

34
24



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

**METODOLOGIA JACKSON PARA EL DESARROLLO
DE SISTEMAS UN CASO PRACTICO**

T E S I S
QUE PARA OBTENER EL TITULO DE
A C T U A R I O
P R E S E N T A :
HUGO HERNANDEZ RAMOS

MEXICO, D. F.

**TESIS CON
FALLA DE ORIGEN**

1991



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

INTRODUCCIÓN	1
CAPÍTULO 1.- ADMINISTRACIÓN DEL PROYECTO	3
1.1.- INTRODUCCIÓN	3
1.2.- ADMINISTRACIÓN DE PROYECTOS	3
1.3.- DESCRIPCIÓN DE LA METODOLOGÍA : PLANEACIÓN DEL PROYECTO	4
CAPÍTULO 2.- METODOLOGÍA JACKSON PARA EL DESARROLLO DE SISTEMAS	18
2.1.- UBICACIÓN DE LA METODOLOGÍA	18
2.2.- CARACTERÍSTICAS GENERALES	19
2.3.- INTRODUCCIÓN A LA METODOLOGÍA	20
2.4.- PROCESO DE DESARROLLO	23
2.4.1.- PASOS DE DESARROLLO	23
2.4.2.- DESARROLLO DE LA ESPECIFICACIÓN	24
2.4.3.- ETAPA DE IMPLEMENTACIÓN	36
CAPÍTULO 3.- CASO PRÁCTICO DE ESTUDIO. FASE DE PLANEACIÓN DEL PROYECTO DE ENTRADAS AL ALMACÉN.	42
CAPÍTULO 4.- METODOLOGÍA JACKSON APLICADA AL SISTEMA DE ENTRADAS AL ALMACÉN.	55
CONCLUSIONES	74
BIBLIOGRAFÍA	76

INTRODUCCIÓN.

El desarrollo de Proyectos de Software cada vez toma mayor importancia, en la actualidad casi todas las empresas utilizan Sistemas de Cómputo con el fin de agilizar procesos y tener un mejor control de la información que manejan; sin embargo, en la tarea de desarrollo de sistemas no se ha logrado tener el éxito esperado.

Al tratar de determinar las principales causas por las cuales los Proyectos de Software no funcionan como el usuario espera que lo hagan, se obtendría una lista enorme, muchas de ellas se eliminarían, o al menos se reducirían, si se siguiera una metodología que guiara el trabajo de desarrollo durante todo el ciclo de vida de un Proyecto de Software.

El objetivo de este trabajo es el de describir una Metodología para el desarrollo de Proyectos de Software, desde su planeación hasta su implementación; éste se divide en dos partes :

- Metodología para la Administración del Proyecto : Planeación del Proyecto, y
- Metodología Jackson para el Desarrollo de Sistemas (JSD, Jackson System Development).

Generalmente el ciclo de vida de desarrollo de un sistema consta de las siguientes etapas :

Análisis,
Diseño,
Implementación, y
Mantenimiento.

Usualmente la primera etapa es la de Análisis, en este trabajo se ha considerado incluir como la primera etapa la de Planeación del Proyecto, en esta etapa se proporcionarán técnicas y recomendaciones importantes para la Administración del Proyecto. Un aspecto muy importante será la participación del usuario, de hecho los proyectos surgen como necesidad de la Empresa y es necesaria la participación activa del usuario como parte del equipo de planeación.

Para las siguientes etapas del desarrollo se seguirá la metodología Jackson. Esta metodología también resalta la importancia que tiene la participación del usuario y considera vital el involucramiento que debe tener la gente de sistemas en el área del usuario, de hecho afirma que el analista deberá ser capaz de percibir la realidad del usuario tal y como él la ve. Se ha elegido esta metodología principalmente por los siguientes factores :

- a).- Modela la realidad del usuario,
- b).- No se comienza a partir de las especificaciones del sistema, sino que con esta metodología se desarrollan dichas especificaciones.

c).- Al tener nuevos requerimientos resulta muy sencillos incorporarlos al sistema que fué desarrollado bajo esta metodología.

Algo que es importante resaltar es que la metodología Jackson no incluye las tareas de Diseño de Base de Datos ni de Programación explícitamente, aunque deja las bases bien fundamentadas para que sea fácil su desarrollo auxiliándose de otras técnicas.

En el capítulo 1 se describirán los pasos a seguir para la Planificación del Proyecto, se incluyen algunas técnicas de estimaciones tales como COCOMO ("Constructive Cost Model"), PUTNAM y una técnica de Esterling.

En el capítulo 2 se describe en forma detallada cada uno de los pasos de la Metodología Jackson, se han incluido varias gráficas con el propósito de apoyar algunos conceptos teóricos.

En los capítulos 3 y 4 se desarrolla un caso práctico, siguiendo esta metodología; el caso es realmente sencillo; sólo pretende mostrar la forma en que deben seguirse paso a paso las metodologías expuestas.

CAPÍTULO 1.- ADMINISTRACIÓN DEL PROYECTO.

1.1.- INTRODUCCIÓN.

En términos generales un proyecto surge con el propósito de alcanzar objetivos específicos bien determinados. Para su desarrollo es necesario contar con alguna técnica de administración, ya que muchos proyectos realmente atractivos fracasan debido a errores u omisiones por la falta de alguna técnica de este tipo.

Los proyectos de software deben ser considerados como cualquier otro proyecto, el cual debe planearse, desarrollarse y controlarse mediante alguna técnica de administración de proyectos.

Un error que generalmente se comete al desarrollar proyectos de software es considerar que la primera etapa del proyecto comienza con la determinación de los requerimientos por parte del usuario. El analista de sistemas trata de satisfacer estos requerimientos mediante un sistema de cómputo, donde normalmente no hay un responsable del proyecto, en la mayoría de los casos el líder del proyecto es el analista del sistema.

Un aspecto muy importante para obtener mejores resultados es el hecho de formar un Equipo de Trabajo en el cual todos los integrantes interactúen. Es importante que los usuarios formen parte de este equipo, ya que de hecho ellos son los dueños del proyecto y la gente de sistemas es solamente un recurso.

Otro aspecto muy importante es que los proyectos a desarrollar deben decidirse en altos niveles de la empresa, con el fin de que éstos sean realmente de interés y para que se cuente con el apoyo necesario.

Así pues, se describirá una técnica de Administración de Proyectos orientada a reducir la magnitud de los problemas antes descritos; ésta será la primera etapa de desarrollo del Proyecto.

1.2.- ADMINISTRACIÓN DE PROYECTOS.

Antes de iniciar la descripción de los pasos de una metodología de Administración de Proyectos es necesario enunciar las definiciones de ambos conceptos: Administración y Proyecto :

Administración : Es un proceso que consiste en planear, organizar, ejecutar y controlar ciertas actividades con el fin de determinar y lograr objetivos, mediante los recursos humanos y de otros medios.

Proyecto : Es un conjunto de actividades planeadas, encaminadas a alcanzar un objetivo específico dentro de un marco de restricciones.

De ambas definiciones se puede concluir que las ideas básicas son :

- Alcanzar Objetivos,

- Planear, y
- Controlar

Así pues cualquier técnica de Administración de Proyectos debe tener como objetivos :

- a).- Planear y establecer las actividades necesarias para alcanzar los objetivos del Proyecto.
- b).- Organizar y dirigir recursos para realizar las actividades planeadas.
- c).- Controlar los procesos para asegurar que las actividades y sus objetivos sean alcanzados dentro del marco de restricciones prevaleciente.

De esta manera se pretende que desde el principio se tengan claros los objetivos y se cuente con herramientas para poder alcanzarlos, asegurando la satisfacción tanto del usuario como de la Empresa en general, tratando en todo momento de realizar las actividades utilizando solamente los recursos necesarios.

1.3.- DESCRIPCIÓN DE LA PRIMERA ETAPA DE DESARROLLO : PLANEACIÓN DEL PROYECTO.

El objetivo de esta etapa es combinar las tareas de investigación, estimación y organización. La investigación permitirá definir el alcance del proyecto. La estimación permitirá elaborar el calendario de trabajo, y la organización permitirá formar el Equipo de Trabajo.

En forma específica los principales objetivos de esta etapa son :

- a).- Determinar si el proyecto coincide con los intereses de la empresa.
- b).- Definir claramente el alcance, objetivos y restricciones del proyecto.
- c).- Asegurar que el proyecto resulte benéfico para la empresa.
- d).- Establecer un plan de desarrollo.
- e).- Organizar efectivamente los recursos.
- f).- Asegurar que todos los integrantes del Equipo de Trabajo entiendan claramente el alcance y objetivos.
- g).- Asegurar que todas las áreas involucradas estén identificadas y estén de acuerdo con su nivel de participación.

Para el seguimiento de esta metodología es necesario desarrollar los siguientes pasos :

- 1.- Determinar el Alcance del Proyecto
- 2.- Organizar el Equipo de Trabajo.
- 3.- Elaborar un Plan de Trabajo.
- 4.- Analizar los Costos y los Beneficios
- 5.- Elaborar un Plan para el Proyecto.

A continuación se describirán cada uno de estos pasos, las actividades, los documentos requeridos para desarrollarlas y los resultados que se esperan obtener.

1.3.1.- DETERMINACIÓN DEL ALCANCE DEL PROYECTO.

En este paso se deben determinar los límites del proyecto e identificar sus objetivos, los principales requerimientos y las restricciones bajo las cuales debe operar. Es importante conocer el medio ambiente del usuario, captar la realidad tal y como él la percibe.

Los principales objetivos de este paso son :

- a).- Conocer el Medio Ambiente que afectará el proyecto
- b).- Definir el alcance del proyecto
- c).- Identificar los problemas y requerimientos
- d).- Asegurar que el proyecto sea viable.

Para obtener tales objetivos, es necesario desarrollar las siguientes actividades :

1.- Investigación del Medio Ambiente.

Con la finalidad de conocer el ambiente que afectará el proyecto, es necesario iniciar una investigación exhaustiva de la realidad del usuario, es decir, se debe conocer con cierto detalle lo que hace el usuario, cómo lo hace, conocer los términos que utiliza. Esta actividad se podría considerar como de alineamiento con el usuario, ya que generalmente la gente de sistemas desconoce o tiene otra visión de las actividades que desarrolla el usuario.

Para hacer esta investigación inicial, es necesario convivir directamente con el usuario en las actividades que diariamente realiza. El objetivo es, como lo menciona Jackson (1), ver la realidad como el usuario la ve. Esta sensibilización es muy importante ya que será fundamental para el desarrollo de todo el proyecto.

2.- Establecimiento del Alcance del Proyecto.

Una vez que se conoce el ambiente del usuario, se debe determinar cual será el alcance y los límites del proyecto, estableciendo todas las áreas involucradas. Se debe ser muy claro en determinar tales límites en base a lo que los usuarios esperan del proyecto.

3.- Establecimiento de los Principales Objetivos y Requerimientos.

En esta actividad deberán enunciarse claramente cuales serán los objetivos del proyecto, los cuales deberán ser pocos y muy específicos.

Además de los objetivos, se deben realizar reuniones de trabajo con el fin de identificar los principales requerimientos que se deben cumplir. En esta actividad se debe asegurar que las áreas funcionales cuyo desempeño es crítico para alcanzar los objetivos, hayan sido incluidas en el alcance del proyecto.

El impacto de los requerimientos sobre el logro de los objetivos y el mejor desempeño en áreas críticas debe evaluarse asignándoles prioridades.

4.- Determinación de Restricciones.

Otra actividad que debe realizarse es la de identificar las principales restricciones dentro de las cuales será desarrollado el proyecto y bajo las cuales operará; éstas pueden ser de diferentes tipos :

- Legales
- Organizacionales
- Económicas
- Técnicas

5.- Presentación de Alternativas de Solución.

Tomando como base los objetivos del proyecto, los requerimientos del usuario y las restricciones bajo las cuales será desarrollado y operará el proyecto; debe evaluarse si el proyecto es viable para ser desarrollado.

Si se considera que el proyecto es viable, deben presentarse al usuario alternativas generales de solución. Dichas alternativas no son un plan detallado de solución, sino propuestas generales sobre la naturaleza de posibles soluciones. La propuesta debe considerar todos los factores externos que son visibles para el usuario, y debe redactarse de manera que permita caminos alternos.

El objetivo de este paso es concientizar al usuario de lo que deberá esperar del proyecto.

En el documento donde se presenten las alternativas de solución se deberán incluir los siguientes aspectos :

- Mejoras en áreas críticas y beneficios asociados
- Requerimientos técnicos y factibilidad asociada
- Impacto organizacional

Los resultados que se obtienen son los siguientes :

- a).- Descripción del Medio Ambiente del Usuario.
- b).- Descripción del Alcance del Proyecto
- c).- Lista de Problemas y Requerimientos
- d).- Lista de Restricciones
- e).- Descripción de Alternativas de Solución.

1.3.2.- ELABORACIÓN DE UN PLAN DE TRABAJO PARA EL PROYECTO.

Antes de comenzar la descripción de las actividades que deben desarrollarse, es pertinente enunciar la siguiente definición :

Plan : Programa, detalle de las cosas que hay que hacer para la ejecución de un proyecto.

Considerando esta definición, los objetivos de este paso son :

a).- Establecer estimaciones para :

- . Requerimientos de Recursos
- . Tiempos
- . Actividades
- . Resultados

b).- Proporcionar un marco de trabajo para dirigir y controlar las actividades y los recursos.

Para elaborar los planes de trabajo es necesario contar con buenas técnicas de estimación de tiempos de duración de las diferentes actividades; generalmente, en esta área, las estimaciones resultan ser muy alejadas de los tiempos reales, debido principalmente a que :

- Las actividades no siempre se ejecutan tal y como se planean.
- Las estimaciones se hacen sin tener, en ese momento, suficiente información, cuando ésta se obtiene ya no se hace una reprogramación del plan original.
- Generalmente se hacen estimaciones de actividades que requieren de la realización de otras actividades o tareas, las cuales en muchas ocasiones ni siquiera están definidas. Esto hace que no se tenga claro lo que implica desarrollar dichas actividades, y conduce a una estimación errónea.

- A menudo no se tiene claro lo que debe hacerse, o bien quien hace las estimaciones no es la persona más adecuada para hacerlas.
- La mayoría de la gente en el momento de hacer estimaciones es muy optimista, no considera tiempos muertos, entrenamiento, otras actividades, etc.

En fin, se podrían enumerar muchas más razones por las cuales casi siempre lo planeado resulta estar muy alejado de la realidad, lo cual trae como última consecuencia que los usuarios pierdan poco a poco la confianza en la gente de sistemas.

Las actividades que deben desarrollarse son las siguientes :

1.- Determinar las Actividades del Proyecto.

Lo primero que debe hacerse para la elaboración de un plan de trabajo es determinar las actividades que deberán desarrollarse dentro del proyecto. En el caso de proyectos de software se deberán determinar las actividades hasta que se obtenga la Implementación del Sistema.

La idea de esta etapa es que las actividades que se determinen sean cortas, ya que de esta manera será mas sencillo controlarlas y las estimaciones serán mas confiables.

Otro aspecto que debe considerarse es la dependencia que hay entre las diferentes actividades, es decir, debe desarrollarse una red de trabajo tipo PERT en donde se indique las actividades secuenciales y las paralelas.

2.- Elaborar estimaciones de duración de cada actividad.

Hacer estimaciones sobre duraciones de actividades es una tarea bastante difícil, ya que existe una gran variedad de factores que influyen y que afectan la duración de una actividad. Para tratar de reducir este problema a continuación se mencionan algunas técnicas que permiten hacer estimaciones :

a).- Modelo COCOMO ("Constructive Cost Model").

Mediante este modelo se puede calcular el esfuerzo de desarrollo de software como función del tamaño del programa y un conjunto de guías de costo que incluye la evaluación subjetiva del producto, del hardware, del personal, y de los atributos del proyecto.

De acuerdo al tamaño de los proyectos, COCOMO distingue tres tipos de proyectos :

- Orgánico : Proyectos pequeños y sencillos en los que participan equipos de trabajo pequeños pero con bastante experiencia.
- Semi-acoplado : Proyectos intermedios en tamaño y en complejidad, en el cual el Equipo de Trabajo está formado por personas de diferentes niveles.

- Empotrado : Proyectos grandes que deberán ser desarrollados en un tipo específico de Hardware.

Además COCOMO evalúa los siguientes factores, asignándoles un valor que va desde "muy bajo" hasta "extra alto", estos factores y sus posibles valores son :

INDICADORES		
ATRIBUTOS DEL PRODUCTO		
Confiabilidad requerida	"MUY BAJO"	"EXTRA ALTO"
Complejidad	0.75	1.40
	0.70	1.65
ATRIBUTOS DEL HARDWARE		
Restricciones de Desempeño del Equipo	1.00	1.66
Restricciones de Memoria	1.00	1.56
Tiempo de respuesta	0.87	1.15
ATRIBUTOS DEL PERSONAL		
Capacidad de análisis	1.46	0.71
Experiencia en el método de desarrollo	1.29	0.82
Experiencia con el equipo de cómputo	1.21	0.90
Experiencia con el lenguaje de programación	1.14	0.95
ATRIBUTOS DEL PROYECTO		
Utilización de herramientas de software	1.24	0.83
Tiempo requerido para el software	1.26	1.10

Para el proyecto en particular, debe evaluarse cada uno de estos atributos asignándoles valores de acuerdo a la tabla anterior. Si la calificación para un atributo es "muy bajo" tomará el valor de la izquierda y si es "extra alto" tomará el de la derecha, las calificaciones intermedias tomarán valores intermedios; se obtienen así indicadores denominados multiplicadores de esfuerzo. El producto de todos los multiplicadores de esfuerzo produce el factor de ajuste de esfuerzo (FAE).

El modelo COCOMO utiliza la siguiente ecuación para la estimación de esfuerzo :

$$E = a_i(KLDC)^{b_i} \times FAE$$

donde :

E : Esfuerzo estimado en personas/mes

KLDC : Número estimado de líneas de código, en miles

a_i, b_i : Dependen del tipo de proyecto

Proyecto	a_i	b_i
Orgánico	2.4	1.05
Semi-acoplado	3.0	1.12
Empotrado	3.6	1.2

FAE : Factor de ajuste de esfuerzo.

b).- Modelo de estimación PUTNAM.

Es un modelo multivariable y dinámico que asume una distribución específica del esfuerzo a lo largo de la vida de un proyecto de desarrollo de software.

La distribución del esfuerzo para grandes proyectos puede ser caracterizada por la curva de Rayleigh-Norden, la cual puede utilizarse para obtener una ecuación que relacione el número de líneas de código esperadas con el esfuerzo y tiempo de desarrollo :

$$K = L^3 / (C_k^3 t_d^4)$$

donde :

C_k : Es una constante que refleja las restricciones que frenan el progreso del programador, puede tomar valores desde 2,000 (entorno pobre de desarrollo) hasta 11,000 (entorno excelente).

K : Esfuerzo de desarrollo estimado (persona-año)

t_d : Tiempo de desarrollo, en años

L : Número de líneas de código esperadas

c).- Modelo de Estimación Temporal.

El proceso de desarrollo de software está afectado por una gran cantidad de factores que resultan de la interacción entre las personas que participan en el proyecto y el entorno en el cual se desarrolló el proyecto.

Esterling afirma que las reuniones de trabajo y otras actividades "no productivas" que se presentan en una jornada de ocho horas de trabajo hacen que las personas no sean tan productivas como se supone. Así pues, considera que estos factores deben incluirse en el modelo de estimación, considera los siguientes parámetros :

a : Fracción media del trabajo diario gastado en trabajos administrativos o indirectos (0-0.5).

t : Duración media de las interrupciones del trabajo, en minutos (1-20).

r : Tiempo de recuperación medio después de la interrupción, en minutos (5-10).

k : Número de interrupciones por día de trabajo de la gente que participa directamente en el proyecto (1-10).

p : Número de interrupciones por día de trabajo por otras causas (1-10).

Esterlin (2) afirma que para estimar el porcentaje de tiempo de trabajo útil por jornada de trabajo por persona, se puede utilizar la siguiente ecuación:

$$w = 0.125 \times (8 - 8a + g - 4r/60 - p(t+r)/60 - k(t+r)/60)$$

donde :

g : Número promedio de horas extras por jornada de trabajo

El caso utópico es cuando no hay tiempos muertos ($a, r, t, k, p = 0$) y no hay tiempo extra ($g = 0$); en este caso $W = 1$, lo cual significa que de una jornada de 8 horas el 100 % se dedica al trabajo.

Además Esterling determina que el coeficiente de tiempo en personas/día necesarias para terminar el proyecto se puede estimar mediante la siguiente ecuación :

$$T = 7/tnw$$

De las técnicas antes descritas, dos de ellas se basan en el concepto de líneas de código, lo cual en algunos casos resulta un poco difícil de evaluar y se presta a controversias; sin embargo son técnicas que se basan en la experiencia y es conveniente considerarlas.

3.- Asignación de Recursos.

Una vez que se tienen las estimaciones para la duración de cada una de las actividades en personas/semana, personas/mes, etc. se asigna el número de recursos que van a desarrollar cada una de las actividades, dependiendo de las fechas de entrega y de la disponibilidad de recursos.

4.- Elaborar la Calendarización del Proyecto.

Ya que se han asignado los recursos y considerado las dependencias que existen entre las diferentes actividades, puede elaborarse fácilmente la calendarización del proyecto.

Para la realización de este paso es necesario :

- Alcance del Proyecto.

Los resultados que se obtendrán son :

- Calendarización del Proyecto.

1.3.3.- ORGANIZACIÓN DEL EQUIPO DE TRABAJO.

El objetivo de este paso es organizar a las personas que participarán directa o indirectamente en el desarrollo del Proyecto. Se

debe garantizar que todos los involucrados conozcan los objetivos del proyecto, el plan de trabajo, su nivel de participación y el tiempo que deberán dedicar al desarrollo del Proyecto, todo esto con el fin de obtener su compromiso.

Como ya se ha mencionado anteriormente el Equipo de Trabajo no sólo deberá estar integrado por gente de sistemas, sino que es indispensable la participación activa del usuario; en este paso se deberán asignar los roles del equipo, entre ellos se deberá nombrar al líder del proyecto.

El Equipo de Trabajo podrá estar asesorado por uno o varios especialistas, tanto técnicos como del área del usuario. Estas personas deberán ser identificadas y notificadas.

Las actividades que deben desarrollarse son las siguientes :

1.- Integración del Equipo de Trabajo.

En este paso se asignan los roles a cada uno de los integrantes del Equipo de Trabajo, se nombra al Líder del Proyecto.

2.- Identificación de Especialistas.

Identificar cualquier especialista (interno o externo), tanto técnico como del área del usuario, requerido en el proyecto.

3.- Identificación de Recursos.

En esta actividad se deben identificar los recursos necesarios para el desarrollo del proyecto; existen tres tipos de recursos a identificar:

a).- Recursos Humanos

Los recursos humanos necesarios ya se han identificado y de hecho en la actividad 1 ya fué integrado el Equipo de Trabajo.

b).- Recursos de Hardware

Se deben considerar tres categorías de Hardware :

- Equipo de Desarrollo : Es el equipo donde se desarrollarán los programas de cómputo.
- Equipo de Producción : Es el equipo de cómputo donde operará el Sistema.
- Equipo Periférico : Impresoras, terminales, reguladores, etc.

No es necesario tener dos computadoras, para desarrollo y para producción; pero es importante manejar dos ambientes diferentes, aún cuando ambos residan sobre la misma computadora.

c).- Recursos de Software.

Se debe identificar el tipo de Software necesario para el desarrollo del Proyecto; existen tres categorías :

- Herramientas orientadas al código. Incluyen compiladores, intérpretes, editores, etc.
- Herramientas de Metodología. Las herramientas de este tipo facilitan las tareas de análisis y diseño de sistemas; entre este tipo de herramientas se encuentran las tipo CASE ("Computer-Aided Software Engineering").
- Herramientas de 4a. Generación. Este tipo de herramientas facilitan enormemente las tareas de programación, reduciendo el tiempo de desarrollo de programas; aunque tienen algunas desventajas de desempeño al momento de ejecución, debido a que consumen muchos recursos de la computadora.

4.- Establecimiento de Procedimientos para la Administración del Proyecto.

Definir los estándares y procedimientos que se van a utilizar en :

- Administración y resolución de problemas
- Administración y control de cambios
- Administración y aseguramiento de calidad

Deben definirse claramente las responsabilidades asociadas a cada una de dichas áreas. Tanto los estándares como los procedimientos deberán comunicarse a todos los miembros y participantes del proyecto.

5.- Desarrollar Planes de Trabajo Individuales.

En esta actividad es conveniente desarrollar planes detallados con el fin de comunicar las expectativas al Equipo de Trabajo para lograr que estén conscientes sobre el alcance y los planes. Para asegurar esto se pueden realizar las siguientes tareas :

a).- Desarrollar planes detallados.

Estos planes deben elaborarse para identificar :

- El trabajo asignado a cada individuo
- La calendarización de actividades para cada miembro del equipo
- Fechas de entrega y fechas de evaluaciones y de revisiones.

b).- Informar al Equipo de Trabajo.

Es necesario asegurar que todos los integrantes del equipo conozcan:

- Antecedentes del proyecto
- Objetivos y restricciones
- Planes
- Método de desarrollo
- Procedimientos de control
- Organización del Equipo de Trabajo

Los documentos requeridos para el desarrollo de este paso son :

- Plan de Trabajo para el Proyecto.

Los resultados a obtener :

- Organización del Equipo de Trabajo
- Calendarización del Proyecto actualizada
- Requerimientos de recursos de apoyo

1.3.4.- ANÁLISIS COSTO/BENEFICIO.

Al desarrollar cualquier proyecto se debe partir del principio básico de que no se invierte dinero en algo que no aporte beneficios superiores a su inversión.

El objetivo de este paso es desarrollar tanto un análisis costo/beneficio; éste debe ser revisado, a detalle y deberá decidirse si el proyecto será desarrollado o no. La decisión debe tomarse en este punto, teniendo todas las bases necesarias para apoyar cualquier decisión.

Para elaborar el Análisis costo/beneficio es necesario considerar tanto los costos de desarrollo como los que se originarán cuando el sistema esté en operación.

Al determinar los beneficios se deben considerar tanto los tangibles como los intangibles. Para evaluar estos últimos es necesario encontrar indicadores mediante los cuales se pueda tener una medición.

Tanto los costos como los beneficios se pueden obtener una sola vez, o bien en forma recurrente.

Los objetivos de hacer un Análisis Costo/beneficio son :

- a).- Establecer la viabilidad del proyecto en términos de costos y de beneficios esperados.
- b).- Identificar :
 - Restricciones de presupuesto y fechas de entrega,
 - Falsas expectativas de beneficios
- c).- Confirmar el criterio de aceptación financiero.

Las actividades a desarrollar son :

1.- Establecimiento de Costos de personal

Determinar los costos del Personal asignado al proyecto (usuarios, costos de desarrollo y de soporte, etc).

Deben incluirse los costos de desarrollo y de conversión originados por los siguientes aspectos :

- Equipo de desarrollo
- Apoyo de especialistas
- Usuarios
- Consultores y contratistas
- Instructores

Y en los gastos operativos :

- Operación
- Mantenimiento y soporte
- Usuarios
- Entrenamiento

2.- Establecimiento de Costos Operativos y de Desarrollo

Estimar los costos de desarrollo y operativos del proyecto, incluyendo :

- Hardware
- Software
- Costos Operativos
- Entrenamiento

3.- Establecimiento de Costos de Desplazamiento

Estimar los costos de desplazamiento, es decir, aquellos costos que van a ser eliminados mediante el sistema propuesto.

4.- Cuantificación de Beneficios

Determinar los beneficios que van a ser obtenidos mediante el sistema propuesto :

- a).- Beneficios directos tangibles
- b).- Costos en los que se incurrirían en caso de que el sistema propuesto no fuera desarrollado
- c).- Beneficios intangibles, es decir, beneficios que se van a obtener pero que no pueden ser cuantificados fácilmente en términos contables. Para tratar de cuantificar el impacto de este tipo de beneficios se pueden determinar beneficios financieros indirectos como :

- . Incremento en ingresos
- . Disminución en costos por menor número de problemas.

5.- Evaluación de Riesgos

Determinar los riesgos asociados con el desarrollo del proyecto:

- a).- Riesgos implícitos en el Análisis costo/beneficio
- b).- Riesgos organizacionales
- c).- Riesgos de planeación
- d).- Riesgos Técnicos

Con base en dicha evaluación de los riesgos, deben tomarse acciones concretas con el fin de minimizarlos, y puede modificarse el Plan de Trabajo para el Proyecto.

6.- Presentación del Análisis Costo/beneficio.

El Análisis Costo/beneficio debe presentarse tanto a los altos niveles del área del usuario como a los directivos de Sistemas. Si el proyecto se aprueba, deben definirse claramente el alcance, los requerimientos, y las restricciones bajo las cuales se va a desarrollar el proyecto.

El formato en que debe presentarse los Análisis Costo/beneficio es el siguiente :

- 1).- Resumen de Costo/Beneficio
 - . Flujo de efectivo/valor presente neto
 - . Costos/beneficios netos
 - . Beneficios intangibles
 - . Principales supuestos
- 2).- Resumen de Costos
 - . Costos de desarrollo
 - . Costos de operación
- 2).- Resumen de Beneficios
 - . Beneficios tangibles
 - . Costos reemplazados
 - . Beneficios intangibles
 - . Incremento de ingresos

Los documentos requeridos son :

- Alcance del Proyecto

- Plan del Trabajo para el Proyecto
- Organización del Equipo de Trabajo

Los resultados esperados son:

- Análisis Costo/Beneficio

1.3.5.- ELABORACIÓN DEL PLAN DEL PROYECTO.

Una vez que se han desarrollado todos los pasos anteriores es necesario integrar un solo documento en el cual se presenten los resultados a los Directivos de la Empresa, o a quien ellos designen, para obtener su aprobación, este documento es llamado Plan del Proyecto.

El formato en que deberá ser presentado el documento es el siguiente:

1.- Introducción.

- Propósito del documento
- Objetivos de la Planeación del Proyecto.

2.- Alcance y Objetivos.

- Alcance del Proyecto
- Objetivos

3.- Estructura Organizacional

- Estructura del Equipo de Trabajo

4.- Requerimientos

- Recursos Humanos
- Hardware
- Software

5.- Análisis Costo/Beneficio

- Análisis Costo/Beneficio

CAPÍTULO 2.- METODOLOGÍA JACKSON PARA EL DESARROLLO DE SISTEMAS

2.1.- UBICACIÓN DE LA METODOLOGÍA.

La Metodología Jackson JSD ("Jackson System Development") es una metodología orientada a la estructura de datos que se basa en transformar la estructura de los datos en una estructura de Software. Se parte de que la "paralelización" de la estructura de los datos de entrada y de salida asegurará un diseño de calidad.

Extensiones más recientes de esta metodología se enfocan a la identificación de entidades de información y de acciones que dichas entidades ejecutan, o bien que son ejecutadas sobre ellas. Tales extensiones son similares en algunos aspectos al método de diseño orientado a objetos.

Las metodologías orientadas a las estructuras de los datos pueden aplicarse con bastante éxito a las aplicaciones que tengan una estructura jerárquica.

Este tipo de metodologías MED (Metodologías orientadas a la Estructura de los Datos) no hacen uso explícito de diagramas de flujos de datos.

Cada metodología del tipo MED tiene sus propias reglas, pero todas poseen las siguientes características :

- a).- Evalúan las características de la estructura de datos, tanto de entrada como de salida con el fin de derivar, a partir de ahí, la estructura del sistema.
- b).- Representan los datos en términos de formas elementales, tales como secuencia, selección y repetición.
- c).- Transforman la representación de la estructura de datos en una jerarquía de control a partir de la cual generan el software estructurado.
- d).- Refinan la jerarquía del Software con el fin de que sea sencilla su implementación.
- e).- Desarrollan finalmente una descripción por procedimientos del Software.
- f).- En este tipo de métodos no está clara la división entre los pasos de Análisis y los de Diseño.

JSD surge como una extensión de JSP ("Jackson Structured Programming"), la cual es una metodología de Jackson para el diseño de programas orientado a la estructura de los datos.

Un programa JSP es un proceso secuencial en el cual sus entradas y sus salidas son vistas como grupos secuenciales de registros. Una forma simplificada del procedimiento JSP es el siguiente :

- a).- Describir la estructura de cada entrada y de cada salida.
- b).- Combinar estas estructuras para formar una estructura de programa.
- c).- Listar las operaciones que el programa debe ejecutar para producir las salidas a partir de las entradas y ubicar cada operación en su lugar en la estructura del programa.
- d).- Escribir el texto del programa, agregando las condiciones necesarias para controlar la ejecución de las iteraciones y selecciones.

2.2.- CARACTERÍSTICAS GENERALES.

La Metodología JSD dirige gran parte del ciclo de vida del Desarrollo de Sistemas, ya sea en forma directa o bien proporcionando un marco de trabajo en el que se pueden utilizar otras técnicas.

La base fundamental de la que parte JSD es el modelado de la realidad a la cual se va a referir el sistema, es decir, para JSD la realidad proporciona el objeto para el sistema. Su objetivo es el asegurar que el sistema refleje correctamente el mundo real, tal y como es.

JSD ayuda a especificar el mundo real, trabajando en cooperación con el usuario, proporcionando conceptos y notaciones de tal manera que la especificación pueda expresarse claramente. El analista debe desarrollar actividades de análisis para lograr un buen entendimiento de las necesidades del usuario, pero también debe tener una alta capacidad de síntesis para lograr obtener una buena abstracción del mundo real y representarlo mediante un modelo.

La idea de JSD es que un buen sistema con un buen diseño estructural será un buen sistema y además, será fácil de mantener (inclusive si la programación es mediocre), pero en cambio, un sistema con un mal diseño estructural siempre será un mal sistema aun cuando la programación sea excelente.

JSD es utilizado para sistemas en donde el orden en el que suceden los eventos es importante.

Para JSD el ciclo de Desarrollo de Sistemas se divide en dos partes: Especificación e Implementación; la fase de diseño que muchas metodologías consideran como una fase en el ciclo de desarrollo, JSD la considera como parte de la Implementación.

JSD no incluye las actividades de Selección de Proyectos, Planeación y Administración de Proyectos, Análisis de Costo/Beneficio ni de Procedimientos para la Aceptación de Sistemas.

JSD consta de seis pasos, de los cuales los primeros cuatro están encaminados a crear una especificación del sistema y los últimos dos son para implementar dicha especificación. JSD no considera que las especificaciones las debe hacer el usuario, sino que es resultado de una de las etapas del ciclo de desarrollo de sistemas.

Una especificación JSD consiste principalmente de una red de procesos secuenciales, en donde cada proceso puede contener sus propios datos locales. La especificación se desarrolla comenzando con un conjunto particular de procesos modelo y posteriormente se van agregando nuevos procesos a la especificación.

2.3.- INTRODUCCIÓN A LA METODOLOGÍA.

Tradicionalmente, el punto de partida en el desarrollo de sistemas es el requerimiento funcional, el cual es la base de la Descomposición Funcional. JSD pospone la consideración de las funciones del sistemas hasta etapas posteriores en el desarrollo.

Una razón para iniciar el desarrollo a través de modelos, en lugar de consideraciones funcionales es que el modelo proporciona las bases conceptuales y computacionales para las funciones del sistema.

En JSD se modela la realidad, en donde el sistema es considerado una simulación de dicha realidad. El modelado consiste en hacer una descripción abstracta donde se incluyen los aspectos relevantes y se omiten los irrelevantes, dicho modelo sirve como contexto para agregar las funciones. JSD comienza haciendo directamente un modelo dinámico basado en una descripción dinámica del mundo real; dicho modelo está compuesto de un cierto número de procesos secuenciales, llamados así ya que no se tiene la noción de concurrencia, paralelismo o multitarea. Sólo puede suceder una cosa en un solo momento dentro del proceso, aunque dentro del modelo pueden estar sucediendo muchas cosas al mismo tiempo.

Existen dos pasos que deben hacerse durante el modelado de la realidad:

- a).- Hacer una descripción abstracta de la realidad bajo estudio, y
- b) Determinar de qué manera dicha descripción abstracta puede ser trasladada en un modelo de trabajo de cómputo.

La descripción abstracta y su traslación proporcionan el contexto para las especificaciones funcionales. Ambas definen un conjunto de palabras que van a ser utilizadas en la especificación de funciones; el significado de cada palabra en el mundo real es dado por las conexiones entre el mundo real y el modelo. De esta manera las funciones pueden hacer referencias no ambiguas a términos previamente definidos.

El modelo define implícitamente un conjunto coherente de posibles funciones del sistema, ya que durante la descripción abstracta se determinan las entidades involucradas en el contexto de la realidad bajo estudio, y sobre estas entidades se definen dichas familias coherentes de funciones.

Las ventajas que JSD tiene sobre los procedimientos funcionales son las siguientes :

- a).- En los procedimientos funcionales el analista no entiende el contexto real del usuario.
- b).- Al desarrollar las especificaciones junto con el usuario como parte del ciclo de desarrollo se garantiza que se ha comprendido lo que realmente el usuario necesita.
- c).- Las especificaciones funcionales son, generalmente, excesivamente técnicas.
- d).- En JSD la comunicación inicial entre el analista y el usuario es sobre el mundo real del usuario, es decir, el analista aprende a ver el mundo como el usuario lo ve.

El modelado del mundo real consiste en describir lo que ya existe, mientras que para definir y especificar funciones se debe crear algo nuevo. JSD reconoce que el analista debe practicar tanto la descripción como la innovación al producir la especificación del sistema.

Para construir un JSD se deben ejecutar los siguientes pasos :

1.- Paso de Acciones y Entidades.

En forma análoga al Diseño Orientado a Objetos, se identifican las entidades : personas, objetos u organizaciones que necesita el sistema para producir o usar información. Y acciones : sucesos que ocurren en el mundo real, que afectan a las entidades. Se debe hacer una descripción breve y en lenguaje natural del problema bajo estudio.

2.- Paso de Estructuración de las Entidades.

Las acciones que afectan a cada entidad son ordenadas de acuerdo al tiempo en el que ocurren y se hace una representación en forma de diagrama.

3.- Paso de Modelado Inicial.

Las entidades y acciones se representan como un modelo del proceso real, es decir, se definen las conexiones entre el modelo y el mundo real.

4.- Paso de las Funciones.

En este paso se especifican las funciones que corresponden a las acciones previamente definidas.

5.- Paso de Sincronización del Sistema.

Se establecen y especifican las características de planificación del proceso. Se especifican las ligaduras de tiempo impuestas en el Sistema.

6.- Paso de Implementación.

Aquí se especifica el Hardware y el Software como un diseño. Los problemas deben descomponerse en estructuras jerárquicas de partes que puedan ser representadas por las tres formas estructurales : Secuencia, Selección e Iteración.

El paso de implementación consista en hacer una serie de consideraciones de acuerdo con las necesidades de cada sistema, tales como :

- Determinar el número de procesadores requeridos o disponibles, y la asignación de especificaciones de procesos a los procesadores.

- Transformar las especificaciones de procesos para poder hacer un control explícito de ellos y construir procesos de propósito especial para programar su ejecución.

Aunque el proceso iterativo en el desarrollo de Software es inevitable, JSD propone las siguientes sugerencias:

- a).- Cada decisión en el desarrollo, tanto en especificación como en implementación, debe hacerse en forma completamente explícita.

- b).- Deben tomarse decisiones sencillas y descriptivas en lugar de decisiones inventivas y difíciles.

- c).- Los errores en decisiones deben detectarse lo más rápido posible.

- d).- Se debe tratar de tomar decisiones en forma independiente.

La programación para JSD está diseminada en varios de los pasos de su metodología : Paso de Estructura de Entidades, Paso de Funciones, Paso de Implementación, etc.

Para JSD la distinción importante no es entre Análisis y Diseño, sino entre Especificación e Implementación.

2.4.- PROCESO DE DESARROLLO.

2.4.1.- Pasos del Desarrollo.

Como ya se ha mencionado anteriormente JSD comienza desarrollando junto con el usuario la especificación, no inicia a partir de ella. Para JSD el ciclo de desarrollo de software se divide en dos etapas: Desarrollo de la Especificación y Desarrollo de la Implementación.

Los pasos a desarrollar utilizando esta metodología se muestran en el siguiente cuadro, los números entre paréntesis indican el orden en que se deben ejecutar cada uno de los pasos.

P R O C E D I M I E N T O	Desarrollo de la Especificación	Especificación del Modelo de la Realidad	Descripción Abstracta de la Realidad	Paso de Entidades y Acciones (1) Paso de Estructura de Entidades (2)
			Definición de un Modelo Trasladado	Paso de Modelado Inicial (3)
		Especificación de las Funciones del Sistema	Paso de Funciones (4) Paso de Sincronización del Sistema (5)	
J S D	Desarrollo de la Implementación	Paso de Implementación (6)		

Dentro de la primera etapa se efectúa la especificación del modelo de la realidad así como las funciones del sistema. En estas etapas se supone que cada instancia de cada entidad tendrá su propio y dedicado procesador, para cada uno de los procesos que ejecuta, es decir, si se tiene la entidad X, se supone que habrá un proceso dedicado y exclusivo para cada uno de los procesos que ejecuta X1, X2, ..., Xn, donde Xi son instancias de la entidad X. Esto se hace con el fin de que las limitaciones de hardware no impidan desarrollar una adecuada especificación del sistema.

A partir de las especificaciones se desarrolla la implementación. En esta fase se introducen las limitaciones del hardware y se adaptan las especificaciones, con el fin de transportarlas y ejecutarlas sobre un solo procesador, eliminando de esta manera el supuesto hecho en la primera fase.

2.4.2.- Desarrollo de la Especificación.

Para obtener la especificación del sistema será necesario desarrollar 5 pasos (según se explicó en el punto anterior, 2.4.1), a continuación se detallan cada uno de ellos.

2.4.2.1.- Paso de Entidades y Acciones.

El punto de partida en JSD es contar con una idea general del propósito del Sistema. Se comienza haciendo, conjuntamente con el usuario, una descripción abstracta del mundo real, la cual debe ser breve y escrita en lenguaje natural.

En este paso se definen las entidades y las acciones que ellas ejecutan o bien, que son ejecutadas sobre ellas.

En base a la descripción abstracta de la realidad se elabora una lista con las entidades y las acciones. De principio todos los sustantivos contenidos en la descripción son candidatos a ser entidades, y todos los verbos son candidatos a ser acciones.

Una acción debe ser considerada atómica, la cual no puede ser descompuesta en sub-acciones. Debe distinguirse entre tipos de entidad e instancias, es decir, una instancia es un objeto particular y un tipo de entidad es general. En JSD deben incluirse tipos de entidades a menos de que diferentes instancias ejecuten, o sufran, diferentes acciones.

Sobre la lista de entidades candidatos se eligen aquellos que se desean incluir en el sistema, esta parte es muy importante ya que delimitará el alcance del sistema, puesto que entidades que no se elijan en este momento no podrán ser explotadas posteriormente.

En JSD una entidad está completamente descrita por sus acciones, todo su ciclo de vida debe ser incluido dentro de los límites del modelo, en JSD una instancia de una entidad no cambia su tipo durante su ciclo de vida.

Una vez que se seleccionan las entidades a ser incluidas en el modelo se debe hacer una lista precisa de las acciones que ejecutan o son ejecutadas sobre ellas. Ya que se tienen ambas listas, de acciones y de entidades, se debe hacer una descripción de cada una de las acciones, listando los posibles atributos que dichas acciones pudieran tener. Estas descripciones y listas de atributos proporcionan el punto de partida para los registros en un diccionario de datos.

Las listas de entidades y de acciones forman un glosario de términos que puede ser utilizado durante todo el ciclo de desarrollo. Ambas listas son tentativas ya que serán modificadas en los pasos de Estructura de Entidades y en el de Modelado Inicial.

El resultado de este paso es una descripción altamente abstracta del mundo real. El sistema será capaz de producir las salidas en términos de su descripción.

2.4.2.2.- El Paso de Estructura de Entidades.

Debido a que las acciones son ejecutadas en un orden determinado, es necesario expresar dicho orden. En general el orden en que suceden las acciones no es completamente fijo, sin embargo están restringidas a seguir ciertos patrones de comportamiento, los cuales deberán ser especificados en esta parte a través de diagramas de estructura de entidades.

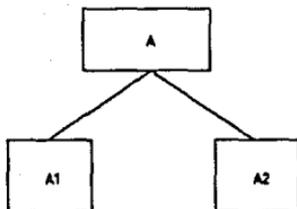
Una sola entidad puede requerir más de un diagrama, en donde pueden surgir entidades o acciones que no se incluyeron en las listas originales.

Un requerimiento fundamental en los diagramas es representar el ciclo de vida completo real de cada entidad, generalmente esto no se hace debido a que el analista piensa en forma anticipada en aspectos de la implementación; aunque más tarde podrán eliminarse aquellas acciones sobre las cuales el sistema no automatizará nada.

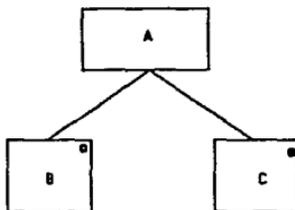
La estructura de los diagramas es de árbol, con una raíz y con hojas, en donde debe haber una y solo una manera de llegar de la raíz a cada una de las hojas. El árbol muestra como están ordenadas las acciones. Las entidades o acciones que no son hojas tienen entidades o acciones asociadas en el siguiente nivel inferior, a las primeras se les denomina padres mientras que a las segundas, hijos.

Una entidad o acción se representa por un rectángulo en cuyo interior se escribe el nombre y en la esquina superior derecha un asterisco, un pequeño círculo o nada cuando es una iteración, una selección o una secuencia, respectivamente:

a).- Secuencia : En el siguiente diagrama se muestra un proceso A, el cual está conformado por los procesos A1 y A2, en este caso el proceso A termina su ejecución cuando se ejecuta el proceso A1 y después el proceso A2. Una secuencia puede tener cualquier número de hijos pero se deben mostrar cada uno de ellos; cuando no se conoce el número de hijos se puede hacer una representación por medio de una iteración en lugar de utilizar una secuencia. Cada acción comienza una vez que la que le antecede haya terminado.

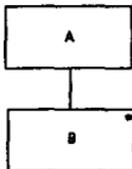


b).- Selección : En este caso sólo se ejecuta un proceso el A o el B, pero no ambos.



En ocasiones cuando una de las opciones no aplica, se esquematiza como una hoja nula.

c).- Iteración : El proceso B se ejecuta varias veces hasta que se cumple alguna condición.



Para simplificar los diagramas de estructura se pueden ejecutar las siguientes operaciones :

- 1).- Quitar las hojas nulas que sean parte de una secuencia.
- 2).- Quitar las hojas nulas que sean parte de una selección que a su vez sea parte de una iteración.
- 3).- Reemplazar las selecciones que contengan sólo una parte , por esta parte.
- 4).- Reemplazar las secuencias que contengan sólo una parte, por ella.
- 5).- Quitar una de las iteraciones cuando haya dos idénticas que sean parte consecutiva de una secuencia.
- 6).- Reducir de dos niveles de iteración a uno solo cuando haya una iteración anidada, es decir, reemplazar B por C cuando A sea una iteración de B, y B sea una iteración de C.

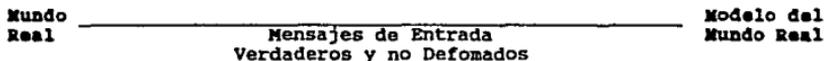
Anexo a los diagramas se pueden incluir notas específicas cuando se crea necesario.

En algunos casos se presentan otro tipo de entidades conocidas como Entidades Marsupiales, las cuales de primera instancia parecen estar empotradas dentro de otras entidades pero que sin embargo en este paso emergen como entidades separadas.

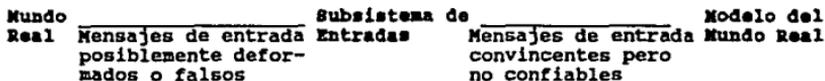
Los diagramas de estructura, así como las listas de entidades y de acciones, y las descripciones de las acciones forman una parte integral de las especificaciones del sistema.

Pueden presentarse inconsistencias entre el patrón de restricciones expresadas en el modelo y los mensajes que se transmiten del mundo real, es decir, que los mensajes enviados del mundo real al modelo se distorsionen de alguna manera. Así pues no se puede suponer que los mensajes de entrada al modelo serán siempre verdaderos; por esta razón debe interponerse un subsistema de entradas entre el mundo real y el modelo, dicho subsistema será construido en el siguiente paso.

Caso Ideal :



Caso más frecuente :



2.4.2.3.- El Paso de Modelado Inicial.

Al desarrollar los pasos anteriores se elabora una descripción abstracta del mundo real en términos de entidades, acciones y el orden en que estas se ejecutan. En este paso se comenzará a construir el sistema a partir de la especificación de un conjunto secuencial de procesos que modelan las entidades del mundo real así como su comportamiento. Las estructuras de entidades en la descripción abstracta se convertirán en procesos secuenciales en el modelo.

Para asegurar que el comportamiento del modelo refleje correctamente los sucesos que ocurren en el mundo real será necesario conectar los procesos modelo, directa o indirectamente, al mundo real.

Para JSD los procesos que ocurren en el mundo real son procesos de nivel 0 y los procesos modelo son procesos de nivel 1, para mayor facilidad en su manejo, se hace referencia a ellos como entidad-0 y entidad-1, respectivamente, para cada entidad en el modelo.

Para cada proceso secuencial del mundo real habrá un proceso secuencial en el sistema capaz de ser ejecutado por una computadora, aunque no se pueden ejecutar directamente en este momento ya que el modelo presenta características inadecuadas, puesto que hay muchos procesos y algunos de ellos tienen una duración muy larga. En el paso de implementación se aplicarán algunas transformaciones para hacer que estos procesos sean ejecutables sobre el hardware y el software disponibles.

El nivel de abstracción que debe tenerse en este momento debe ser bastante alto, se puede pensar que cada proceso del modelo correrá sobre su propio procesador, el cual tendrá dispositivos apropiados para conectarse con el mundo real, así como con otros procesadores.

El hecho de mencionar que los procesos serán ejecutados por una computadora no excluye la posibilidad de que en el paso de implementación se decida que algunos procesos deberán ser ejecutados manualmente.

El objetivo en este paso es el de describir la forma en que se conectan los procesos del mundo real con los procesos del modelo, y describir la forma en que se deban tratar los problemas que surjan debido a las conexiones, las cuales son, inevitablemente, imperfectas. Dichas descripciones se representan mediante Diagramas de Especificación del Sistema ("SSD, Specification System Diagram").

Tanto en el paso de Modelado Inicial como en el de Funciones, existen sólo dos formas de conectar procesos entre sí : a través de mensajes o señales, o bien a través de vectores de estado.

a).- Conexión a través de Mensajes :

Las conexiones a través de mensajes o señales consiste en que un proceso envía un conjunto ordenado de mensajes o registros a otro

proceso para que éste los lea. Las conexiones a través de mensajes se representa mediante un círculo :



El proceso del mundo real para la entidad A se conecta a través de mensajes con el proceso modelo de la misma entidad, es decir, el modelo deberá responder ante mensajes que se generen en la realidad.

En el caso en el que se desee hacer una representación de muchos a uno se representa la multiplicidad mediante una doble barra del lado de los procesos que sean varios, ejemplo :



significa que varios procesos B-1 están conectados con un proceso A-1. La doble barra solo especifica multiplicidad en la conexión y no tiene ninguna relación con el número de entidades.

Cuando dos procesos A y B están conectados a través de mensajes (DS) se dan las siguientes características :



- 1).- La iniciativa parte de A para determinar el contenido y el orden de los registros de DS, este conjunto de mensajes se puede deducir considerando sólo el comportamiento de A.
- 2).- B tiene que leer todos los registros escritos por A en el mismo orden en que A los escribe.
- 3).- En el caso de que B se retrase, o pare temporalmente, los registros de DS esperarán en un buffer hasta que B continúe su ejecución y siga leyéndolos.
- 4).- Si B trata de leer DS pero no hay registros disponibles en el buffer, la ejecución de B no puede continuar hasta que haya un registro disponible en DS para que pueda ser leído.
- 5).- Se considera que la capacidad del buffer es ilimitada.

En resumen, se puede decir que B se comporta en la forma en que A lo determina.

b).- Conexión a través de Vector de Estado :

Cuando dos procesos están conectados a través de un vector de estado significa que un proceso inspecciona directamente las variables internas (vector de estado) de otro proceso, las conexiones mediante inspección del vector de estado se representan por un rombo :



En este caso se tiene una conexión a través de una inspección del vector de estado. El modelo deberá hacer una inspección de hechos que se dan en la realidad, el mundo real no envía ningún mensaje al modelo.

Las conexiones a través de vectores de estado entre procesos son muy diferentes, éstas presentan las siguientes características :



1).- La iniciativa para la comunicación parte de Q. No existe ninguna operación en P que origine que Q inspeccione el vector de estado. A la operación de que Q inspeccione el vector de estado de P se le denomina "getsv".

2).- Las operaciones de "getsv" (para inspeccionar el vector de estado) son ejecutadas por Q.

3).- Si Q se alenta o suspende temporalmente su ejecución, entonces los valores actuales del vector de estado pueden cambiar en cualquier momento sin que Q los haya leído, esto ocasiona que se pierda información contenida en los vectores de estado.

Este tipo de conexiones está sujeto a la velocidad con la que se ejecute el proceso Q, lo que puede originar inconsistencias en los vectores de estado ya que Q puede ejecutar un "getsv" antes de que P haya terminado de actualizar el vector de estado correspondiente a una transacción; por este motivo se impone la restricción de que ningún proceso pueda inspeccionar el vector de estado de P a menos de que :

- P esté en el punto de una operación de lectura del buffer de mensajes, o
- P esté en el punto de una operación de escritura en el buffer de mensajes, o
- P esté en el punto de una operación de inspección el vector de estado de otro proceso.

Para implementar esta restricción el apuntador de texto de P deberá indicar los puntos en el texto en donde se ejecuten operaciones de lectura, escritura y/o inspección.

A menos de que existan limitaciones técnicas, siempre se tratará de conectar procesos de nivel 1 con procesos de nivel 0 mediante conexiones de mensajes, ya que de esta manera se asegurará que los procesos modelo reflejen el comportamiento de las entidades en la realidad.

Tanto en las conexiones a través de mensajes como a través de vectores de estado se tienen problemas debido a la velocidad con que se ejecutan los procesos; para evitar esto se pueden ir almacenando tanto los mensajes como los diferentes valores del vector de estado.

JSD considera a las bases de datos como agrupaciones de vectores de estado, en donde el contenido de cada vector de estado está determinado por las necesidades de cada proceso, más tarde durante la implementación del sistema se introduce el concepto de base de datos. Cuando dos procesos están conectados a través de vectores de estado, los contenidos de éstos serán determinados por el proceso que los inspecciona.

Quando un proceso tiene que leer diferentes tipos de mensajes provenientes de diferentes procesos, éste debe leer todos siguiendo algún procedimiento previamente establecido, JSD llama a esto "merge" de mensajes de entrada; existen 4 tipos de Merge :

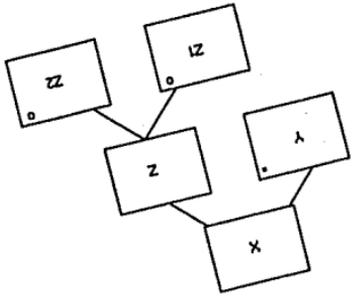
1.- Merge Fijo.- El proceso lee los mensajes de las diferentes entradas de acuerdo a reglas fijas, las cuales se determinan al momento de hacer las especificaciones del sistema.

2.- Merge Simple de Datos.- Este tipo de merge de datos consiste en que el proceso tiene que leer los mensajes de todas las entradas y entonces comenzar su ejecución. Este tipo de merge implica que el proceso quede bloqueado hasta que todos los mensajes de las diferentes entradas hayan sido generados y estén disponibles.

3.- Merge Periódico de Datos.- Este tipo de merge parte de la misma idea del merge simple, solo que el proceso no queda bloqueado hasta que todos los mensajes estén disponibles, sino que se determina un periodo de tiempo que el proceso esperará los mensajes y entonces iniciar su ejecución. Cuando se utiliza este tipo de merge se dice que se tiene un sistema síncrono.

4.- Merge Áspero.- Mediante este tipo de merge el proceso determina cuál mensaje está disponible para ser leído, chequea todos los mensajes para determinar si están vacíos o no. La "asperidad" se debe a que existe una dependencia sobre la velocidad de los procesos que generan los mensajes y sobre la velocidad de transmisión hacia los buffers. Este tipo de Merge se representa de la siguiente manera :

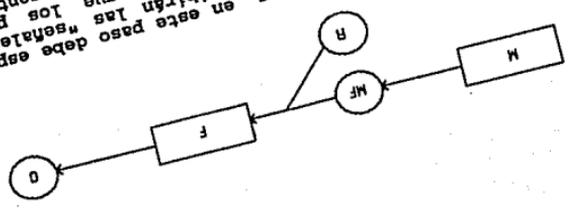
- a) -- Las hojas del diagrama estructurados se representan por sus nombres seguidos por punto y coma.
- b) -- El nodo secuencia X se representa :

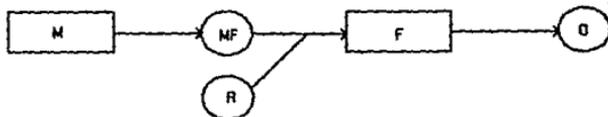


Haciendo uso de estos conceptos, en este paso debe especificarse la forma en que los procesos Modelo recibirán las "señales" o mensajes provenientes del exterior con el fin de que los procesos puedan ejecutarse; el resultado de este análisis se presenta mediante un Diagrama de Especificación del Sistema ("SSD", Specification System Diagram).

Además de elaborar el Diagrama de Especificación del Sistema debe expresarse el texto estructurado de cada uno de los procesos modelo desarrollados en el paso anterior. Debe especificarse dentro del texto estructurado la forma en que se recibirán las "señales" o mensajes provenientes del exterior, supongase que se llena el siguiente diagrama :

Con el fin de mostrar la forma de obtener el texto estructurado a partir de un diagrama de estructura, supongase que se llena el siguiente diagrama :

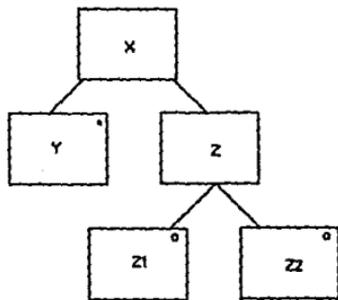




Haciendo uso de estos conceptos, en este paso debe especificarse la forma en que los Procesos Modelo recibirán las "señales" o mensajes provenientes del exterior con el fin de que los procesos puedan ejecutarse; el resultado de este análisis se presenta mediante un Diagrama de Especificación del Sistema ("SSD, Specification System Diagram").

Además de elaborar el Diagrama de Especificación del Sistema debe expresarse el texto estructurado de cada uno de los procesos modelo desarrollados en el paso anterior. Debe especificarse dentro del texto estructurado la forma en que se recibirán las "señales" o mensajes provenientes del exterior.

Con el fin de mostrar la forma de obtener el texto estructurado a partir de un diagrama de estructura, supóngase que se tiene el siguiente diagrama :



a).- Las hojas del diagrama estructurado se representan por sus nombres seguidos por punto y coma.

b).-El nodo secuencia X se representa :

X seq

.
.
.

X end

c).- El nodo iteración Y :

Y itr while (condición Z)

...
Y end

d).- La selección de dos partes Z :

Z sel

...
Z alt
...
Z end

la primera parte de la selección se especifica entre Z sel y Z alt y la segunda (la alternativa) entre Z alt y Z end.

De esta manera el texto estructurado correspondiente al diagrama anterior se puede escribir así :

X seq

Y itr

.
.
.

Y end

Z sel

.
.
.

Z alt

.
.
.

Z end

El uso de los nombres de las entidades y de las acciones durante la especificación del texto estructurado permite que se correlacione con el diagrama de estructura asociado.

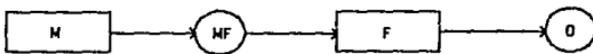
Así pues, los resultados de este paso son: la descripción de las formas de conexión entre el mundo real y los procesos modelo, los diagramas de especificación del sistema (SSD, System Specification Diagrams) y los textos estructurados de los procesos modelo.

2.4.2.4.- El Paso de las Funciones.

Con el paso anterior se obtuvo un Diagrama de Especificación del Sistema compuesto por procesos modelo, los cuales están conectados a entidades reales. Para cada proceso se tiene un diagrama de estructura así como texto estructurado. Hasta este momento no se tienen definidas, todavía, las salidas del sistema, las entradas son mensajes que son señales que indican que ha ocurrido algún evento en el mundo real.

El objetivo de este paso es el de definir las salidas del sistema. Cada función estará basada sobre el modelo y podrá especificarse en forma independiente de otras funciones. En JSD las funciones pueden ser definidas de la siguiente manera :

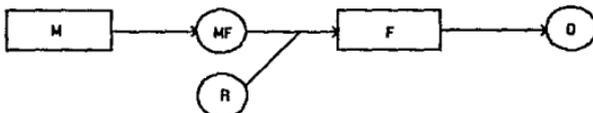
Quando se da un conjunto de circunstancias "C" se produce la salida "O" para los datos "D".



donde :

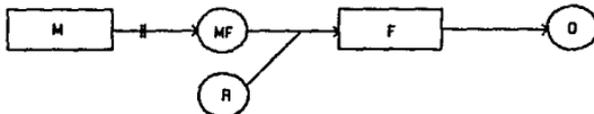
MF es un conjunto de mensajes a través de los cuales M indica que los eventos C han ocurrido.

En ocasiones el conjunto de circunstancias C no dependen del modelo M, sino que dependen de alguna ocurrencia externa R.



En este caso la función F necesita la ocurrencia de un conjunto de eventos del modelo M así como de la ocurrencia de un evento externo, para que pueda producir la salida O.

Quando una función necesita información de varios procesos modelo se indica mediante una doble barra, ejemplo :



En este caso, F necesita obtener información de varios procesos modelo M y de la ocurrencia de un evento externo para que pueda generar la salida O.

En ocasiones algunas funciones pueden incluirse fácilmente en los procesos modelo ya que resultan de la misma estructura de dichos procesos; por ejemplo si se deseara incluir en el modelo una función que llevara un contador dentro de uno de los procesos, sería muy sencillo incluirla, bastaría con agregar una línea al texto estructurado de dicho proceso; a este tipo de funciones JSD las denomina Funciones Empotradas.

El resultado de este paso es una especificación de cada una de las funciones requeridas del sistema. Esta especificación debe estar documentada en tres formas :

- a).- Una re-especificación del requerimiento informal, en términos del modelo inicial.
- b).- Una re-elaboración del modelo inicial, mostrando cómo se construyen las funciones dentro de los procesos modelo, o bien como dichas funciones se conectan con los procesos modelo.
- c).- Textos estructurados que muestren la especificación detallada de cada función.

2.4.2.5.- El Paso de Sincronización del Sistema.

En el Paso de las Funciones, el analista no considera directamente los aspectos de sincronización de las salidas de las funciones; es precisamente en este paso donde el analista debe hacerlo, ya que las restricciones de sincronización del sistema deberán ser documentadas antes del paso de implementación.

Los requerimientos de sincronización pueden ser muy variados, algunos pueden ser los siguientes :

- 1.- Algunas salidas del sistema deberán generarse en un periodo de tiempo especificado por el usuario (fijo).
- 2.- Algunos procesos modelo de nivel 1 deben ejecutar las operaciones de "getsv" (leer el vector de estado) de los procesos de nivel 0 en el mundo real, en un tiempo y frecuencia especificados.
- 3.- Algunos procesos modelo cuyos vectores de estado serán inspeccionados por otros procesos, deberán estar actualizados al momento de ser leídos.
- 4.- Los procesos en los cuales la salida de uno es la entrada de otro deberán estar perfectamente sincronizados.

5.- Las diferentes partes del sistema pueden estar sujetas a diferentes retrasos, así que tanto los procesos modelo como los procesos de las funciones pueden implementarse de tal forma que se logre su ejecución, considerando dichos retrasos.

6.- En este paso el analista considera, junto con el usuario, cuáles deberán de ser los tiempos de respuesta aceptables y cuáles serán los retrasos tolerables.

7.- En algunos casos será necesario especificar muy precisamente las relaciones de tiempo entre los diferentes procesos del sistema. Esto puede hacerse en JSD introduciendo procesos de sincronización en los diagramas de estructura SSD.

De hecho la documentación puede ser un poco informal, puede hacerse a manera de notas, las cuales serán de gran importancia en el paso de implementación.

2.4.3.- Etapa de Implementación.

Esta etapa solamente consta de un paso, el cual se detalla en el siguiente punto.

2.4.3.1.- El Paso de Implementación.

El desarrollo de la especificación se ha desarrollado bajo el supuesto de que se tiene un procesador exclusivo y dedicado para cada instancia de cada una de las entidades en el modelo. El objetivo primordial de este paso es el de transportar las especificaciones desarrolladas en la fase anterior para su ejecución en un solo procesador. La esencia de la implementación JSD consista en que la especificación del sistema sea retenida durante la implementación.

Con el fin de explicar en forma clara la forma en que se trasladarán las especificaciones para su implementación en un solo procesador y eliminar el supuesto anterior sin perder la esencia de las especificaciones se hará en dos etapas :

1.- Ejecución de todos los procesos de cada instancia en un solo procesador.

Cuando se tiene un solo procesador es esencial que exista un proceso que controle la ejecución de cada uno de los procesos incluidos en el sistema, dicho proceso controlador se denomina "Scheduler", el cual debe leer todos los mensajes de entrada del sistema y determinar qué proceso debe ejecutarse y en que momento debe suspenderse para que se ejecute otro de los procesos.

La importancia del proceso "Scheduler" es vital, ya que si algún proceso leyera los mensajes de entrada al sistema bloquearía los demás procesos en espera de que algún mensaje estuviera disponible.

Para ejecutar dos procesos en un solo procesador, debe intercalarse su ejecución, es decir, se ejecuta parte de uno de los procesos, se

suspende en un momento determinado y se inicia la ejecución del otro, la cual también será suspendida para continuar la ejecución del primero; así se continúa hasta finalizar la ejecución de ambos procesos. JSD denomina a este procedimiento inversión.

Supóngase que se tienen dos procesos A y B conectados a través de Mensajes (M) :



La ejecución de cada proceso (A y B) puede ser esquematizada de la siguiente manera :

a).- Ejecución del proceso A :



b).- Ejecución del proceso B :

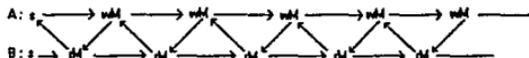


donde wM y rM indican las operaciones de "write" y "read" sobre M.

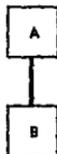
Se desea implementarlos sobre un solo procesador ejecutándose en forma intercalada; para hacer esto existen dos maneras :

a).- Iniciando con la ejecución del proceso B.

En este caso inicia el proceso B hasta alcanzar el primer "read", en ese momento inicia la ejecución de A hasta alcanzar el primer "write", entonces continúa la ejecución de B hasta alcanzar el siguiente "read"; este proceso continúa hasta terminar la ejecución intercalada de ambos procesos. Gráficamente, se tiene lo siguiente :



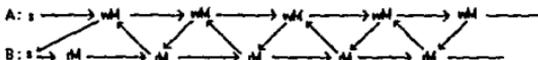
En este caso se dice que A se invierte con respecto de M y se implementa como una rutina de B, esto se representa de la siguiente manera :



La doble línea indica que el proceso B ha sido invertido.

b).- Iniciando con la ejecución del proceso A.

En este caso, se inicia el proceso con la ejecución de A hasta que alcanza el primer "write", en ese momento se inicia la ejecución de B, en este caso hasta alcanzar el segundo "write", el proceso continúa intercalando la ejecución de ambos procesos, esquemáticamente se puede visualizar así :



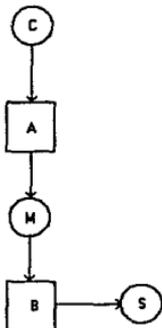
En este caso B se ha invertido con respecto de M y se ha implementado como rutina de A, análogamente la representación gráfica es la siguiente :



Así pues, con el fin de que el "Scheduler" pueda controlar la ejecución intercalada de los procesos, éstos deben invertirse e implementarse como rutinas de dicho proceso controlador. Gráficamente "Scheduler" se diferencia de los demás procesos por una línea vertical en la parte derecha del rectángulo.



Supóngase que en la fase de desarrollo de la especificación se ha obtenido el siguiente Diagrama de Estructura del Sistema :

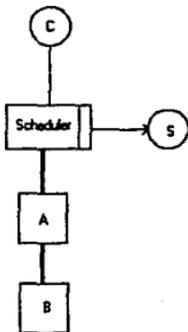


Entonces, para lograr su implementación en un solo procesador se debe hacer lo siguiente :

a).- Los procesos A y B deben ejecutarse en forma intercalada por lo cual B se invierte con respecto de M y se implementa como rutina de A.

b).- Se adiciona el proceso "Scheduler", el cual leerá los mensajes de entrada provenientes de C y de S y controlará la ejecución del proceso A. La ejecución del proceso B estará controlada por el proceso A.

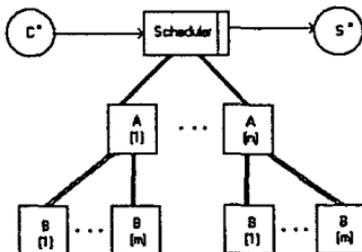
Siguiendo la notación antes mencionada, se obtiene el siguiente diagrama:



Este diagrama se denomina Diagrama de Implementación del Sistema ("SID, System Implementation Diagram"), el cual generalmente se representa junto con el Diagrama de Estructura del Sistema con el fin de que se relacionen visualmente de fácil manera.

2.- Ejecución en un solo procesador de todos los procesos de todas las instancias.

Si se representara en un solo Diagrama de Implementación la ejecución de todos los procesos para todas las instancias de cada entidad se obtendría el siguiente diagrama :

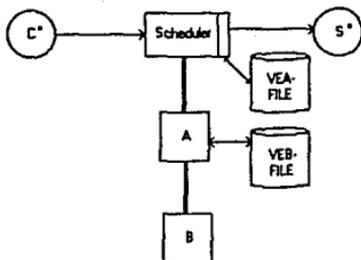


en este caso se tiene la ejecución de n procesos A, uno para cada una de las instancias, y dentro de cada proceso A se tiene la ejecución de m procesos B. C* representa el conjunto de mensajes de entrada C para cada una de las instancias, S* representa el conjunto de salidas.

Tratar de implementar esto sería bastante complicado, además de que no resultaría práctico. Para implementar este sistema no es necesario tener tanta copia de los procesos A y B corriendo para cada una de las instancias, basta con tener una copia de cada uno de ellos y con almacenar los vectores de estado en algún lugar accesible para todos los procesos.

De esta manera surge en JSD el concepto de base de datos como agrupaciones de vectores de estado para diferentes instancias de una misma entidad. La representación gráfica dentro del Diagrama de Implementación es la representación tradicional de "tambor".

Al agrupar los vectores de estado que utilizan los procesos A y B se obtiene VEA-FILE y VEB-FILE (Vectores de Estado para el proceso A y B, respectivamente), finalmente se obtiene el Diagrama Final de Implementación del Sistema, el cual se muestra en la siguiente figura :



JSD considera el proceso de implementación hasta este punto; claro que hay bastante más por hacer pero deja las bases bastante claras para que a partir de ellas se desarrolle tanto el diseño de la base de datos como la programación en forma explícita, ya que se va desarrollando poco a poco; para desarrollar los programas de cómputo deben utilizarse técnicas especializadas.

CAPÍTULO 3.- CASO PRÁCTICO DE ESTUDIO. FASE DE PLANEACIÓN DEL PROYECTO DE ENTRADAS AL ALMACÉN.

Con el propósito de mostrar las bondades de las metodologías expuestas en los capítulos 1 y 2 se ha seleccionado un caso práctico: Entradas al Almacén de una fábrica; este caso es muy sencillo y el único objetivo es mostrar paso a paso las metodologías expuestas.

En este capítulo se desarrollará la Planeación del Proyecto, y en el capítulo 4 se desarrollarán tanto la especificación como la implementación siguiendo la metodología Jackson (JSD).

3.1.- PASO 1 : DETERMINACIÓN DEL ALCANCE DEL PROYECTO.

Actividades :

1.- Investigación del Medio Ambiente.

El procedimiento de suministro de materias primas a una fábrica se efectúa de la siguiente manera :

- a).- En base a los planes de producción de la fábrica se elaboran reportes mensuales en donde se especifican las materias primas necesarias para satisfacer dichos planes durante el siguiente mes; tales requerimientos se especifican en base diaria. Estos reportes son conocidos como Reportes Mensuales de Requerimientos Diarios.
- b).- De acuerdo con los Reportes Mensuales de Requerimientos Diarios se elabora un plan de pedidos, en donde se especifica la materia prima requerida, la cantidad y la fecha de entrega.
- c).- Estos pedidos son entregados al proveedor con el fin de que los surta en la fecha especificada.
- d).- Cuando las materias primas llegan a la fábrica son recibidas en el almacén por un Almacenista , quien es responsable de la recepción.
- e).- El Almacenista checa que efectivamente la materia prima haya sido solicitada, para hacer esto debe comunicarse con el área responsable. En caso de que no haya sido solicitada para ese día, simplemente no se recibe.
- f).- En caso de que la materia prima haya sido solicitada para ese día se recibe y el Almacenista elabora un documento denominado Recibo de Entrega, en dicho documento se especifica la materia prima, la cantidad y la fecha de llegada.
- g).- Con este Recibo de Entrega la materia prima es enviada al laboratorio, con el fin de aplicar pruebas de control de calidad, si la materia prima no cumple con las normas de la fábrica, se rechaza. Si la

materia prima es aprobada, se especifica en el documento de Recibo de Entrega correspondiente.

h).- Una vez que se tiene el documento de Recibo de Entrega completo se captura la información en el Sistema de Inventarios, con el fin de registrar la entrada a la fábrica.

i).- Una copia del Recibo de Entrega es entregada al Proveedor con el fin de que pueda reclamar su pago.

2.- Establecimiento del Alcance del Proyecto.

Se pretende que el proyecto ayude agilizar algunas de las tareas manuales del Almacenista.

El proyecto abarcará las siguientes áreas :

- a).- La parte de pedidos, ayudará a llevar un control de las materias primas que han sido solicitadas a los proveedores.
- b).- Ayudará al Almacenista a validar si las materias primas han sido solicitadas o no.
- c).- Se hará el registro automático en el Sistema de Inventarios de las materias primas que han sido aceptadas por el laboratorio.

No se incluye la parte de pago a proveedores.

3.- Establecimiento de los Principales Objetivos y Requerimientos.

Objetivos del Proyecto :

- a).- Contribuir a mantener los inventarios en tiempo real en lo referente a entradas de materia prima a la fábrica.
- b).- Contribuir a reducir los costos ocasionados el registro tardío del ingreso de las materias primas al almacén.
- c).- Desarrollar un Sistema que se apegue a la realidad del usuario.

Requerimientos :

Los requerimientos de los usuarios se presentan en la siguiente tabla :

REQUERIMIENTO
Registrar en inventarios todas las materias primas en cuanto se reciban en la fábrica.
Agilizar el trabajo del Almacenista eliminando procesos manuales.
Desarrollar un sistema que se apegue al ambiente real de trabajo del usuario.
Desarrollar un sistema que sea fácil de mantener.

4.- Determinación de Restricciones.

Las principales restricciones de este proyecto son :

a).- De la Empresa :

- Como política de la Empresa no se permite recibir materias primas que no hayan sido solicitadas para el día en que llegan a la fábrica.

b).- Técnicas :

- Las computadoras que se utilicen deberán ser VECTRAS, ya que es el estándar de la Empresa.

5.- Presentación de Alternativas de Solución.

De acuerdo con los Objetivos del Proyecto, el alcance, los requerimientos y las restricciones del mismo, se propone la siguiente solución general :

a).- Una vez que se tengan los Reportes Mensuales de Requerimientos Diarios se pueden registrar en el sistema, especificando las materias primas, cantidades y fechas de entrega.

b).- Cuando llegue la materia prima al almacén el Almacenista checará en el sistema si la materia prima ha sido solicitado para ese día.

c).- En caso de que se haya solicitado la materia prima para ese día, el Almacenista podrá capturar en el sistema el Recibo de Entrega.

d).- El Laboratorio checará en el sistema cuáles son las materias primas que deberá analizar, sus resultados los podrá especificar directamente en el sistema.

e).- Cuando el laboratorio determine que una materia prima ha pasado las pruebas de control de calidad, lo especificará en el sistema y en ese momento se registrará en el Sistema de Inventarios.

f).- Este sistema registrará la información referente a Recibos de Entrega al Sistema de Pagos para que se elabore el pago.

3.2.- PASO 2 : ELABORACIÓN DE UN PLAN DE TRABAJO PARA EL PROYECTO.

Actividades :

1.- Determinar las Actividades del Proyecto.

Este proyecto será desarrollado básicamente siguiendo la metodología JSD para el desarrollo tanto de la especificación como de la implementación. Se ha adicionado la parte de Planeación del Proyecto debido a que se considera indispensable utilizar una metodología desde el momento en que se decide invertir recursos para la consecución de objetivos específicos, es decir, desde el momento en que surge un proyecto.

JSD permite que se utilicen otras técnicas, así que al adicionar la parte de Planeación del Proyecto se proporcionan las herramientas administrativas necesarias para contribuir a alcanzar los objetivos del proyecto.

Se ha decidido utilizar JSD debido principalmente a los siguientes factores :

- JSD modela la realidad del usuario y su objetivo es el asegurar que el sistema refleje el mundo real, el cual es un objetivo también de este proyecto.
- JSD no comienza el ciclo de desarrollo a partir de las especificaciones del sistema, sino que el desarrollo de ellas es parte de su ciclo.
- Para JSD es importante el orden en el que suceden los eventos y este proyecto tiene esa característica.

Para JSD existen básicamente dos etapas :

- a).- Desarrollo de la Especificación y
- b).- Desarrollo de la Implementación.

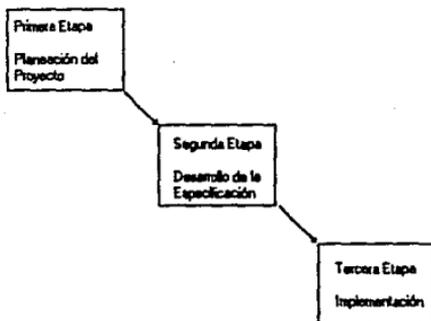
Para la primera etapa es necesario desarrollar las siguientes actividades:

- a).- Paso de Entidades y Acciones
- b).- Paso de Estructura de Entidades
- c).- Paso de Modelado Inicial
- d).- Paso de Funciones
- e).- Paso de Sincronización del Sistema

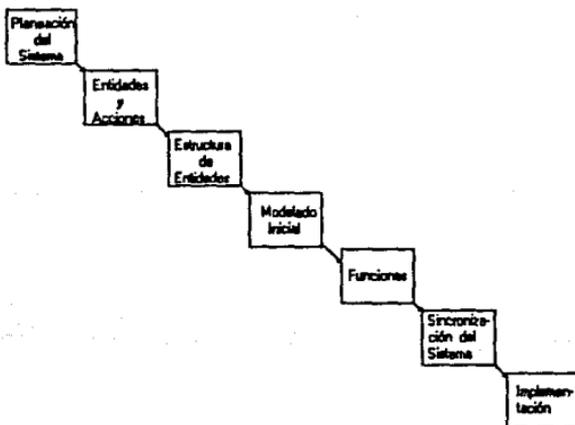
En la etapa de Implementación solo es necesario desarrollar una sola actividad :

- a).- Paso de Implementación.

Así pues el proyecto tendrá tres etapas :



Las actividades serán secuenciales, tal y como se muestran en la siguiente figura :



2.- Elaborar Estimaciones de Duración de Cada Actividad.

Para hacer la estimación de los esfuerzos requeridos, se hace una estimación para cada una de las actividades de cada etapa. La etapa de Planeación del Proyecto no se incluye debido a que terminará en el

momento en que se presente el Plan del Proyecto. Las primeras estimaciones se hicieron en base a la experiencia y se presentan en la siguiente tabla :

ETAPA	ACTIVIDAD	ESFUERZO ESTIMADO
Desarrollo de la Especificación	Entidades y Acciones	5 días/hombre
	Estructura de Entidades	3 días/hombre
	Modelado Inicial	5 días/hombre
	Funciones	5 días/hombre
	Sincronización del Sistema	5 días/hombre
Desarrollo de la Implementación	Implementación	30 días/hombre

Como se mencionó en el Capítulo 1, Esterling afirma que el tiempo productivo se ve afectado por una serie de factores que dependen de la interacción de los miembros del equipo de trabajo con su entorno, afirma que para estimar el tiempo de trabajo útil por jornada de trabajo por persona se puede estimar mediante la siguiente fórmula :

$$w = 0.125 \times (8 - 8a + g - 4r/60 - p(t+r)/60 - k(t+r)/60)$$

donde :

- a : Fracción media del trabajo diario gastado en trabajos administrativos o indirectos (0-0.5).
- t : Duración media de las interrupciones del trabajo, en minutos (1-20).
- r : Tiempo de recuperación medio después de la interrupción, en minutos (5-10).
- k : Número de interrupciones por día de trabajo de la gente que participa directamente en el proyecto (1-10).
- p : Número de interrupciones por día de trabajo por otras causas (1-10).
- g : Número promedio de horas extras por jornada de trabajo

Así pues considerando que con el equipo de trabajo y en el entorno en el cual se desarrollará el proyecto se tendrán los siguientes valores :

- a = En promedio 1/2 de las 8 horas de trabajo son utilizadas en tareas administrativas, esto representa el 25 %.
- t = Se cree que el tiempo medio de interrupciones será de 15 minutos.

r = El tiempo de recuperación medio después de cada interrupción se estima en 2 minutos.

k = Se cree que el promedio de interrupciones diarias para una persona serán 3.

p = Se estima que habrá dos interrupciones por otras causas.

Suponiendo además que no se trabajarán horas extras y que el equipo de trabajo estará integrado por 7 personas, al aplicar la fórmula de Esterling se obtiene que el tiempo de trabajo útil por jornada de trabajo es el siguiente :

$$w = 80 \%$$

Esto significa que de las 8 horas de trabajo diario, solamente se aprovecha el 80 %, entonces el esfuerzo estimado de cada actividad debe incrementarse en un 25 % obteniendo la siguiente tabla :

ETAPA	ACTIVIDAD	ESFUERZO ESTIMADO	ESFUERZO AJUSTADO
Desarrollo de la Especificación	Entidades y Acciones	5 días/hombre	6.25 días
	Estructura de Entidades	3 días/hombre	3.75 días
	Modelado Inicial	5 días/hombre	6.25 días
	Funciones	5 días/hombre	6.25 días
	Sincronización del Sistema	5 días/hombre	6.25 días
TOTAL ETAPA			28.75 días
Desarrollo de la Implementación	Implementación	30 días/hombre	37.5 días
TOTAL ETAPA			37.5 días

3.- Asignación de Recursos.

En este caso el proyecto no tiene una fecha de entrega determinada, se considera que es un proyecto importante, así es que la duración del proyecto estará sujeta a la disponibilidad de recursos. La disponibilidad de los recursos es la siguiente : Una persona estará trabajando en este proyecto el 90% de su tiempo, en la primera etapa. En la etapa de implementación hay dos recursos disponibles, uno dedicará el 90 % de su tiempo, mientras que el otro solamente le dedicará el 50 %. Los resultados de estas disponibilidades se muestran en la siguiente tabla :

ACTIVIDAD	ESFUERZO AJUSTADO	RECURSOS ASIGNADOS	DISPONIBILIDAD	DURACIÓN ACTIVIDAD
Entidades y Acciones	6.25 días	1	90 %	6.875 días
Estructura de Entidades	3.75 días	1	90 %	4.125 días
Modelado Inicial	6.25 días	1	90 %	6.875 días
Funciones	6.25 días	1	90 %	6.875 días
Sincronización	6.25 días	1	90 %	6.875 días
DURACIÓN DE LA ETAPA				31.625 días
Implementación	37.5 días	1	90 %	
		1	50 %	26.875 días
DURACIÓN DE LA ETAPA				26.875 días

Para el desarrollo de este proyecto prácticamente no se tienen altos riesgos, lo único que podría retardar un poco es el uso de las metodologías, ya que serán utilizadas por primera vez.

Así pues se ha determinado que para las actividades de la etapa de desarrollo de la especificación se incrementa su duración en un 10 % y para la etapa de implementación se haga en un 15 %.

Considerando además que durante el año se tienen las siguientes interrupciones :

- 2 semanas de vacaciones
 - 2 semanas por días festivos
 - 1 semana por enfermedad
 - 2 semanas por entrenamiento
- 7 semanas no efectivas

Se tiene un porcentaje de 15 % (52/45 - 1) de tiempo no efectivo durante el año. Haciendo ambas consideraciones, la duración final de cada actividad se especifica en la siguiente tabla :

ACTIVIDAD	DURACIÓN (días)	DURACIÓN EN SEMANAS	AJUSTE POR RIESGOS	AJUSTE POR TIEMPO NO-EFECTIVO	DURACIÓN FINAL
Entidades y Acciones	6.875	1.375	10 %	15 %	2 semanas
Estructura de Entidades	4.125	0.825	10 %	15 %	1 semana
Modelado Inicial	6.875	1.375	10 %	15 %	2 semanas
Funciones	6.875	1.375	10 %	15 %	2 semanas
Sincronización del Sistema	6.875	1.375	10 %	15 %	2 semanas
TOTAL ETAPA					9 SEMANAS
Implementación	26.875	5.375	15 %	15 %	7 SEMANAS
TOTAL ETAPA					7 SEMANAS

4.- Elaborar la Calendarización del Proyecto.

Con base a la tabla anterior y considerando que deben incluirse en el proyecto actividades enfocadas a la revisión y evaluación del desarrollo así como actividades propias de la administración del proyecto.

En este caso se considera que deben incluirse puntos de revisión al final de cada etapa.

De esta manera, la calendarización final del proyecto se muestra en la siguiente gráfica :

ACTIVIDAD	5	6	7	8	9	10	11	12	13	14	15	...	21	22	Duración
DESARROLLO DE LA ESPECIFICACIÓN															Total
Entidades y Acciones	X	X													2 Semanas
Estructura de Entidades			X												1 Semana
Modelado Inicial				X	X										2 Semanas
Funciones					X	X									2 Semanas
Sincronización							X	X							2 Semanas
REVISIÓN										X					1 Semana
DESARROLLO DE LA IMPLEMENTACIÓN															
Implementación											X	...	X		7 Semanas
REVISIÓN														X	1 Semana

3.3.- PASO 3.- ORGANIZACIÓN DEL EQUIPO DE TRABAJO.

ACTIVIDADES :

1.- Integración del Equipo de Trabajo.

- a).- Líder del Proyecto : En este caso el líder del proyecto es la persona representante de los usuarios, es quién sugirió el proyecto.
- b).- Usuarios : Los usuarios se han comprometido a dedicar una persona de tiempo completo durante la etapa de Planeación del Proyecto, y a dedicar un 30 % del tiempo de dos personas durante las siguientes etapas, con el compromiso de que en caso de ser necesario, se dedicará un poco más de tiempo.
- c).- Sistemas : Por parte de Sistemas se han dedicado dos recursos :

- uno, con disponibilidad del 90 % para la etapa de Desarrollo de la Especificación.
- dos para la etapa de Implementación, uno con disponibilidad del 90 % y otro del 50 %.

2.- Identificación de Especialistas.

En cuestión técnica se han identificado los siguientes especialistas:

- Especialista del Sistema de Inventarios
- Distribuidor de Equipo necesario para el desarrollo del proyecto.

En lo que se refiere al área del usuario, los especialistas son los Almacenistas así como la persona responsable de mantener los inventarios.

3.- Identificación de Recursos.

a).- Recursos Humanos.

Ya fueron identificados en el paso anterior.

b).- Recursos de Hardware.

En base al alcance del proyecto se requiere :

2 VECTRAS RX
3 Impresoras
2 Reguladores

Ya que no se cuenta con suficientes recursos como para tener dos computadoras, una para desarrollo y otra para producción, se crearán dos ambientes de trabajo en la misma computadora.

c).- Recursos de Software.

El desarrollo se hará internamente, ya se tienen compiladores, editores y todo lo necesario para el desarrollo del proyecto.

4.- Establecimiento de Procedimientos para la administración del Proyecto.

En este caso, como el equipo de trabajo no es muy grande, no existen grandes problemas de comunicación, sin embargo es necesario establecer los siguientes acuerdos :

- a).- Los documentos generados serán de común acuerdo entre los usuarios y el equipo de sistemas.
- b).- Al final de cada etapa se hará una revisión con todo el equipo de trabajo.

c).- Cualquier cambio después de la aprobación requerirá ser sometido a juicio del Equipo de Trabajo y de ser aprobado se hará una reprogramación de actividades.

5.- Desarrollar Planes de Trabajo Individuales.

En base a la Calendarización del Proyecto se elaborarán planes individuales de trabajo, en este caso se omiten por considerarse de poco interés para la presentación de este trabajo. Esta actividad no requerirá gran esfuerzo en este proyecto, sin embargo se debe mantener informado a todo el equipo de trabajo de la situación del proyecto.

3.4.- PASO 4 : ANÁLISIS COSTO/BENEFICIO.

Actividades :

1.- Establecimiento de costos de Personal.

a).- Costos de Desarrollo :

	NÚMERO DE PERSONAS	SUELDO MENSUAL	TIEMPO (SEMANAS)	DISP.	TOTAL
EQUIPO DE DESARROLLO	1	\$3'000,000	22	90 %	\$14'850,000
	1	\$3'500,000	8	50 %	\$ 3'500,000
ESPECIALISTAS	1				\$ 3'000,000
	1				\$ 2'000,000
USUARIOS	1	\$3'000,000	10	30 %	\$ 2'250,000
	1	\$3'000,000	8	10 %	\$ 600,000
TOTAL					\$26'200,000

b).- Costos de Operación.

En este caso no se consideran gastos de personal adicionales durante la operación, ya que este sistema se lleva actualmente y el personal asignado actualmente operará el nuevo sistema.

2.- Establecimiento de costos operativos y de desarrollo.

Los costos de Hardware son los siguientes :

VECTRAS RX	\$18'000,000	x 2	=	36'000,000
Impresoras	\$ 3'600,000	x 3	=	10'800,000
Reguladores	\$ 1'000,000	x 2	=	2'000,000
TOTAL				\$48'800,000

El desarrollo del software se hará todo internamente, por lo cual el costo del desarrollo ya está incluido en los costos de personal.

Los costos operativos son gastos que se tendrán serán periódicos ya que se tendrán que desembolsar en forma mensual durante la operación del sistema :

Papel	\$ 200,000 x 12 =	2'400,000 anuales
Consumibles	\$ 150,000 x 12 =	1'800,000 anuales
Contrato de Servicio VECTRAS RX		7'200,000 anuales
TOTAL		\$11'400,000 anuales

3.- Establecimiento de costos de Reemplazamiento.

Al reemplazar el sistema actual por el nuevo se tendrán los siguientes costos de reemplazamiento :

- El sueldo de 2 secretarias que apoyan la labor administrativa del almacenista. El ahorro será de \$ 5'000,000 mensuales.
- 2 escritorios con valor de \$1'500,000.

4.- Cuantificación de Beneficios.

a).- Beneficios directos tangibles.

Uno de los objetivos de este proyecto es el de disminuir los costos por no tener los inventarios actualizados. El departamento contable determinó que si los inventarios estuvieran actualizados al día, se tendrían ahorros por \$ 30'000,000 mensuales.

b).- Costos en que se incurrirían si el sistema no fuera desarrollado.

Estos costos serían los considerados en el inciso anterior.

Considerando estos costos y beneficios se obtiene la siguiente tabla de flujo de efectivo :

AÑO	COSTO	BENEFICIO	BENEFICIO ANUAL NETO	ACUMULADO
1	79'900	30'000	-49'900	-49'900
2	6'400	30'000	23'600	-26'300
3	6'400	30'000	23'600	-2'700
4	6'400	30'000	23'600	20'900
5	6'400	30'000	23'600	44'500
6	6'400	30'000	23'600	68'100
7	6'400	30'000	23'600	91'700
TOTAL	118'300	270'000		91'700

De esta tabla de flujo de efectivo se puede decir que la inversión se recuperará en 3.2 años

3.5.- Paso 5 : ELABORACIÓN DEL PLAN DEL PROYECTO.

Como ya se mencionó anteriormente el objetivo de este paso es integrar un documento en el cual se presenten los resultados obtenidos en este paso a los Directivos de la Empresa con el fin de obtener su aprobación para el Desarrollo del Proyecto. El formato del documento se describió en el Capítulo 1.

En este caso se omite la presentación del documento debido a que se considera innecesario, ya que toda la información se obtiene en los pasos anteriores, en este paso solo es el formateo del documento.

CAPÍTULO 4.- METODOLOGÍA JACKSON APLICADA AL SISTEMA DE ENTRADAS AL ALMACEN.

4.1.-ETAPA DE DESARROLLO DE LA ESPECIFICACIÓN.

4.1.1.- PASO 1.- PASO DE ENTIDADES Y ACCIONES.

De acuerdo con la descripción del ambiente de trabajo del usuario se listan las entidades candidatos a ser incluidas en el sistema, haciendo una breve descripción e indicando cuáles serán seleccionadas.

Materia Prima.- Materia prima necesaria para la elaboración de producto terminado, esta entidad es relevante ya que en torno a ella gira el proyecto; se elige.

Fábrica.- Se refiere al lugar en donde se elaboran productos terminados y a la cual entra materia prima y salen productos terminados, no es relevante ya que no se hará ningún análisis sobre la fábrica como tal.

Planes de Producción.- Se refiere a la cantidad de productos a producir de cada tipo en un período de tiempo determinado, queda fuera debido a que no es parte del alcance del proyecto.

Producción.- Se refiere a los planes de producción, queda fuera.

Proveedor.- Es a quien se le solicitan las materias primas y quien se compromete a entregarlas en una fecha determinada, pudiera quedar dentro de los límites del proyecto ya que es deseable levantar los pedidos a los proveedores que ofrezcan mejores precios y tengan mejores tiempos de entrega, pero como no se automatizará ninguna de sus acciones, queda fuera.

Almacén.- Es el lugar en donde se efectúa la operación de recepción de materias primas, ahí el almacenista elabora el Recibo de Entrega; queda fuera debido a que se automatizarán solo algunas de las funciones que ahí se desarrollan.

Almacenista.- Es el responsable de la recepción de las materias primas, verifica que se trate de una entrega programada y elabora el Recibo de Entrega; finalmente registra el documento en el Sistema de Inventarios; está dentro de los límites del sistema ya que se tratará de automatizar algunas de las acciones que él realiza.

Recibo de Entrega.- Es el documento en el cual se especifica la materia prima, el proveedor, la fecha de llegada, el número de pedido, y la cantidad recibida. Este documento es la prueba de que la materia prima ha sido entregada por el proveedor, y en base a él se reclama el pago; es un documento que no puede ser tratado como una entidad ya que mas bien una entidad de salida, queda fuera.

Inventarios.- Relación de las cantidades de cada producto o materia prima que posee la fábrica en su almacén, queda fuera.

Laboratorio.- Es la entidad responsable de analizar las materias primas que llegan a la fábrica y decidir si se aceptan o se rechazan, es el

área de Control de Calidad, tentativamente queda incluido dentro de los límites del sistema.

Sistema de Inventarios.- Es el sistema actual de inventarios en donde están registradas las cantidades en existencia de cada Materia Prima, queda fuera.

Sistema de Pagos.- Es el sistema mediante el cual se efectúan los pagos a los proveedores, queda fuera.

Así pues, las entidades seleccionadas son :

Materia Prima,
Almacenista, y
Laboratorio

Enseguida se describen las actividades que ejecutan o que son ejecutadas sobre estas entidades.

Materia Prima :

- a).- Se solicita.- Se refiere a la acción de colocar un pedido con un proveedor determinado.
- b).- Llega a la Fábrica
- c).- Control de Pedidos.- Cuando llega una materia prima a la fábrica se debe checar si realmente se ha solicitado o no.
- d).- Se registra la entrada.
- e).- Control de Calidad.- La materia prima debe ser analizada bajo ciertos criterios de control de calidad.
- f).- Se Consume
- g).- Se Paga.

Almacenista :

- a).- Se regresa una Materia Prima.- Cuando la materia prima no ha sido solicitada para esa fecha, se regresa al proveedor.
- b).- Se acepta una materia prima.

Laboratorio :

- a).- Se rechaza una Materia Prima.- Cuando una materia prima no cumple con las normas de Control de Calidad se rechaza.
- b).- Se acepta una Materia Prima.

4.1.2.-PASO 2.- PASO DE ESTRUCTURA DE ENTIDADES.

En base a los resultados obtenidos en el paso anterior se construyen los diagramas de estructura. La figura 1 muestra la secuencia en que cronológicamente son ejecutadas las acciones para cada materia prima, se muestra todo el ciclo de vida, aunque no necesariamente todas las acciones serán incluidas dentro de los límites del sistema.

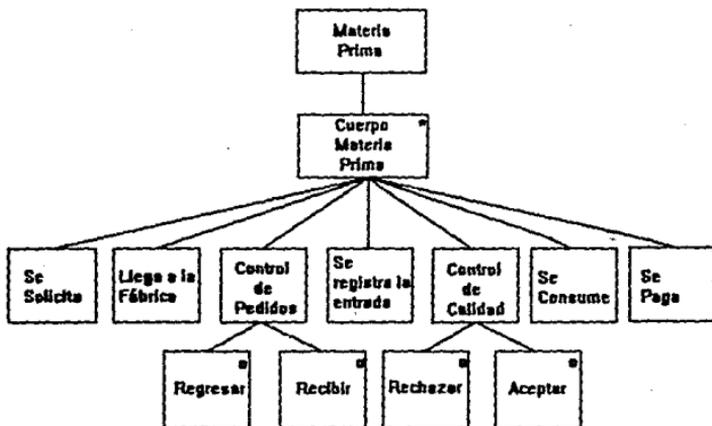


Figura 1.- Diagrama de Estructura de la Entidad Materia Prima

En la figura 2 se muestra el Diagrama de estructura de la Entidad Almacenista, sus actividades son :

- a).- Regresar una cierta materia prima "a", o bien
- b).- Recibir otra materia prima "b".

donde las entregas de la materia prima "a" y de la materia prima "b" son diferentes, es decir las actividades diarias del Almacenista consisten en recibir varias materias primas y en regresar otras. No se puede dar el caso de que reciba y regrese la misma entrega.

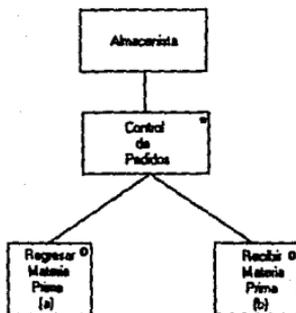


Figura 2.- Diagrama de Estructura de la Entidad Almacenista

La estructura Almacenista tiene asociada una Estructura Marsupial, la cual emerge de ella, esta estructura (Al-Mat-Prima) se muestra en la figura 3. Cuando llega una entrega de materia prima el almacenista solamente tiene dos opciones : recibirla o regresarla. Como puede observarse esta estructura es un caso particular de la estructura Almacenista.

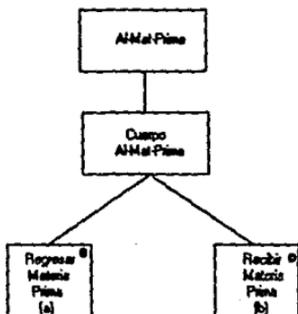


Figura 3.- Estructura Marsupial de Almacenista Caseta (Al-Mat-Prima).

Otra de las entidades seleccionadas fue Laboratorio, esta entidad es la responsable del control de calidad, sus actividades son :

- a).- Rechazar cierta materia prima "a", o bien
- b).- Aceptar otra materia prima "b".

El diagrama de Estructura es análogo al de la Entidad Almacenista, se muestra en la figura 4.

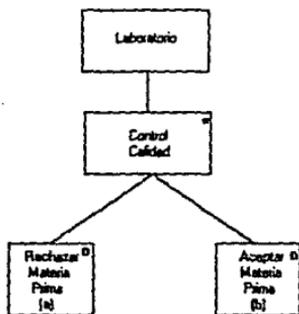


Figura 4.- Diagrama de Estructura de la Entidad Laboratorio

En forma análoga a la Entidad Almacenista, esta estructura tiene asociada una Estructura Marsupial (Lab-Mat-Prima), es decir, para una materia prima específica, el Laboratorio solo tiene dos opciones : la rechaza o la acepta. El Diagrama de Estructura para la Entidad Marsupial Lab-Mat-Prima se muestra en la figura 5.

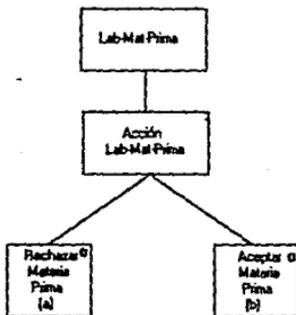


Figura 5.- Estructura Marsupial de la Estructura Laboratorio (Lab-Mat-Prima)

Como puede observarse el Diagrama de Estructura de Materia Prima (figura 1) contiene intercaladas las estructuras marsupiales Al-Mat-Prima y Lab-Mat-Prima, por lo que no es necesario considerarlas por separado.

Por otro lado, al Laboratorio realmente no se va a automatizar ninguna de sus acciones, por lo cual se dejará fuera de los límites del sistema; originalmente se incluyó con el fin de mostrar que el proceso de desarrollo de la especificación es un proceso de refinamiento.

De esta manera quedan solamente dos entidades dentro de los límites del sistemas, ellas son : Materia Prima y Almacenista.

El siguiente paso es quitar de los Diagramas de Estructura, de cada Entidad seleccionada, las acciones que no serán automatizadas, es decir, que no entrarán dentro del alcance del sistema.

Al quitar las acciones que no entran dentro del alcance del sistema se obtiene el Diagrama de Estructura de la Entidad Materia Prima mostrado en la figura 6.

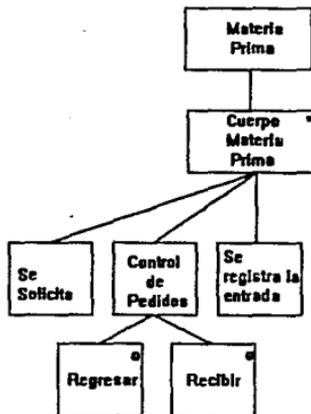


Figura 6.- Diagrama de Estructura de la Entidad Materia Prima

Este diagrama puede refinarse un poco más, ya que el proceso de registro de la entrada sólo se ejecuta cuando la Materia Prima ha sido recibida, por lo tanto en el diagrama debe indicarse esto tal y como se muestra en la figura 7.

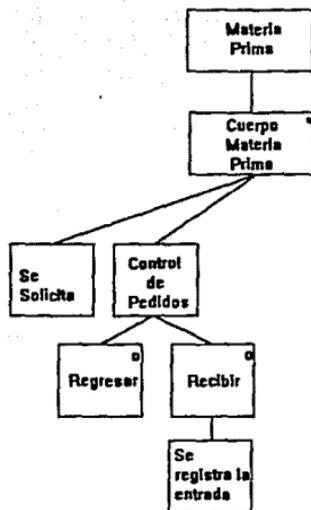


Figura 7.- Diagrama de Estructura de la Entidad Materia Prima

Es importante checar, una vez que se tengan los Diagramas de Estructura de Entidades cada una de las reglas de simplificación enunciadas en el Capítulo 2, sección 2.4.2.2 ya que si no se simplifican dicho diagramas, cuando exista la posibilidad de hacerlo, se "arrastrarán" procesos innecesarios hasta la fase de implementación.

Con el fin de simplificar el Diagrama de Estructura de la Entidad Materia Prima mostrado en la Figura 7, se puede observar lo siguiente :

En este caso se tiene una selección (Recibir) que tiene solamente una parte, en este caso se puede utilizar la regla 3 enunciada en el Paso 3 de Estructura de Entidades del Capítulo 2, que dice :

"Para simplificar un Diagrama de Estructura se pueden reemplazar las selecciones que contengan sólo una parte, por esta parte".

De esta manera se obtiene el Diagrama Simplificado de Estructura de la Entidad Materia Prima (figura 8), el cual es equivalente al que aparece en la figura 7, pero sin embargo éste es mas sencillo.

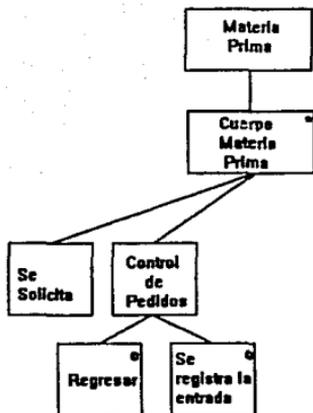


Figura 8.- Diagrama Simplificado de Estructura de la Entidad Materia Prima

De esta manera, el resultado final de este paso son los Diagramas de Estructura de las Entidades Materia Prima (figura 8) y Almacenista (figura 2).

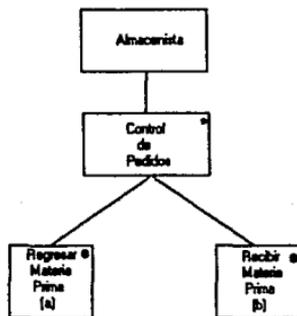


Figura 2.- Diagrama de Estructura de la Entidad Almacenista

4.1.3.- PASO 3.- PASO DE MODELADO INICIAL

El objetivo de este paso es el determinar la forma en que los Procesos Modelo recibirán información del mundo real, para su ejecución.

a).- Materia Prima :

Para que el proceso modelo Mat-Prima-1 inicie su ejecución será necesario que los responsables de solicitar las Materias Primas indiquen que desean solicitar una Materia Prima, cuando ocurra esto en el mundo real, se iniciará la ejecución del proceso "Se Solicita". Así pues la forma en que se hará la conexión será a través de mensajes, cada mensaje indicará que ha ocurrido alguna acción en el mundo real. La representación en forma de diagrama de esta conexión se muestra en la figura 9.



Figura 9.- Diagrama de especificación de Materia Prima

b).-Almacenista :

El proceso modelo Almacenista-1 iniciará su ejecución en el momento en que una Materia Prima llegue a la fábrica, por lo tanto la conexión con el mundo real será a través de mensajes, en donde cada mensaje indicará al proceso modelo que ha llegado una Materia Prima. La conexión se muestra en la figura 10.



Figura 10.- Diagrama de Especificación de Almacenista

Cuando llega una Materia Prima, el Almacenista debe validar si la Materia Prima realmente ha sido solicitada, para hacer esto es necesario hacer una inspección del vector de estado para esa Materia Prima, de esta manera se obtiene el Diagrama de Especificación del Sistema mostrado en la figura 11.

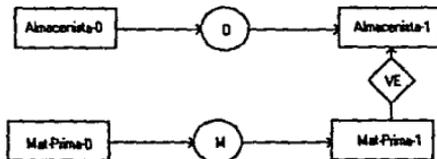


Figura 11.- Diagrama de Especificación del Sistema (SSD)

El Texto Estructurado para cada una de las entidades es el siguiente :

a).- Materia Prima :

```
Mat-Prima-1- seq
  leer M
  Cuerpo-Mat-Prima-1
  Se Solicita; leer M;
  Control-Pedidos sel (regresar)
    Regresar;
    leer M;
  Control-Pedidos alt (aceptar)
    Aceptar;
    leer M;
  Control-Pedidos end;
  Se Registra Entrada;
  leer M;
Mat-Prima-1 end.
```

b).- Almacenista :

```
Almacenista-1 seq
  leer O;
  getsv (VE);
  Control-Pedidos sel (regresar)
    Regresar;
    leer O;
  Control-Pedidos alt (aceptar)
    Aceptar;
    leer O;
  Control-Pedidos end;
Almacenista end.
```

4.1.4.- PASE 4.- PASE DE LAS FUNCIONES.

Las funciones definidas con el usuario son las siguientes:

- 1.- Notificación por escrito al Proveedor de que una entrega de Materia Prima ha sido regresado, cuando éste sea el caso (F1).
- 2.- Elaboración de un documento escrito que haga constar que se ha recibido la Materia Prima (F2).
- 3.- Generar una lista de las entregas de Materia Prima recibidas por mes (F3).
- 4.- Determinar las cargas de trabajo de los diferentes Almacenistas en base a las entregas de Materia Prima que reciben o que regresan (F4).

Función 1.- Notificación por escrito al Proveedor de que una entrega de Materia Prima ha sido regresada, cuando éste sea el caso.

Se trata de una función empotrada ya que se puede incluir fácilmente dentro del diagrama de estructura de la entidad Materia Prima. El Diagrama de estructura se presenta en la siguiente figura :

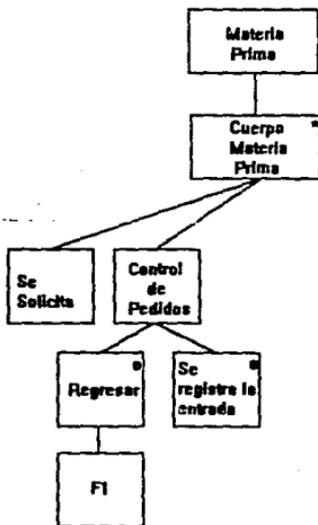


Figura 12.- Inclusión de la Función FI

El texto estructurado queda de la siguiente manera :

```

Mat-Prima-1- seq
  leer M
  Cuerpo-Mat-Prima-1
  Se Solicita; leer M;
  Control-Pedidos sel (regresar)
    Regresar;
    imprimir encabezado;
    imprimir líneas detalle;
    leer M;
  Control-Pedidos alt (aceptar)
    Aceptar; leer M;
  Control-Pedidos end;
  Se Registra Entrada; leer M;
Mat-Prima-1 end.
  
```

Función 2.- Elaboración de un documento escrito que haga constar que se ha recibido la Materia Prima.

Al igual que la función 1, se trata de una función empotrada, el diagrama de estructura queda así:

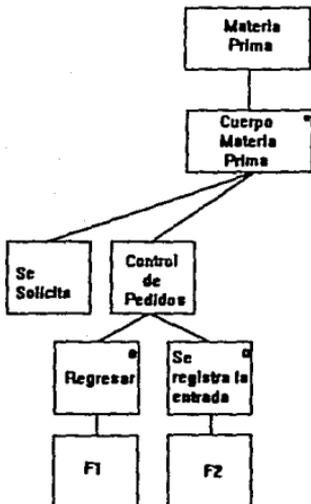


Figura 13.- Inclusión de la Función F2

Y el texto estructurado es el siguiente:

```

Mat-Prima-1- seq
  leer M
  Cuerpo-Mat-Prima-1
  Se Solicita; leer M;
  Control-Pedidos sel (regresar)
    Regresar;
    imprimir encabezado;
    imprimir líneas detalle;
    leer M;
  Control-Pedidos alt (aceptar)
    Aceptar;
    imprimir encabezado;
    imprimir líneas detalle;
    leer M;
  Control-Pedidos end;
  Se Registra Entrada; leer M;
Mat-Prima-1 end.
  
```

Función 3.- Generar una lista de las entregas de Materia Prima recibidas por mes.

Esta función es incompatible con la estructura de Materia Prima, por lo que es necesario agregar un proceso de funciones (Rep-Men-Mat) el cual tiene como entradas :

- a).- Varios mensajes provenientes de la entidad Mat-Prima-1, y
- b).- Un requerimiento externo(RM) el cual determinará el momento en el que se deberá generar el reporte, en este caso tiene una periodicidad mensual.

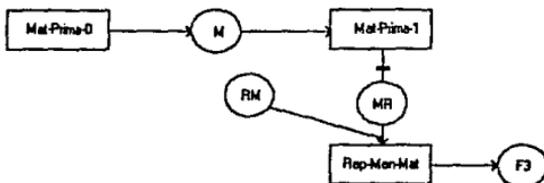


Figura 14.- Inclusión de la Función F3

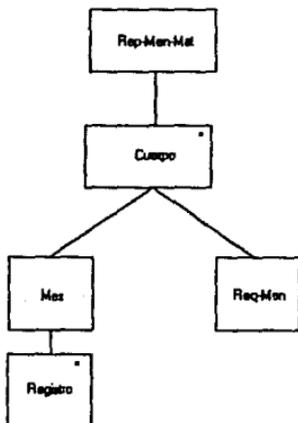


Figura 15.- Diagrama de Estructura de la función Rep-Men-Mat

Rep-Men-Mat tendrá que hacer un Merge Áspero para unir los mensajes contenidos en MR y el requerimiento externo contenido en RM, como ya se mencionó anteriormente existe la posibilidad que llegue un mensaje MR y

uno de RM casi al mismo tiempo y que por ello el proceso función considera un mensaje MR adicional, en este caso no se considera tan importante que en el reporte del mes registrara una entrega adicional, ya que ésta no se contabilizará el mes siguiente.

El Diagrama de estructura de Rep-Men-Mat se muestra en la figura 15, su texto estructurado es el siguiente :

```

Rep-Men-Mat seq
  mes:=1;
  contador:=0;
  leer MR&RM;
  Cuerpo-Rep-Men-Mat itr
    Mes seq
      cont-mes:=0;
      Cuerpo-Mes itr while (Registro)
        Registro seq
          cont-mes:=cont-mes +1;
          contador:=contador + 1;
          leer NR&RM;
        Registro end
      Cuerpo-Mes end
    Req-Men seq
      Imprime Encabezado;
      Imprime Detalles;
    Req-Men end
    mes:=mes + 1;
  Mes end
  Cuerpo-Rep-Men-Mat end
Rep-Men-Mat end
  
```

Función 4.- Determinar las cargas de trabajo de los diferentes Almacenistas en base a las entregas de Materia Prima que reciben o que regresan.

Como cada entidad Almacenista-1 modela el comportamiento de un solo Almacenista de caseta en el mundo real, entonces el Diagrama de especificación del Sistema para esta parte queda de la siguiente manera:

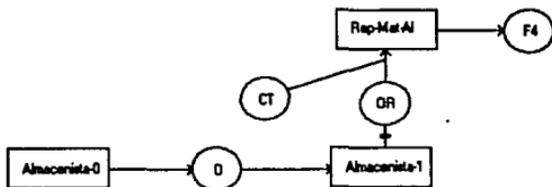


Figura 16.- Inclusión de la Función F4

El Diagrama de Estructura para Rep-Mat-Al es el siguiente (figura 17):

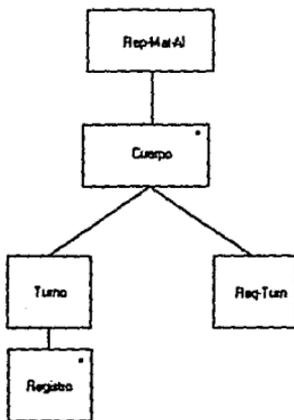


Figura 17.- Diagrama de Estructura de la función Rep-Mat-Al

El texto estructurado para Rep-Mat-Al es el siguiente :

```

Rep-Mat-Al seq
  turno:=1;
  cont:=0;
  leer OR&CT;
  Cuerpo itr
    Turno seq
      cont-turno:=0;
      Cuerpo-turno itr while (Registro)
        Registro seq
          cont-turno:=cont-turno + 1;
          cont:= cont + 1;leer OR&CT;
        Registro end
      Cuerpo-turno end
    Req-turn seq
      Imprime encabezado;
      Imprime detalles;
      leer OC&CT;
    Req-turn end
    turno:=turno + 1;
  Turno end
  Cuerpo end
Req-Mat-Op end.
  
```

Finalmente, al integrar los diagramas mostrados en las figuras 14 y 16 se obtiene el Diagrama de Especificación del Sistema en el cual se incluyen los procesos para las funciones requeridas por el usuario :

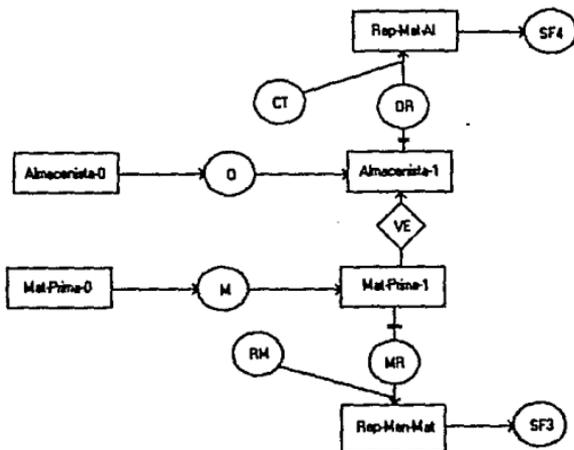


Figura 18.- Diagrama de Especificación del Sistema con procesos de funciones incluidas.

4.1.5.- PASO 5.- PASO DE SINCRONIZACIÓN DEL SISTEMA.

El objetivo de este paso es el documentar las restricciones de sincronización de las salidas del sistema.

En este caso, como ya se mencionó anteriormente, en las funciones 3 y 4 se utilizará un Merge Áspero, lo cual no implica graves problemas, ya que lo peor que podría suceder es el hecho de considerar un registro adicional, el cual será compensado el siguiente mes.

Por otro lado, para generar las salidas de las funciones no es necesario que exista sincronización entre los procesos, por lo tanto no se hará ninguna consideración especial a este respecto.

Los resultados de esta etapa, que de hecho son la especificación del sistema son los siguientes :

a).- Diagrama de Especificación del Sistema, figura 18

b).- Diagrama de Estructura de los procesos :

- Materia Prima, figura 8
- Almacenista, figura 2
- Rep-Men-Mat (función F3), figura 15
- Rep-Mat-Al (función F4), figura 17

c).- Textos estructurados para cada uno de los procesos incluidos en el inciso anterior.

4.2.- ETAPA DE DESARROLLO DE LA IMPLEMENTACION

4.2.1.- PASO 1.- PASO DE IMPLEMENTACIÓN.

En la etapa anterior se desarrolló la especificación del sistema bajo el supuesto de que cada procesos de cada instancia de cada entidad se ejecutaría en un procesador diferente, el cual estaría dedicado todo el tiempo a la ejecución de un solo proceso.

Como ya se mencionó anteriormente el objetivo de esta etapa es el de llevar estas especificaciones a ejecutarse en un solo procesador.

En el Diagrama de Especificación del Sistema, figura 18, puede observarse que las entradas al sistema son :

- O : Contiene los mensajes que indican que el Almacenista ha sido notificado de que una Materia Prima ha llegado.
- M : Contiene los mensajes que indican que la Materia Prima ha sido solicitado.
- RM : Contiene los mensajes de requerimiento del Reporte de Entregas de Materias Primas recibidas por mes (función F3).
- CT : Contiene los mensajes de requerimiento para generar el Reporte de Materias Primas que se reciben y de las que se regresan.

Al introducir el proceso "Scheduler" que será encargado de controlar la ejecución de todos los procesos sobre un solo procesador, deberá leer todos estos mensajes de entrada, tal y como se observa en la figura 19 :

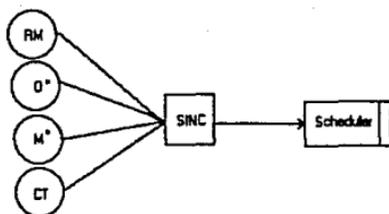


Figura 19.- Lectura de Mensajes de Entrada

O* y M* representan todas las entradas provenientes de todos los Almacenistas y de todas las Materias Primas.

SINC es el proceso que lee directamente del exterior los mensajes que estén disponibles y los pasa a "Scheduler", el Diagrama de Estructura de este proceso se muestra en la figura 20.

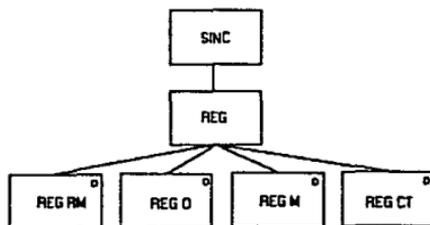


Figura 20.- Diagrama de Estructura del proceso SINC

Los procesos Almacenista-1 y Mat-Prima-1 deben ejecutarse en cuanto se origine una señal del exterior, es decir, en cuanto se tenga un registro disponible para ser leído por "Scheduler", ambos procesos son invertidos implementándose como una rutina de "Scheduler", con el fin de que él controle su ejecución.

El proceso Rep-Men-Mat será controlado por Mat-Prima-1 ya que este proceso se ejecuta a partir de las veces que una Materia Prima ha sido recibido, una vez que ha pasado por Control de Calidad; estos mensajes son enviados a través de MR (figura 15), de esta manera Rep-Men-Mat se invierte con respecto a MR y se implementa como rutina de Mat-Prima-1.

En forma análoga, Rep-Mat-A1 se invierte con respecto a OR y se implementa como rutina de Almacenista-1, en este caso OR contiene los

mensajes que indican que un Almacenista ha recibido o regresado una Materia Prima.

El Diagrama de Implementación se muestra en la figura 21. VEM-FILE es la agrupación de los vectores de estado de las instancias de la entidad Materia Prima, tal y como se explicó en el Capítulo 2.

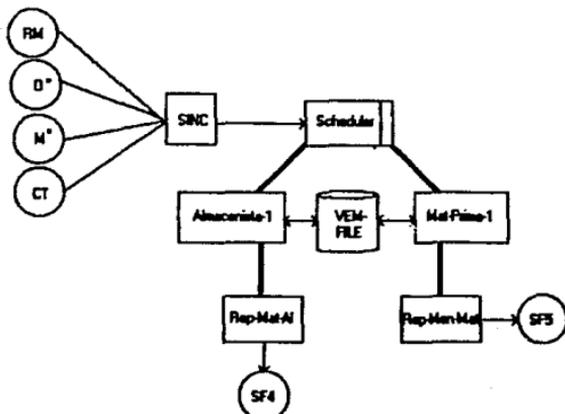


Figura 21.- Diagrama de Implementación del sistema

Generalmente se representan en una misma hoja los Diagramas de Implementación del Sistema (figura 18) y el Diagrama de Especificación del mismo (figura 21), con el fin de poder hacer referencia en forma rápida a cualquiera de los dos.

CONCLUSIONES.

El haber finalizado el desarrollo del presente trabajo me permite enunciar las siguientes conclusiones :

1.- La participación del usuario es básica para el Desarrollo del Proyecto. En términos generales la experiencia que los usuarios tienen en el área de Sistemas no es buena, por lo que la labor de convencimiento debe ser. Creo que es muy importante ir ganando terreno poco a poco. Algo muy importante para el usuario es que se cumpla con lo que se ofrece, es decir, una vez que se haya hecho algún ofrecimiento al usuario habrá que cumplirlo.

2.- El utilizar la técnica de Planeación del Proyecto no fué sencillo, resulta complicado organizar grupos de trabajo, hacer estimaciones y desarrollar actividades que aparentemente no son importantes, sin embargo los resultados que se obtuvieron se debieron en buena parte a esta técnica. Generalmente personas con formación matemática nos mostramos un poco renuentes a utilizar técnicas de este tipo.

3.- Considerar tiempos improductivos para hacer estimaciones utilizando el método de Esterling resultó muy práctico, y no es raro puesto que las consideraciones que hace son bastante reales.

4.- Considero indispensable que cuando se va a desarrollar un Proyecto de Software se asigne un analista de tiempo completo a conocer el ambiente del usuario, como si fuera parte de su grupo de trabajo, el hacer esto nos da una experiencia que difícilmente se puede obtener por otro medio, aprendemos a ver la realidad del usuario tal y como él la ve.

5.- La metodología Jackson comienza en una forma muy natural, realmente modela los procesos de la realidad del usuario, los conecta con la realidad y finalmente los implementa. Aparentemente la etapa de implementación es complicada, pero adentrándose un poco nos podemos dar cuenta que realmente es muy sencilla, una vez que se entiende el entorno teórico.

6.- Los programas en JSD realmente se van desarrollando poco a poco, ya que desde el paso de Modelado Inicial se crea el texto estructurado, y a partir de ese momento se la van añadiendo nuevas instrucciones; en el paso de funciones se crean los procesos de funciones junto con su texto estructurado.

7.- Algo que es muy importante para JSD es que se define bien el alcance del proyecto, ya que si algunas entidades no se incluyen en el paso de Modelado Inicial, difícilmente podrán ser incluidas en etapas o pasos posteriores; sin embargo de las entidades incluidas, resulta muy sencillo agregar nuevas funciones.

8.- Aunque JSD no incluye técnicas de Diseño de Bases de Datos, resulta muy sencillo incorporar alguna técnica de este tipo, de hecho deja

marcadas las estructuras de datos como agrupaciones de vectores de estado.

9.- Con el desarrollo JSD resulta muy sencillo trasladar la implementación utilizando lenguajes de cuarta generación, ya que existen varias analogías con la programación orientada a objetos; en JSD el desarrollo de la especificación parte del modelado de entidades de información y de acciones que dichas entidades ejecutan.

10.- Con la metodología JSD no se llega hasta la implantación final del sistema, sino al desarrollar la especificación y al generar la implementación construye un marco sólido a partir del cual se puede concluir el proyecto.

11.- Finalmente, es necesario mencionar que las metodologías por sí solas no garantizan el éxito del proyecto, es necesario el entendimiento del problema, el conocimiento de las metodologías, el involucramiento de cada una de las áreas afectadas y el interés que cada una de ellas tenga sobre el proyecto.

BIBLIOGRAFÍA

1.- SYSTEM DEVELOPMENT

M.A. JACKSON
PRENTICE/HALL INTERNATIONAL
1983.

2.- INGENIERÍA DE SOFTWARE

ROGER S. PRESSMAN
MC GRAM HILL
SEGUNDA EDICIÓN, 1988.

3.- SISTEMAS DE INFORMACIÓN TEORÍA Y PRÁCTICA

JOHN G. BURCH JR.
FELIX R. STRAYER JR.

4.- PRINCIPIOS DE ADMINISTRACIÓN

GEORGE R. TERRY
EDITORIAL LIMUSA.