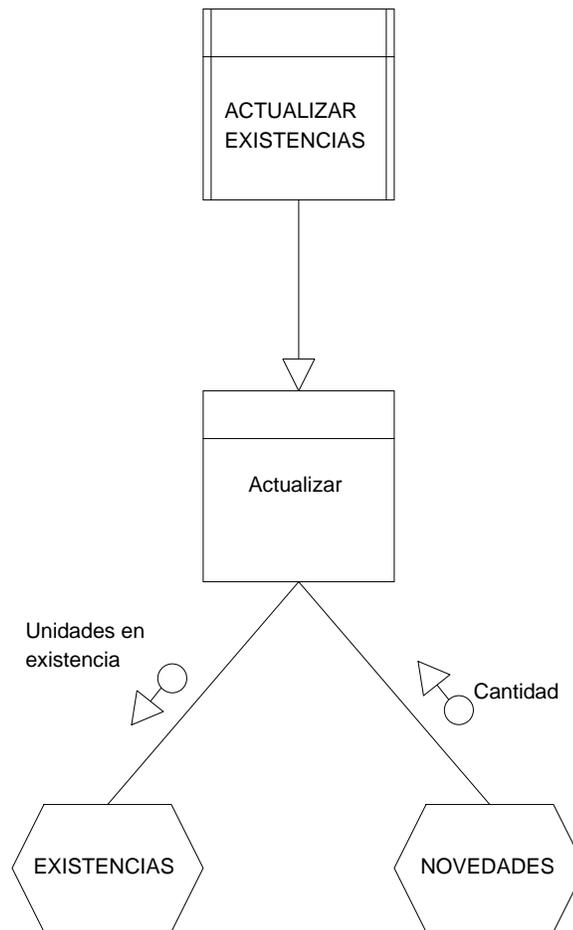
Dominio:

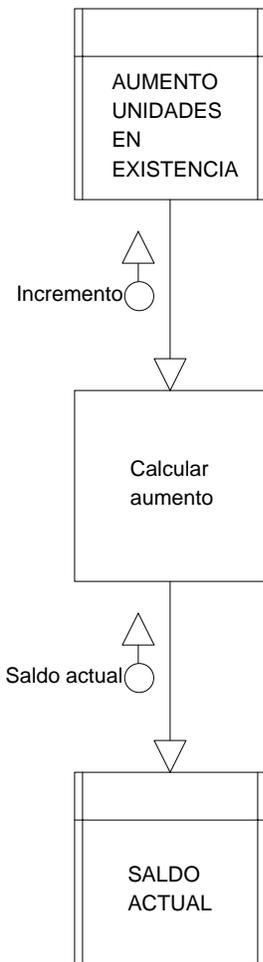
Movimiento = {Ingreso, Egreso}

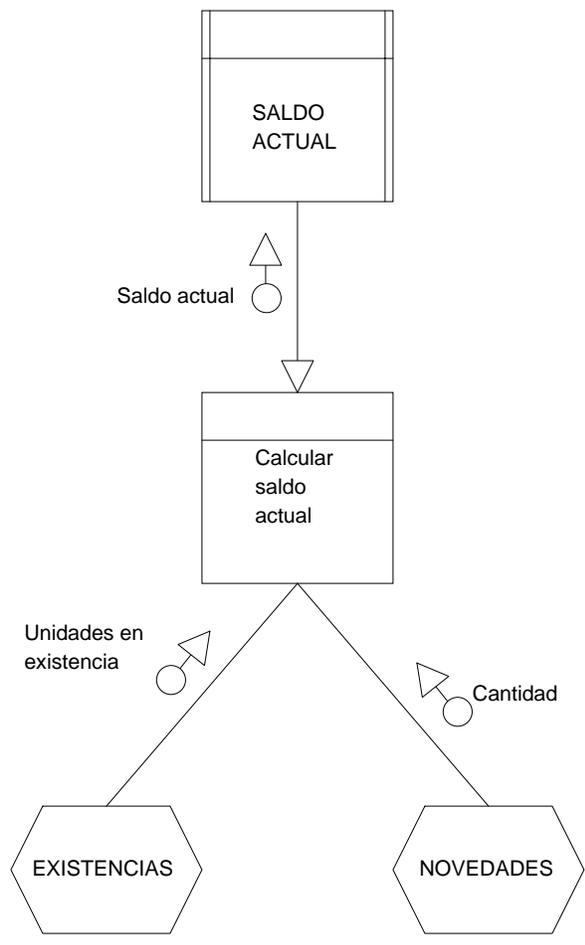
5 Diseño del modelo relacional de datos

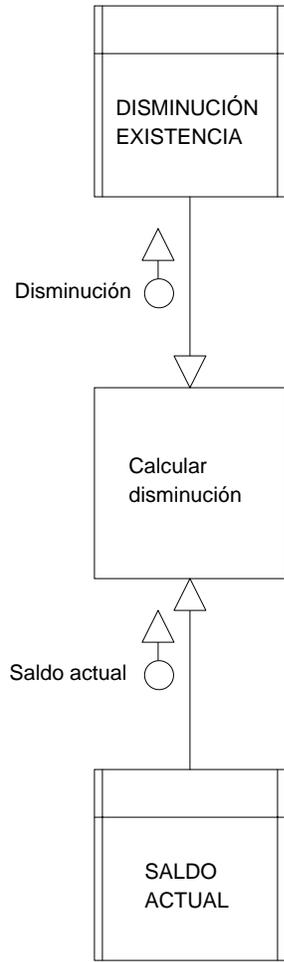


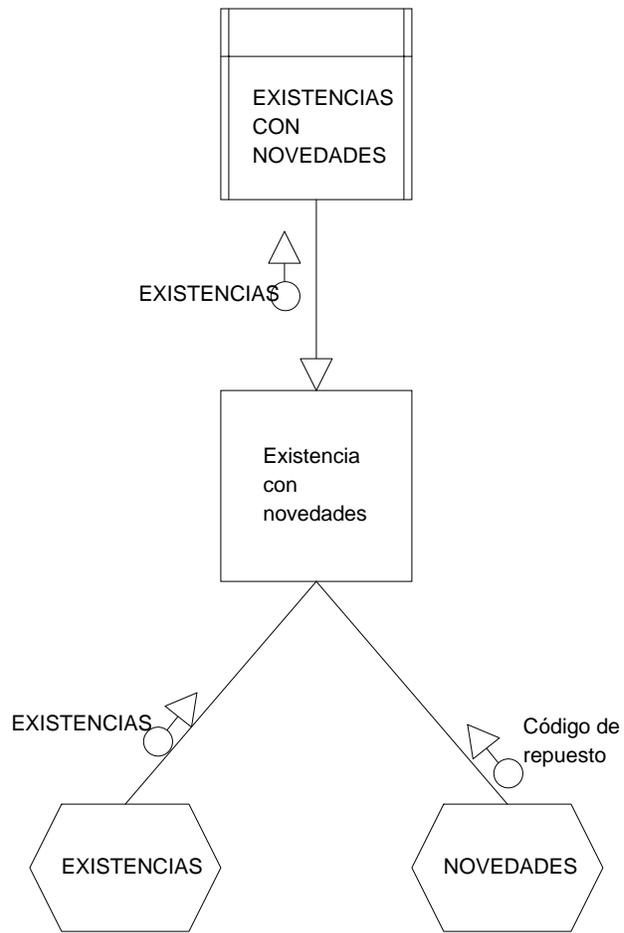
Módulos



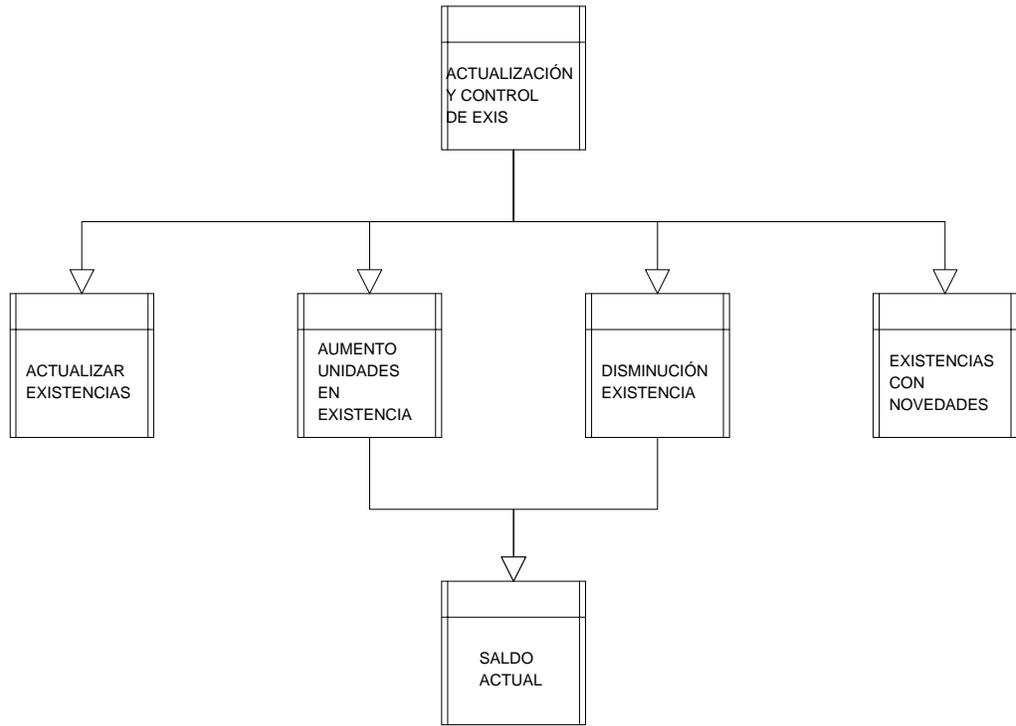








Árbol del sistema



EJERCICIO Nº:4

1 Procedimiento

Actualización y control de existencias

2. Operaciones

1. Actualizar el archivo **PRODUCTOS**, según los datos provenientes del archivo **NOVEDADES DE PRODUCTOS**:
 - 1.1. La Cantidad en existencia;
 - 1.2. El Costo unitario (precio promedio ponderado).
2. Calcular:
 - 2.1. La Cantidad total de productos ingresados, detallados por Código del producto;
 - 2.2. El Promedio general de costo por unidad.
3. Imprimir:
 - 3.1. La Cantidad total de productos ingresados, detallados por Código del producto
 - 3.2. El Promedio general de costo por unidad.

3 Algoritmos

Existencia valorizada:

Si Movimiento = Ingreso entonces: Cantidad en existencia = Cantidad en existencia + Cantidad

Costo unitario = (Cantidad en existencia x Costo unitario + Cantidad x Precio de ingreso) / (Cantidad en existencia + Cantidad)

Promedio general de costo por unidad = suma(Costo unitario x Cantidad en existencia) / suma(Cantidad en existencia)

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PRODUCTOS(Código del producto, Denominación, Costo unitario, Cantidad en existencia)

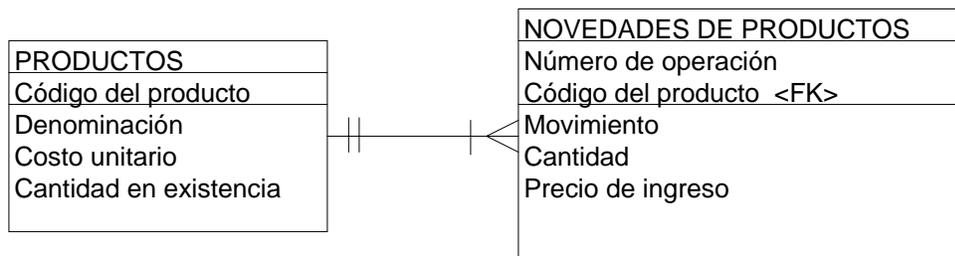
ARCHIVO DE TRANSACCIONES:

NOVEDADES DE PRODUCTOS(Número de operación, Código del producto, Movimiento, Cantidad, Precio de ingreso)

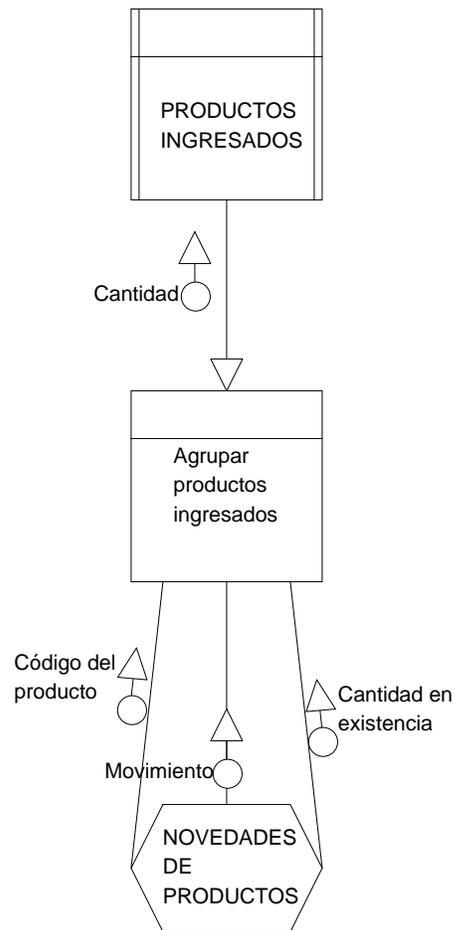
Dominios:

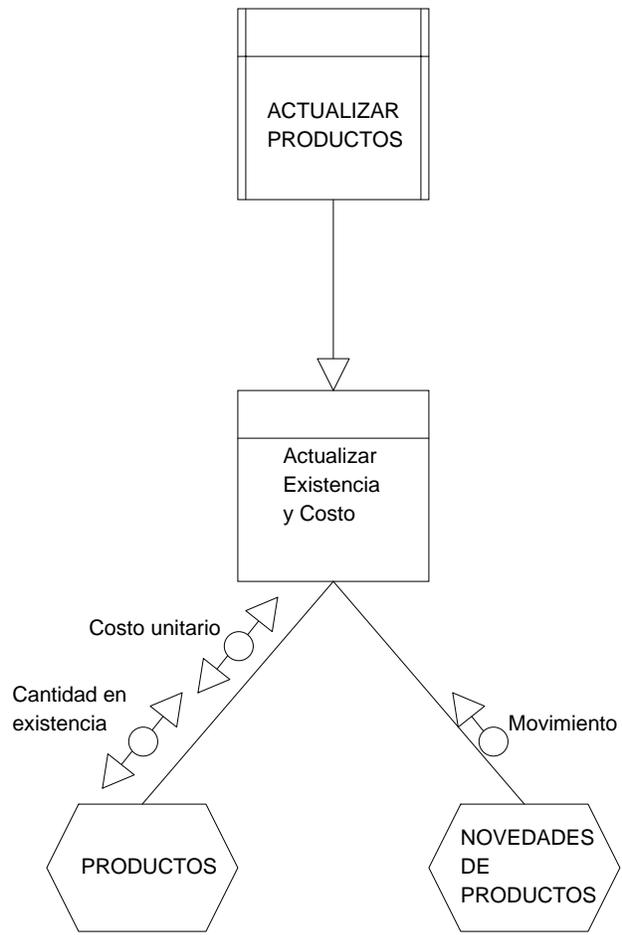
Movimiento = { Ingreso, Egreso }

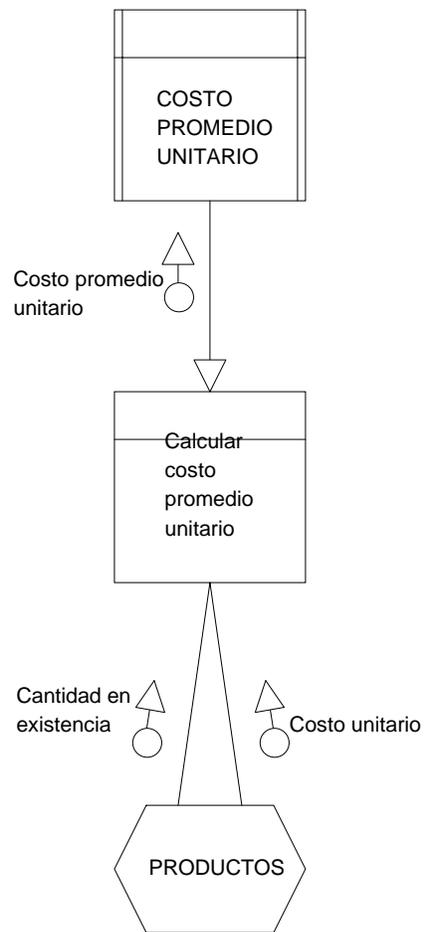
5 Diseño del modelo relacional de datos



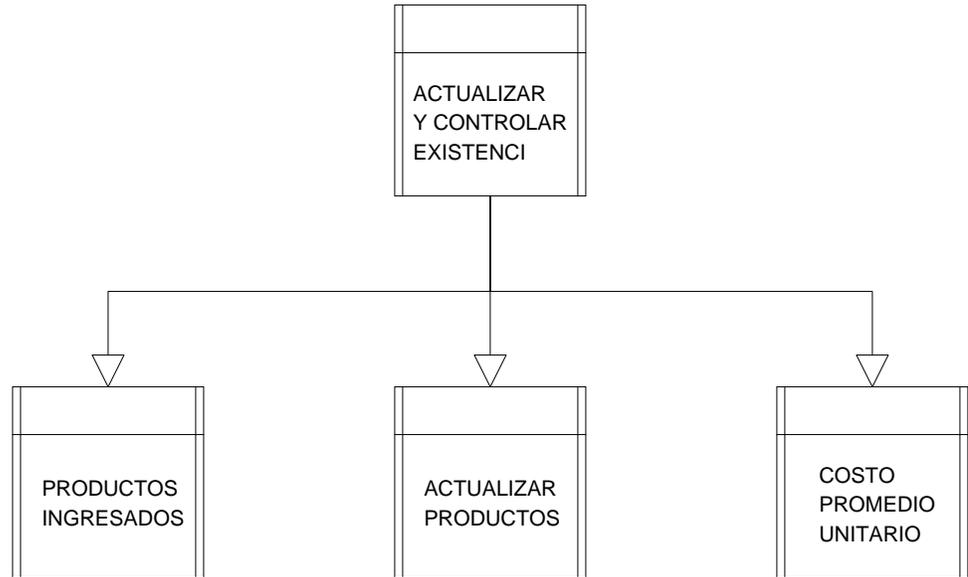
Módulos







Árbol del sistema



EJERCICIO Nº:5

1 Procedimiento

Liquidación de jornales.

2. Operaciones

1. Calcular.
 - 1.1. Importe bruto.
 - 1.2. Aporte del personal.
 - 1.3. Aporte patronal.
 - 1.4. Importe neto.
2. Totalizar.
 - 2.1. Acumulado de: Importe bruto, Aporte del personal, Aporte patronal , e Importe neto.
3. Imprimir recibos.
4. Imprimir un informe de totales.

3 Algoritmos

Importe bruto = Cantidad de horas trabajadas x Valor hora
Aporte del personal = Importe bruto x 10%
Aporte patronal = Importe bruto x 20%
Importe neto = Importe bruto - Aporte del personal.

4 Diseño de los Archivos

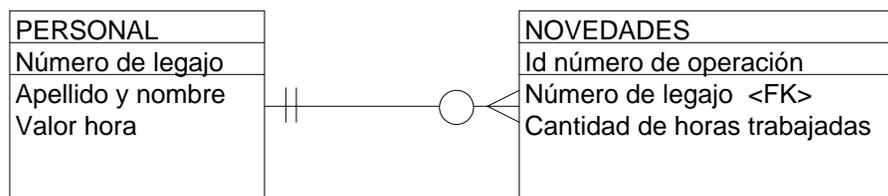
ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora).

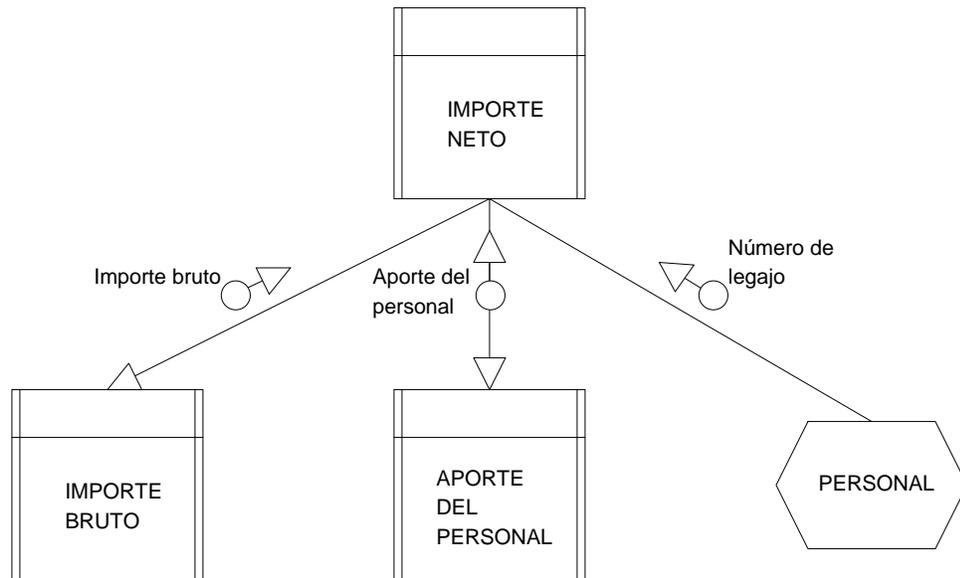
ARCHIVO DE TRANSACCIONES:

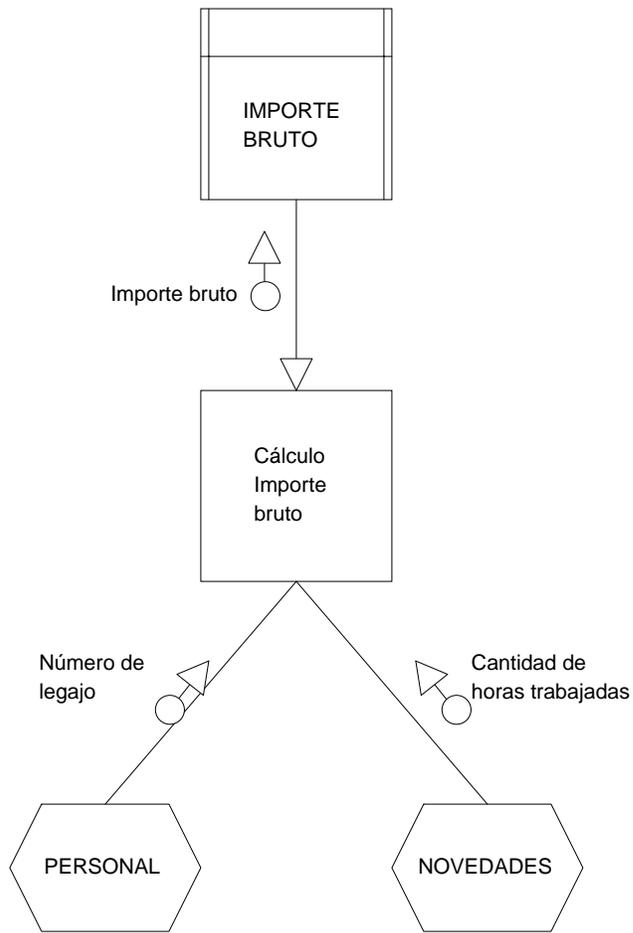
NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas)

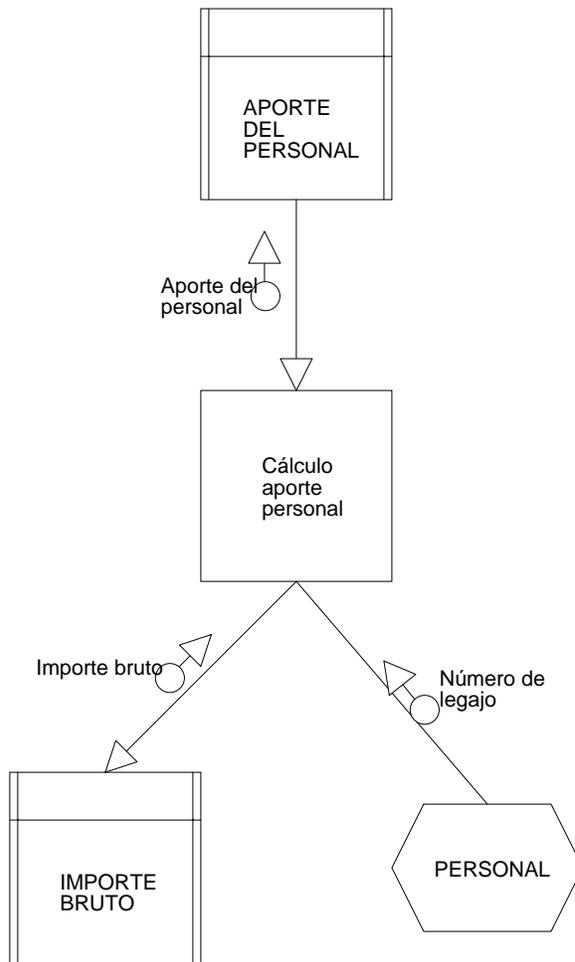
5 Diseño del modelo relacional de datos

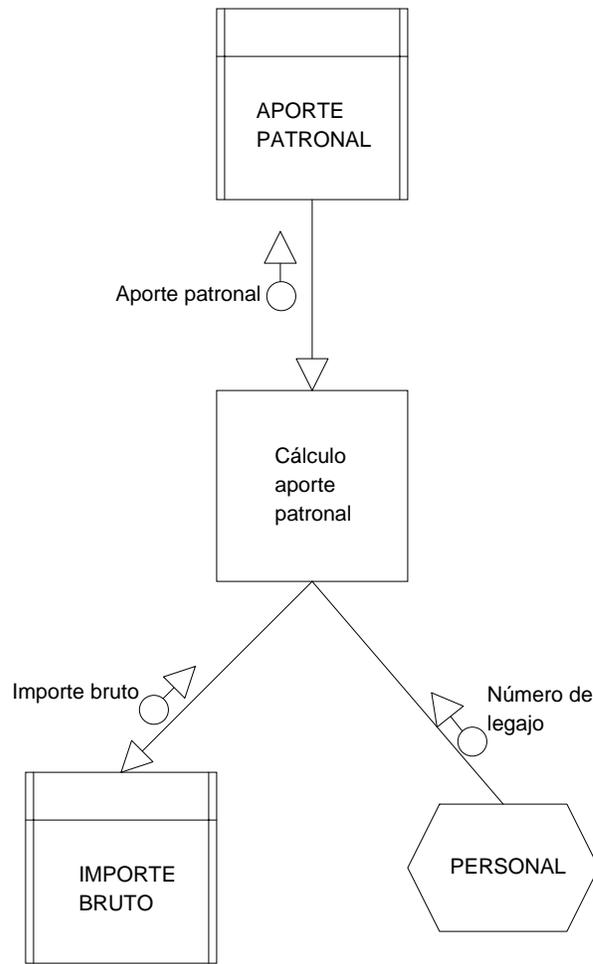


Módulos

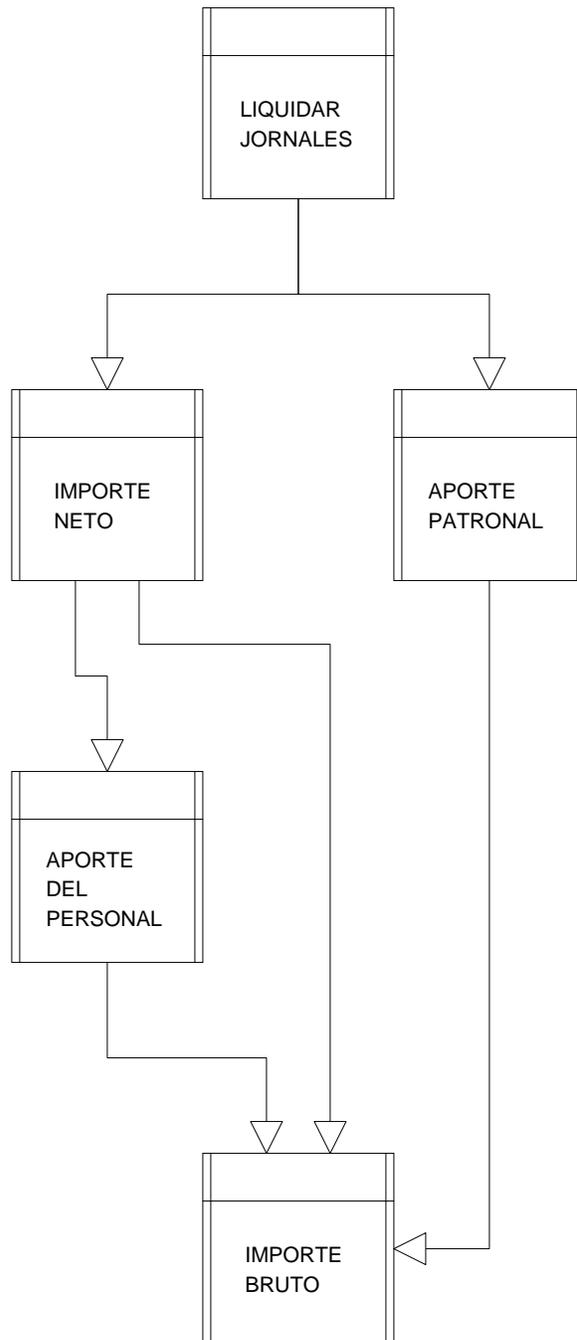








Árbol del sistema



EJERCICIO Nº:6

1 Procedimiento

Liquidar jornales

2. Operaciones

1. Calcular:
 - 1.1. Total bruto de la liquidación
 - 1.2. Total neto de la liquidación
 - 1.3. Totales de los aportes patronales
 - 1.4. Totales de los aportes personales.
2. Imprimir:
 - 2.1. Recibos
 - 2.2. Informe el total de aportes personales y patronales

3 Algoritmos

Bonificación por antigüedad = 10\$ x Años de antigüedad

Importe bruto = (Cantidad de horas trabajadas x Valor hora) + Bonificación por antigüedad

Aporte del personal = Importe bruto de cada liquidación x 10%

Aporte patronal = Importe bruto de cada liquidación x 20%

Si Estado civil = casado, entonces: Asignación por cónyuge = 10 \$

Asignación por hijo = 16\$ x Cantidad de hijos

Importe neto = Importe bruto - Aporte del personal + Asignación por cónyuge + Asignación por hijo.

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora, Cantidad de hijos, Estado civil,

Años de antigüedad)

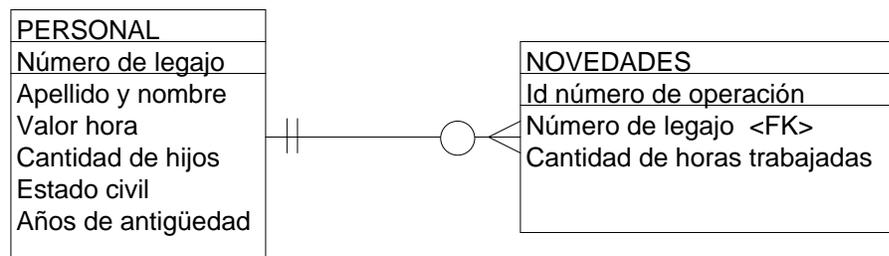
Dominio:

Estado civil = {casado, soltero}

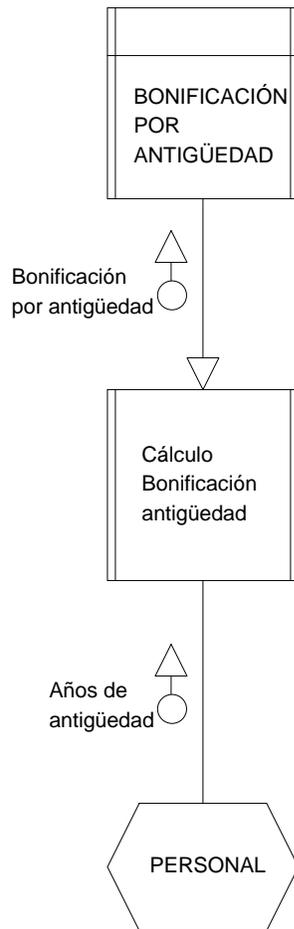
ARCHIVO DE TRANSACCIONES:

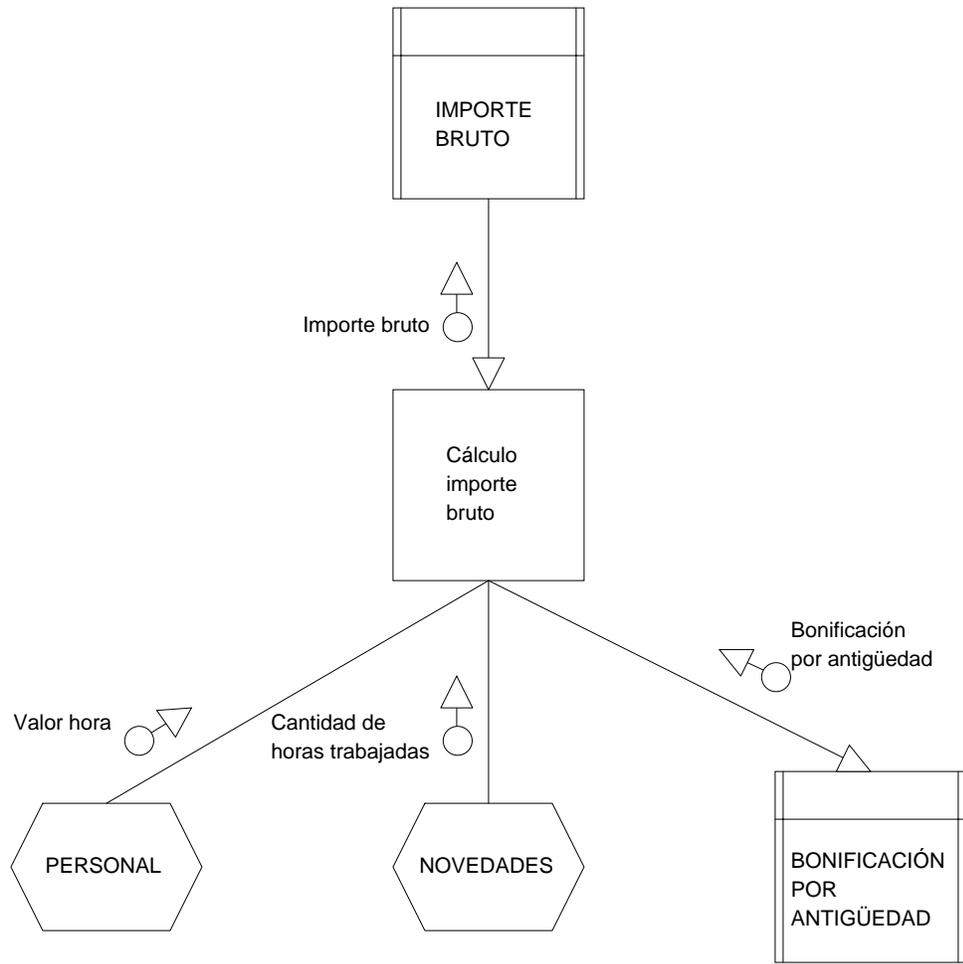
NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas)

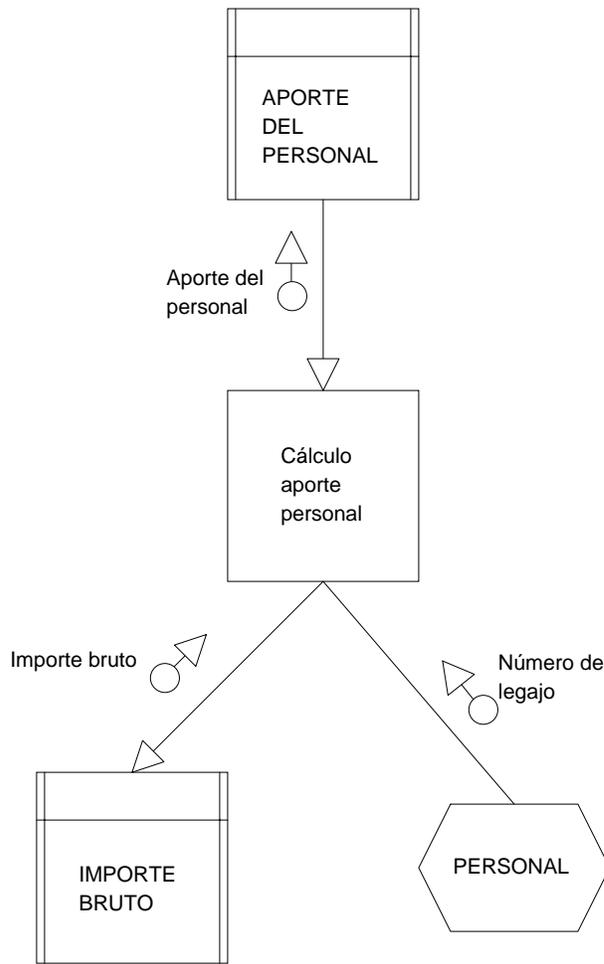
5 Diseño del modelo relacional de datos

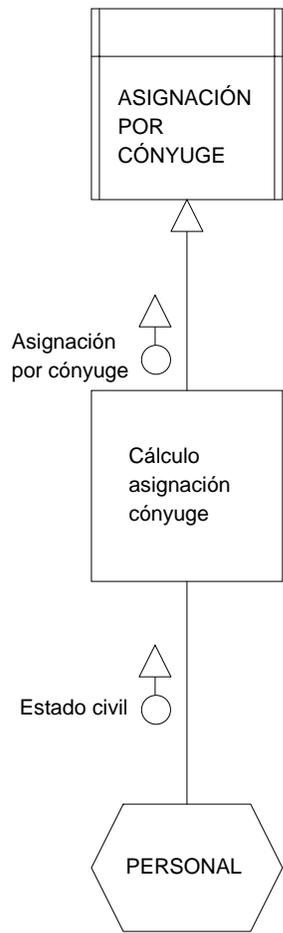


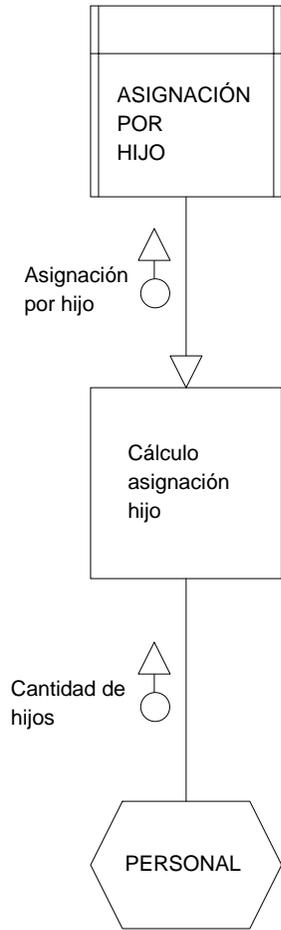
Módulos

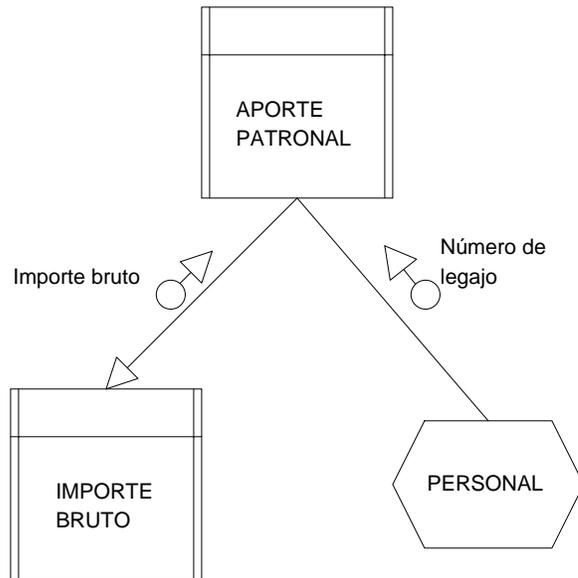
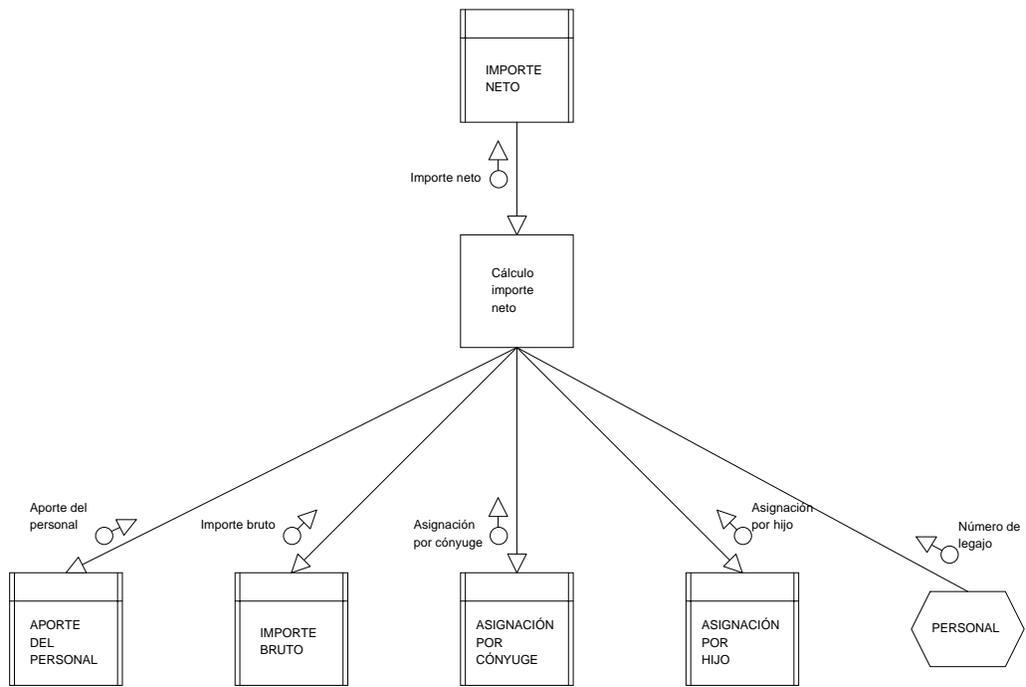




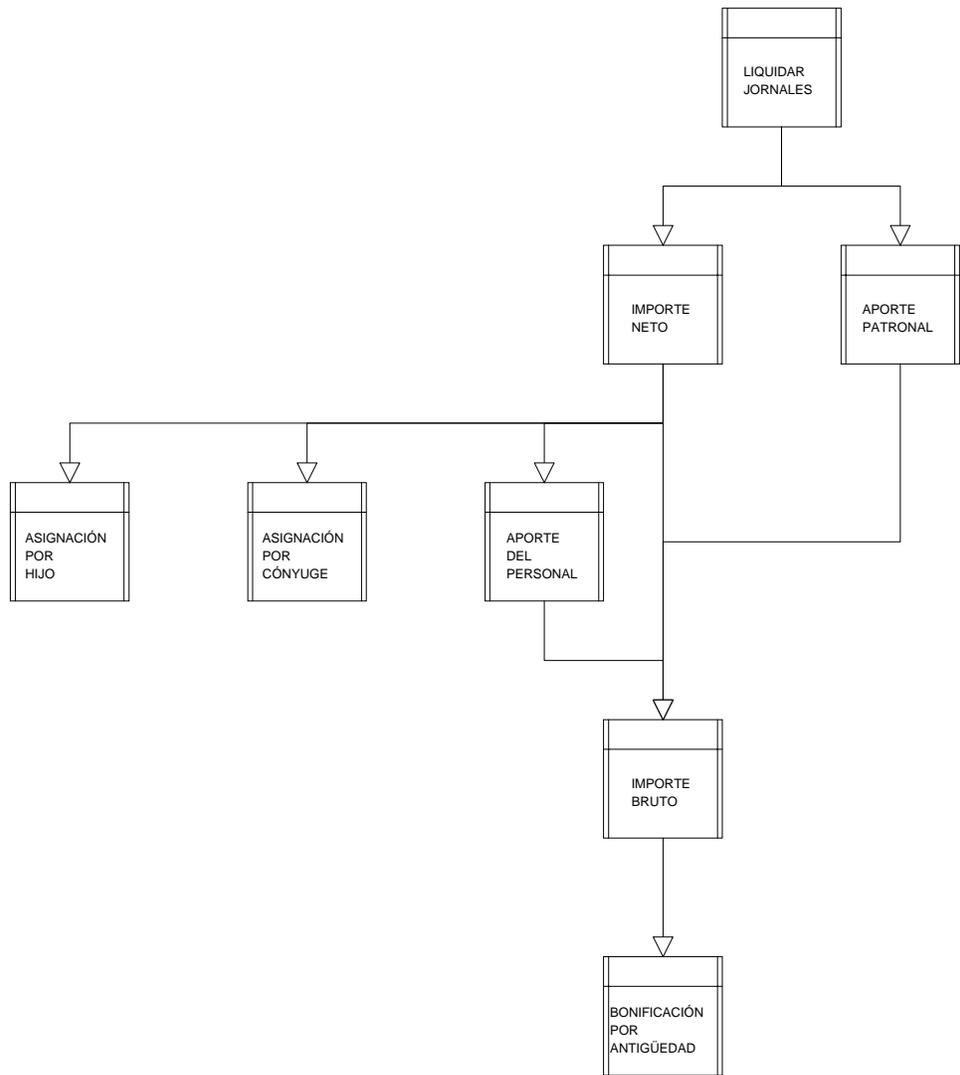








Árbol del sistema



EJERCICIO Nº:7

1 Procedimiento:

Liquidar jornales

2. Operaciones

1. Calcular
 - 1.1. Total bruto de la liquidación
 - 1.2. Totales de los aportes patronales
 - 1.3. Totales de los aportes personales
 - 1.4. Total neto de la liquidación
2. Actualizar la acumulación, del Acumulado importe bruto para aguinaldo, para el cálculo del sueldo anual complementario
3. Imprimir recibos

3 Algoritmos

Importe bruto = Cantidad de horas trabajadas x Valor hora

Aporte del personal = Importe bruto de cada liquidación x 10%

Aporte patronal = Importe bruto de cada liquidación x 20%

Importe neto = Importe bruto - Aporte del personal.

Acumulado sueldo anual complementario = Importe bruto

4 Diseño de los Archivos

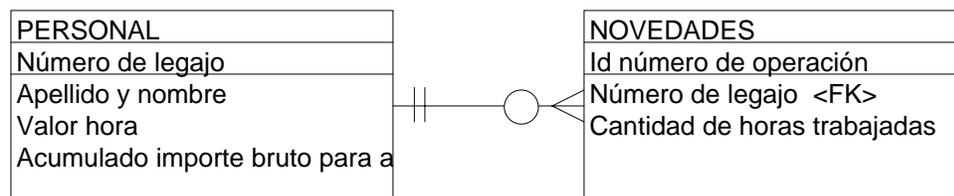
ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora, Acumulado importe bruto para aguinaldo)

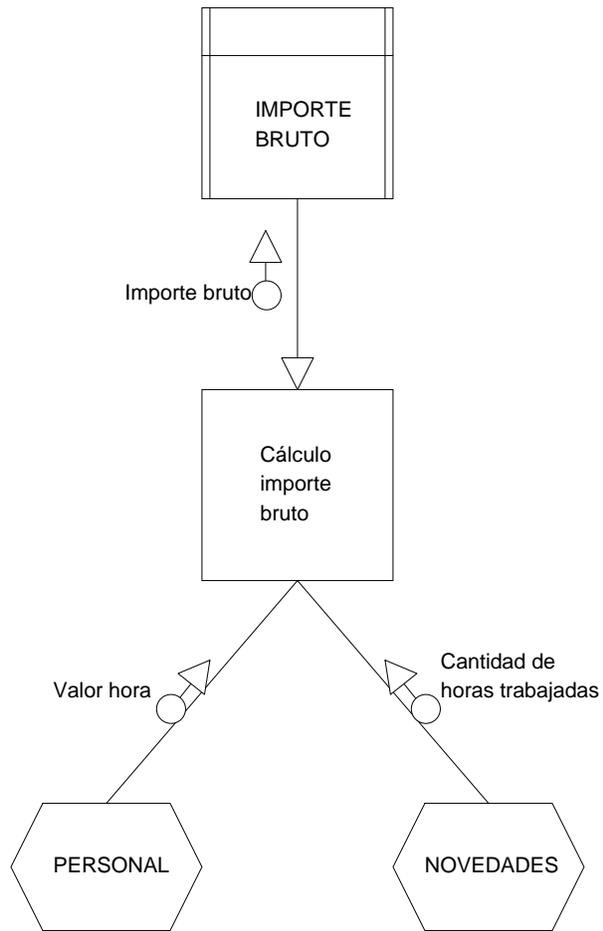
ARCHIVO DE TRANSACCIONES:

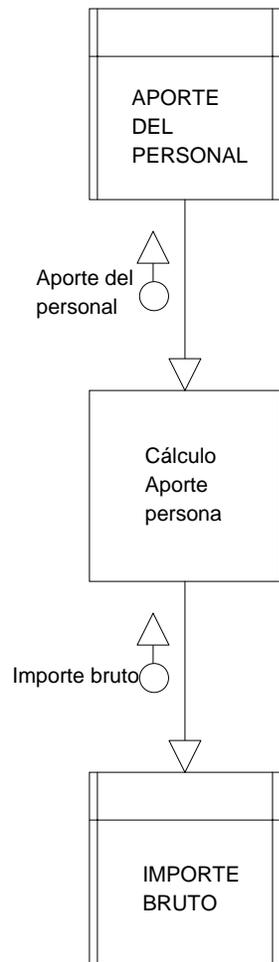
NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas)

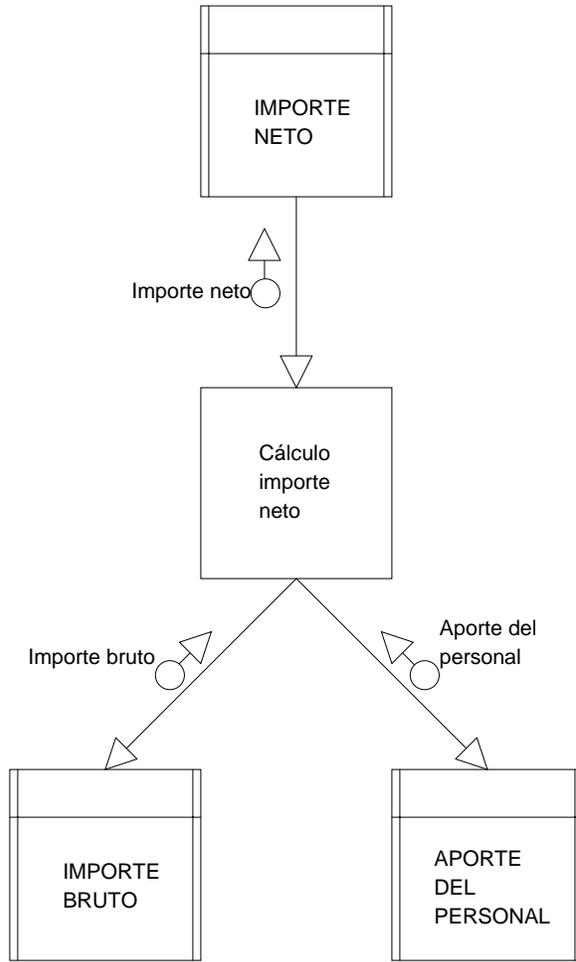
5 Diseño del modelo relacional de datos

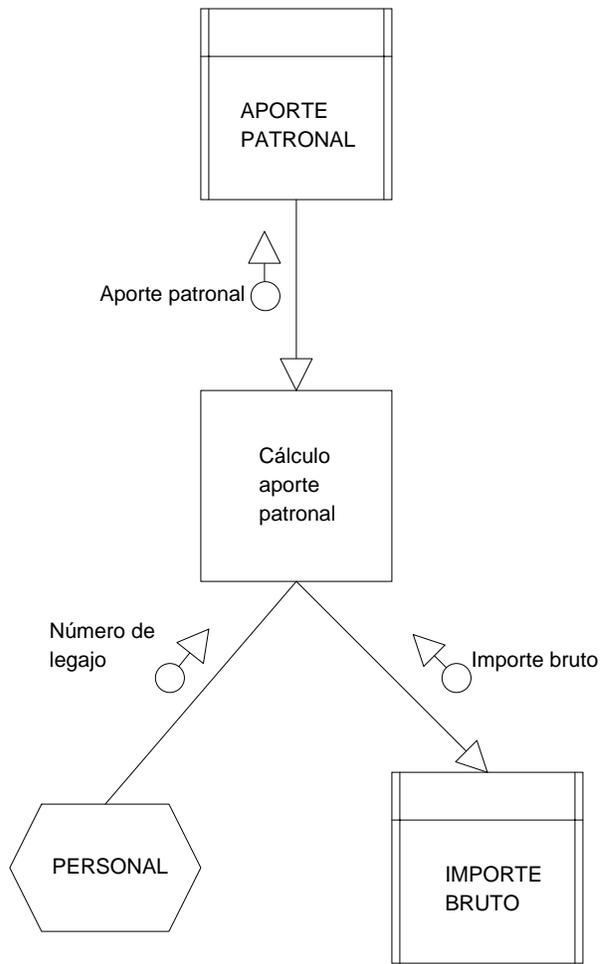


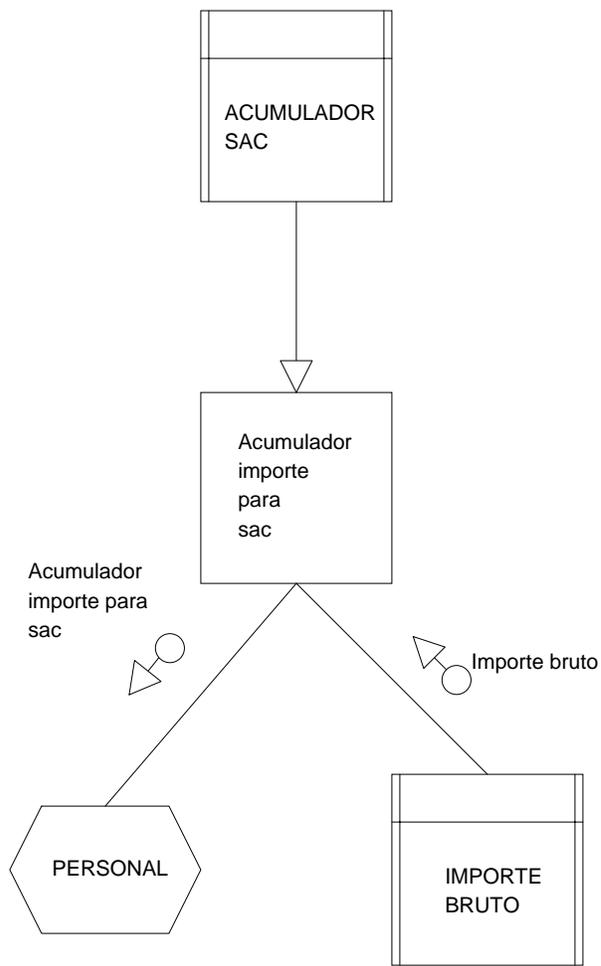
Módulos



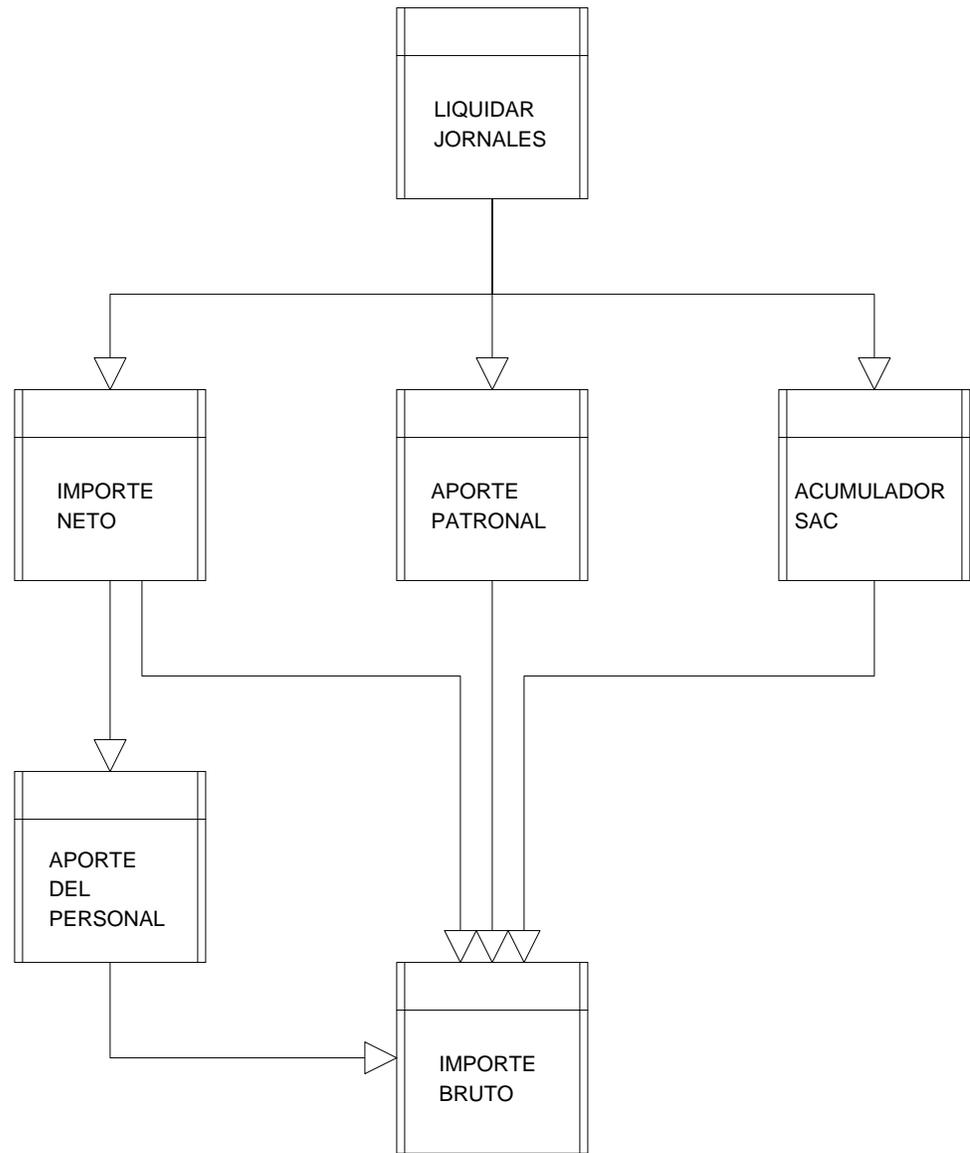








Árbol del sistema



EJERCICIO Nº:8

1 Procedimiento:

Liquidar jornales

2. Operaciones

1. Calcular:
 - 1.1. Total bruto de la liquidación
 - 1.2. Total neto de la liquidación
 - 1.3. Totales de los aportes patronales
 - 1.4. Totales de los aportes personales
2. Actualizar la acumulación, del Acumulado importe bruto para aguinaldo, para el cálculo del sueldo anual complementario
3. Imprimir recibos

3 Algoritmos

Bonificación por antigüedad = Años de antigüedad*10\$

Importe bruto = Cantidad de horas trabajadas x Valor hora + Bonificación por antigüedad

Aporte del personal = Importe bruto x 10%

Aporte patronal = Importe bruto x 20%

Si Estado Civil = casado, entonces: Asignación por cónyuge = 10\$

Asignación por hijo = 16\$ x Cantidad de hijos

Importe neto = importe bruto - Aporte del personal + Asignación por cónyuge + Asignación por hijo.

Sueldo anual complementario = Cantidad de horas trabajadas x Valor hora = Importe bruto

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PERSONAL = (Número de legajo, Apellido y nombre, Valor hora, Acumulado importe bruto para aguinaldo, Cantidad de hijos, Estado Civil, Años de antigüedad)

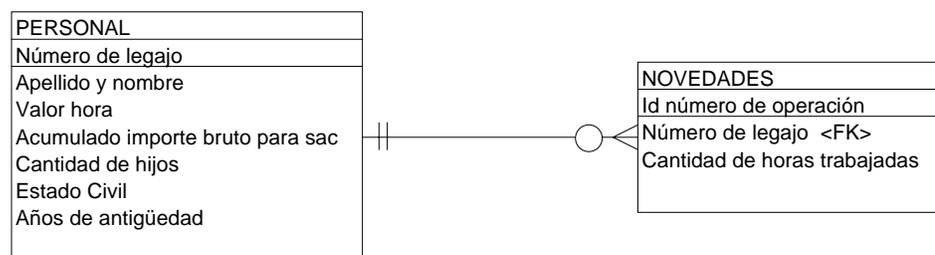
Dominio:

Estado civil = { casado , soltero }

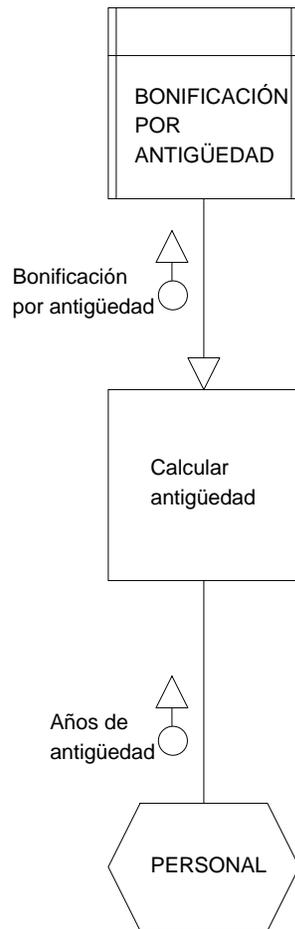
ARCHIVO DE TRANSACCIONES:

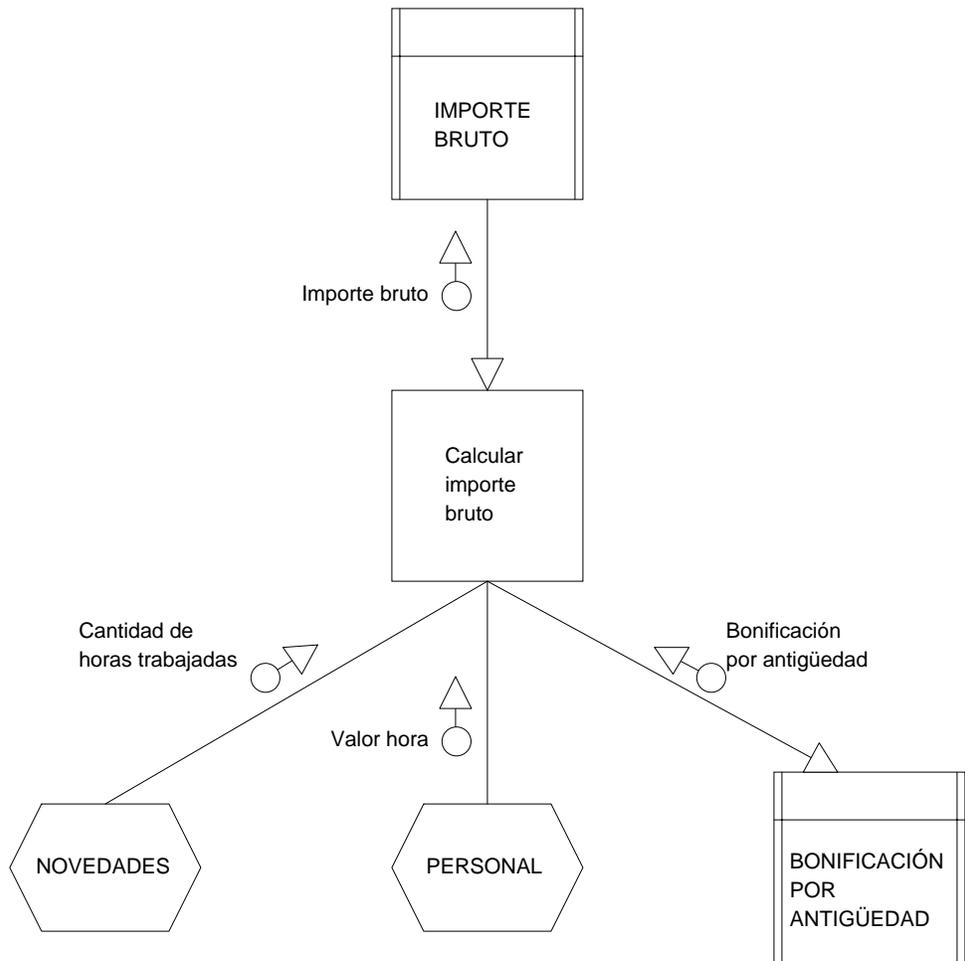
NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas)

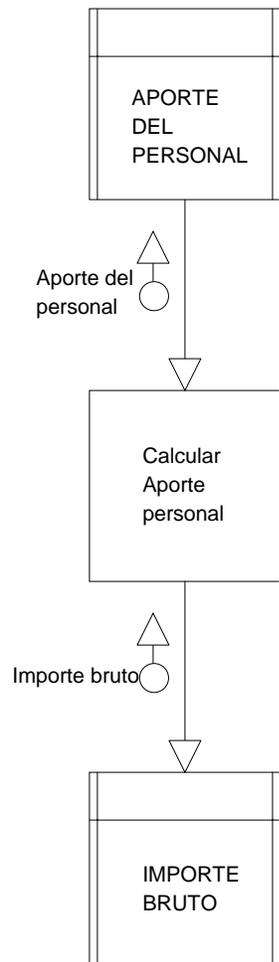
5 Diseño del modelo relacional de datos

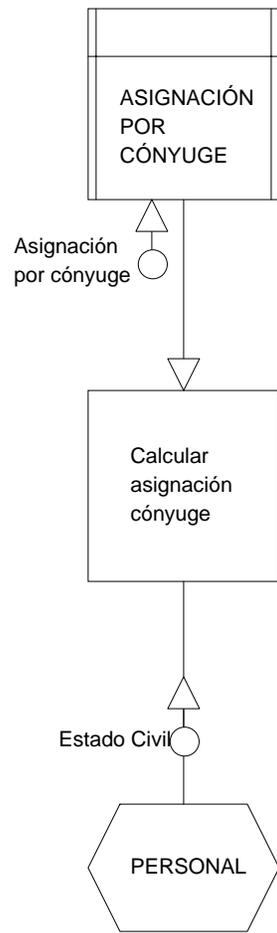


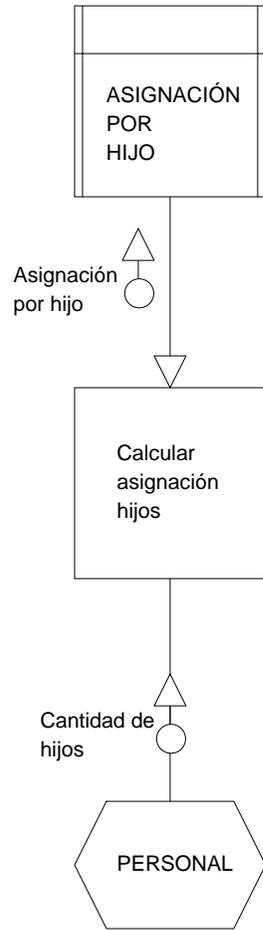
Módulos

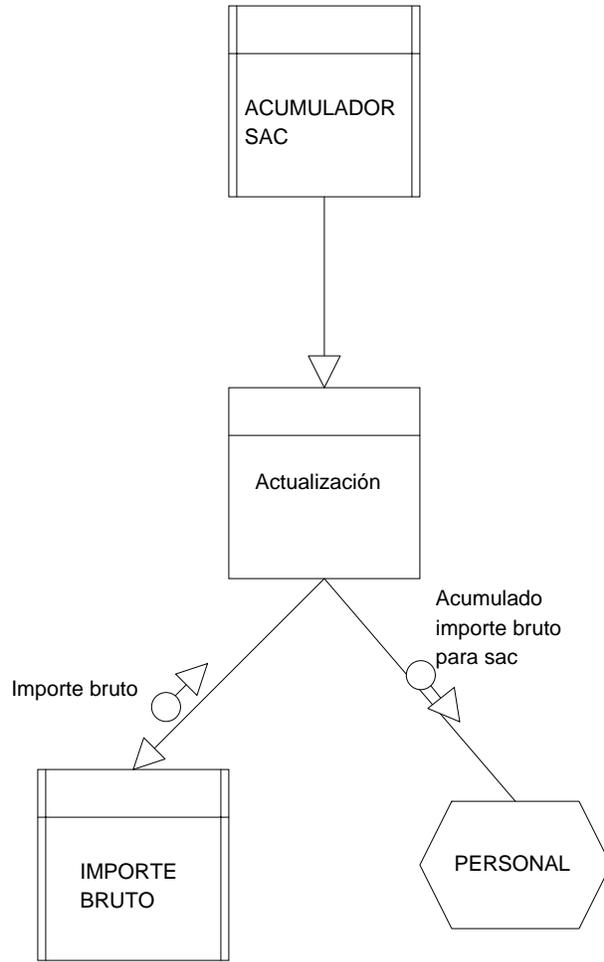


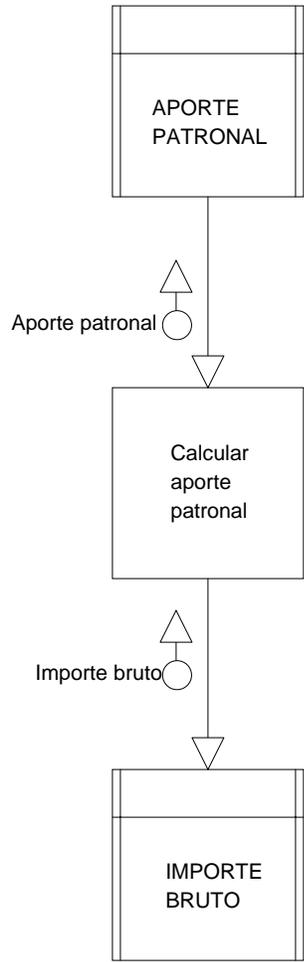


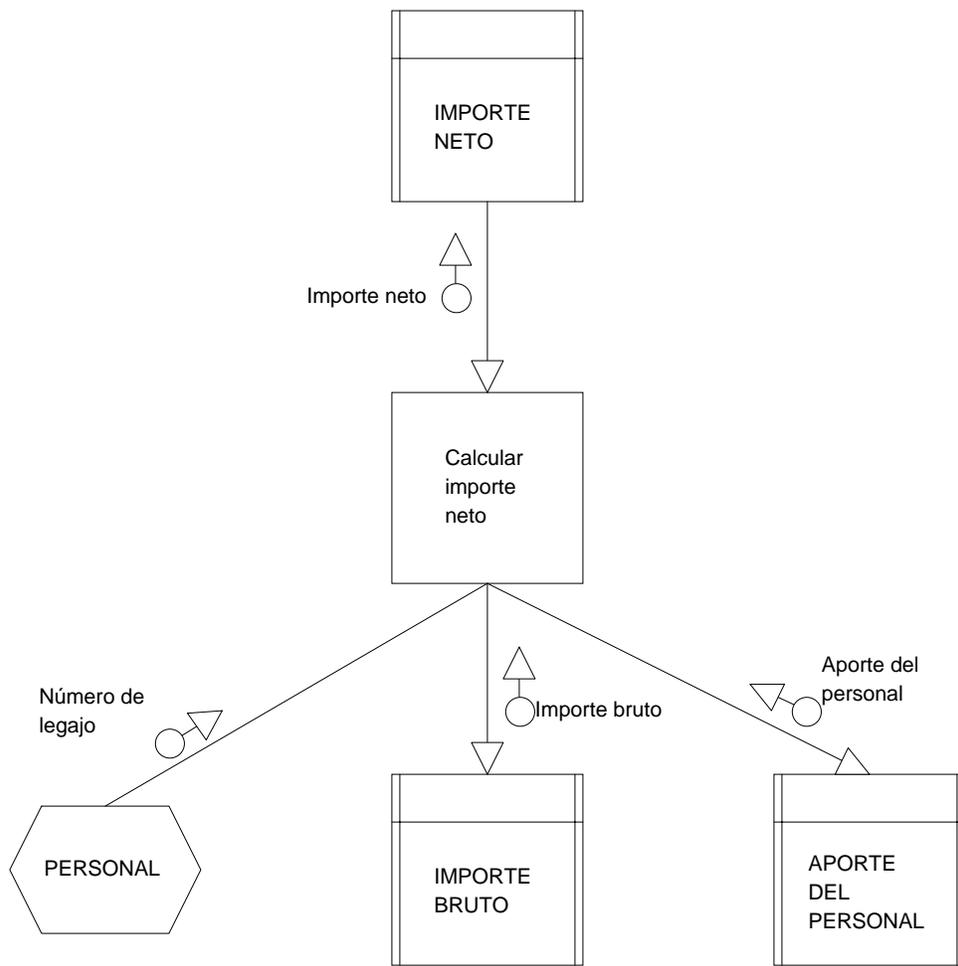




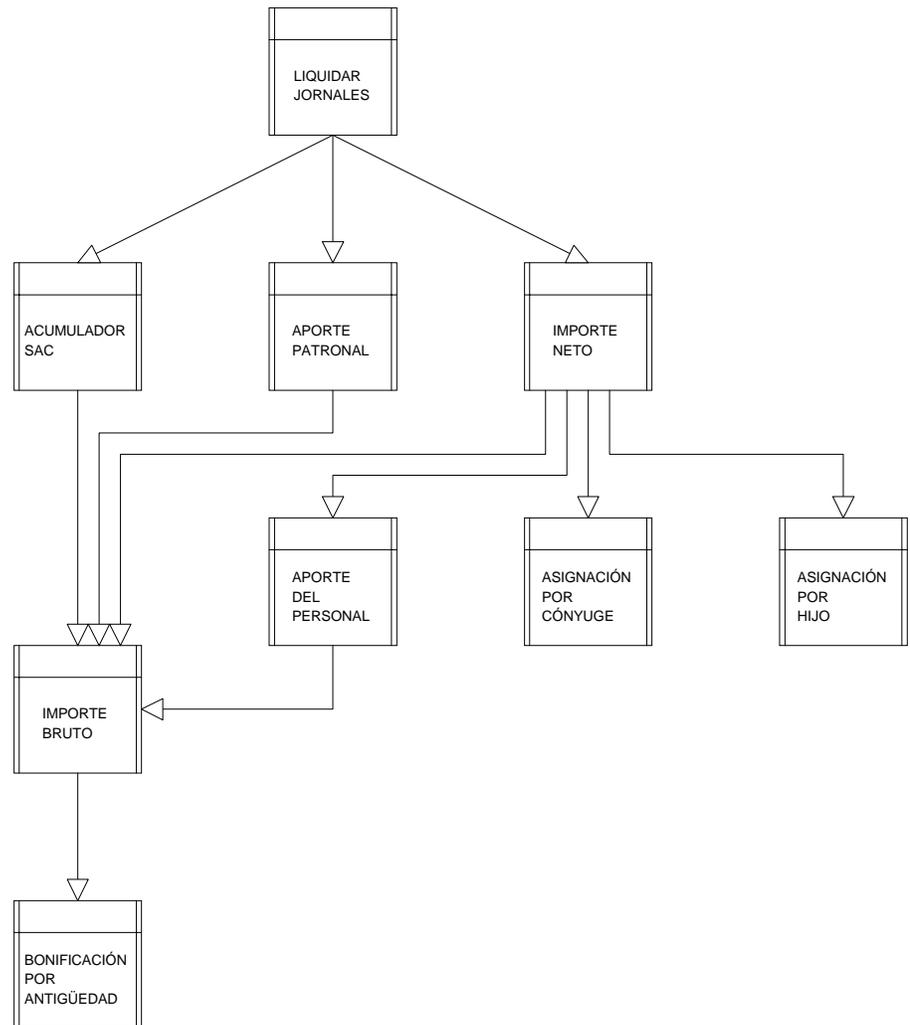








Árbol del sistema



EJERCICIO Nº:9

1 Procedimiento:

Liquidar jornales

2. Operaciones

1. Calcular:
 - 1.1. Total bruto de la liquidación
 - 1.2. Totales de los aportes patronales
 - 1.3. Totales de los aportes personales
 - 1.4. Total neto de la liquidación
2. Imprimir recibos

3 Algoritmos

Importe bruto = Cantidad de horas trabajadas x Valor hora x Código de hora
(un mismo legajo puede incluir horas trabajadas normales y distintos tipos de horas extras)
Aporte del personal = Importe bruto x 10%
Aporte patronal = Importe bruto x 20%
Importe neto = Importe bruto - Aporte del personal.

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora).

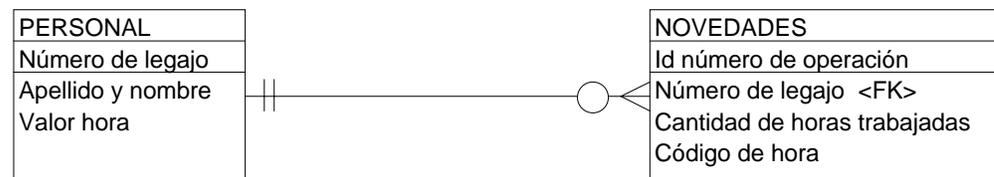
ARCHIVO DE TRANSACCIONES:

NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas, Código de hora)

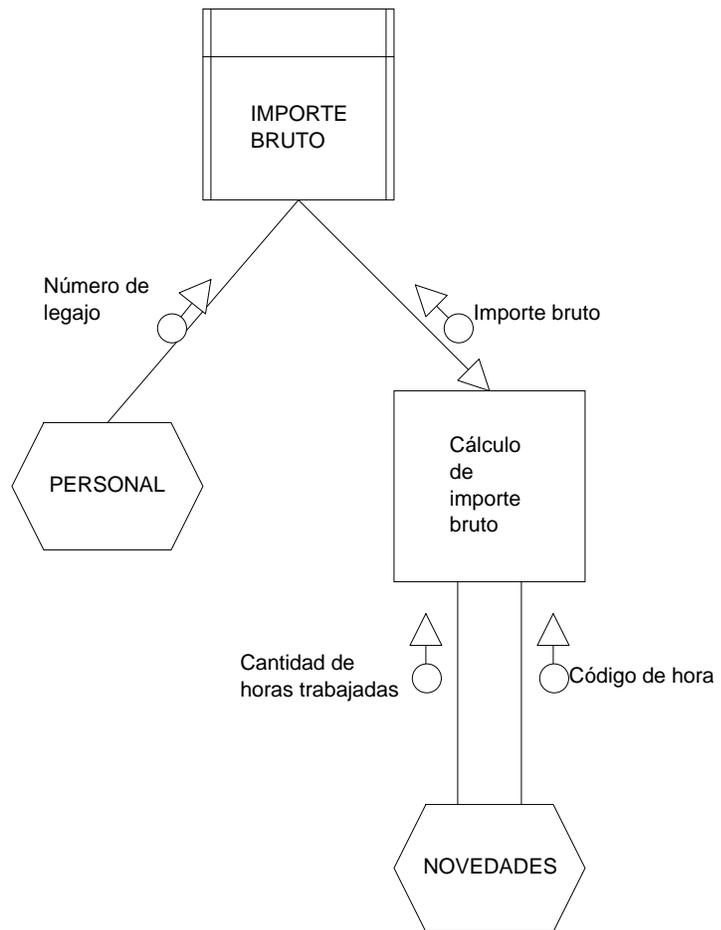
Dominio:

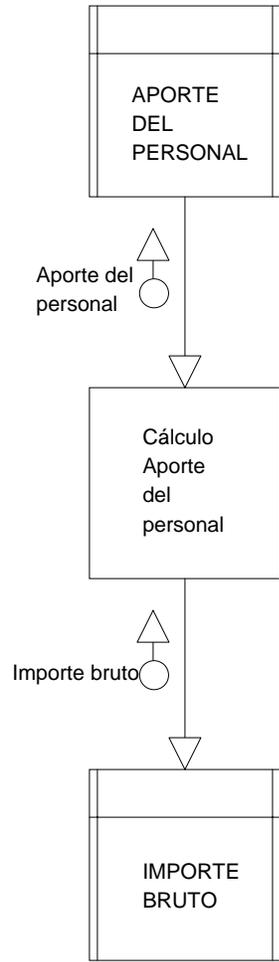
Código de hora = { 1=Valor hora, 2= Valor hora x 0,5, 3= Valor hora x 2 }

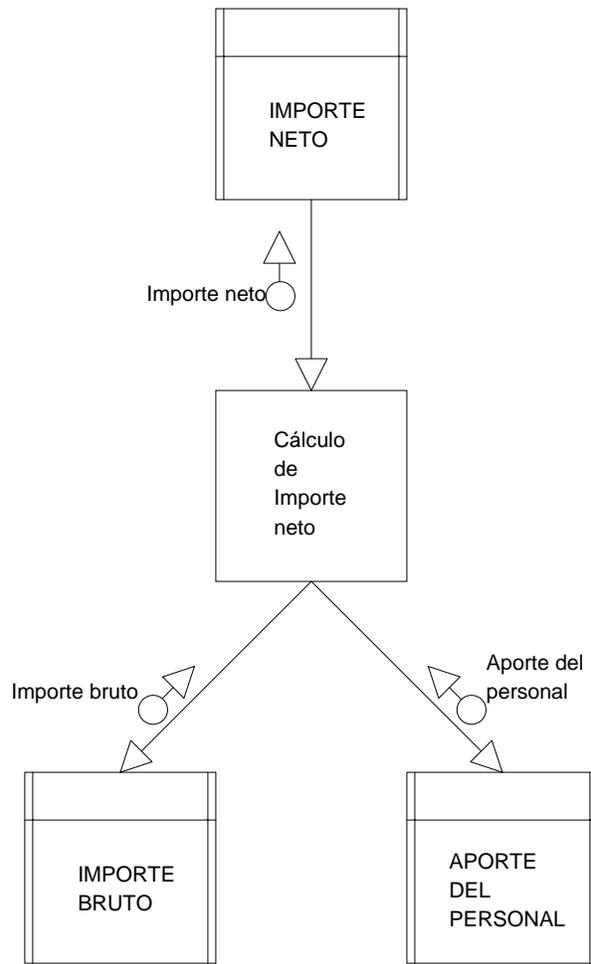
5 Diseño del modelo relacional de datos

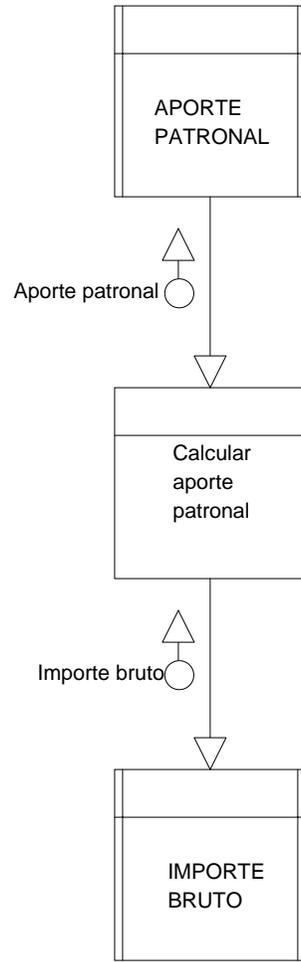


Módulos

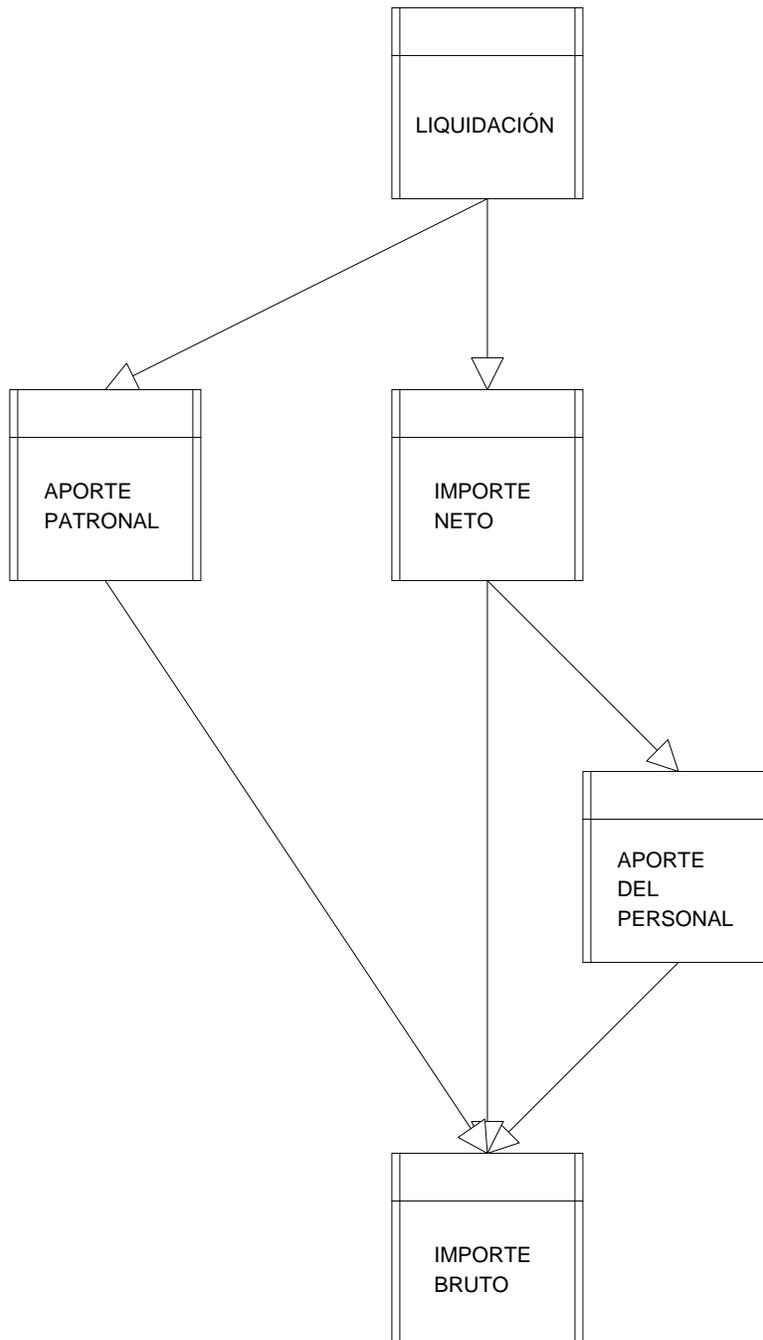








Árbol del sistema



EJERCICIO Nº:10

1 Procedimiento:

Liquidar jornales

2. Operaciones

1. Calcular:
 - 1.1. Total bruto de la liquidación
 - 1.2. Total neto de la liquidación
2. Imprimir recibos

3 Algoritmos

Bonificación por antigüedad = Años de antigüedad x 10\$

Importe bruto = \sum (Cantidad de horas trabajadas x Valor hora x Código de la hora)+
Bonificación por antigüedad

Aporte del personal = Importe bruto de cada liquidación x 10%

Aporte patronal = Importe bruto de cada liquidación x 20%

Si Estado Civil = casado, entonces: Asignación por cónyuge = 10\$

Asignación por hijo = 16\$ x Cantidad de hijos

Importe neto = Importe bruto - Aporte del personal + Asignación por cónyuge + Asignación por hijo.

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora, Cantidad de hijos, Estado Civil,

Años de antigüedad).

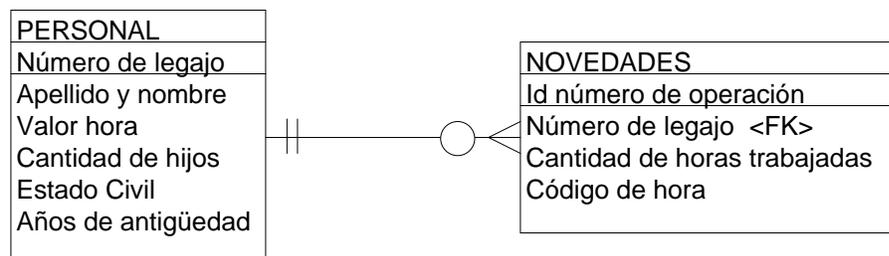
Dominio:

Estado civil = {casado , soltero }

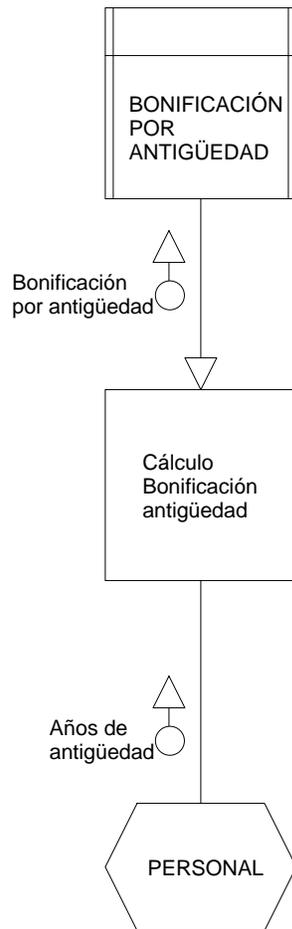
ARCHIVO DE TRANSACCIONES:

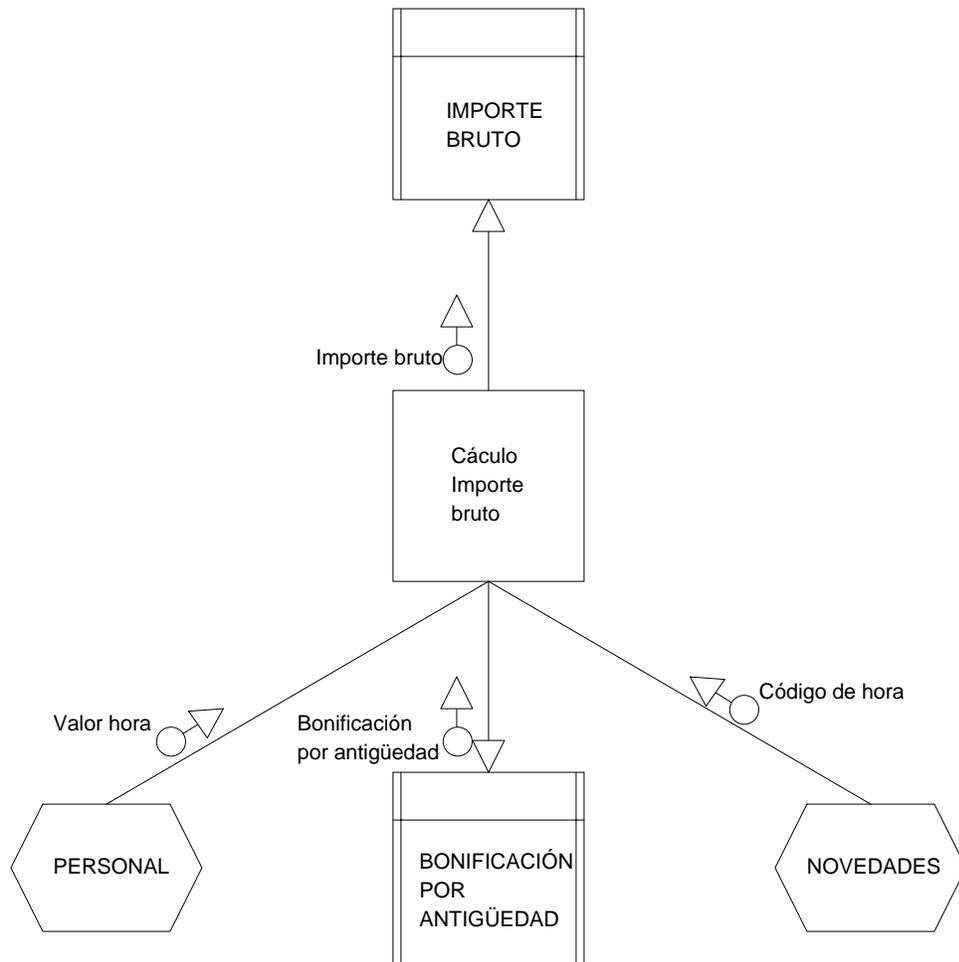
NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas, Código de hora)

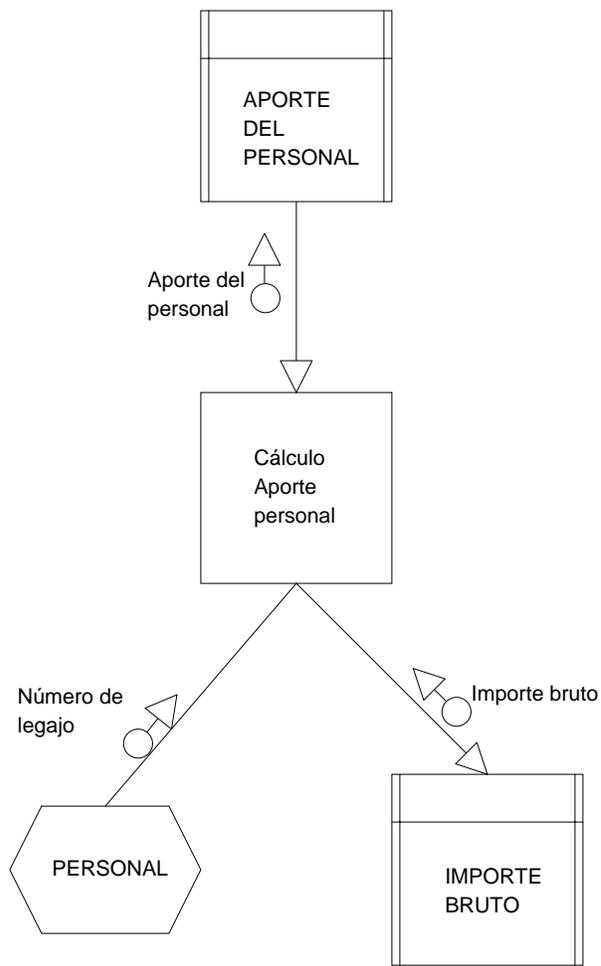
5 Diseño del modelo relacional de datos

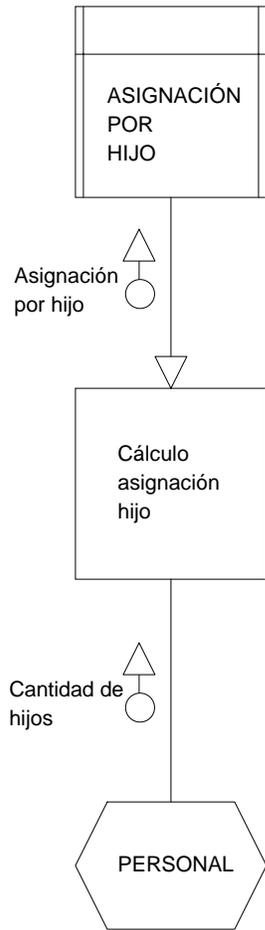


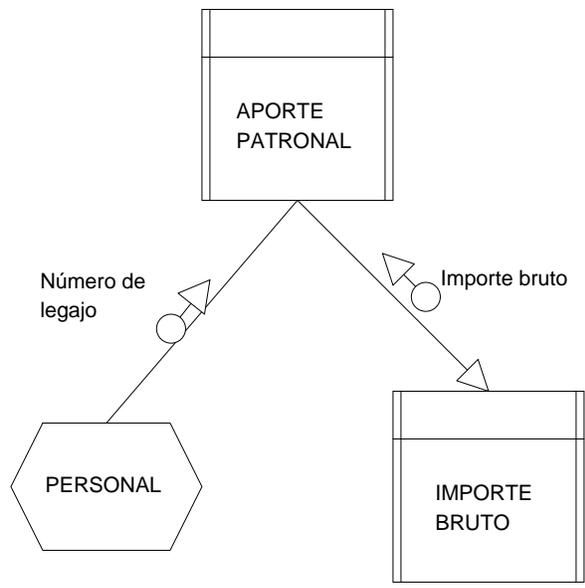
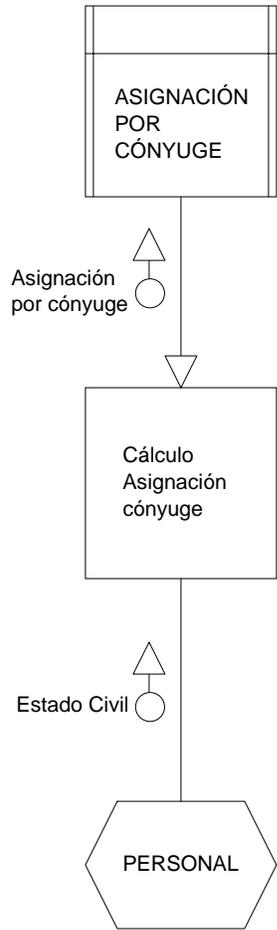
Módulos

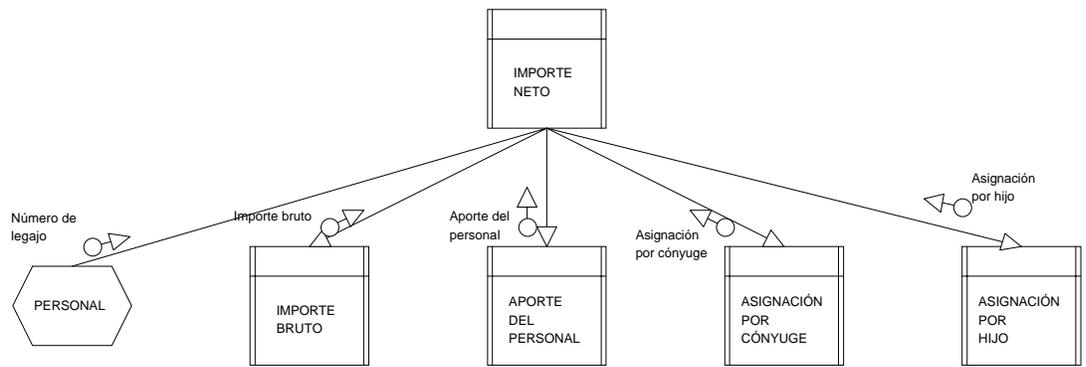




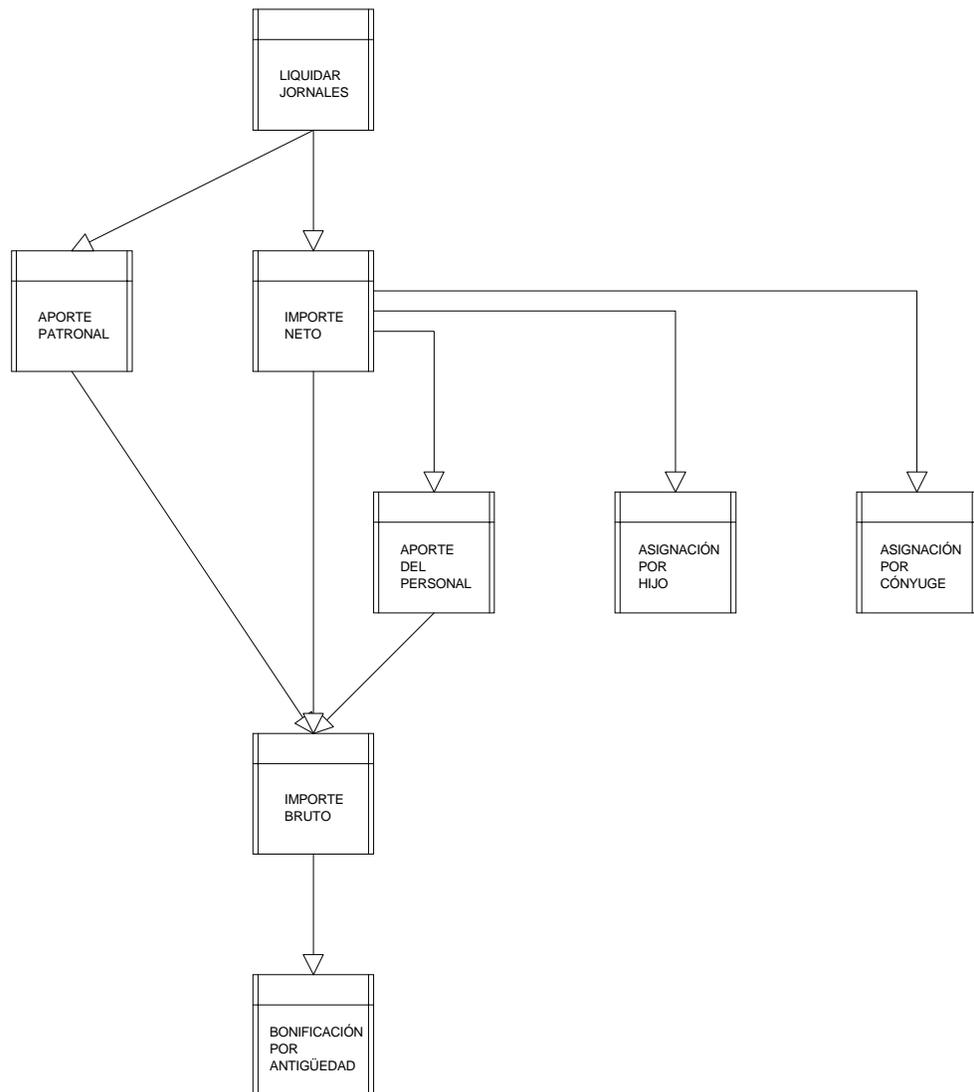








Árbol del sistema



EJERCICIO Nº:11

1 Procedimiento:

Liquidar jornales

2. Operaciones

1. Calcular:
 - 1.1. Total bruto de la liquidación
 - 1.2. Total neto de la liquidación
 - 1.3. Totales de los aportes patronales
 - 1.4. Totales de los aportes personales
2. Actualizar la acumulación, del Acumulado importe bruto para aguinaldo, para el cálculo del sueldo anual complementario
3. Imprimir recibos

3 Algoritmos

Bonificación por antigüedad = Años de antigüedad x 10

Jornal = \sum (Cantidad de horas trabajadas x Valor hora x Código de la hora)

Importe bruto = Jornal + Bonificación por antigüedad

Aporte del personal = Importe bruto x 10%

Aporte patronal = Importe bruto x 20%

Si Estado Civil = casado, entonces: Asignación por cónyuge = 10\$

Asignación por hijo = 16\$ x Cantidad de hijos

Importe neto = Importe bruto + Asignación por cónyuge + Asignación por hijo - Aporte del personal.

Sueldo anual complementario = \sum (Cantidad de horas trabajadas x Valor hora x Código de la hora) (Jornal)

Código de la hora = { 1=Valor hora; 2=Valor hora x 1,5 ; 3=Valor hora x 2 }

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora, Acumulado importe bruto para aguinaldo, Cantidad de hijos, Estado Civil, Años de antigüedad)

Dominio:

Estado civil = { casado , soltero }

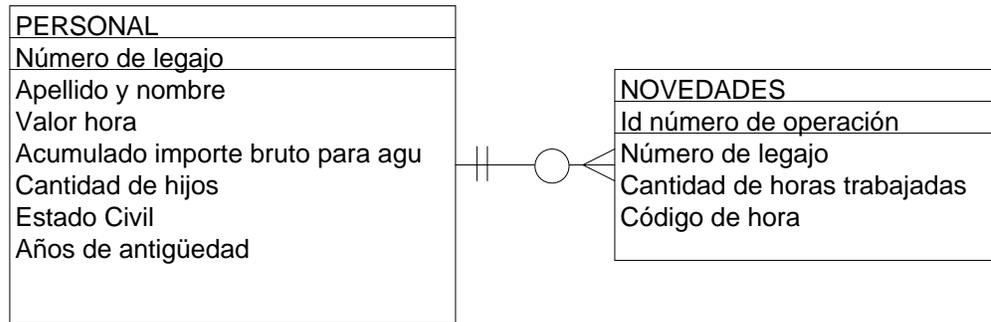
ARCHIVO DE TRANSACCIONES:

NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas, Código de hora)

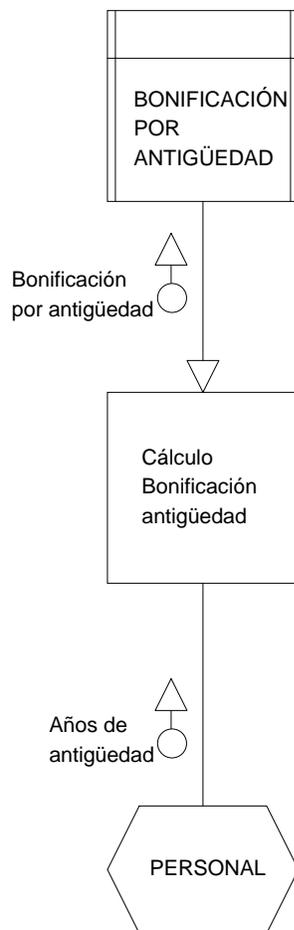
Dominio:

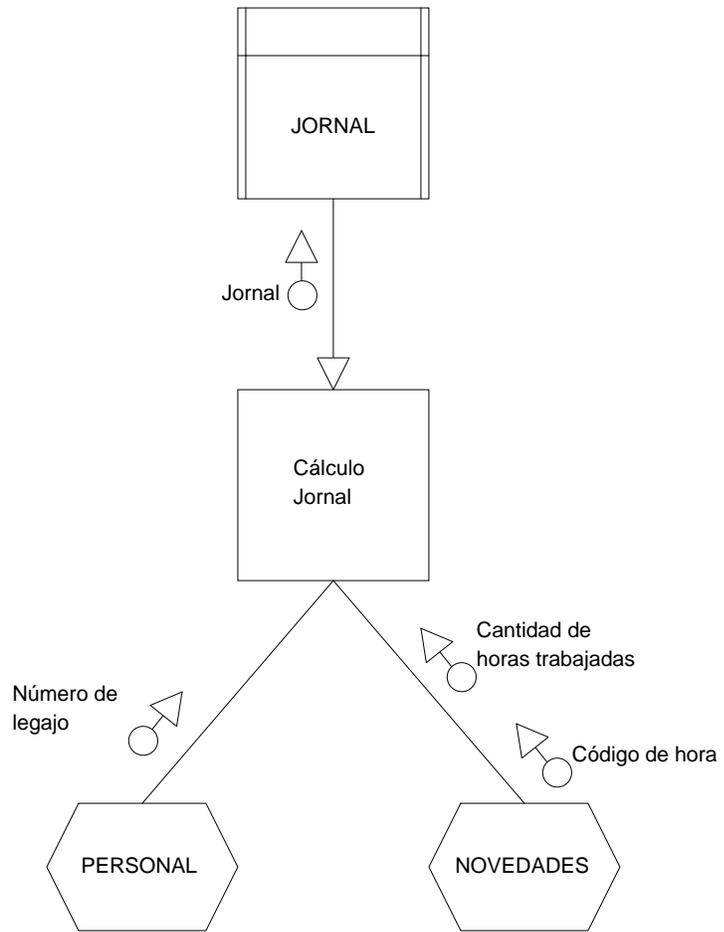
Código de hora = { 1; 2; 3 }

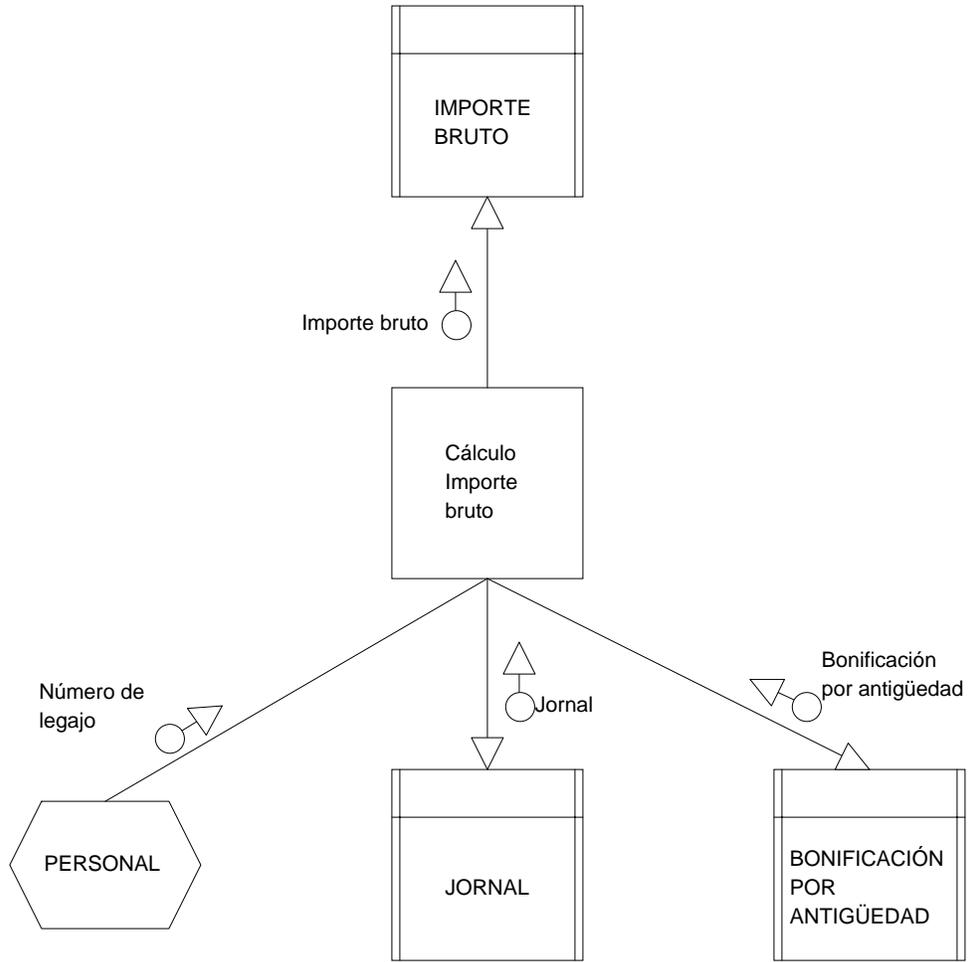
5 Diseño del modelo relacional de datos

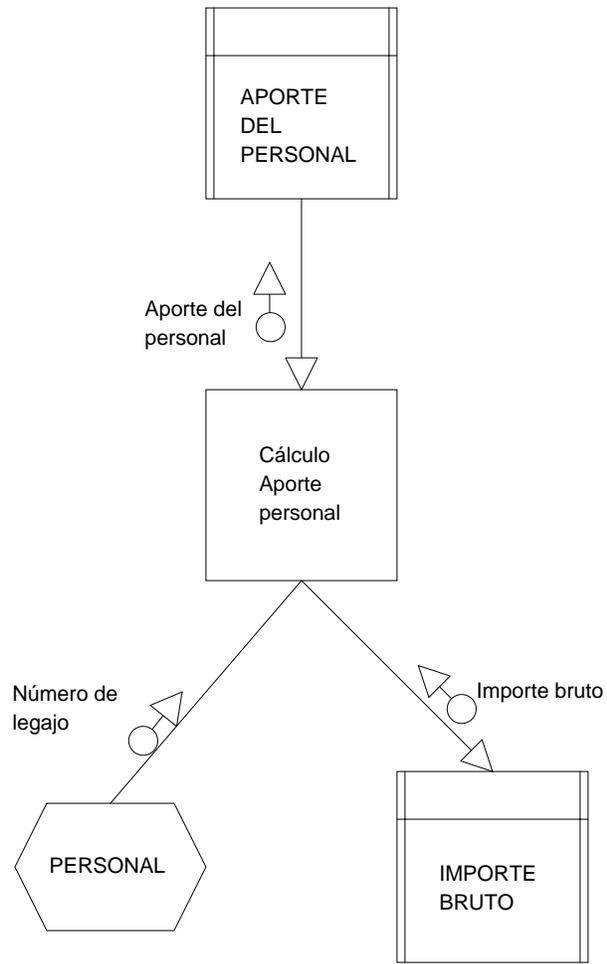


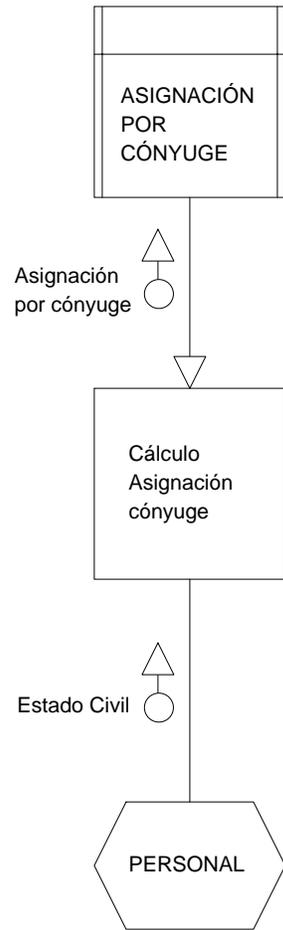
Módulos

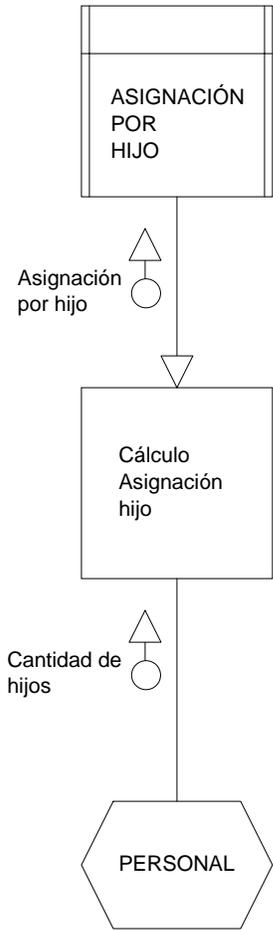


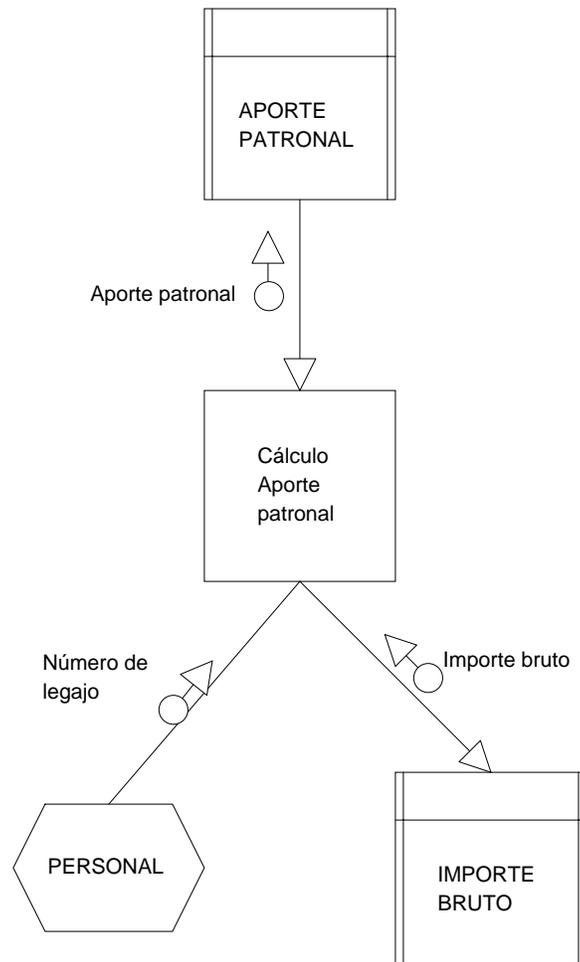


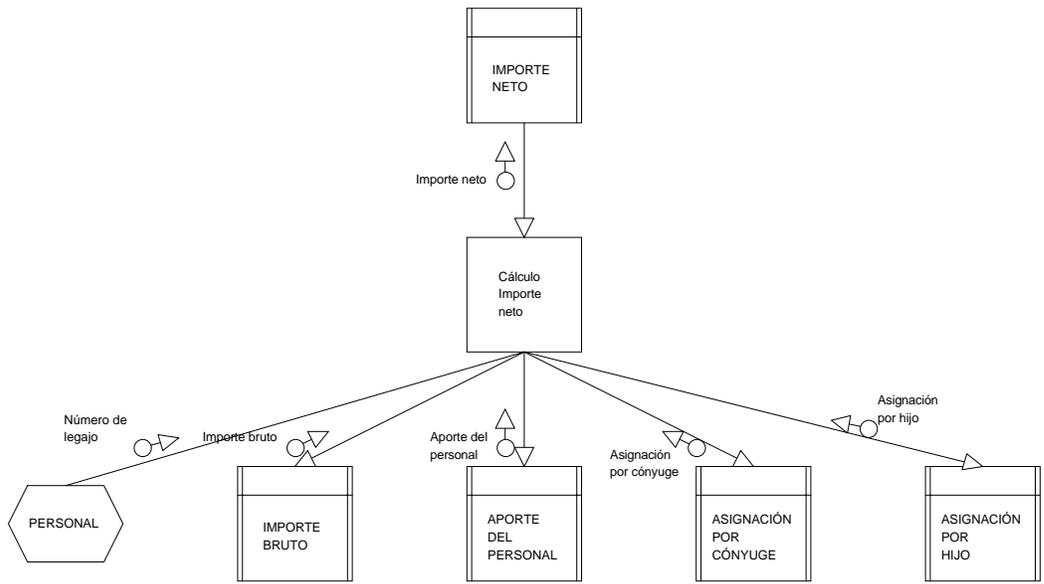


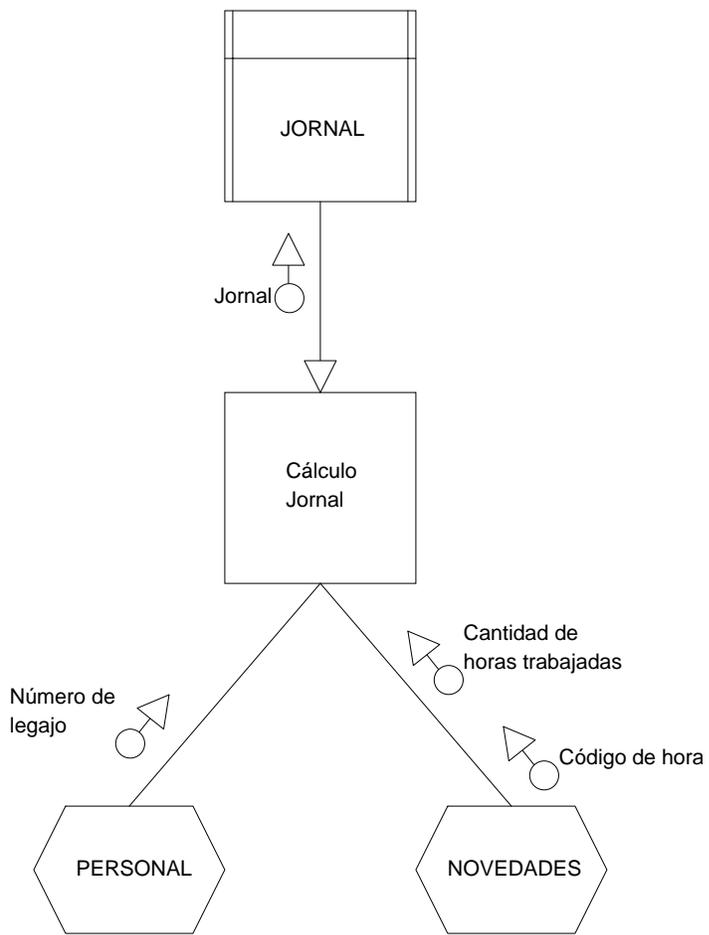


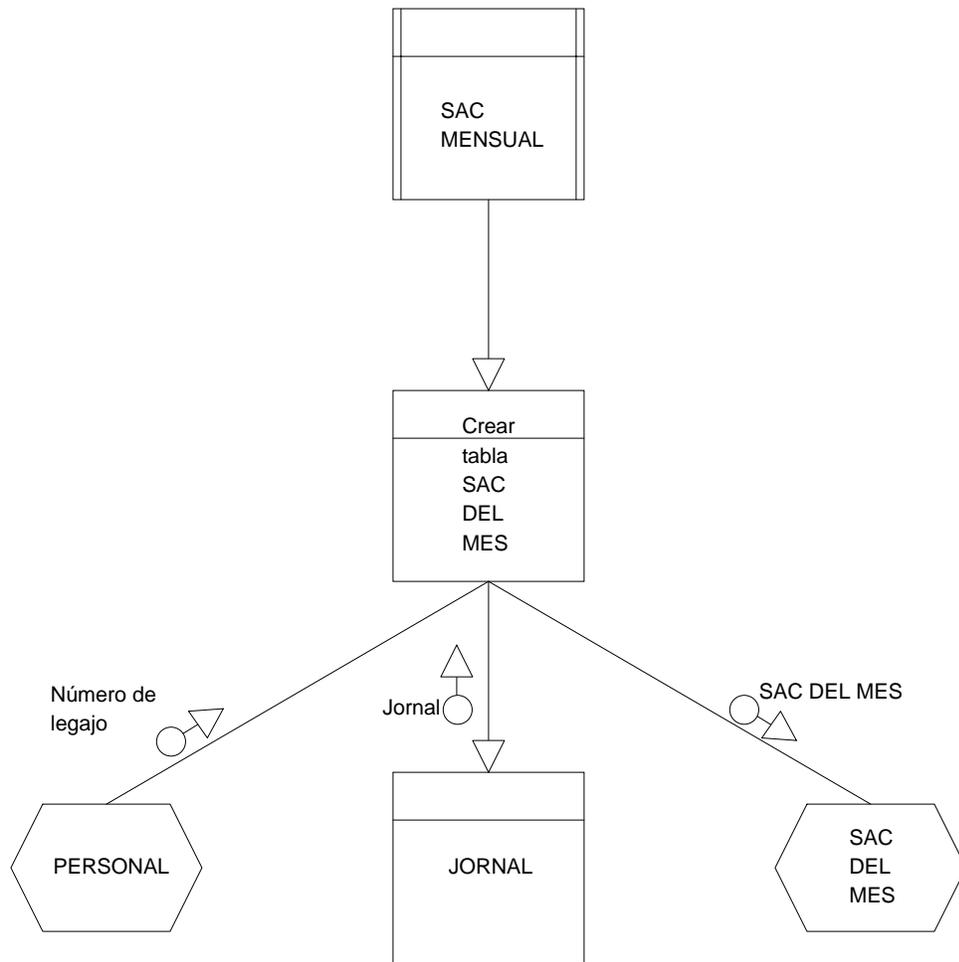


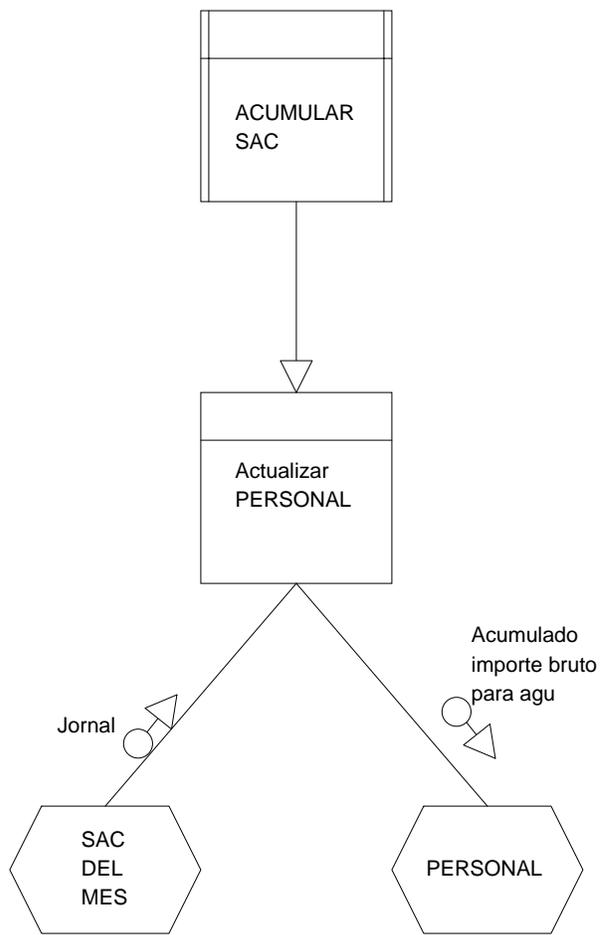




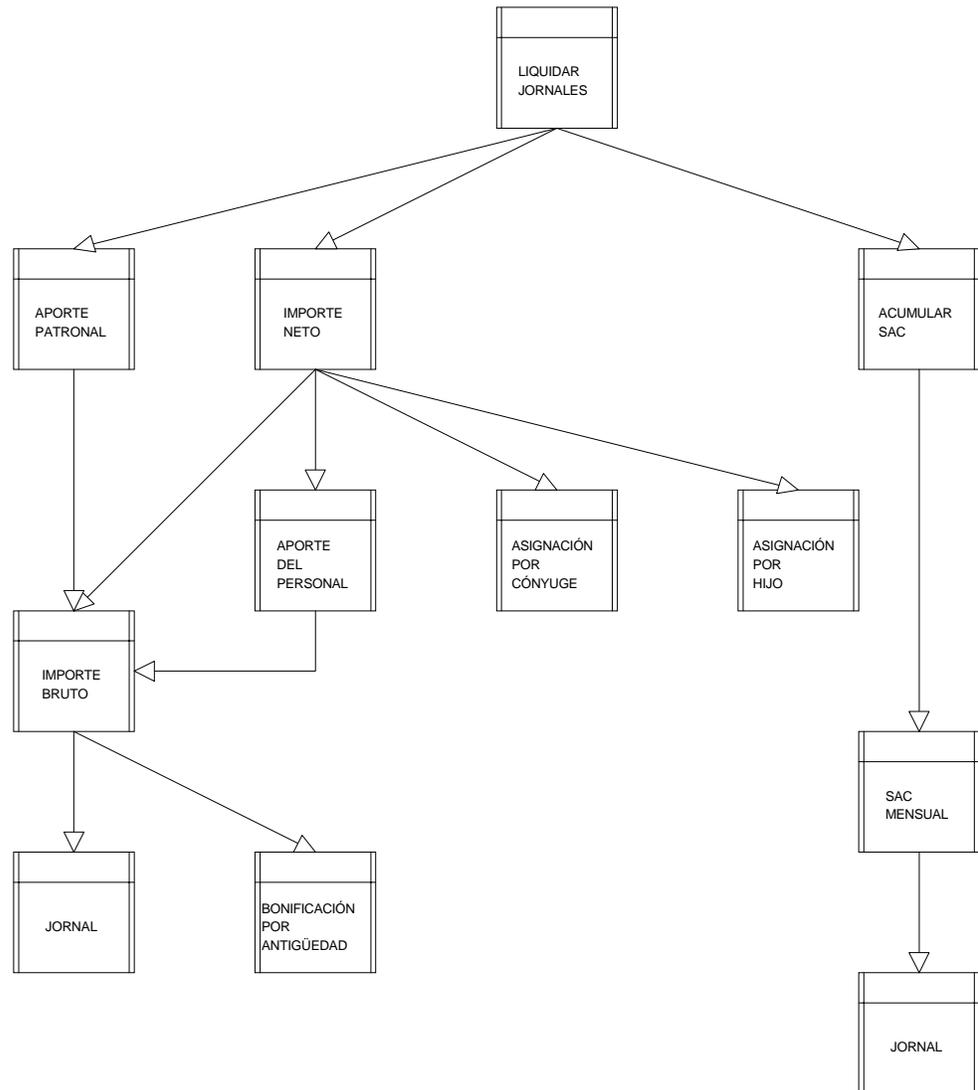








Árbol del sistema



EJERCICIO Nº:12

1 Procedimiento:

Liquidar jornales

2. Operaciones

1. Calcular:
 - 1.1. Total bruto de la liquidación
 - 1.2. Total neto de la liquidación
 - 1.3. Totales de los aportes patronales
 - 1.4. Totales de los aportes personales
2. Actualizar la acumulación, del , Acumulado importe bruto para aguinaldo, para el cálculo del sueldo anual complementario
3. Imprimir recibos

3 Algoritmos

Importe bruto = \sum (Cantidad de horas trabajadas x Valor hora x Código de la hora)

Aporte del personal = Importe bruto de cada liquidación x 10%

Aporte patronal = Importe bruto de cada liquidación x 20%

Importe neto = Importe bruto - Aporte del personal.

Sueldo anual complementario = Cantidad de horas trabajadas x Valor hora x Código de la hora

Código de la hora = { 1=Valor hora; 2=Valor hora x 1,5 ; 3=Valor hora x 2 }

4 Diseño de los Archivos

ARCHIVO MAESTRO:

PERSONAL (Número de legajo, Apellido y nombre, Valor hora, Acumulado importe bruto para aguinaldo)

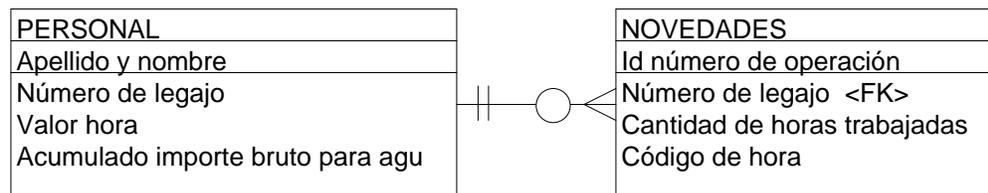
ARCHIVO DE TRANSACCIONES:

NOVEDADES (Id número de operación, Número de legajo, Cantidad de horas trabajadas, Código de hora)

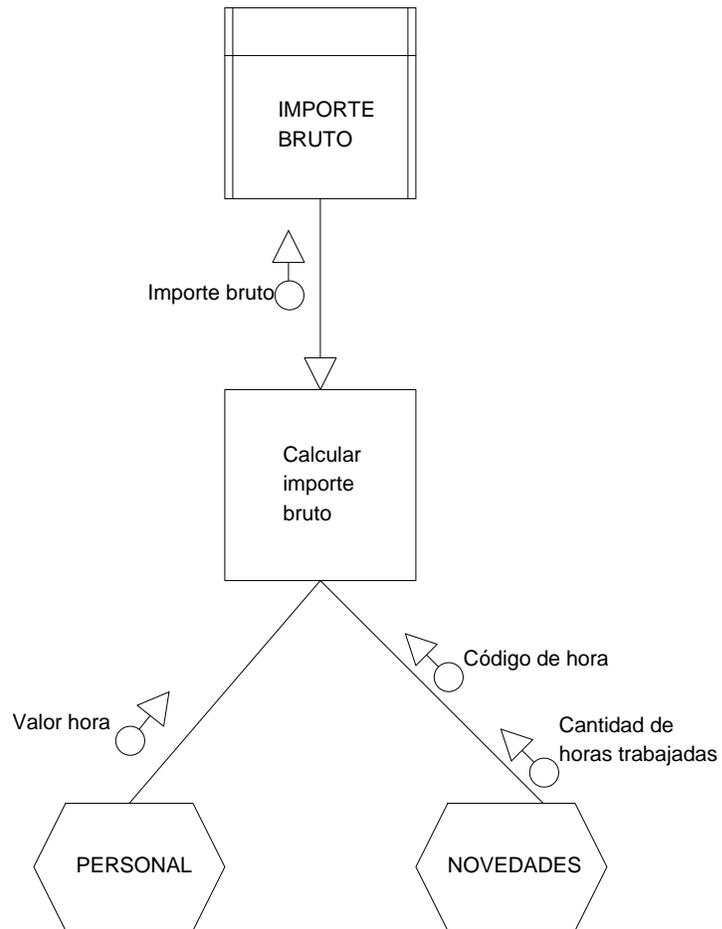
Dominio:

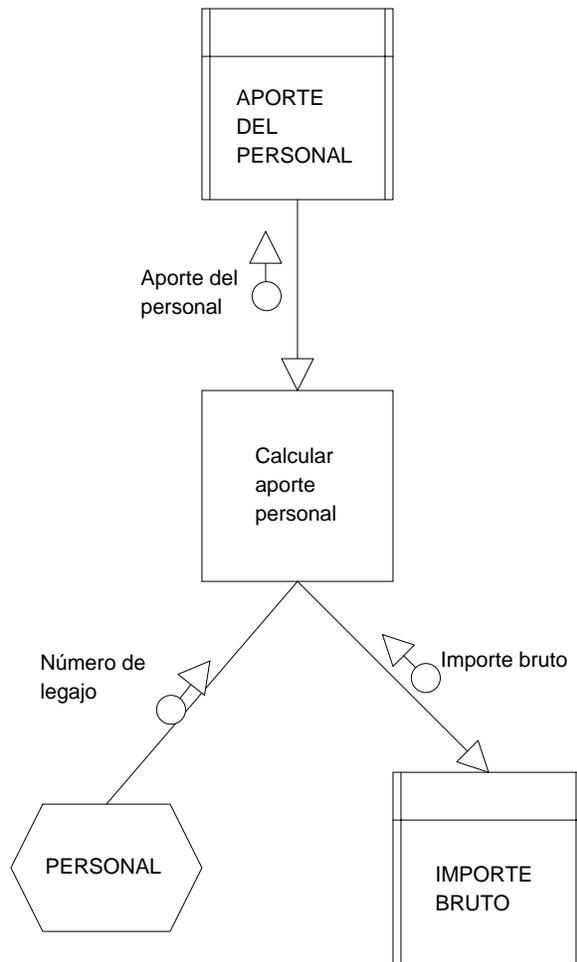
Código de la hora = {1; 2; 3}

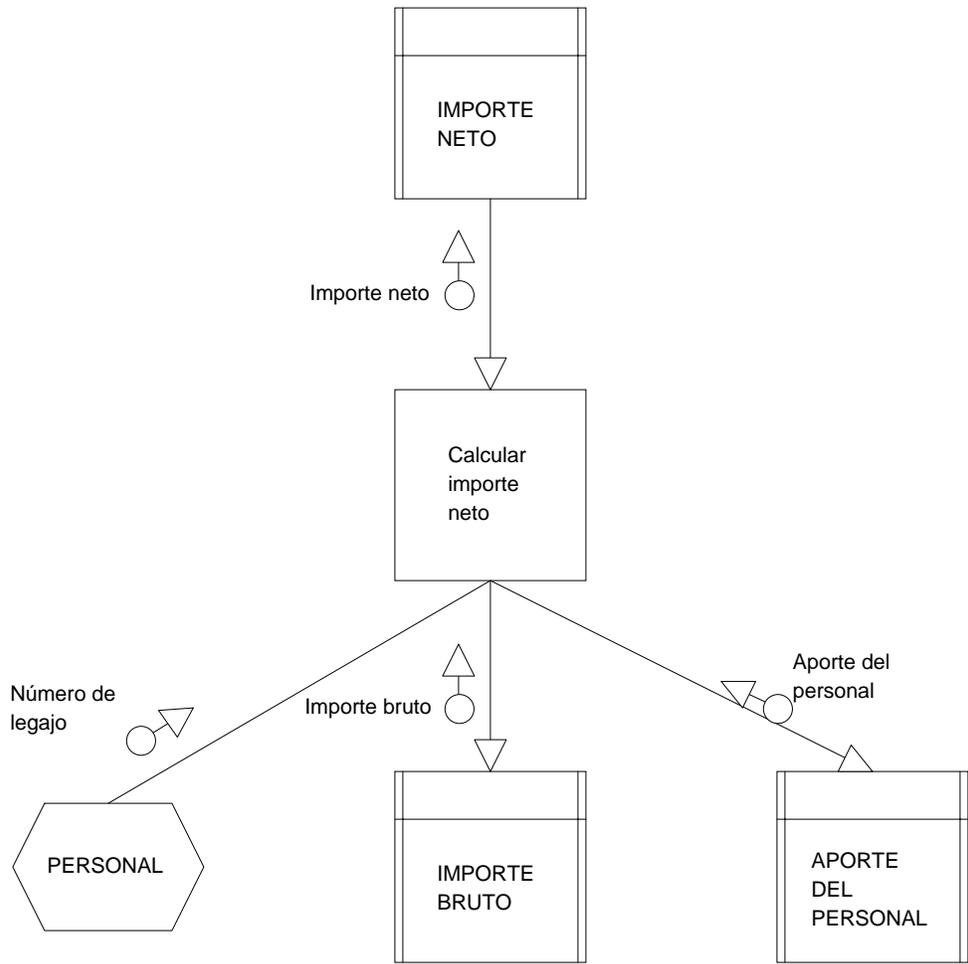
5 Diseño del modelo relacional de datos

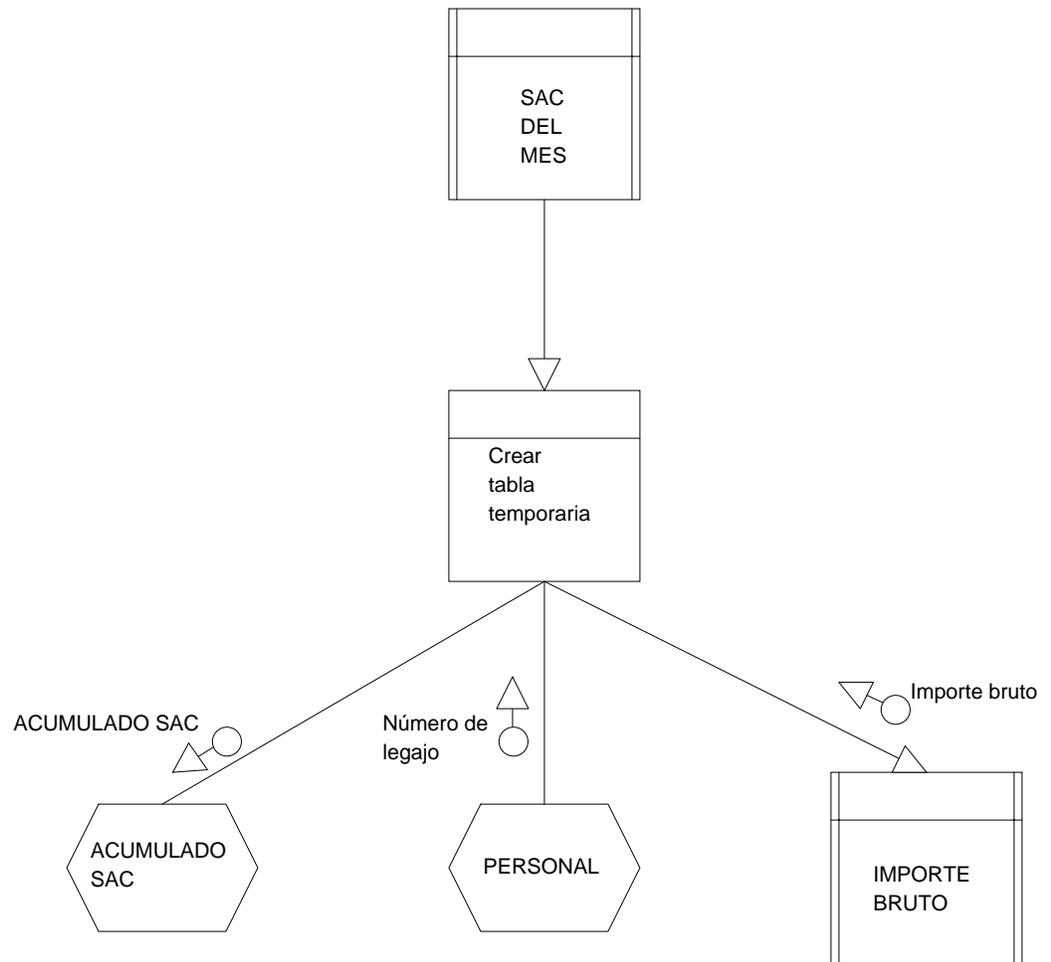


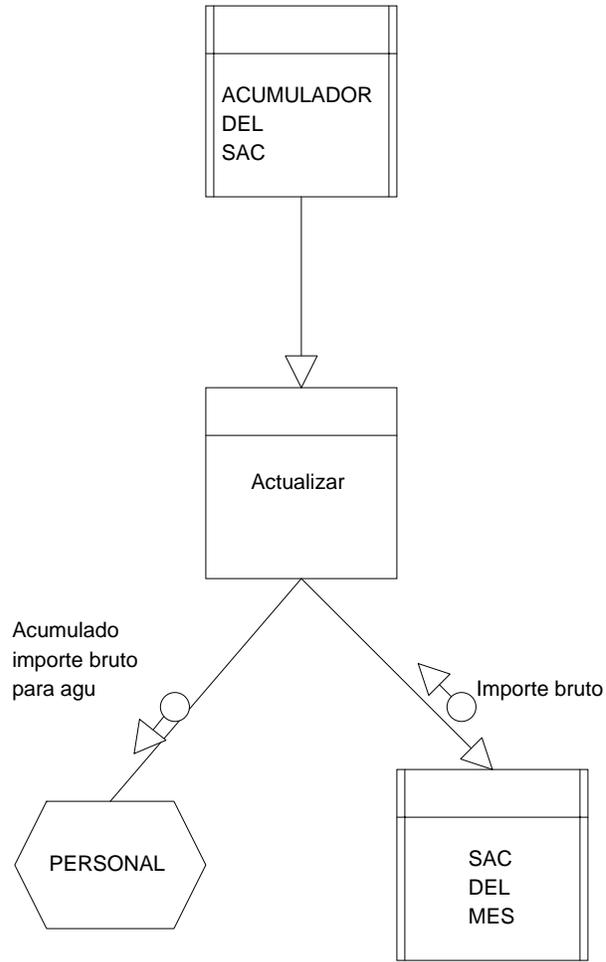
Módulos

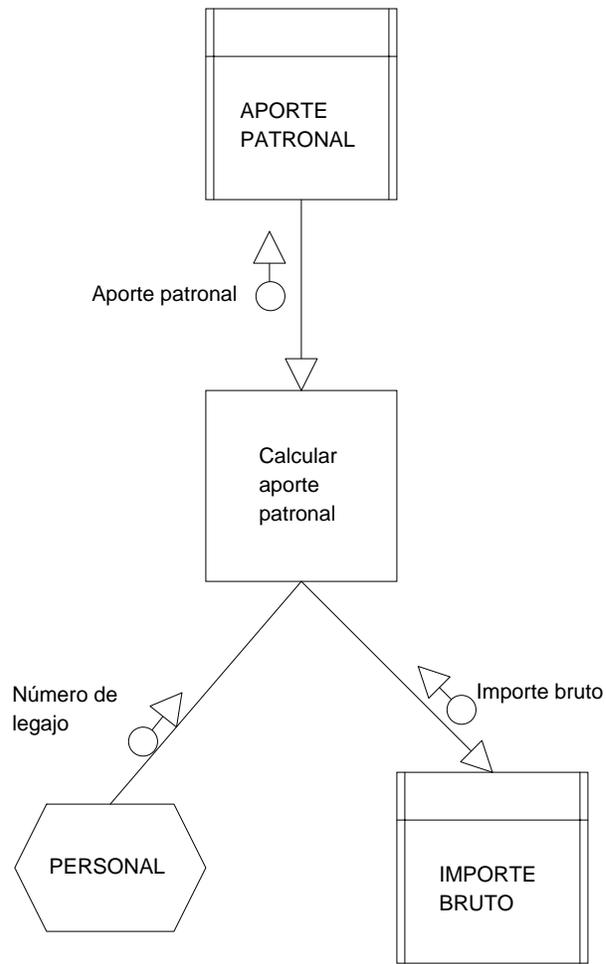




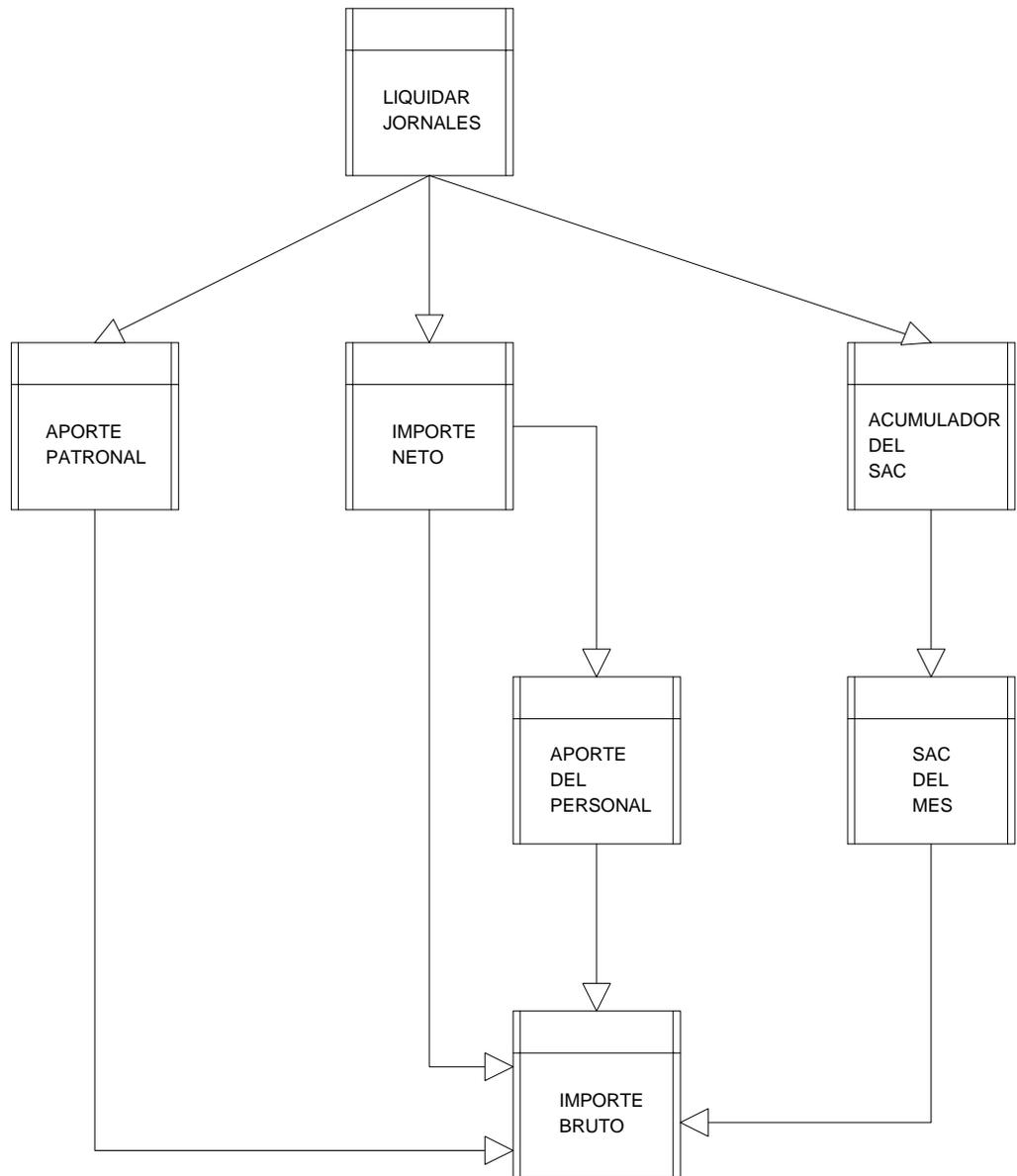








Árbol del sistema



EJERCICIO Nº:13

1 Procedimiento

Consulta al padrón de empleados

2 Operaciones

1. Contar registros de EMPLEADOS
2. Imprimir la cantidad total de empleados registrados
3. Consultar el último domicilio informado por cada empleado
4. Consultar el nivel de estudios alcanzado por cada empleado

Para ello cuenta con tres archivos clasificados por número de documento de identidad. Los tres archivos son completos, es decir que los datos de cada empleado se encuentran en los tres archivos.

3 Diseño de los archivos

MAESTRO:

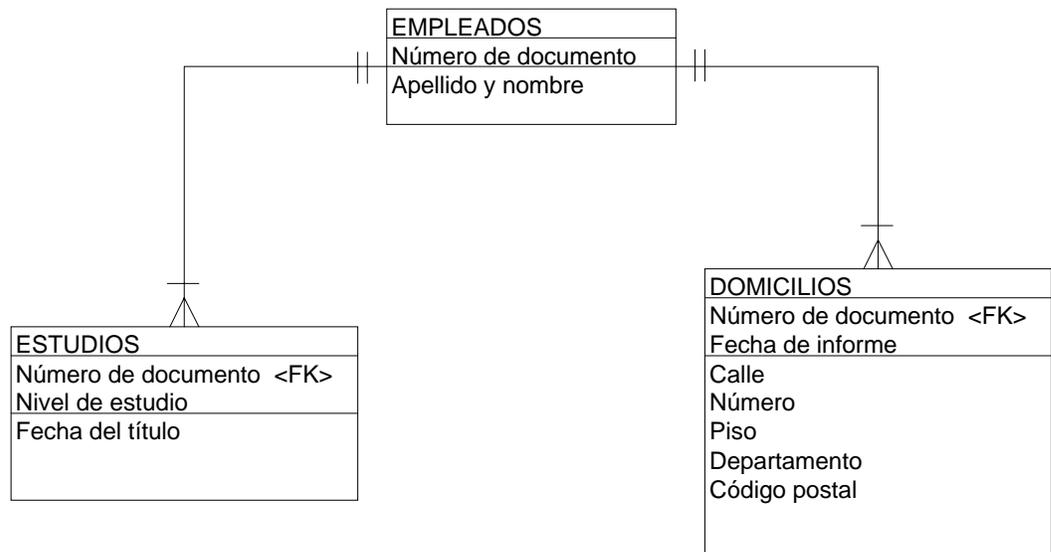
PERSONAL(Número de documento, Apellido y nombre)

TRANSACCIONES:

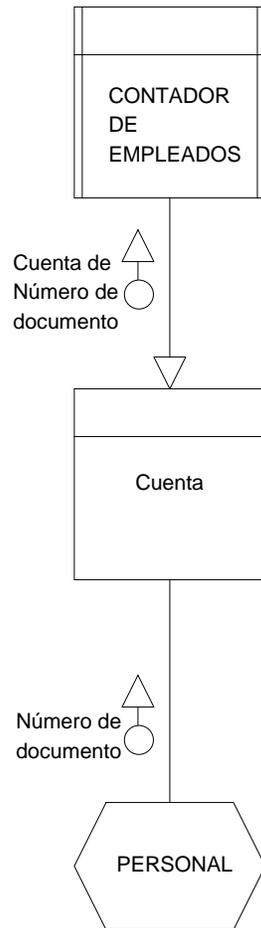
DOMICILIOS(Número de documento, Fecha de informe, Calle, Número, Piso, Departamento, Código postal)

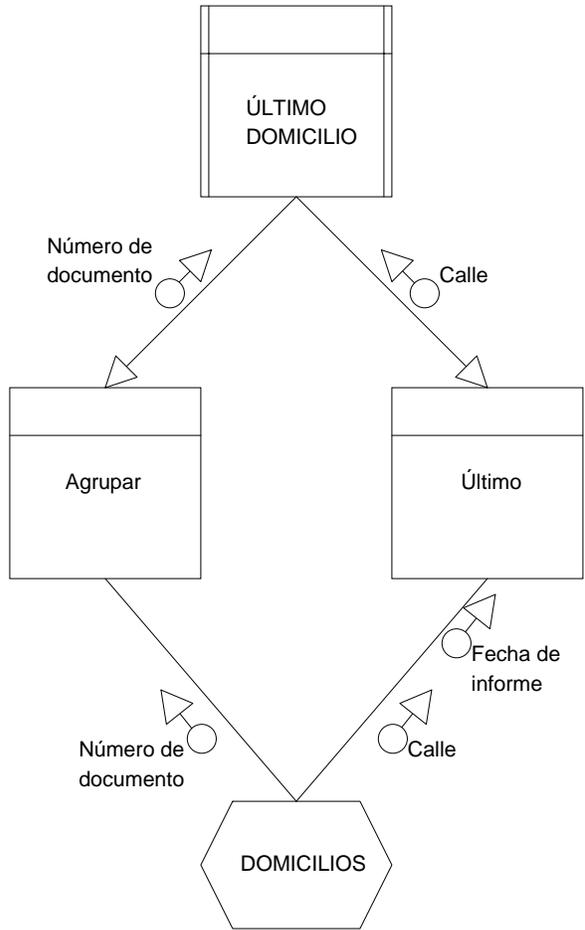
ESTUDIOS(Número de documento, Nivel de estudio, Fecha del título)

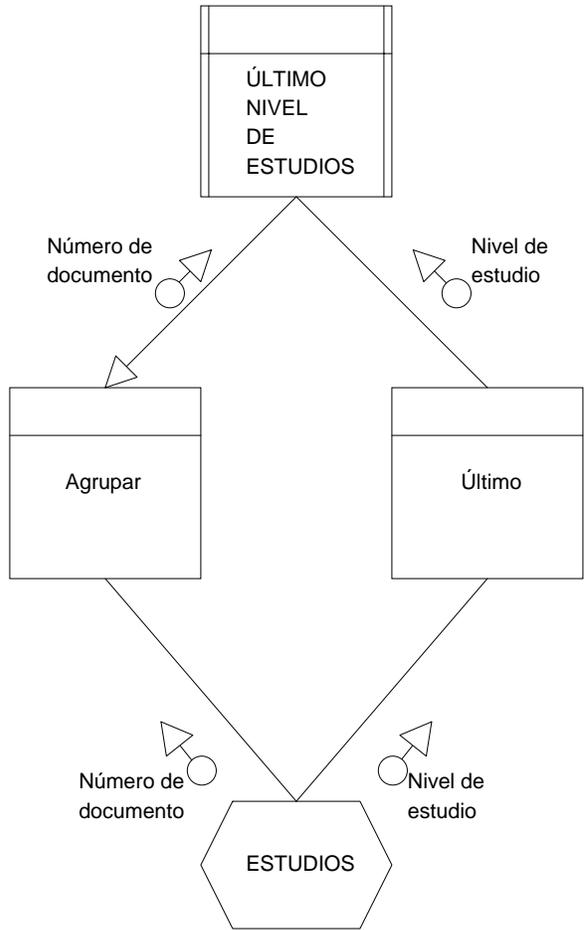
4 Diseño del modelo relacional de datos



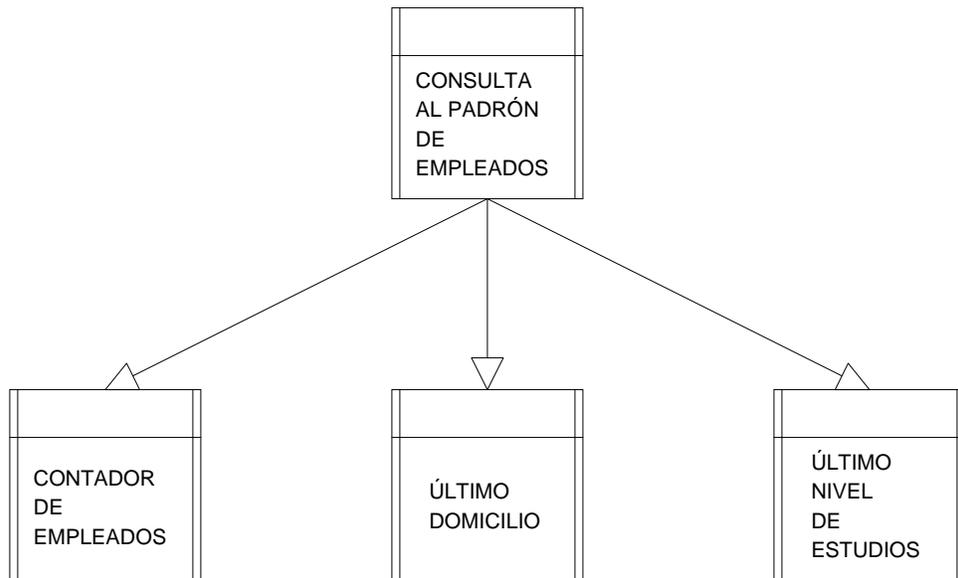
Módulos







Árbol del sistema



EJERCICIO Nº:14

1 Procedimiento

Consulta al padrón de empleados

2 Operaciones

1. Listar un padrón con los datos de los empleados.
2. Imprimir, cuántos empleados se tienen registrados.

Para ello cuenta con tres archivos: EMPLEADOS, DOMICILIOS Y ESTUDIOS. Clasificados por número de documento de identidad.

Los archivos de EMPLEADOS Y DOMICILIOS son completos, es decir que los datos de cada empleado se encuentran en los dos archivos.

El archivo ESTUDIOS puede contener datos faltantes en el campo Nivel de estudio.

Este último punto implica que el campo puede contener valores nulos; para identificar los valores nulos, se definirá la propiedad del campo como longitud cero, y en el formato del campo se le asignará la leyenda: "NO ENTREGÓ TÍTULO"

3 Diseño de los registros

ARCHIVO MAESTRO:

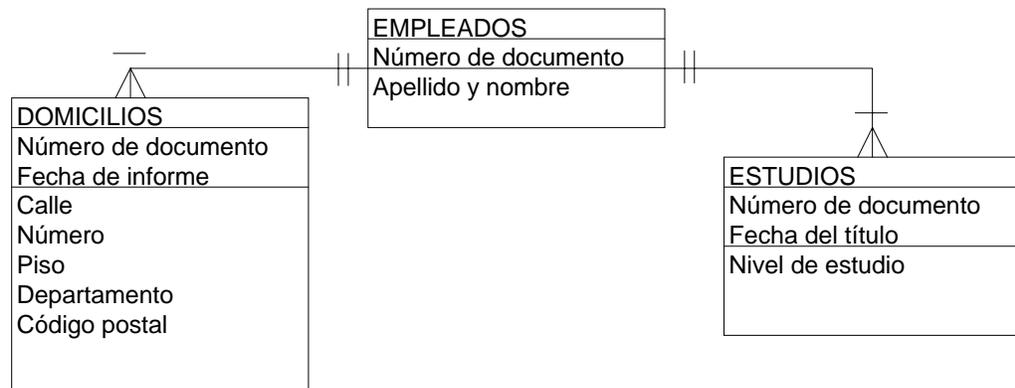
PERSONAL(Número de documento, Apellido y nombre)

ARCHIVOS DE TRANSACCIONES:

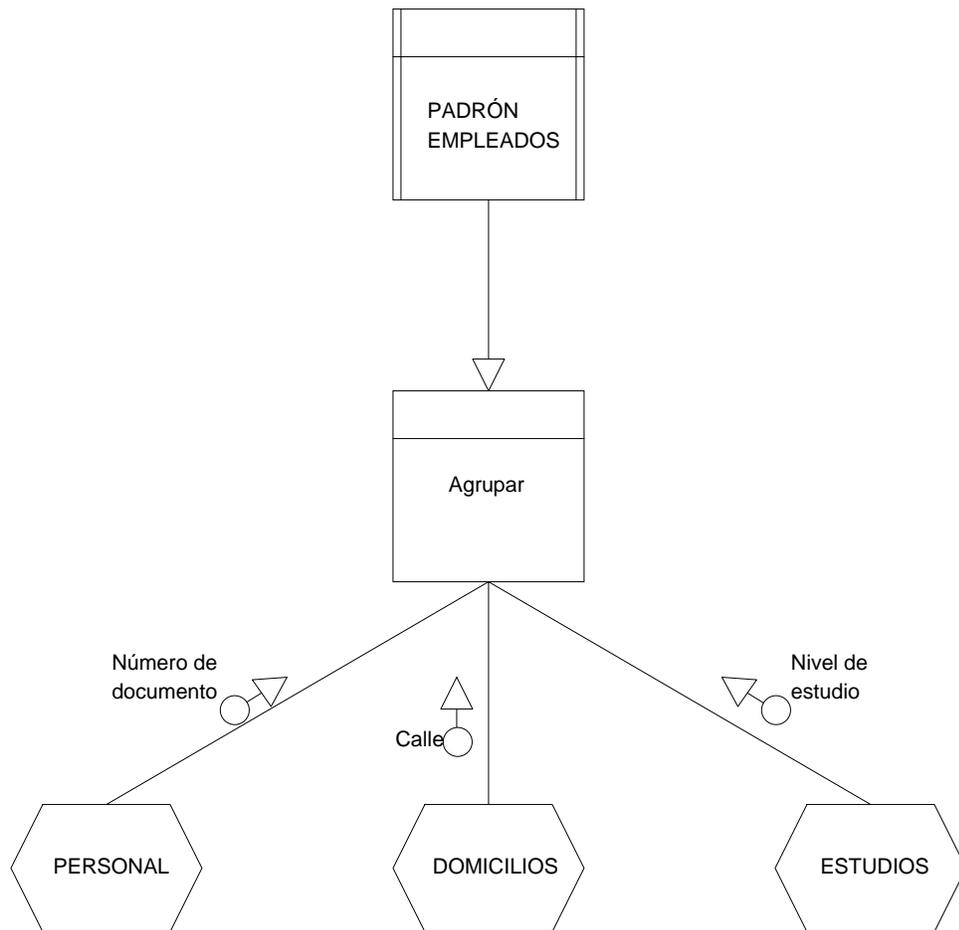
DOMICILIOS(Número de documento, Fecha de informe, Calle, Número, Piso, Departamento, Código postal)

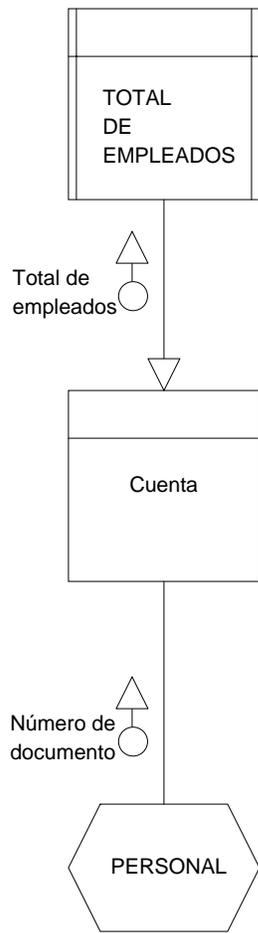
ESTUDIOS(Número de documento, Fecha del título, Nivel de estudio)

4 Diseño del modelo relacional de datos

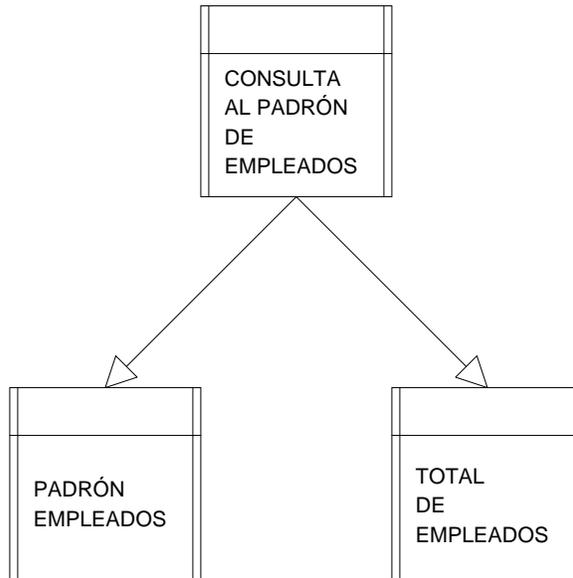


Módulos





Árbol del sistema



EJERCICIO Nº:15

1 Procedimiento

Actualización y control de existencias:

2. Operaciones

1. Calcular
 - 1.1. Stock a valor de mercado
 - 1.2. Costo de la mercadería
 - 1.3. Utilidad
 - 1.4. Importe total de ventas
2. Actualizar su archivo de stock
3. Emitir un listado de stock actualizado; respecto a los artículos que se realizaron ventas; se imprimirá:
 - 3.1. la existencia actualizada, valorizada al precio de venta,
 - 3.2. costo de la mercadería, y
 - 3.3. la diferencia entre ambas.
4. Imprimir el Importe Total correspondiente a las ventas realizadas.

3 Algoritmos

Stock a valor de mercado = Cantidad en existencia x Precio unitario

Costo de la mercadería = Cantidad en existencia x Costo unitario

Utilidad = Costo de la mercadería - Existencia actualizada

Importe total de ventas = suma(Cantidad vendida x Precio unitario)

Todos los archivos contienen un registro por cada artículo, y están clasificados por número de artículo en forma secuencial ascendente.

4 Diseño de los Archivos

MAESTRO:

STOCK (Número de artículo, Descripción, Precio unitario, Cantidad en existencia, Costo unitario)

TRANSACCIONES:

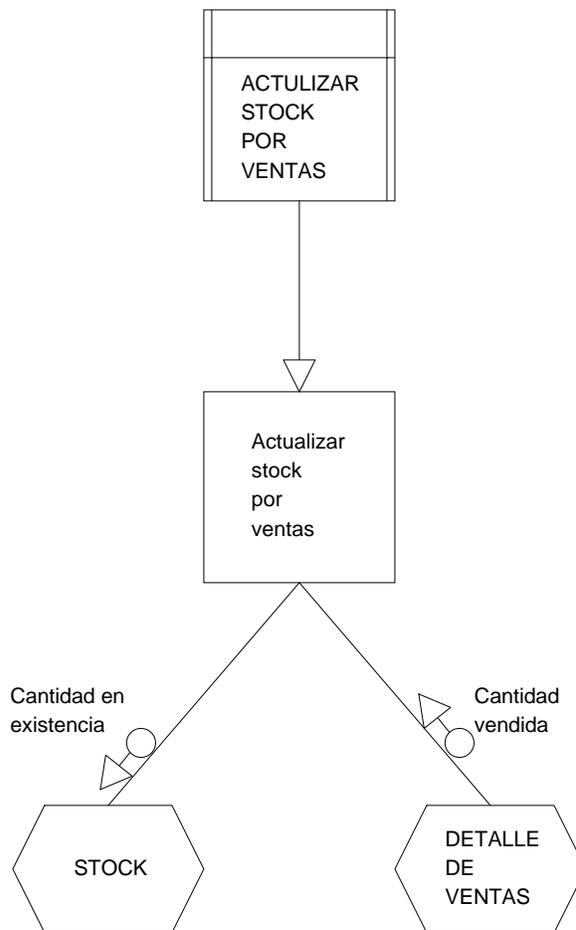
DETALLE DE VENTAS (Número de venta, Número de artículo, Cantidad vendida)

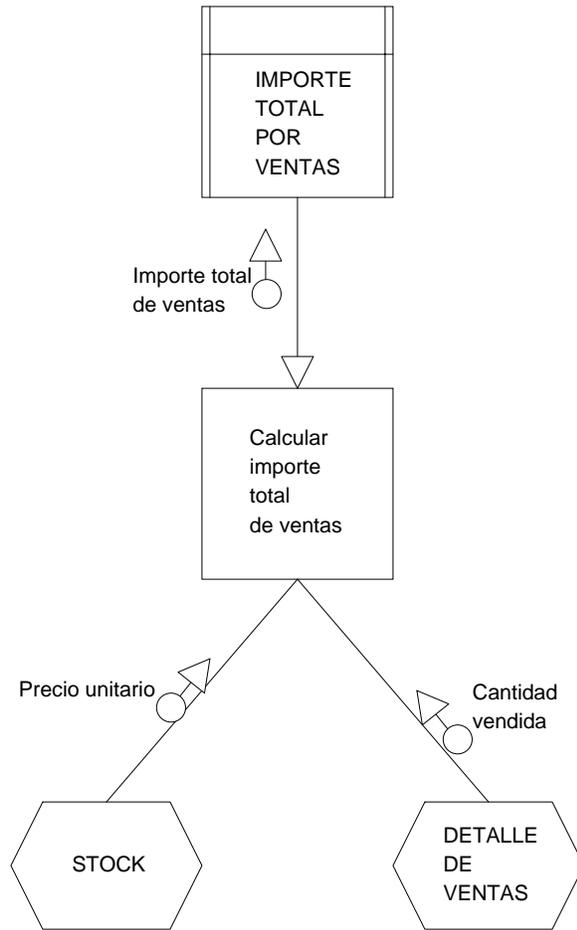
COSTO POR ARTÍCULO (Número de artículo, Costo unitario)

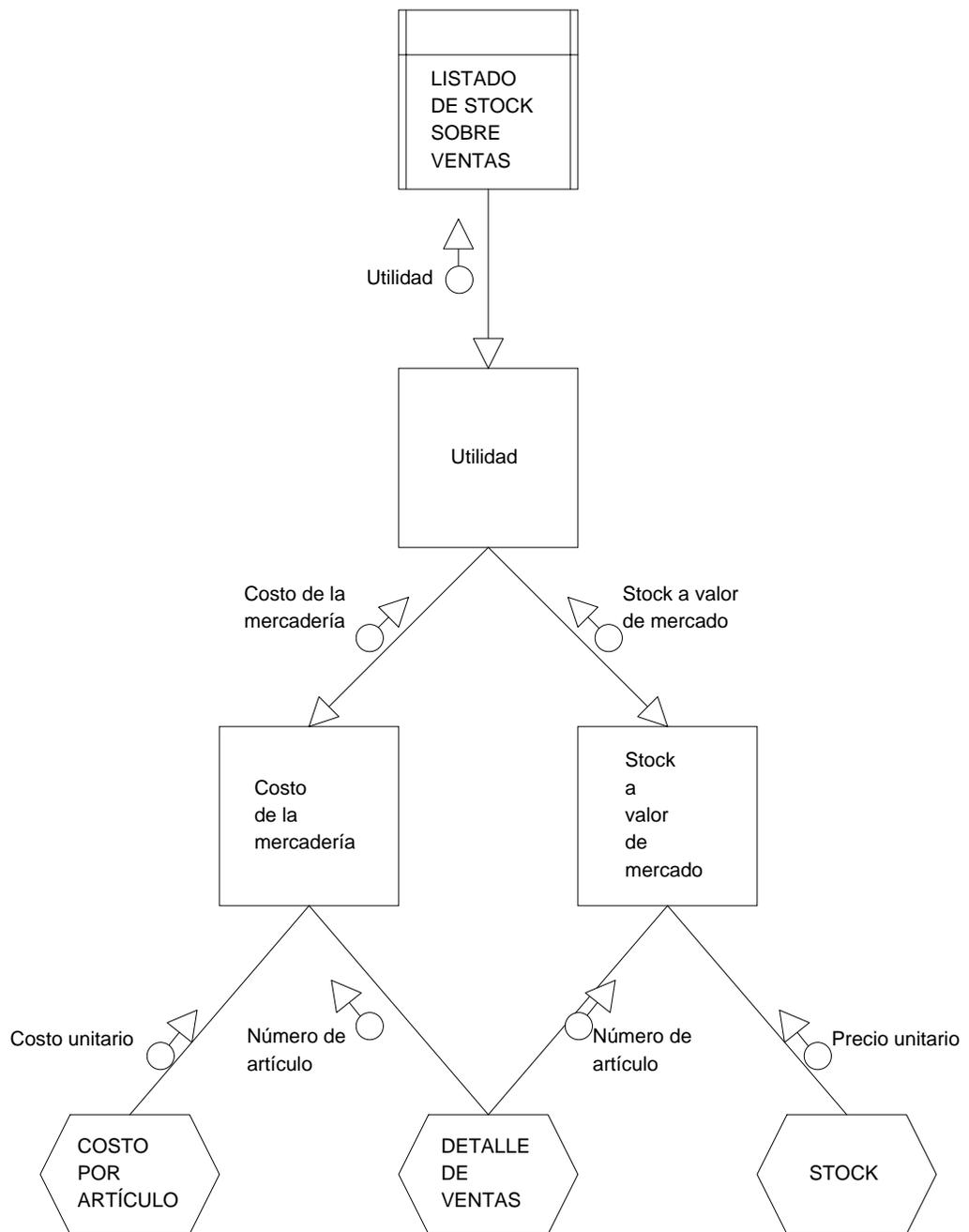
5 Diseño del modelo relacional de datos



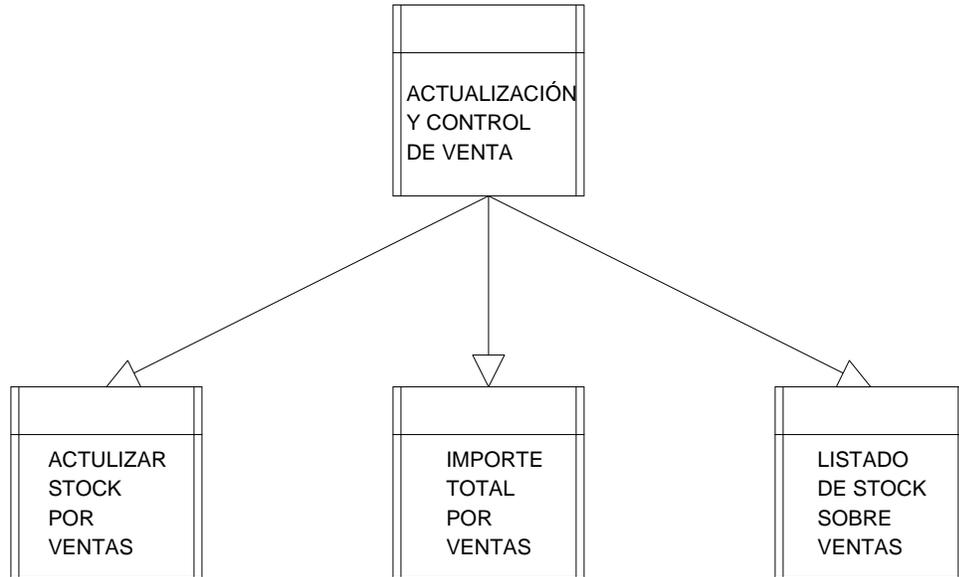
Módulos







Árbol del sistema



EJERCICIO Nº:16

1 Procedimiento

Control de la producción

2. Operaciones

Se desea:

1. Listar diariamente:
 - 1.1. los partes de producción de los distintos productos finales que elabora
2. Calcular:
 - 2.1. El Costo total por producto,
 - 2.2. El Costo de la orden y
 - 2.3. El Costo total de la producción del día.

Para ello cuenta con datos almacenados en los archivos cuyo detalle de contenido se muestra más abajo. Los registros de los archivos PARTES DE PRODUCCIÓN y el maestro de PRODUCTOS se encuentran ordenados por número de producto.

3 Algoritmos

Costo total = Suma(Costo unitario x Cantidad elaborada) de cada Número de producto

Costo de la orden = Suma(Costo unitario x Cantidad elaborada) de todos los Números de productos

Costo total de la producción del día = Suma (Costo de la orden) de todas las órdenes de un mismo día.

4 Diseño de los Archivos

MAESTROS:

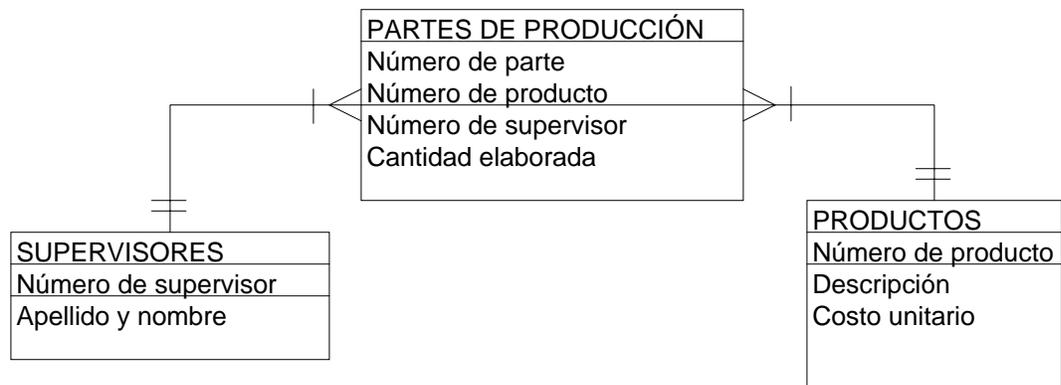
PRODUCTOS(Número de producto, Descripción, Costo unitario)

SUPERVISORES(Número de supervisor, Apellido y nombre)

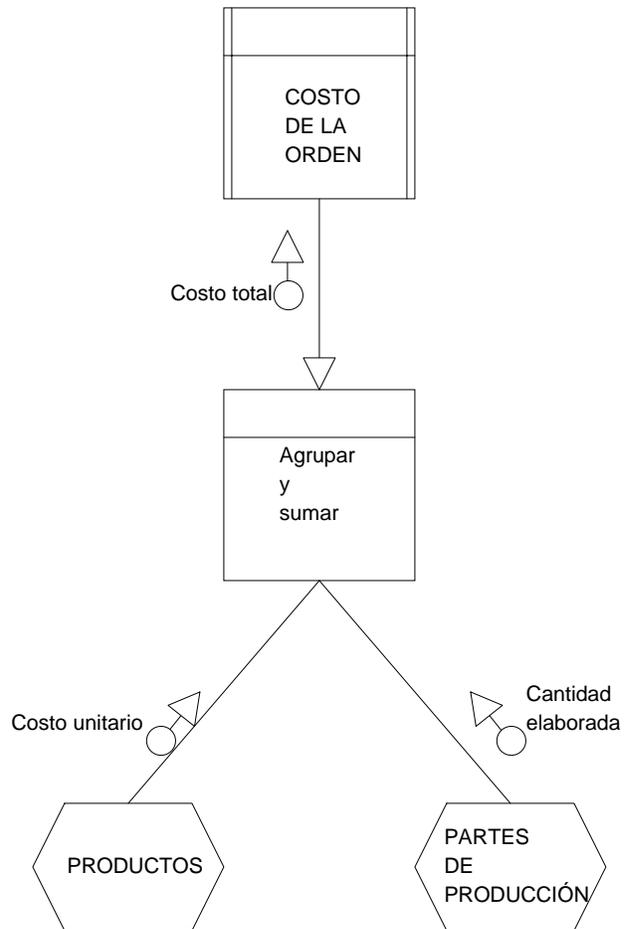
TRANSACCIONES:

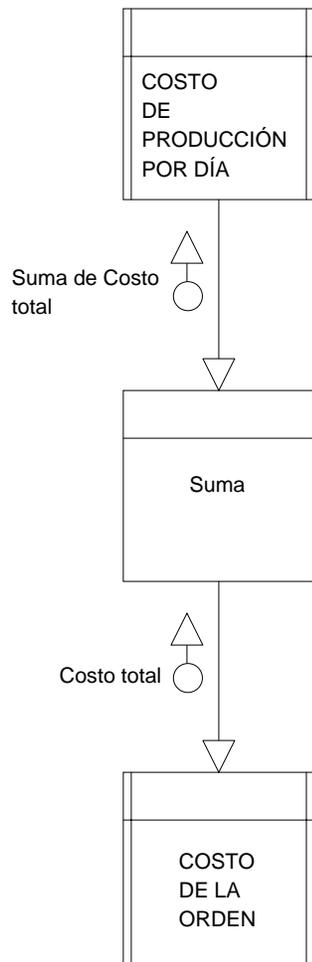
PARTES DE PRODUCCIÓN (Número de parte, Número de producto, Número de supervisor, Cantidad elaborada)

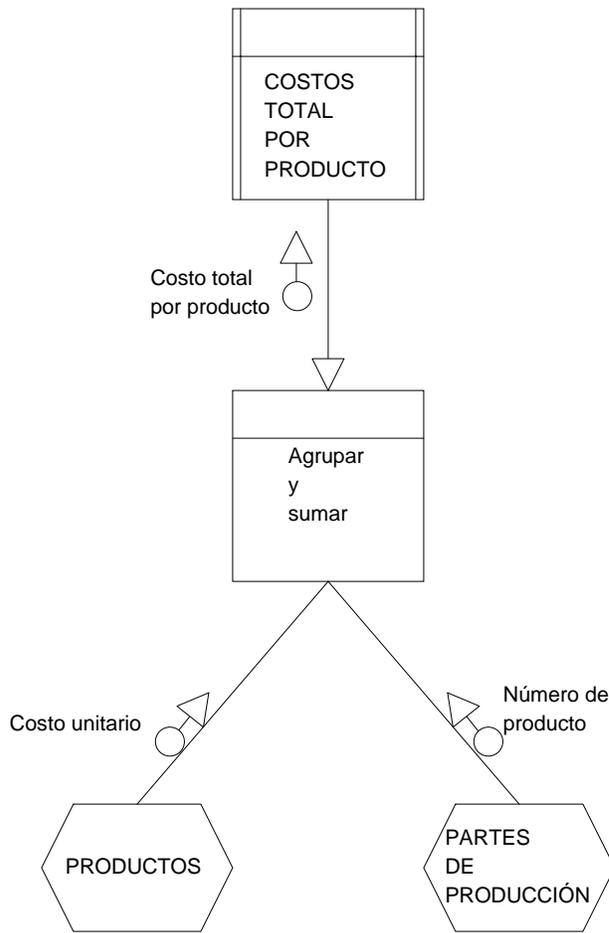
5 Diseño del modelo relacional de datos

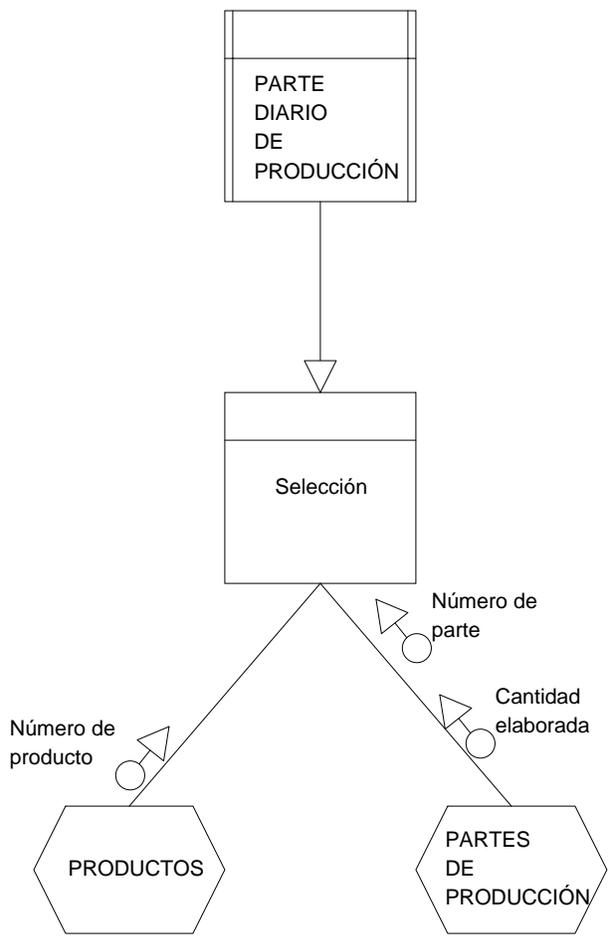


Módulos

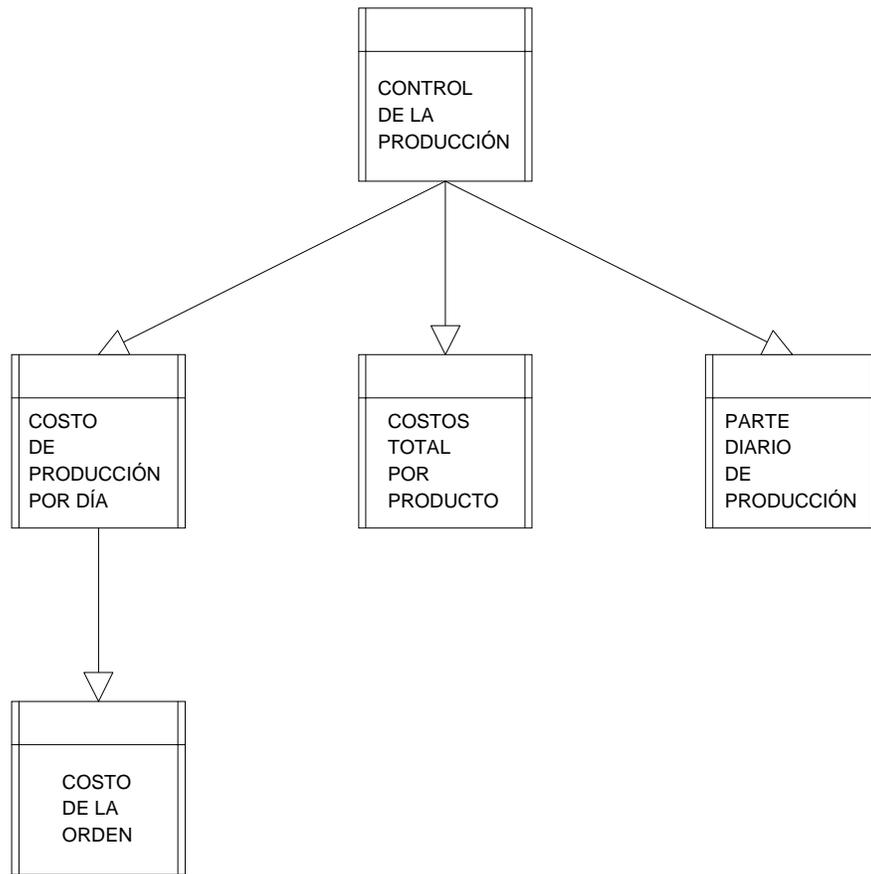








Árbol del sistema



EJERCICIO Nº:17

1 Procedimiento

Actualización de stock y control de la producción.

2. Operaciones

Se desea:

1. Calcular:
 - 1.1. Costo de cada orden,
 - 1.2. Costo total por producto utilizado y
 - 1.3. Costo de la producción del día.
2. Actualizar el archivo maestro de **PRODUCTOS** con la cantidad producida indicada en el archivo de **PARTE DE PRODUCCIÓN**.
3. Listar diariamente los **PARTES DE PRODUCCIÓN** de los distintos productos finales que elabora

3 Algoritmos

Para el Costo de cada orden: Primero agrupar por número de orden; segundo Suma(Costo unitario standard x Cantidad elaborada)

Para el Costo total por producto: Primero agrupar por número de producto; segundo Suma(Costo unitario standard x Cantidad elaborada)

Para el Costo de la producción del día: Primero agrupar por fecha; segundo Suma(Cantidad elaborada x Costo unitario standard)

Actualizar el archivo maestro de **PRODUCTOS** = Cantidad - Cantidad elaborada

4 Diseño de los Archivos

TRANSACCIONES:

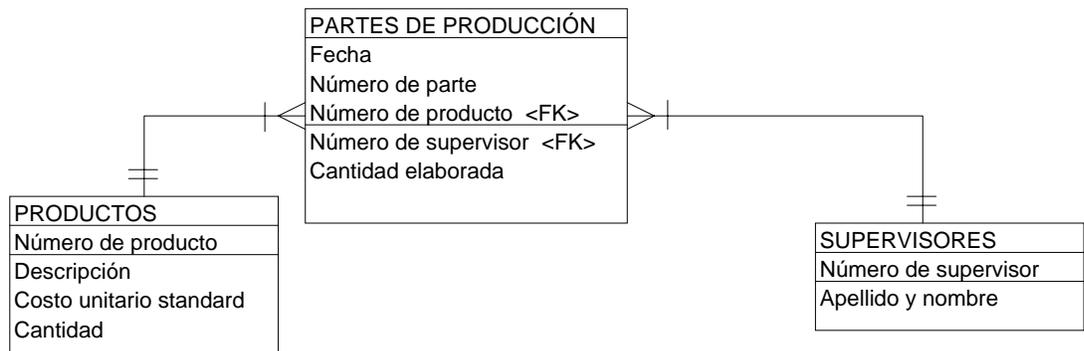
PARTES DE PRODUCCIÓN (Fecha, Número de parte, Número de producto, Número de supervisor, Cantidad elaborada)

MAESTROS:

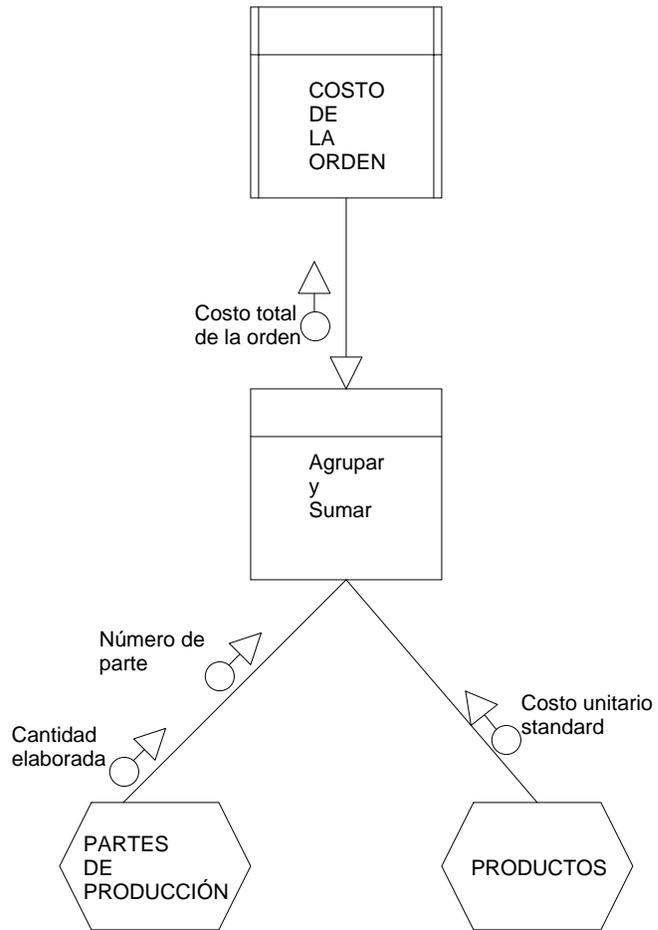
PRODUCTOS (Número de producto, Descripción, Costo unitario standard, Cantidad)

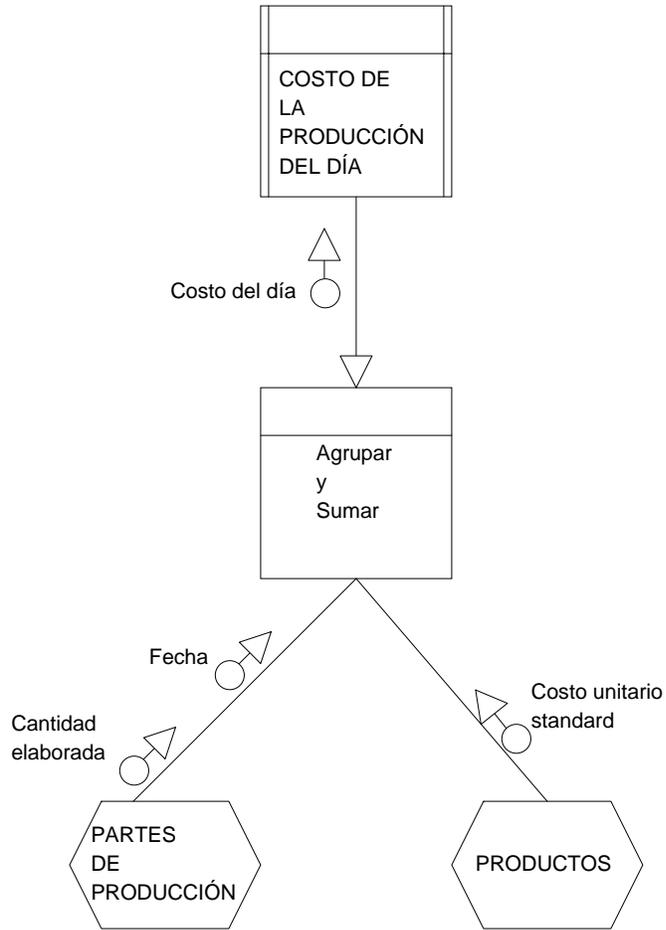
SUPERVISORES (Número de supervisor, Apellido y nombre)

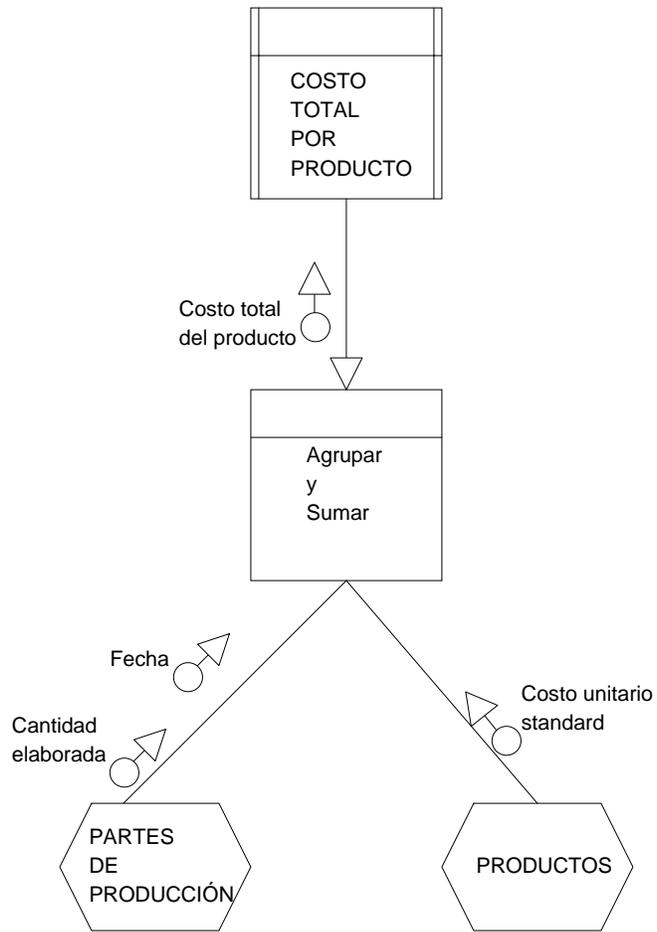
5 Diseño del modelo relacional de datos

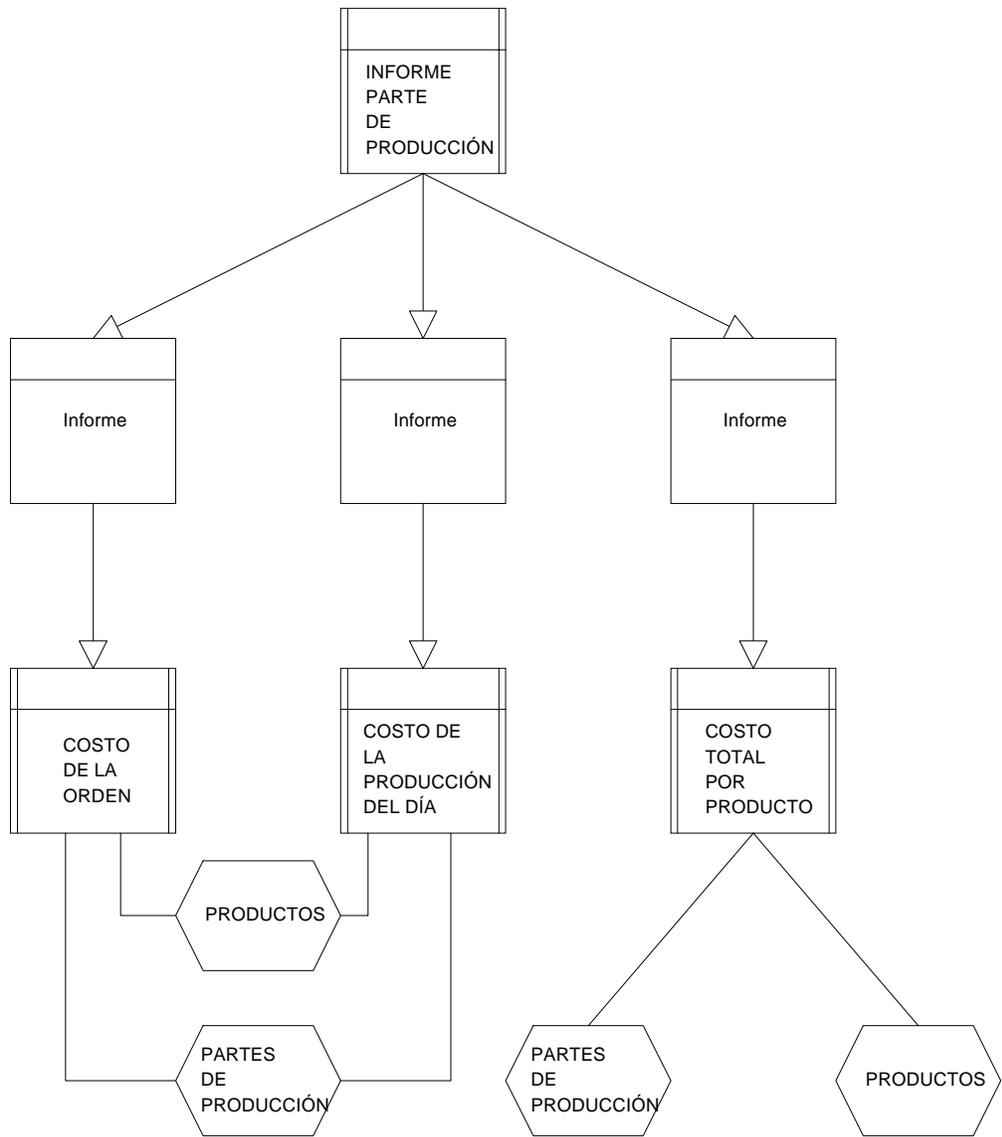


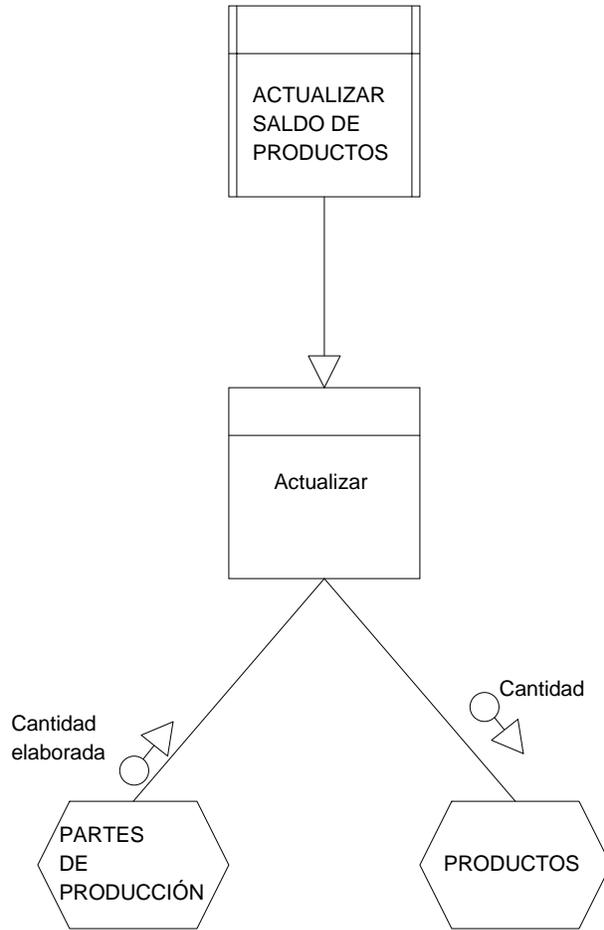
Módulos



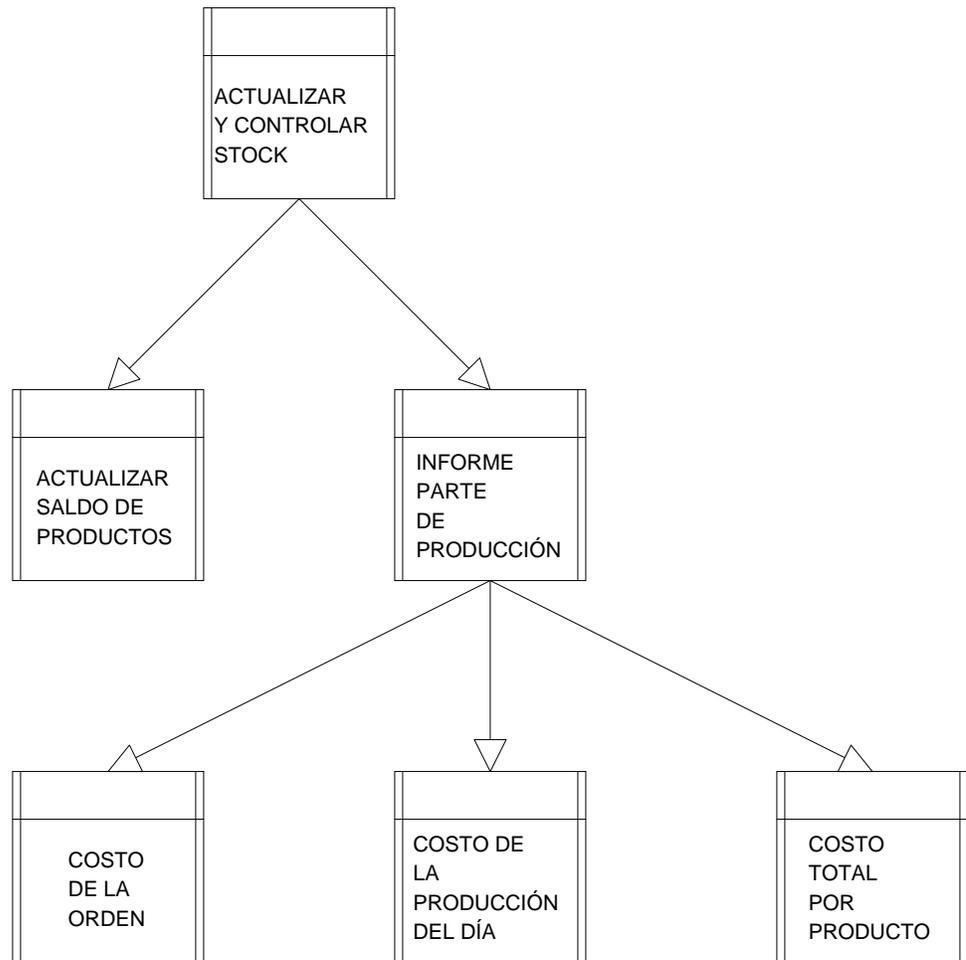








Árbol del sistema



EJERCICIO Nº:18

1 Procedimiento

Facturación.

2. Operaciones

1. Emitir su facturación por computadora.
2. El departamento de contabilidad de la empresa necesita conocer, además de la emisión de la facturación, el importe de la facturación total de cada proceso.
3. Actualizar archivo **SOTCK**.

3 Algoritmos

Para emitir la factura: $\text{Precio total} = \text{Cantidad} \times \text{Precio unitario}$

Para la facturación total de cada proceso: $\text{Suma precio total} = \text{suma} (\text{Cantidad} \times \text{Precio unitario})$

Para actualizar SOTCK: $\text{Saldo} = \text{Saldo} - \text{Cantidad}$

4 Diseño de los Archivos

MAESTROS:

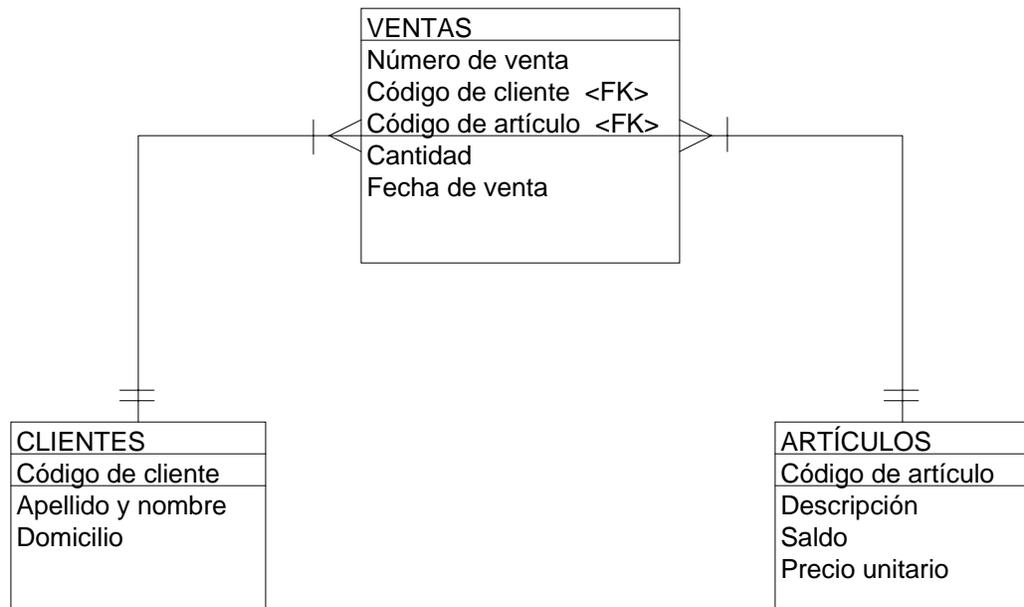
CLIENTES(Código de cliente, Apellido y nombre, Domicilio)

ARTÍCULOS(Código de artículo, Descripción, Saldo, Precio unitario)

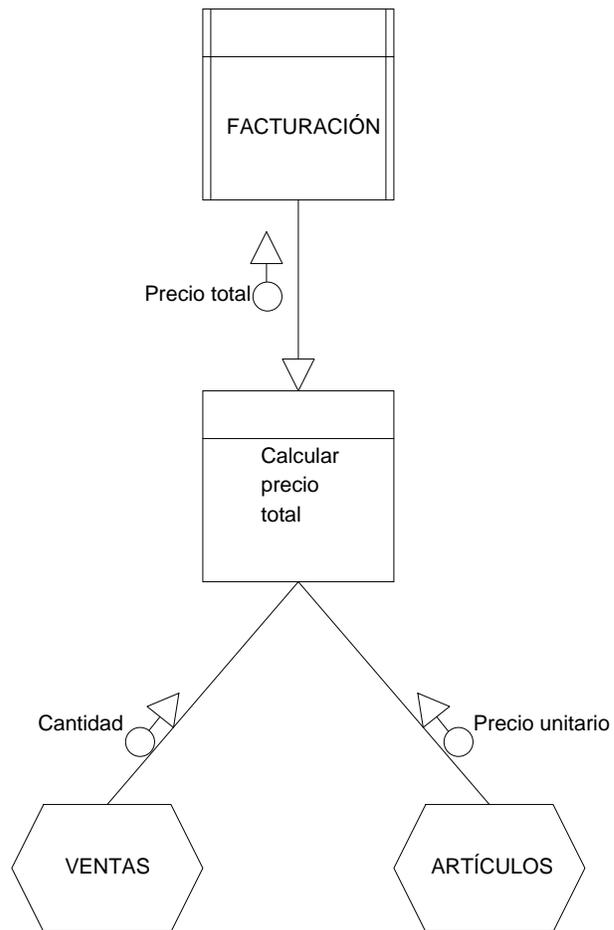
TRANSACCIONES:

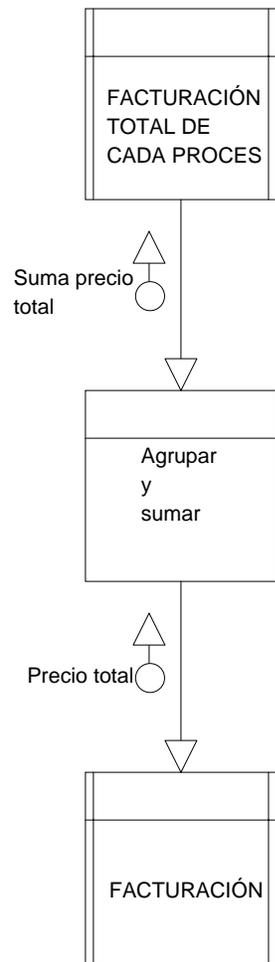
VENTAS(Número de venta, Código de cliente, Código de artículo, Cantidad, Fecha de venta)

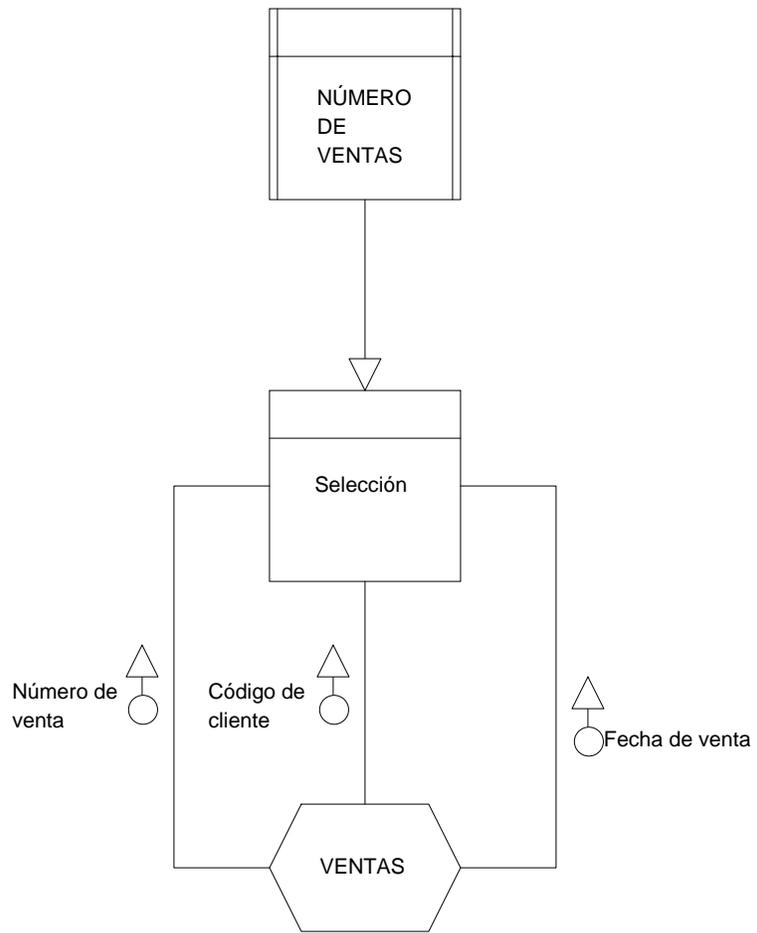
5 Diseño del modelo relacional de datos

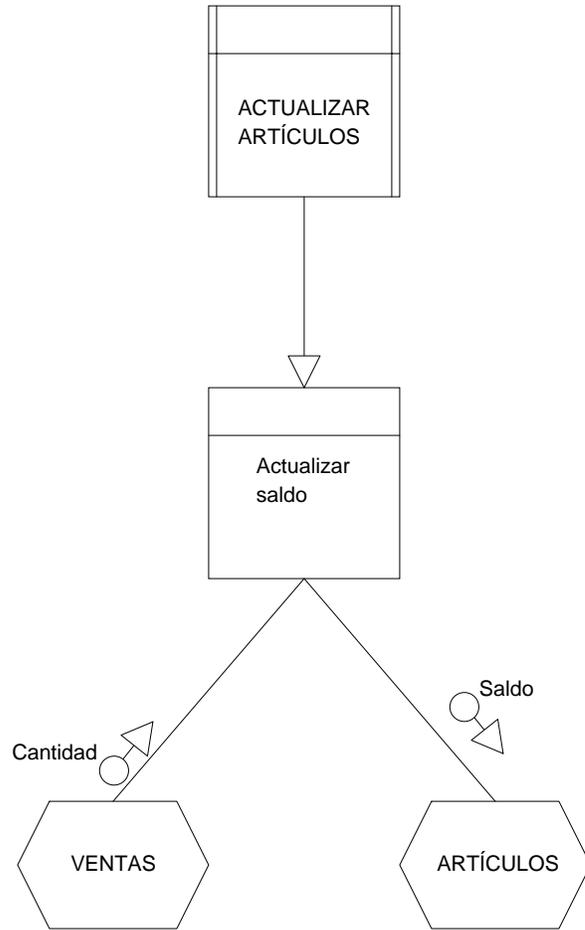


Módulos

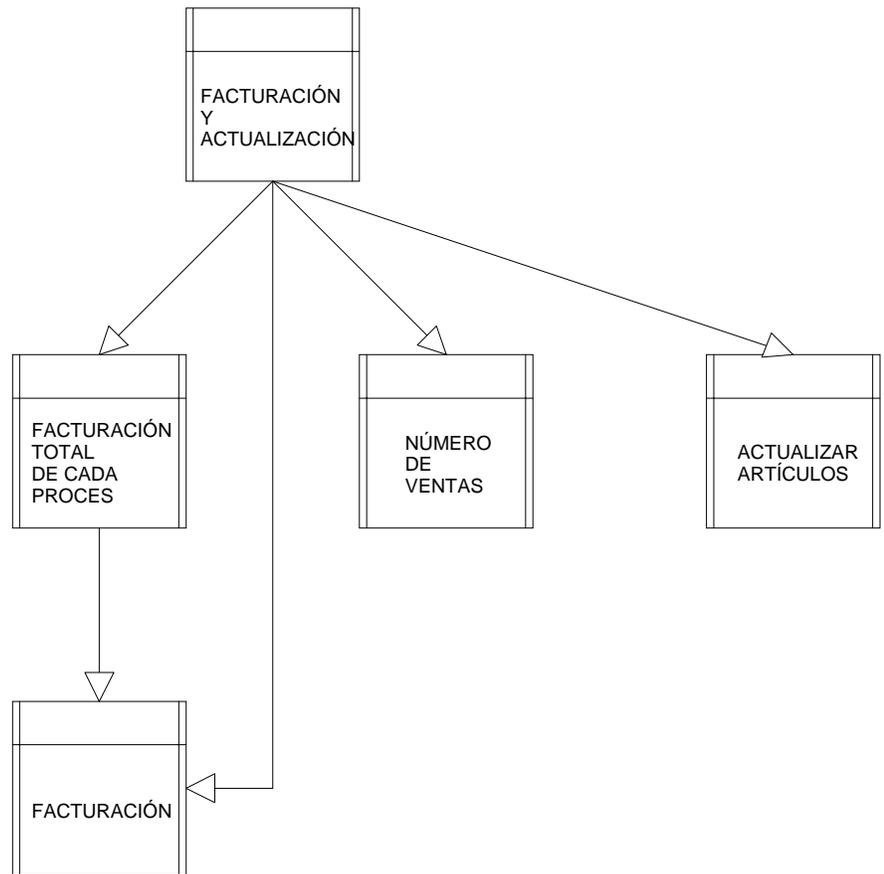








Árbol del sistema



BIBLIOGRAFÍA

- Ackoff Rusell L.; *Planejamento empresarial*, Livros técnicos e científicos editora.
- Ackoff Rusell, Vergara Finnel E., Gharajedaghi J. , *Guía para controlar el futuro de la empresa*, Limusa
- Allen Pau C., *Técnicas estructuradas efectivas*, El Ateneo Buenos Aires 1996.
- Batini Carlos, Stefano Ceri y Shamkent B. Nabathe, *Diseño conceptual de bases de datos*, Addison-Weslwy/Diaz de Santos, 1994.
- Deming , *Total Quality*
- Diaz, Alberto, *Descripción de operaciones típicas de una Empresa*, Editorial Club de estudio, Buenos Aires Fascículo 2.
- Edit Grassini Lino de Campos, Judith Pavón, Paulo S. Cugnasca; *Análise Comparativa de Ferramentas Gráficas de Manipulação de Banco de Dados*; 2da. Jornada USP-SUCESU-SP de informática e telecomunicações. Mayo 1994
- Gorry, G.A. y Scott Morton, M:S., *A framework for management information systems*, in *Management Review*, otoño 1971.
- Henry C. Lucas Jr., *Sloan Management Review*, invierno 1978, *The evolution of an information system: from key-man to every person*.
- Hertz, David B.;"Risk Anaysis in Capital Investment", *harvard Business Review* 42 (1): 95:106.1964
- ISO 9004, 5.3.3 *Quality plans* 1987
- Gary L. Lilien y Philip Kotler, *Marketing decision Making A Model-Building Approach*. Harper & row Publishers, inc., segunda edición1992
- Lardent A.,Gomez Echarren Manuel A., Loro Alberto; *Técnicas de Organización Sistemas y Métodos*, Club de estudio 1984.
- Lardent Alberto, *Diagramación lógica*, Club de estudio, 1987.
- Lucas H., *The evolution of an infromation system: from key-man to every person.*, Sloan Managemente Review.
- Martin James, *Sistemas de Información*, El Ateneo.
- Martin Merle P.; *Business Information System*, secon editition; Pretince Hall 1.995.
- Martin, J. *Conceptual Strcutres: Information Processing in Mind and Machine*. Addison Wesley, California 1984.
- Novelli Filho, Aristides; Grigas Varella Ferreira, María A.; *Case e Metodología para Projetos de pequeno porte* ; 2ª jornada USP-SUCESU-SP de informática e telecomunicações 1994.
- Parro Nereo Roberto, *Administración de Empresas*, T III, pg. 289.
- Pereira,F.C.N.&Warren, D.H.D *Definite Clause Grammanrs for Language Analsis-A Surveyof the Formalism and a Comparison with Augmented Transition Networks"*. *Artificial Intelligence* 13,231-278.
- Pressman Roger S. *Ingeniería del Software* McGraw-Hill.
- Randolph Alan W., Posner Barry Z. Grijalbo 1991
- Richard Barker, *el modelo entidad - relación CASE * METHOD*, Addison - Wsley / Diaz de santos 1994.

- Senn, James A., *Análisis y diseño de sistemas de información*, McGraw-hill, 1992, segunda edición.
- Fred L luconi , Thomas W. Malone, Michaels s. Scott Morton, *Sistemas expertos: el próximo desafío para los gerentes*, Administración de empresas.
- *Tecnología CASE*, Cuaderno de SADIO, serie Informática N° 9 noviembre de 1993., Communication of de ACM, case IN THE 90'.
- Torné Juan P., *Proyectos informáticos*, Parainfo 1994.
- Vidal, Antonio Geraldo, , *Introducción al proyecto y desenvolvimiento de sistemas de Información FEA, USP*, 1998.
- Yourdon E., *Análise Estruturada Moderna*, Editora Campus 1990.
- Zwicker Ronaldo, *Aprendizagem e Uso de Sistemas*. Anais do XXIV Congresso Nacional de Informática, Sao Paulo, 1991.

¹ Parro Nereo Roberto, *Administración de Empresas*, T III, pg. 289

² Randolph Alan W., Posner Barry Z. Grijalbo 1991

³ Deming , *Total Quality*

⁴ ISO 9004, 5.3.3 Quality plans 1987

⁵ Torné Juan P., *Proyectos informáticos*, Parainfo 1994

⁶ *Sistemas expertos: el próximo desafío para los gerentes*, Fred L luconi , Thomas W. Malone, Michaels s. Scott Morton, *Administración de empresas Tomo XVII* pag. 965

⁷ Henry C. Lucas Jr., *Sloan Management Review*, invierno 1978, pg. 39; *The evolution of an information system: from key-man to every person*.

⁸ Gorry, G.A. y Scott Morton, M:S:, *A framework for management information systems*, in *Management Review*, otoño 1971. Pag. 55/70.

⁹ Novelli Filho, Aristides; Grigas Varella Ferreira, María A.; *Case e Metodología para Projetos de pequeno porte ; 2ª jornada USP-SUCESU-SP de informática e telecomunicações* 1994 Pg56.

¹⁰ Martin Merle P.; *Business Information System*, secon edittion; Pretince Hall 1995

¹¹ Ackoff Rusell L.; *Planejamento empresarial*, Livros técnicos e científicos editora. Pg. 2

¹² Pressman Roger S. *Ingeniería del Software* McGrawHill pg.114

¹³ Lardent A., Gomez Echarren Manuel A., Loro Alberto; *Técnicas de Organización Sistemas y Métodos*, Club de estudio 1984.

¹⁴ Hertz, David B.; "Risk Anaysis in Capital Investment", harvard Business Review 42 (1): 95:106.1964

¹⁵ *Diario la Nación*, 9 de octubre de 2000, suplemento informática pg. 12.

¹⁶ Zwicker Ronaldo, *Aprendizagem e Uso de Sistemas*. Anais do XXIV Congresso Nacional de Informática, Sao Paulo, 632-638, 1991.

¹⁷ Martin, J. *Conceptual Strcutres: Information Processing in Mind and Machine*. Addison Wesley, California 1984.

¹⁸ Ackoff Rusell, Vergara Finnel E., Gharajedaghi J. , *Guía para controlar el futuro de la empresa*, Limusa

¹⁹ Gary L. Lilien y Philip Kotler, *Marketing decision Making A Model-Building Approach*. Harper & row Publishers, inc., segunda edición 1992, pg28:30)

-
- ²⁰ Hillier y Lieberman y Bieman, Bonini y Hausman
- ²¹ Batini Carlos, Stefano Ceri y Shamkent B. Nabathe, Diseño conceptual de bases de datos, Addison-Weslwy/Diaz de Santos, 1994.(pg.100:104)
- ²² Daniel Cassanay, la cocina de la escritura,Página148, Editorial Anagrama, S.A., 1995, Barcelona ISBN 84-339-1392-1
- ²³ Allen Paul C., Técnicas Estructuradas Efectivas, El Ateneo, Buenos Aires, 1996.
- ²⁴ Pereira,F.C.N.&Warren, D.H.D. (1980).Definite Clause Grammanrs for Language Analsis-A Surveyof the Formalism and a Comparison with Augmented Transition Networks". Artificial Intelligence 13,231-278.
- ²⁵ Daniel Cassanay, la cocina de la escritura, Anfibología, Página130, Editorial Anagrama, S.A., 1995, Barcelona ISBN 84-339-1392-1
- ²⁶ Yourdon E., Análise Estruturada Moderna, Editora Campus 1990.
- ²⁷ Vidal, Antonio Geraldo, Introducción al proyecto y desenvolvimiento de sistemas de Información FEA, USP, 1998.
- ²⁸ Vidal, Antonio Geraldo, , Introducción al proyecto y desenvolvimiento de sistemas de Información FEA, USP, 1998.
- ²⁹ Vidal, Antonio Geraldo, , Introducción al proyecto y desenvolvimiento de sistemas de Información FEA, USP, 1998.
- ³⁰ Edit Grassini Lino de Campos, Judith Pavón, Paulo S. Cugnasca; Análise Comparativa de Ferramentas Gráficas de Manipulação de Banco de Dados; 2da. Jornada USP-SUCESU-SP de informática e telecomunicações. Mayo 1994
- ³¹ Bueno Zamboni, Euriclea M. y Grassiani Lino de Campos Edit. Uma Ferrmaenta para Normalizações (FNR).
- ³² Lardent Alberto, *Diagramación lógica*, Club de estudio, Buenos Aires 1987
- ³³ *Tecnología CASE*, Cuaderno de SADIO, serie Informática N° 9 noviembre de 1993., Communication of de ACM, case IN THE 90'(vol 35 N° 4 april 1992)
- ³⁴ Martin James, *Sistemas de Información*, El Ateneo, Pg.178
- ³⁵ Lucas H., *The evolution of an infromation system: from key-man to every person.*, Sloan Managemente Review.