

80
2j



**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA AUTOMATICO DE CONTROL DE MOLDES
PARA UNA INYECTORA DE PLASTICOS**

T E S I S
Para obtener el Título de
INGENIERO EN COMPUTACION
p r e s e n t a

MARIO MONGE REYES



Director: Ing. Adolfo Millán N.

MEXICO 1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



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 esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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

A mis padres:

María Elena y Mario

Por el apoyo y consejo incondicionales durante toda mi vida.

A mis abuelos:

Margarita y Amando, Gabina y José

Por el impulso que me dieron para seguir adelante cuando decaía el ánimo.

**En agradecimiento a maestros,
amigos y familiares, por su
estímulo y colaboración en mi
formación profesional.**

PROLOGO

Este trabajo de tesis está enfocado a la aplicación y utilización de herramientas y metodologías para el desarrollo de sistemas. Dichas herramientas y metodologías se aplicaron en una empresa de transformación consiguiendo resultados satisfactorios.

El objetivo específico de este trabajo, es diseñar y desarrollar un sistema que automatice las funciones de administración de moldes de inyección de plástico y la producción obtenida de ellos, utilizando para ello una metodología de desarrollo de sistemas (SLC Systems Life Cycle) y dentro del análisis, la utilización de una herramienta case (IEW Information Engineering Workbench).

Este trabajo escrito esta compuesto por varios temas y a continuación se explicará brevemente que contiene cada uno de ellos.

En la introducción se define el problema que se atacó, así como el proyecto en sí mismo, en esta última definición también se tomaron en cuenta los recursos disponibles para desarrollarlo. Una vez que se definieron el problema y el proyecto en sí, se estableció el ambiente de desarrollo, es decir, el hardware y software. Es importante señalar que se utilizaron componentes que darán al sistema resultante vigencia tecnológica y posibilidad de crecimiento futuro.

En el segundo tema se explica la metodología utilizada para llevar a cabo el análisis del problema. Esta actividad incluye distintas actividades, como la documentación de la situación actual por medio de entrevistas con el cliente, elaboración del diagrama de contexto, obtención de los requerimientos de alto nivel y de negocio, modelado de funciones y datos, hasta la obtención de un modelo entidad-relación y una primera definición de los elementos de datos.

Durante el diseño estructurado del sistema, se llevan a cabo distintas actividades, en algunas de ellas interviene intensamente el cliente y en otras se realiza trabajo interno, utilizando las herramientas de creación de base de datos, desarrollo de programas y definiendo la arquitectura necesaria para el funcionamiento del sistema.

Profundizando en el contenido, se diseña el comportamiento del sistema y uno de los productos más importantes es la carta de estructura, que hace las veces de un plano que se utiliza para construir un edificio. También se diseñan las entradas-salidas, es decir, pantallas y reportes y las nuevas funciones físicas (seguridad, calendarización de procesos, etc.). Otra actividad es el diseño de organización o la manera en que se relacionarán los usuarios con el sistema.

Por otro lado, el diseño incluye otras actividades que son internas, en donde no se tiene una participación directa del cliente y tratan con aspectos meramente técnicos. Dentro de éstas se encuentran el diseño de arquitectura y el diseño físico de datos.

Posteriormente se desarrolla el tema de la construcción del sistema. Aquí se explican las distintas actividades que la componen, dentro de las que se encuentran, el establecimiento de estándares de programación y documentación, el desarrollo de procedimientos relacionados, la programación en sí y una fase de pruebas unitarias, que aparentemente puede parecer que esta fuera de lugar, pero se verá que está correctamente ubicada.

Finalmente, se dan puntos importantes para el desarrollo de la documentación, que contenido deben incluir los distintos productos y cual es la secuencia adecuada de éstos.

El siguiente tema es el de las pruebas del sistema. Se explican los diferentes tipos de pruebas a aplicar según las características del proyecto. Se considera que las pruebas sirven para dos cosas, determinar si el sistema efectivamente cubre las necesidades expresadas por el cliente durante la fase de análisis y para demostrarle que efectivamente el sistema está funcionando listo para utilizarse. Se explican los pasos relacionados para llevar a cabo el proceso de pruebas y como se puede conseguir una aprobación por parte del usuario.

El siguiente tema es la implantación del sistema. Se explica como se planea una implantación, hay una lista de actividades tipo que se pone a consideración. También se incluyen otras actividades como la elaboración del plan de contingencia. Dentro de la instalación propia del sistema, se incluye el hardware, conversión/carga de datos, instalación de la aplicación, capacitación y soporte al arranque del sistema. Se incluyen otras actividades post-implantación, como el monitoreo del sistema, revisión de éste y la entrega al cliente.

Posteriormente vienen las conclusiones del trabajo de tesis. Las conclusiones giran en torno a todos los elementos que intervinieron durante el desarrollo de la tesis, es decir, la metodología, las herramientas desarrollo y la infraestructura, así como la responsabilidad de los profesionistas con nuestro país y el rol que debemos tomar dentro del proceso de globalización que esta sufriendo el mundo.

Finalmente, se incluyen unos apéndices que sirven de apoyo en algunos de los temas desarrollados y la bibliografía.

INDICE

1. INTRODUCCION	
Definición del problema	1-1
Definición del proyecto	1-3
2. ANALISIS ESTRUCTURADO DEL SISTEMA	
Situación actual	2-1
Definición de requerimientos de negocio	2-9
Análisis de datos	2-11
3. DISEÑO ESTRUCTURADO DEL SISTEMA	
Diseño de negocios	3-1
Diseño de organización	3-6
Selección de plataforma de aplicación	3-7
Diseño técnico	3-7
Diseño de arquitectura	3-8
Diseño físico de datos	3-9
Diseño técnico del sistema	3-9
4. CONSTRUCCION DEL SISTEMA	
Organización	4-1
Programación	4-2
Pruebas unitarias	4-6
Desarrollo de documentación	4-9
Prueba de componentes no programa	4-12
5. PRUEBAS AL SISTEMA	
Organización	5-2
Prueba del sistema	5-3
Pruebas de aceptación de usuario	5-10
6. IMPLANTACION	
Organización	6-1
Planeación de implantación	6-3
Instalación	6-6
Post instalación	6-9
7. CONCLUSIONES	7-1
8. APENDICE	
A Terminología y simbología de diagramas entidad relación	A-1
9. BIBLIOGRAFIA	9-1

1 INTRODUCCION

El objetivo de este trabajo de tesis es el de mostrar la solución a una problemática real de una empresa de transformación. Para llegar a la solución se desarrollará un sistema utilizando la metodología "Ciclo de Vida de los Sistemas" (SLC).

Pero qué hace distinta esta metodología a las ya existentes, pues la diferenciación básica es que está enriquecida con experiencias de grupos de trabajo que están inmersos en el medio, en el fondo, se basa en los mismos principios que otras, pero definitivamente tiene conceptos de negocio que ayudan a asegurar el éxito del proyecto.

El SLC (Systems Life Cycle) es una metodología efectiva para la administración de proyectos de sistemas de información. Hoy en día los clientes demandan que los sistemas sean desarrollados eficientemente y tengan mantenimiento bajo, a la vez que debe haber una relación costo-eficiencia. Esta demanda requiere una metodología que este documentada y sea repetible y mensurable.

Para el análisis de este sistema se utilizará una herramienta CASE cuyo nombre es IEW. Como sabemos, este tipo de herramientas apoyan o están basadas en distintas metodologías para el desarrollo de sistemas, que podemos nombrar como "Ingeniería de Software". El apoyo de estas herramientas disminuye la posibilidad de error en el sentido de omisiones o descuidos cuando el tamaño de la aplicación a desarrollar es muy grande. El impacto económico es muy fuerte, ya que como sabemos, a medida que más adelantados estemos en el desarrollo de un sistema más caro resultará hacer correcciones.

1.1 Definición del Problema

A continuación se muestra un panorama general del proyecto y la definición del problema.

Una compañía inyectora de plásticos se dedica a transformar esta materia prima en piezas o componentes que forman parte de otras más complejas a través de moldes y máquinas de inyección.

Una máquina de inyección de plásticos tiene una utilización que podríamos llamar general, es decir, puedo utilizar una máquina para inyectar varios tipos de plástico y le puedo "montar" distintos moldes.

Las características básicas de una máquina son:

- Límite en la capacidad de inyección (cantidad plástico)

- Tipo de inyección (uno o más colores al mismo tiempo y tipo de colada)

Cuando se adquiere alguno de estos equipos se sabe de entrada las características que tiene. El otro elemento que participa en el proceso de inyección es el molde. Si algunos de estos elementos faltara no sería posible obtener el resultado final, que sería la pieza o parte, pero podemos considerar que el elemento fundamental para poder obtener el producto final es el molde. Cada molde tiene asociado características inherentes a sí mismo, como su peso, tiempo de enfriamiento, tiempo de inyección, presión de inyección, etc. Estas características no se encuentran formalmente descritas en la empresa de cuyo estudio es este trabajo, lo que tiene como consecuencia que se tengan grandes desperdicios de plástico al tratar de poner en producción algún molde.

El proceso de elaboración de un molde puede llegar a ser tan largo y complejo como sea la forma de la pieza o como se decida la durabilidad de éste. Por el trabajo pesado al que son sometidos los moldes tienen que ser construidos con materiales como acero o zamac, por lo que por su peso se hace difícil su manejo.

Sin entrar a tanto detalle podemos decir que un molde es una inversión considerable que en algunos casos puede tardar varios años en ser recuperada.

Por lo general este tipo de compañías tienen muchos clientes que son dueños de varios moldes y es difícil identificar alguno específico. Por otro lado, el manejo administrativo como entradas y salidas se vuelve un problema cuando no hay un control automatizado.

Por otro lado, se tienen clientes que requieren de una producción no fija, es decir, pueden llamar para averiguar si se tienen determinadas piezas en almacén ya que está a punto de terminarse lo que están empaquetando. Este control no sería difícil llevarlo manualmente si no se tratará con más de 200 moldes distintos y se tiene que controlar la producción de varias máquinas que tienen 3 turnos diarios cada una.

Como podemos ver, la problemática de esta compañía es muy específica, se requiere básicamente de controles automatizados para identificar de manera única cada molde que se recibe o se entrega, poderlo encontrar fácilmente dentro de la bodega en caso que se requiera, obtener las especificaciones de operación de cada molde por máquina y asociar la producción de las máquinas a cada molde para poder obtener rápidamente la existencia y ofrecer un servicio más eficiente a los clientes.

1.2 Definición del Proyecto

Justificación

A pesar que existen en el mercado paquetes para necesidades generales, algunas veces se dan situaciones especiales bajo las cuales no son aplicables.

En este caso, se tienen situaciones muy específicas como el almacenamiento de parámetros de inyección, entre los que se encuentran: tiempo de enfriamiento, presión y tiempo de inyección, peso de la carga a inyectar. Cada una de estas variables es distinta de acuerdo a la pieza a inyectar y al material y calidad de éste.

El costo y la oportunidad de tener identificado plenamente moldes de inyección de plástico es considerable, y más aún cuando se tienen más de 200 que pertenecen a distintos clientes. Algunos de los moldes tienen más de 2 años sin utilizarse. El control de entradas y salidas, así como su colocación dentro de la bodega es plenamente justificable tomando en cuenta que el costo de un molde de inyección puede sobrepasar varios miles de nuevos pesos.

Otro costo asociado, es el propio de algunos materiales, que en combinación con el peso de la pieza y el tiempo a invertir, puede ser alto. Se desea eliminar el proceso de prueba y error para dejar funcionando un molde que previamente ya había sido montado. Es un hecho que cuando no se cuenta con el valor de estas variables la primera vez que se monta el molde, se tiene que recurrir a este proceso empírico.

Por otro lado, la oportunidad de saber en un momento la existencia de piezas por color es muy importante tomando en cuenta que algunas veces los clientes tienen líneas de empaque y no pueden detenerlas.

El utilizar herramientas automatizadas facilita estas actividades y pueden ayudar con oportunidad al proceso productivo de las empresas.

Ambiente de Desarrollo

Aunque más adelante durante el desarrollo de este trabajo de tesis se determinará el momento en donde se puede especificar el ambiente de desarrollo de un sistema, podemos decir que se trata de una aplicación que se desarrollará con las siguientes herramientas y tendrá las siguientes características:

- Desarrollo en PC
- Herramienta de análisis IEW (modelado de datos y procesos)
- Diseño físico de base de datos utilizando ERwin
- Herramienta de programación Visual Basic
- Ambiente Windows
- Utilización de base de datos (SQLBase de Gupta)
- Utilización de comandos SQL a través de ODBC para acceder la base de datos (drivers de QELib)
- Reporteador (Q+E de Intersolv)

El desarrollo de la aplicación comprenderá la utilización de los componentes mencionados anteriormente y el resultado funcionará bajo el ambiente establecido.

Este ambiente de desarrollo se determinó con base en la disponibilidad del cliente respecto al hardware y a aspectos de tecnología que podrán dar mayor vigencia a la aplicación.

Otra característica de la aplicación es que puede funcionar en red, con la limitante de utilizar la misma base de datos. Para el punto anterior se tienen que evaluar las restricciones de ésta última. En dado caso de querer utilizar otra distinta, se tendrá que evaluar la compatibilidad de los comandos de SQL.

2 ANALISIS ESTRUCTURADO

En esta fase de la metodología se determinan las oportunidades o problemas del área de estudio, se definen los requerimientos del sistema para poder resolver dichos problemas o para darle la oportunidad al cliente de tomar ventaja del conocimiento, identificación y su posible solución.

Como sabemos, gran parte del éxito de un proyecto de sistemas depende que tan bien se entiende la problemática por parte del analista. Es común que un análisis no sea correctamente documentado por algunas situaciones como la utilización de términos muy específicos de una industria, tomar como obvio alguna condición o proceso que para el usuario lo es, falta de experiencia en la industria, etc. Estas son sólo algunas situaciones, pero podríamos encontrar varias más.

Otro punto muy importante a considerar es que en la definición del sistema debe haber congruencia entre los objetivos planteados al inicio del proyecto y las metas del cliente. Además se debe analizar cada una de las peticiones de servicio para determinar su impacto en el proyecto.

Otras actividades que son importantes al inicio del proyecto son el establecimiento de estándares de documentación y procedimiento de control de cambios. Los estándares sirven para dar imagen de orden y profesionalismo al cliente, además que no se pierde tiempo tratando de definir un formato cada vez que se elabore un documento. El control de cambios sirve para controlar desviaciones que pueden afectar el alcance y por ende la duración del proyecto.

2.1 Situación Actual

El propósito del análisis de situación actual es el de comprender el medio en el que se desempeña el cliente. Se documentan las funciones actuales que tienen sus sistemas, archivos de datos, componentes de Hardware incluyendo redes, estructura organizacional, flujos de trabajo e interfaz con otros sistemas. No se debe escatimar en documentar esta situación actual. Se hace uso de la información escrita que exista tal como manuales de usuario, documentación del sistema y organigramas, no olvidar que el cliente quiere una solución, no documentación sobre su problema.

Analizar la Situación Actual

Esta actividad consiste en el estudio del medio ambiente del negocio del cliente, en los aspectos de utilización de sistema(s) operativo(s), sistemas automáticos, y recursos computacionales, así como funciones manuales y automatizadas.

A través de entrevistas, se entienden las funciones principales y las características que debe tener el nuevo sistema. En caso que ya exista un sistema hay que documentar las funciones más populares de este, para que si aplica, se incluyan en el nuevo o por lo menos se especifique como serán cubiertas.

Se levanta un inventario del hardware actual y los componentes de red (si hay). Este inventario sirve para determinar la disponibilidad de bienes para que sean considerados cuando se determine la solución o para saber los recursos en los que se desarrollará el sistema.

En el análisis se llevaron a cabo entrevistas en distintos niveles. En estos casos lo más recomendable es comenzar las entrevistas desde un punto de vista TOP-DOWN, por lo que la primera entrevista fué con el dueño de la empresa, en esta entrevista básicamente se preguntó sobre las expectativas que tiene del proyecto, su punto de vista respecto a su problemática, los planes a futuro respecto al negocio y también se determinó quién sería responsable del proyecto por parte de ellos.

El responsable del proyecto es la persona que va a coordinar las entrevistas con el resto de las personas que van a intervenir en éste, por lo general, esta persona determina quién va a participar, aunque en algunos casos se requiere alguna entrevista con alguna otra persona que consideremos puede aportar información importante al proyecto.

Para el caso de la entrevista inicial, se extraen los comentarios más importantes y se integran en un documento que contiene los objetivos del proyecto que están directamente relacionados con la problemática, los requerimientos de alto nivel, los planes a futuro, incluyendo únicamente los que están directamente relacionados con el proyecto, así como la designación del responsable del proyecto por parte de ellos. También se determina la periodicidad de la presentación del reporte de avance del proyecto, este reporte debe ser validado por el responsable del proyecto y posteriormente entregado al dueño para su revisión.

Un documento muy importante que se requiere siempre es un organigrama de la empresa. El organigrama sirve entre otras cosas para ayudar a visualizar líneas de mando, responsabilidades y para poder determinar posibles flujos de información.

No como receta, pero como guía se especifican los siguientes pasos para el análisis de situación actual:

Elaborar Diagrama de Contexto

Después de entrevistar al jefe de producción pudieron identificar todos las entidades que intervienen dentro del proceso que está en estudio, así como los principales flujos de datos, con lo que se pudo determinar el diagrama de contexto.

En este diagrama se podrán ver las entidades que van a delimitar el alcance del sistema. Así como las partes de la organización que se ven afectadas por los problemas que tienen actualmente. La mayoría de estos procesos son llevados de forma manual en formas de papel que son susceptibles de tener errores y de perderse.

Otro producto que se puede obtener de este diagrama es la lista de entidades. Esta lista tiene un explicación de la función que desempeñan o como intervienen en el sistema.

CLIENTE	Son el objetivo de la existencia de esta empresa. Mandan producir piezas de sus moldes.
TALLER MECANICO	Es donde se mandan reparar los moldes. La manera en que afectan al proceso es que pueden tener en su poder moldes.
AREA PRODUCCION	Es la encargada de tener funcionando las máquinas que tienen montadas los moldes. Saben cual ha sido la producción por máquina y turno.
BODEGA	Es la encargada de controlar la producción disponible para ser entregada. Debe tener en todo momento la existencia por pieza.

Elaborar lista de Eventos-Respuesta

Esta lista sirve para documentar la interfaz entre el sistema y el medio ambiente. Estos eventos se incluyen en los requerimientos de negocio.

La siguiente lista es una selección de los principales eventos-respuesta que suceden dentro del ciclo de negocio.

EVENTO	RESPUESTA
Cliente Entrega Molde	Emitir recibo de entrada con fecha y hora, estado general Registrar la localización del molde dentro del área de producción
Cliente Recoge Molde	Emitir recibo de salida con fecha y hora, estado general Emitir la localización del molde dentro del área de producción
Taller Mecánico Recoge Molde	Emitir orden de trabajo, con descripción del estado general Emitir localización del molde dentro del área de producción
Taller Mecánico Regresa Molde	Emitir recibo de entrada con fecha, hora y estado general Registrar la localización del molde dentro del área de producción
Area de Producción Registra Variables para Inyectar	Ingresar las variables indicando la clave de la inyectora
Area de Producción Registra Producción por Molde	Registrar producción del molde por turno y empleado, así como observaciones Acumular la producción a la ya existente (si hay)
Area de Producción Registra Localización de Molde	Ingresar código de localización dentro del área de producción
Area de Producción Solicita Localización de Molde	Ingresar identificador de molde y recibir la localización
Area de Producción Solicita Variables para inyectar	Ingresar código de molde y recibir reporte de variables
Bodega Consulta Existencias	Ingresar identificador de Molde/Pieza y recibir por pantalla existencia
Bodega Solicita Reporte de Inventario	Generar reporte ordenado por molde/pieza
Bodega Registra Ajuste al Inventario	Generar registro de ajuste Emitir reporte de ajustes Ejecutar ajustes

Determinar los requerimientos de Alto Nivel

El determinar estos requerimientos sirven de guía para obtener requerimientos más detallados.

Los requerimientos de alto nivel son los siguientes:

- Poder determinar la localización de un molde que haya entrado en la fábrica, esto incluye la posibilidad que este dentro, haya sido devuelto a su dueño o se encuentre en reparación
- Poder saber en cualquier momento los valores de las variables de las máquinas para poder inyectar cada molde
- Tener en cualquier momento el inventario de piezas producidas
- Utilizar los recursos de hardware con los que se cuenta actualmente

Revisar la documentación actual

En caso que exista un sistema en producción, es recomendable enterarse de los atributos y funcionalidad que tiene, es decir, que funciones cubre y para que fue diseñado. Esta documentación se evalúa respecto a la vigencia que tenga ya que es muy común que no esté actualizada.

Documentar funciones actuales

Para comprender las prácticas actuales del negocio que se incluirán en el nuevo sistema, se documentan los sistemas manuales y automáticos. Se determinan las funciones relevantes que se incluirán en el nuevo sistema. Es recomendable documentar también, espacio ocupado en disco, tiempos de proceso y responsabilidades organizacionales de las funciones documentadas.

El objetivo de determinar las funciones automatizadas actuales es el establecer cuales de éstas deberán ser incluidas en el nuevo sistema. Esto es muy importante ya que si el Usuario no ve cubiertas todas las actividades con las que contaba en ese momento opondrá resistencia al cambio.

No cuentan con ningún sistema relacionado con el objetivo de este proyecto, por lo que no se llevará a cabo ningún mapeo de funciones sistema actual-nuevo.

Las funciones lógicas son los procesos y/o características críticas para el Usuario que debe cubrir el sistema. El objetivo de determinar estas funciones es el no perder de vista cuales son las áreas del negocio del cliente en que nosotros intervenimos y que deben quedar perfectamente cubiertas. Esto es muy importante, ya que aunque no estén especificadas dentro de los criterios de éxito,

la manera en que las cubramos serán en gran medida el parámetro en que evaluarán nuestro trabajo.

Podemos decir que aunque el sistema abarcará funciones muy importantes dentro de la operación de la empresa no quedaría paralizada en caso de que fallara el sistema. Pero esto se debe al tamaño actual de ésta, en cuanto aumente el volumen de operaciones con las máquinas y los moldes, será un sistema vital.

Documentar la estructura de datos actual

Se documenta la estructura de datos y el contexto en el que se encuentra funcionando. Los datos son la representación de conceptos, reglas o instrucciones representados de tal manera que son adecuados para comunicación, interpretación o procesamiento manual o automático. La documentación puede incluir estructuras de datos existentes, tablas, columnas, relaciones, modelo entidad-relación, diccionario de datos, volúmenes de información, restricciones de seguridad, etc.

Podemos decir que no cuentan con sistemas para sostener la operación respecto al objetivo de este proyecto por lo que podemos resumir que la situación actual es la siguiente:

- Registro manual de variables de inyección de algunos moldes, dicho registro es susceptible de perderse ya que se tiene en un cuaderno al que todo el personal de operación tiene acceso
- No hay registro de las entradas de los moldes, únicamente se elabora un documento que especifica la entrega de algún molde a su dueño
- No hay registro de la producción, cuando se quiere saber la existencia de alguna pieza en específico se recurre a un conteo físico

Inventario de hardware, software y componentes de red

Desarrollar o modificar diagramas o gráficas que muestren la localización del hardware, componentes de red, aplicaciones y los departamentos que utilizan el sistema actual.

Es importante conocer los recursos con los que cuenta la empresa de nuestro estudio ya que este inventario junto con las políticas tecnológicas nos dará la pauta a seguir sobre la solución que debemos de proponer en un proyecto. Es mucho más fácil que se apruebe un proyecto en el que se venda la idea que se tendrá continuidad con los planes tecnológicos de la empresa que si se propone un cambio radical, claro, esto es en el caso que no sea esto mismo otro objetivo a alcanzar en un proyecto.

Actualmente cuentan con un equipo ACER 386 SX/33 con 4 MB RAM, 210 MbHD, Mouse, Drive de 3.5 in de alta densidad, cuentan con aproximadamente 140 MB de HD libre, monitor VGA a colores.

No cuentan con ninguna tarjeta especial.

Dentro del software se incluyen manejadores de bases de datos, versiones de sistemas operativos y manejadores de red.

Se cuenta con un sistema de contabilidad desarrollado en clipper, windows 3.1, word para windows 2.0, excel 4.0 y la versión del sistema operativo es la DOS 5.0

Documentar la organización actual

Esta información debe incluir la estructura actual, políticas operativas y descripciones de puestos. También podemos incluir canales de comunicación, políticas de éstos, así como la localización física de los participantes en el proyecto.

Posteriormente al leer las entrevistas o llevar a cabo otras, se obtiene información muy importante como los problemas/oportunidades relacionados con el objeto de estudio que de alguna manera sirven para medir al final cuando se esta implementando el sistema si este cumple con las expectativas. También analizando de manera global el impacto dentro de la organización de la empresa se pueden determinar cuáles áreas, servicios o productos serán afectados.

A continuación se muestran los problemas detectados durante las entrevistas, así como los observados en la operación diaria.

- Desperdicio de material al montar un molde que ya ha sido previamente maquinado
- Se invierte mucho tiempo al montar un molde ya que fué montado previamente
- No se puede determinar en el momento la cantidad de piezas que se tienen en el almacén
- Dificultad para localizar un molde específico en el área de producción
- No se puede determinar en el momento la localización física de un molde específico

La lista anterior tiene los problemas más importantes que serán cubiertos dentro de la solución que ofrece el desarrollo del sistema.

Es muy importante determinar que problemas quedarán cubiertos con el sistema y cuales con procedimientos. Los procedimientos son muy importantes ya que apoyan la implantación del nuevo sistema.

Por otro lado, se determina el efecto que tendrá en la operación el nuevo sistema, esto con el objetivo de llevar a cabo las acciones pertinentes o incluirlas en el plan de implantación.

Producto/ Servicio/ Operaciones/ Organizaciones Afectados	Situación Actual
Hoja de Producción	Se llena después de cada turno y se entrega a la secretaria de control de producción
Cardex de Producción	Se lleva un cardex en papel para determinar la existencia de alguna pieza
Requisición de servicio en taller mecánico	Verbalmente se dan órdenes de trabajo
Entrega/Recepción de Moldes	No se da ningún comprobante, ni se lleva ningún control administrativo.
Montaje de Molde	Lleva mucho tiempo "calibrar" la inyectora ya que sólo se cuenta con un aproximado del peso de la pieza
Búsqueda de Molde	Lleva mucho tiempo ya que no se encuentran plenamente identificados y ubicados.
Bodega	Sólo localiza las piezas y saca las cantidades pedidas para ser despachadas.
Solicitud de existencia de piezas	Se verifica el cardex, pero no es muy confiable ya que es susceptible a tener errores en las adiciones y salidas.

Finalmente, se pueden obtener los criterios de solución que se deberán considerar para hacer una recomendación final. En este caso, ya sabemos que se requiere un sistema para automatizar las funciones, pero se puede dar el caso que la recomendación sea seguir trabajando como se encuentra actualmente todo, no siempre un sistema automático es la solución a los problemas.

Los criterios de éxito son considerados como sinónimo de lo que el cliente espera del proyecto. Estos criterios pueden establecerse directamente o indirectamente.

El propósito de determinar este criterio de éxito es documentar el conjunto de productos, funciones, características que el sistema debe tener o desempeñar, para que el cliente pueda aprobar el proyecto.

Es muy común que estos criterios de éxito sean subjetivos e intangibles. Es responsabilidad nuestra ponerlos de tal manera que sean mensurables y si pueden quedar determinados en forma de requerimientos puede facilitar nuestro trabajo. Los criterios de solución son:

- Poder determinar en el momento la localización de un molde
- Disminuir el costo asociado al montar un molde cuando ya fué montado previamente en la compañía
- Poder determinar en el momento el inventario de piezas producidas.
- Aprovechar los recursos de hardware con los que se cuenta actualmente.

El determinar el criterio de éxito es crucial para cualquier proyecto. Se puede tener la mejor documentación y sistema terminados pero el proyecto puede no ser exitoso porque no se cubrieron las expectativas del cliente.

2.2 Definición de requerimientos de negocio

Utilizando técnicas de análisis estructurado se identifican las funciones de negocios que el cliente requiere en el nuevo sistema. El establecer estos requerimientos enfatiza los requerimientos funcionales, da una base para el modelado de datos, diseño, pruebas, y evaluación del sistema propuesto; y establece un criterio de aceptación de usuario.

Requerimientos de Negocio Actuales

Podemos definir esta actividad como la identificación de las funciones actuales de negocios que el sistema debe incluir; identificar políticas, condiciones e influencias internas y externas que puedan afectar la aplicación.

Las actividades a desarrollar son las siguientes:

Expandir los requerimientos de alto nivel

Los requerimientos de alto nivel establecen el contexto del proyecto. Las áreas funcionales que intervendrán deberán ser discutidas con el cliente. Se obtendrá mayor detalle de los requerimientos.

Revisar los requerimientos de negocio actuales

Es necesario entender los estándares y regulaciones a las que está sujeta la industria del cliente, así como la interfaz con otros sistemas o entidades.

De ser necesario se llevan a cabo nuevas entrevistas para revisar los resultados, de tal manera que se esté seguro que ya estén completos.

Al final se documenta toda la información resultante para que este a la mano durante el resto del análisis y diseño.

Algunos productos son :

Funciones actuales relacionadas con el sistema

- Localizar un molde en donde quiera que se encuentre, esto incluye la fábrica, que haya sido devuelto a su dueño o se encuentre en reparación
- Obtener en cualquier momento los valores de las variables de las máquinas para poder inyectar cada molde
- Tener en cualquier momento el inventario de piezas producidas

Influencias externas

Considerar que hay distintas marcas de inyectoras en el mercado y cada proveedor tiene sus propias especificaciones. Por otro lado la empresa no compra equipo de una sola marca, por lo que se deberán de considerar distintas variables de inyección.

Desarrollar especificaciones de formas, pantallas y reportes

Se incluye una lista del propósito de este(a), contenido y características de los reportes, formas y pantallas; inicialmente no se incluyen atributos físicos como colores, encabezados o colocación de campos, lo anterior es agregado más adelante a medida que se avanza en el análisis y el diseño; algunas veces los lay-out actuales son incluidos en las especificaciones.

En este punto las especificaciones son de alto nivel y serán actualizadas a medida que avance el proyecto. El contenido de esta especificación puede incluir algunos de estos elementos:

REPORTES	FORMAS	PANTALLAS
nombre	nombre	nombre
propósito	propósito	propósito
frecuencia de generación	frecuencia de uso	tiempo de respuesta
estimado de páginas		frecuencia de uso
distribución		nivel de seguridad
cortes		funciones
totales		ayuda
títulos		mensajes de error
orden (sort)		tamaño de campos
		defaults

Especificaciones de desempeño

Estas especificaciones describen el tiempo de respuesta y los requerimientos de salidas del sistema. Estas especificaciones pueden generar restricciones para determinar los componentes de éste.

2.3 Análisis de Datos

En esta actividad se analizan los datos existentes del cliente, para identificar, definir y organizar la información requerida para dar soporte a los objetivos de negocio del cliente. Posteriormente, esta información se liga con la obtenida en la Definición de Requerimientos de Negocio.

A continuación se describen los pasos a seguir para obtener el modelo de datos completo especificándose en donde se debe de utilizar el IEW. Los productos a desarrollar son:

1. Modelo de Eventos Respuesta y Diagrama de Contexto
2. Modelo de Procesos de Alto Nivel
3. Modelo Entidad-Relación

Reglas Generales:

1.- Modelo de Eventos Respuesta y Diagrama de Contexto

El diagrama de contexto se usa para definir los límites y áreas de influencia con las cuales interactúa el sistema.

Las actividades a realizar son:

- 1.a Desarrollar el diagrama de contexto con el fin de especificar el alcance del sistema.

En el diagrama de contexto se especifican sólo los eventos o respuestas que se relacionan con entidades externas al sistema.

Los eventos-respuesta serán representados como flujos en el diagrama de contexto.

Para el desarrollo del diagrama de contexto se recomienda observar las siguientes guías:

- Debe ser una representación gráfica del alcance de la aplicación
- Debe definir los límites de la aplicación
- Debe mostrar las interfases hacia otros ambientes externos, tales como aplicaciones, departamentos, etc.
- Debe contener sólo un proceso, el cual es la aplicación
- Debe mostrar todas las fuentes de eventos que alimentan a la aplicación y los destinos de la respuestas producidas por la aplicación
- El diagrama debe tener un nombre que indique el área de negocio o aplicación que representa
- La aplicación se representa con un rectángulo redondeado y con su nombre dentro
- Las entradas y salidas se representan con líneas terminadas en una flecha la cual indica el sentido
- Las entidades externas se representan usando un rectángulo con su nombre dentro de él
- Nombrar de una manera descriptiva y clara todas las entidades externas que interactúen con la aplicación utilizando exclusivamente letras mayúsculas
- Todas las entradas y salidas deben tener un nombre que describa perfectamente el flujo de información. Este nombre puede estar escrito con letras mayúsculas y minúsculas

1.b Desarrollar lista de evento(s) respuestas

Un evento debe especificarse de acuerdo a la siguiente sintaxis:

Sujeto-Verbo-Objeto

Deben incluirse eventos:

- Originados por una entidad externa
- Originados por el paso del tiempo
- Originados por una secuencia o conjunción de otros eventos

1.c Documentar diagrama de contexto y eventos respuesta en IEW

Usando la facilidad de DFD (diagrama de flujo de datos) documentar el diagrama de contexto.

1.d Documentar los eventos y su transformación al medio

Para cada evento y respuestas se especificará su relación con formas, comunicaciones, reportes, su volumen y distribución, con el objetivo de tener un punto inicial de dimensionamiento. Estos datos se documentarán en formas de papel.

2.- Modelo de Procesos de Alto Nivel

Se particionará el diagrama de contexto en varios diagramas de flujo de datos, para modelar cada uno de los eventos del diagrama de contexto. Constituyéndose cada uno de estos modelos en un proceso de alto nivel.

Para cada uno de estos flujos se usarán las siguientes reglas de sintaxis.

Para nombrar procesos se deberá usar un *verbo*

Para nombrar flujos de datos se usará:

Sustantivo + Adjetivo descriptivo

Para almacén de datos se usará:

Sustantivo + Adjetivo

Para entidad externa se usará un Sustantivo. El cual puede indicar Gente, Organización, proceso fuera del alcance del sistema.

Un proceso debe realizar una transformación sobre los datos, por lo que el flujo de datos debe tener un nombre diferente después de la transformación.

Estos procesos de alto nivel se documentarán usando los diagramas de flujo de datos de IEW.

3.- Modelo Entidad Relación

3.1 Se desarrollarán los diagramas de entidad relación, siguiendo las guías generales especificadas en el apéndice A.

3.2 Se documentará el diagrama de entidad relación utilizando la utilería correspondiente de IEW

Dentro de esta utilería deberá especificarse para cada entidad la siguiente información:

Nombre:

Definición:

Clasificación: (Fundamental, Asociativa, Atributiva)

Comentarios:

Se utilizarán los tipos de entidad fundamental y asociativa para especificar relaciones regulares. Para entidades del tipo característica se utilizará la clasificación atributiva, para subtipo y colección se utilizará la clasificación otros y se especificará en comentarios (Ver apéndice A para clasificación de entidades y relaciones).

Para entidades asociativas se indicará además si la entidad es compleja o histórica.

Para cada relación se especificará:

Nombre:

Definición:

Número mínimo:

Número máximo

Comentarios:

El número máximo indica la cardinalidad de la relación (1 ó M).

El número mínimo indica su participación ya sea total (1) ó parcial (0).

Para cada atributo o elemento de datos se especifican reglas de validación, fórmula de cálculo en caso de ser un dato derivado, los significados en caso de ser una tabla o bandera. Además se debe especificar el dominio.

3 DISEÑO ESTRUCTURADO DEL SISTEMA

Se considera que la fase de diseño está dividida en dos partes: el diseño de negocios y el diseño técnico. En el diseño de negocios se define como se cubren las necesidades de negocio desde el punto de vista del cliente. Al mismo tiempo, en el diseño técnico se define como se cubrirán las necesidades de negocio del cliente desde el punto de vista técnico.

En el diseño de negocios se diseñan pantallas, reportes y procesos para cubrir los requerimientos del cliente. Durante el diseño pueden surgir conflictos con el cliente debido a la factibilidad técnica de algunos requerimientos. Es importante involucrar a todos los participantes en el proceso.

El diseño técnico tiene que ver básicamente con el diseño del sistema de tal manera que éste se implante en la tecnología seleccionada. El objetivo primario es desarrollar el diseño con la perspectiva de obtener programas y la estructura de datos (flujo de datos del sistema).

3.1 Diseño de Negocios

Esta es la fase de diseño que requiere la mayor participación del cliente, inclusive más que en el caso del diseño técnico. En la fase de análisis, el problema u oportunidad se descompone hasta llegar a obtener requerimientos. En ambas fases de diseño, los requerimientos se conjuntan para obtener especificaciones, a partir de las cuales se puede desarrollar el producto durante la fase de construcción. El diseño de negocios y técnico se desarrollan al mismo tiempo.

El diseño de negocios es el proceso de definición de los requerimientos lógicos y de diseño del sistema, con el objetivo de satisfacer los requerimientos de negocio del cliente.

El objetivo de esta fase es cubrir los requerimientos del cliente a través de la definición del sistema que está esperando.

Podemos decir que en la fase de Análisis se determinó *Qué* se tiene que hacer y en las fases de Diseño se determina *Cómo* hacerlo.

Diseño del comportamiento del sistema

El comportamiento del sistema a diseñar es muy importante. Es conveniente invertir tiempo obteniendo los modelos de comportamiento porque muestran las pequeñas partes del sistema y la manera en que los distintos procesos del mismo se sincronizan.

Hay muchas maneras de modelar el comportamiento de un sistema. Los diagramas de flujo de datos son la base del diseño de un sistema y pueden estar complementados con flujos de control y procesos. Los diagramas de estados son útiles para modelar sistemas que son puramente reaccionarios, como los sistemas en tiempo real. Los modelos de transacción sincronización están orientados hacia usuarios finales, además de ser la base para desarrollar gráficas de la estructura del sistema. Finalmente los prototipos, que son realmente modelos automáticos sincronizados, se pueden construir para demostrar el comportamiento del sistema al usuario.

Para elaborar el diseño del sistema podemos desarrollar las siguientes actividades:

Documentar el flujo de control del sistema utilizando los nuevos requerimientos de negocio

Estos requerimientos pueden ser diagramas de flujo de datos, diagramas de transición de estado y diagramas entidad-relación, y se pueden utilizar para desarrollar el modelo de comportamiento del sistema. En sistemas de pequeños a medianos se puede utilizar únicamente el diagrama de flujo de datos con anotaciones sobre las relaciones entre los procesos. Los sistemas grandes requerirán de diagramas de transición de estado o modelo de sincronización de transacciones.

Analizar el modelo de comportamiento

Determinar relaciones entre funciones. Es necesario identificar la relación entre módulos que utilizar la misma información. Una vez identificadas estas relaciones se pueden tomar precauciones sobre la responsabilidad de uso de funciones.

Crear la carta de estructura de los procesos on-line

Es la representación visual de la estructura de menús.

Obtener la aprobación del cliente

La aprobación puede ser formal o informal pero hay que incluirla.

Para el proyecto se obtuvieron dos modelos, uno de ellos fué por medio de prototipos debido a las facilidades de la herramienta para generarlos. Sin embargo, para especificar un mayor detalle en el aspecto metodológico, a continuación se detalla que debe mostrar un modelo de comportamiento del sistema.

Un modelo de comportamiento del sistema debe:

- Describir el flujo de control dentro del sistema
- Demostrar las acciones tomadas por el sistema en respuesta a un evento
- Describir la interacción del operador con un sistema en línea
- Identificar funciones y datos del diseño ya sea para acciones manuales o automatizadas
- Determinar la secuencia de las funciones
- Definir la entrada de datos y el medio de salida
- Agregar cualquier función extra requerida por aspectos de tecnología

Se obtuvo otro modelo a partir de los diagramas de flujo de datos generados por el IEW, complementados con los procedimientos escritos para la utilización de las opciones del sistema. Por ejemplo:

Ajuste de Inventario

1. Registrar el monto o cantidad del ajuste en la opción del sistema "Registro de Ajustes al Inventario". Se debe repetir esta acción tantas veces como ajustes se tengan.
2. Ejecutar la opción "Reporte de Ajustes al Inventario" para verificar que los registros sean los deseados. Este paso es opcional.
3. Ejecutar la opción "Afectación de Ajustes al inventario", este proceso afecta todos los movimientos que se encuentran registrados sobre el inventario existente.

El procedimiento anterior acompañado con el diagrama de flujo de datos muestran de una manera muy sencilla como se comporta el sistema y cuales son los pasos a seguir para esperar un funcionamiento adecuado.

Un modelo de comportamiento debe ser lo más claro y sencillo posible ya que en grandes proyectos debe ser entendible por cualquiera de los integrantes del equipo.

Diseño de entradas-salidas

Una actividad muy relacionada con el desarrollo del modelo de comportamiento de un sistema es el diseño de las entradas-salidas del sistema. Esta actividad incluye la forma y el contenido de datos que fluye entre los procesos. El objetivo primario es determinar exactamente cuál será la interfase hombre-máquina, que en otras palabras son las formas

codificadas, impresiones y pantallas interactivas. Las interfases máquina/máquina se ocupan de cómo pasar mensajes y de los protocolos a utilizar y cómo registrar la información en archivos leibles por la máquina. Estas interfases deberán de cubrirse durante el diseño de arquitectura. En la aprobación de las entradas-salidas pueden intervenir desde el usuario directo, hasta el responsable de un sistema con el cual se tiene alguna interfaz.

Dentro de esta actividad se deben considerar los estándares que se utilizarán en el diseño para que el sistema sea consistente y por lo tanto fácil de entender, probar, documentar y mantener.

Los pasos son los siguientes:

Examinar los procesos definidos en los nuevos requerimientos de negocio

Se determina que funciones serán pantallas, reportes o procesos. Es necesario involucrar al personal del cliente designado para ello. La decisión acerca de las funciones se toman con base en la frecuencia de utilización, frecuencia de actualización, necesidades de horarios, volumen de información y consideraciones de desempeño.

Se consultaron los nuevos requerimientos de negocio ya que sirven para determinar que funciones serán implementadas en pantallas en línea, reportes y procesos Batch. Otro punto a considerar en esta designación es la frecuencia de acceso, necesidades de tiempo, volumen de la información y consideraciones de desempeño.

Diseñar físicamente las pantallas y reportes

En esta etapa, probablemente las pantallas sean dibujos en papel. Al diseñarlas se considera el estándar establecido al inicio del proyecto.

Este proceso es iterativo, el cliente especifica los campos que quiere ver en pantalla y el ingeniero la presenta. El cliente la revisa y sugiere adiciones o modificaciones y el ingeniero las efectúa y las vuelve a presentar.

Además de ver físicamente como es la pantalla/reporte se recomienda incluir una descripción funcional general que ayude a detectar posibles diferencias entre la información a desplegar/imprimir y lo que espera el usuario y al funcionamiento mismo.

Documentar interfases

Obtener los formatos de lay-outs de como otros sistemas requieren la información a recibir.

Obtener la aprobación del cliente

La aprobación puede ser formal o informal pero hay que incluirla.

Diseñar nuevas funciones físicas

Los modelos de comportamiento de un sistema no son modelos que operen en la realidad, sino que están basados en condiciones de operación óptimas. Para hacer funcionar el modelo en las condiciones reales en las que trabajará, requiere que lo modifiquemos agregándole algunos procesos no esenciales, tales como módulos de seguridad y calendarización.

Los módulos de seguridad permiten que el sistema se proteja contra el ingreso de datos ilegales. Los calendarizadores son procesos que permiten el ingreso de datos en el momento adecuado. Reconociendo que los humanos y las computadoras podemos fallar, se deben incluir procesos de control para monitorear la funcionalidad interna del sistema, esto con el objeto de asegurar la integridad de los datos y la misma funcionalidad. Se debe considerar la hostilidad del medio ambiente incluyendo procedimientos de control de accesos para proteger el sistema de su uso sin autorización. Es decir, no olvidar que son usuarios finales quienes utilizarán el sistema.

Las nuevas funciones físicas muestran, en un nivel alto de abstracción, como se combinan los modelos de comportamiento dentro de un sistema. También muestran como un sistema está compuesto de procesos y como éstos interactúan con otros.

En otras palabras, estas funciones físicas muestran quienes pueden ingresar datos al sistema, qué tipo de datos, el momento en que deberán ingresarse y los procesos que los transformarán en información

Para el proyecto se incluyó un módulo de seguridad ya que se dividieron las funciones del sistema para que sean distintas personas las que tengan acceso al sistema.

Definición del impacto en el ambiente actual

"Los requerimientos físicos de un sistema tendrán un impacto en el ambiente en el que operará. No se deben perder de vista los requerimientos del hardware, software (base de datos), monitoreo de teleproceso, líneas de comunicación, dispositivos de almacenamiento de datos y los requerimientos del site" (1).

El objetivo de esta actividad es asentar este impacto en algún lado y sumarizar los requerimientos para que pueda funcionar el sistema. Estos requerimientos se recolectaron

durante el diseño de todos los procesos que conforman el sistema y se usan durante los diseños de arquitectura y plataforma.

Para el proyecto se debe mencionar que respecto a este punto se presentó una configuración sugerida, en la que se incluyen únicamente PC's. No se espera tener un impacto importante por los requerimientos de espacio y operación del equipo.

3.1.1 Diseño de la Organización

El diseño de la organización define las relaciones entre el personal de la empresa (usuario) y los procesos manuales requeridos por el sistema.

Para lograr un buen diseño de organización hay que determinar el impacto de los requerimientos del proceso manual del nuevo sistema en la estructura organizacional actual, como la descripción de puesto, equipo de oficina y documentos.

Definición de funciones del trabajo

Se requiere analizar todas las funciones manuales del nuevo sistema y las funciones de negocio que va a soportar. Los procesos manuales se comparan con los anteriores y se agrupan de tal manera que den sentido de negocios. Muchos puestos definidos de esta manera requerirán de ciertas habilidades y programas de capacitación que deberán revisarse y utilizarse cuando sea necesario.

Esta actividad ayuda a determinar las nuevas funciones de los empleados, por ejemplo, cuando se pedía la existencia de una determinada pieza, el encargado del almacén tenía que efectuar un conteo físico, ahora únicamente hará una consulta a las existencias o en su defecto la secretaria consultará una pantalla.

Diseño de documentos

Diseñar documentos requiere de una completa revisión de todas las interfases manuales del sistema. Un documento puede ser un archivo electrónico o documentos usados para preparar interfases del sistema. Se debe considerar almacenamiento y tecnología de explotación. Los archivos físicos donde se almacenen estos documentos tienen que ser accesibles para los usuarios.

Se incluye en el proyecto el diseño de algunas formas que serán utilizadas para ingresar información al sistema.

Definición de requerimientos de capacitación

Se analizan las especificaciones de trabajo para determinar qué tiene que poder hacer el futuro usuario del sistema para operarlo de manera correcta, es decir, se determinan las

características o el conocimiento necesario para desempeñar el trabajo. Estas habilidades se comparan con las que tienen los usuarios actuales para determinar los nuevos requerimientos de capacitación. Se pueden determinar otros requerimientos durante el resto del desarrollo del sistema y complementar los ya existentes.

El objetivo de esta actividad es determinar las necesidades de capacitación que en este caso son desde aprender el manejo de una computadora y utilizar windows hasta el propio sistema.

3.1.2 Selección de plataforma de aplicación

Seleccionar la plataforma de aplicación es determinar el hardware y el sistema(s) operativos necesario(s) para soportar la operación del sistema.

El objetivo que se persigue con esta actividad es seleccionar el mejor hardware/software que soportará el funcionamiento del sistema, considerando performance, compatibilidad y costo. Se debe considerar la inversión existente en hardware, software y entrenamiento antes de hacer una recomendación.

Aunque en este proyecto se tiene como requerimiento el utilizar el hardware disponible, es bueno recordar este punto de la metodología.

Definición de criterios de selección del hardware/software

Existen muchos criterios de selección para una plataforma, es muy raro encontrar un sistema que sea tan distinto o que tenga características tan específicas que no se deban tener consideraciones comunes. El criterio de selección dará peso o determinará una característica que debe ser obligatoria. Algunos criterios comunes son:

- Equipo actual, ya sea comprado o en renta
- Habilidades y capacitación del personal
- Exclusividad de licencias de vendedores
- Requerimientos de velocidad y volumen
- Compatibilidad con otros sistemas, hardware y redes

Evaluación de alternativas de hardware/software

Esta evaluación debe estar sustentada en un análisis costo-beneficio. Hay que mostrar al cliente los distintos factores a considerar. Este proceso puede ser "evolutivo" es decir, se comienza con amplias alternativas, posteriormente se evalúan, se seleccionan y se redefinen hasta que queda una alternativa viable.

3.2 Diseño Técnico

El diseño técnico es una de las dos fases del diseño, como sabemos la otra fase es el diseño de negocios. El diseño de negocios, como su nombre lo indica, se enfoca al aspecto del negocio, se incluyen muy pocos aspectos técnicos. En el diseño técnico no se requiere de tanta participación del usuario. Por ejemplo, en el diseño de negocios, se asegura que el sistema cubra con requerimientos de presentación de información y de desempeño de funciones, por otro lado, en el diseño técnico el enfoque es totalmente distinto y trata de asegurar que se puedan cubrir los requerimientos del usuario con la tecnología que se determinó. Ambas fases del diseño se desarrollan paralelamente.

3.2.1 Diseño de Arquitectura

El diseño de arquitectura es la definición específica de la configuración de hardware incluyendo la localización física y software así como sus parámetros de funcionamiento.

El objetivo de esta parte del diseño es proveer el ambiente necesario para desarrollar y operar el sistema, tomando en cuenta la estrategia tecnológica del cliente.

Análisis de arquitectura

Para asegurarnos que se cuenta con los recursos adecuados para el nuevo sistema y para evitar gastos innecesarios en hardware y software, se cuantifica el hardware, software e instalaciones de teleproceso.

Configurar es el proceso de decisión sobre el tipo de hardware y software necesario, así como el lugar donde se instalará cada uno de ellos, incluyendo por supuesto el nuevo sistema. La configuración incluye decisiones que tienen que ver como se conectarán las distintas partes del equipo, así como la interfaz entre el hardware y el software. La configuración del nuevo sistema debe satisfacer los requerimientos funcionales y de desempeño.

Es evidente que lo extenso de este análisis dependerá del tamaño de la aplicación, las localidades en las que se instalará o utilizará y el hardware necesario. La actividad que siempre se lleva a cabo es levantar un inventario.

Selección de otros componentes

Muy ligado al diseño de la arquitectura se seleccionan otros componentes que deben ser compatibles entre sí, verificando que cubran las expectativas del cliente. Algunos ejemplos de estos componentes pueden ser:

Sistema de teleproceso.- Verificar que sea compatible con la red actual, con los dispositivos y protocolos.

Base de datos.- Se deben de tomar en cuenta las consideraciones del diseño lógico y físico de datos. También debe permitir, cuando sea requerido, compartir e intercambiar datos entre la organización del cliente, los procesos existentes y el resto de los sistemas.

Para los casos anteriores también se consideran estándares, la planeación estratégica y el plan de informática. En este caso, debido a la plataforma a utilizar, restricciones de presupuesto y al mismo tiempo para seguir las tendencias tecnológicas, se seleccionó una base de datos que funciona en ambiente PC cuyo nombre es SQLBase. Esta base de datos es relacional, funciona a través del ODBC y permite elaborar un diseño bastante abierto basado en sentencias estándar de SQL. Con cambios menores, la aplicación podrá ser migrada a una base de datos más robusta que inclusive funcione en otro sistema operativo. Esta es una característica muy importante ya que cuando la empresa tenga crecimiento, podrá fácilmente aumentar su capacidad de almacenamiento y procesamiento de información así como su potencial de proporcionar servicios de red para todos sus departamentos.

3.2.2 Diseño Físico de Datos

En el diseño físico de datos se crean las definiciones de datos que se utilizarán por el sistema de acceso a archivos o el DBMS (Data Base Management System) para almacenar y acceder los datos.

El objetivo que se persigue es habilitar los programas y usuarios de un sistema para almacenar y sustraer datos de manera eficiente.

Diseño físico de base de datos/archivos

El diseño físico de las definiciones de datos que serán utilizados por el nuevo sistema, es el proceso de trasladar los requerimientos lógicos de datos, vistas lógicas de datos y las especificaciones de arquitectura en lay-outs físicos, determinando tanto los requerimientos de almacenamiento de datos, como los métodos de acceso.

En esta parte del diseño del sistema se utilizó la herramienta ERwin, que a partir de diagrama entidad-relación obtenido en el IEW se ingresó y se agregaron las columnas de las tablas. Posteriormente, se establecieron las relaciones entre las entidades.

Para efectos de facilitar el desarrollo del sistema se obtuvieron "áreas de estudio", que son vistas del diagrama entidad-relación, en donde se incluyeron las tablas y relaciones que son accedidas por módulos del sistema.

Una vez creado el modelo con las columnas y las relaciones se genera el script de creación de la base de datos. Una vez generado, se conecta con ésta y se genera el modelo físicamente.

3.2.3 Diseño Técnico del Sistema

El diseño técnico del sistema consiste en construir toda la estructura del sistema en la computadora, así como las funciones individuales de una manera consistente y eficiente.

Se busca desarrollar la estructura del sistema y el flujo de éste. Diseñar los programas que cubran las especificaciones funcionales, además de los requerimientos de performance. También incluyen los procedimientos de operación que serán necesarios para ejecutar los programas.

Diseño de la estructura y flujo del sistema

El diseño técnico de un sistema es el desarrollo del modelo de la estructura de un sistema que cubra la implantación de las funciones. Usando técnicas de diseño estructurado, se avanza en el proyecto considerando las necesidades funcionales de primer nivel (Top-Down) y las limitaciones externas. Las necesidades funcionales de primer nivel son por lo general factores de desempeño y relaciones generales de las funciones. Las características anteriores indican un tipo de implantación (on-line, tiempo real o batch) y conforman subsistemas y aplicaciones. En general los pasos son los siguientes:

Documentar la estructura del sistema

Se revisa el modelo de comportamiento del sistema, que puede estar compuesto de diagramas de flujo de datos, diagrama entidad-relación o diagramas de transición de estado, esto con el objetivo de identificar las áreas funcionales más grandes del sistema. La estructura del sistema es la representación top-down del sistema que lo divide en subsistemas. Cada subsistema representa un área funcional del sistema.

Los subsistemas se identifican como la estructura organizacional del cliente o un grupo de actividades de negocio.

Hay otros criterios para agrupar las funciones como la cohesión y el acoplamiento. El acoplamiento se refiere a la interdependencia entre módulos. Los módulos que son relativamente independientes se dice tienen poco acoplamiento. Esta característica es deseable ya que disminuye los problemas de mantenimiento.

La cohesión se refiere a la interdependencia de las actividades realizadas en un módulo en particular. Un módulo con alta cohesión es aquel que realiza una función perfectamente definida. Siempre es deseable tener alta cohesión en un

módulo. Un módulo con cohesión baja es aquel que realiza más de una función. Un módulo que realiza varias funciones que no tienen relación entre ellas tiene baja cohesión y es difícil de mantener.

Examinar las interfaces externas para actualizar el modelo de comportamiento del sistema

Se determina la dirección de los flujos de datos externos, ya sea para adentro o para afuera. Determinar quién es responsable de establecer y mantener la interfaz. Actualizar la documentación de éstas. Establecer exactamente las características, lay-out de archivos o tablas.

Diseñar el flujo del sistema

Revisar la estructura del sistema y el modelo de comportamiento para desarrollar el flujo del sistema. Se documenta los programas on-line y batch, sus funciones, datos de entrada y salida y métodos de cálculo.

Al utilizar la herramienta de análisis IEW, se obtiene la descomposición funcional del sistema ya que se modela la funcionalidad y las necesidades del área/sujeto de estudio, y además de obtener esta funcionalidad se muestran los flujos de información, que son parte del modelo de comportamiento del sistema.

Se determinan puntos lógicos de control, basados en las necesidades de datos (tablas) o en la similaridad de las funciones de usuarios finales. Estos aspectos encabezan el criterio fundamental para la interacción de un sistema y la seguridad.

Los puntos de control sugieren la necesidad de utilizar programas que pueden ser módulos comunes o interfaces para comunicarse con los procesos.

"La habilidad de la estructura del sistema para emular el modelo de comportamiento del sistema requerido se llama Flujo del Sistema. Hablar del flujo del sistema en un nivel alto (Arquitectura del Sistema) puede requerir de calendarización que puede ser satisfecha a través del operador o lenguaje de control. Esta calendarización es conocida como Calendario Maestro, que se incluye como una parte del manual de usuario y operación" (2). Lo anterior se refiere a que para concluir un proceso determinado, es probable que se tenga que alimentar datos previos, ejecutar procesos de validación, etc., y éstos deban tener algún orden que deba *calendarizarse*.

Un ejemplo sencillo y que se aplica en el proyecto es el proceso de ajustes a inventario. Para poder hacer un ajuste se debe primero registrar y posteriormente ejecutar un proceso de afectación. Si se ejecutara primero el proceso de afectación y posteriormente se capturarán los ajustes, no se tendría ningún efecto.

Diseño de programas

Una vez que se ha definido la estructura del sistema, las funciones lógicas se pueden convertir en rutinas o programas. Las funciones repetidas se pueden desarrollar en módulos comunes o copiando el original a los nuevos programas. Las interfaces y secuencias de control son definidas de tal manera que cada programa se comunique en el ambiente del sistema. También se incluyen en los programas las pantallas, reportes, algoritmos y vistas lógicas. Lo anterior conforma el primer esbozo de una especificación. Cuando se cuente en un proyecto con personal con habilidades muy buenas de programación o cuando se van a utilizar generadores de programas esta especificación preliminar será suficiente para la construcción.

A las especificaciones detalladas se les agrega texto (explicaciones), modelos o pseudocódigo para orientar a los programadores. Los modelos pueden ser usados como referencia sobre la lógica que debe seguir un programa antes de codificarlo. Es recomendable utilizar técnicas de programación estructurada.

Como se utilizaron prototipos para el diseño de negocios y técnico, las especificaciones no tienen gran explicación ya que si uno mismo diseña y construye la aplicación, no es necesario gran detalle.

Diseño de procedimientos de operación

El Calendario Maestro documenta los requerimientos del sistema para la intervención externa y el soporte de los operadores. Este es detallado y documentado como *Procedimientos de Operación* que formarán parte de la Guía de Operación. Los procedimientos describen las acciones que se deben seguir para responder a errores, ejecutar procesos y manejar archivos.

En este caso para el tamaño del sistema no es necesario un manual de operación ya que el usuario podrá hacer uso del sistema previa capacitación y con el apoyo del manual de usuario.

3.2.4 Plan de Conversión

El plan de conversión determina la manera en que se pasa del sistema existente al nuevo.

Se planea la nueva localización y configuración del hardware que utilizará el nuevo sistema. También la conversión de los programas existentes para implementar las especificaciones funcionales, así como los requerimientos de performance. Finalmente, se incluye la conversión de datos.

Determinar alcance de la conversión del hardware

Todo el hardware con el que se cuente actualmente es candidato potencial para utilizarse en el nuevo sistema. Se debe considerar que durante el tiempo que dure la instalación no se podrá utilizar por ninguno de los sistemas, el viejo o el nuevo. En este caso no hay sistema viejo y el tiempo que no se podrán utilizar las PC's será casi nulo ya que únicamente se instalará la aplicación.

Determinar alcance de la conversión de programas

Se identifican que programas del sistema anterior pueden ser utilizados en el nuevo. Posteriormente se determinan los cambios necesarios para que estos programas puedan funcionar dentro del ambiente del nuevo sistema. Los cambios deben ser documentados.

Al no tener sistema anterior no se llevará a cabo ninguna conversión de este tipo.

Determinar alcance de la conversión de datos

Esta conversión debe ser considerada como un proceso crítico, es esencial que se mantenga la integridad de los datos actuales. Utilizando el diseño físico de datos actual y la definición física de la base de datos, se determinan que datos pueden ser convertidos al nuevo sistema. Se puede dar el caso que no se necesiten algunos datos con los que se cuente en el sistema viejo, pero también que no se cuente con alguno y sea necesario capturarlo.

Otras consideraciones que se deben tener son:

- Respaldos de la información antes de la conversión
- Datos históricos (cuanto atrás)
- Pruebas, pistas de auditoría y controles (lo que sea necesario para asegurarse y verificar la exactitud de la conversión)
- Estimaciones (tiempo a invertir para llevar a cabo la conversión, reprocesos, etc.)

Básicamente la actividad de conversión que se llevará a cabo será de captura ya que no se cuenta con sistema actualmente.

Referencias del tema

- 1 SLC Manager's Guide pag 4-13
- 2 SLC Manager's Guide pag 5-20

4 CONSTRUCCION DEL SISTEMA

La actividad principal de la construcción es traducir el Diseño y Especificaciones en componentes del sistema. Los productos principales son programas que cuentan con pruebas unitarias, procedimientos de ejecución y documentación.

4.1 Organización

Participación del cliente

Cualquier interfaz externa que sea responsabilidad del cliente se documenta y al mismo tiempo se le da seguimiento. Los usuarios también pueden apoyar a los ingenieros de sistemas que están integrando los planes de prueba, mediante la consulta a ellos sobre los posibles datos a ingresar.

Desarrollo de estándares y procedimientos

Se desarrollan por los ingenieros de sistemas expertos en el lenguaje de programación. Hay algunas reglas para establecer estos estándares y procedimientos, pero pueden variar de acuerdo al lenguaje de programación. Los siguientes son estándares que se pueden establecer en cualquier proyecto:

Programación: de acuerdo a la herramienta se crean estándares como de tipos de variables, objetos tipo, funciones, acumuladores, contadores, constantes, etc.

Nomenclatura: se utiliza para librerías, procedimientos de compilación, copys o includes de módulos, programas batch, programas en línea, nombres de jobs, archivos o tablas.

Librerías de aplicación: por lo general se utilizan un conjunto de librerías tales como carga de módulos y documentación del sistema.

Librerías misceláneas: estas librerías se utilizan para almacenar documentación y procedimientos para programas.

Módulos comunes: estándares para módulos comunes, código reutilizable o rutinas. Son de gran ayuda ya que son un factor de incremento de productividad.

Cambios a la base de datos: se establecen estándares y procedimientos para coordinar estos cambios. Los ingenieros son responsables de: explicar el cambio,

identificar todos los subsistemas y programas que se pueden ver afectados por el cambio y entregar una notificación del mismo a todo el equipo.

Todos los cambios a la base de datos se canalizan a través de una sola persona que es DBA. Lo anterior es muy importante ya que ayuda a tener un ambiente de desarrollo controlado.

Desarrollar esqueletos

Sirven para no teclear información fija tal como, formatos de planes de prueba, definiciones de programas y formatos de ayuda.

Mantener el control de cambios del sistema

Los ingenieros son responsables de mantener el control de los cambios al sistema, actualizando la documentación, manteniendo al día las especificaciones y actualizando el log de modificaciones de los programas. También se debe estar al pendiente de los cambios del ambiente de pruebas tales como cambios a los nombres de archivos y contenido de los mismos.

4.2 Programación

La programación es la transformación del diseño detallado en programas. Una vez que se tienen las especificaciones se puede escribir el pseudocódigo para ayudar a documentar con mayor detalle la lógica del programa. Después la lógica se traduce en el lenguaje de programación adecuado. Los ingenieros deben conocer el lenguaje de programación de tal manera que se utilice efectivamente. También se conocen los métodos de acceso a archivos/tablas, frecuencia de ejecución del programa, demanda del usuario sobre el programa y la manera más eficiente de acceder los elementos de datos necesarios. Cuando se desarrolle un programa hay que considerar que alguna otra persona le dará mantenimiento y que las funciones que se encuentren dentro deben estar claramente documentadas.

Escribir, compilar y documentar programas

Los programas deben ser estructurados. Dichos programas son más entendibles y mantenibles. Uno de los propósitos de la programación estructurada es hacer que la lógica del programa sea fácil de entender para alguien que no es familiar con éste. La programación estructurada permite que los cambios sean más rápidos y que sean ejecutados por ingenieros que no son familiares con el código.

Se tienen que entender claramente todas las interfases internas del sistema. Una interfaz interna la podemos definir como una serie de datos que son compartidos por una o más partes del sistema. Ambas interfases, internas y externas se monitorean continuamente. Si

se requieren archivos externos se analiza cuidadosamente el lay-out para asegurar que se está enviando la información completa en el formato adecuado.

La programación es un proceso iterativo. En la medida en que se desarrolla el código se prueba utilizando, ya sea técnicas top-down, botton-up o una combinación de ambas, en realidad es como se obtengan mejores resultados. Después de cualquier cambio se ejecutan todos los casos prueba considerados, esto es para asegurar que el cambio no haya afectado alguna parte del código que ya estaba funcionando. El ingeniero debe ser capaz de comprobar que el programa cumpla con todos los requerimientos y estándares.

Pasos generales:

Analizar el problema

Las especificaciones contienen toda la información que el ingeniero necesite para codificar el programa, por lo que para elaborarla hay que tener un idea clara sobre lo que se tiene que hacer. También debe existir plena comunicacón entre el diseñador y el ingeniero encargado de codificar el programa, en el caso que la especificación no haya sido formalmente escrita.

Desarrollar rutinas comunes y copiar miembros

Una rutina común puede ser utilizable en varias partes dentro de un programa o en varios programas dentro de un sistema. Centralizar la función en una rutina permite repetir su uso sin duplicar el esfuerzo utilizado en desarrollo, pruebas o mantenimiento.

Los Copys de los miembros (estandarizar el código reutilizable) se crean en la librería del proyecto reservada para dicho efecto. También se utiliza el diccionario de datos para los copys.

Estas librerías se utilizan para aprovechar los recursos existentes, incrementar la productividad y la calidad.

Escribir el código

Es recomendable contar con un esqueleto de programa que contenga los estándares. El código se ajusta a los requerimientos de la especificación, debe tener lógica sencilla, ser eficiente y mantenible.

Se utilizan técnicas de programación estructurada. Un programa estructurado tiene una sola entrada y salida. El programa se divide en distintas funciones (módulos) o subrutinas y el flujo es de arriba hacia abajo. Las ventajas de la programación estructura son las siguientes:

- Fácil de probar y
- Fácil de seguir y entender la lógica, entre otras

El número de opciones dentro de un módulo es limitado para reducir la complejidad e incrementar la mantenibilidad. El incrementar la complejidad incrementa dramáticamente el costo del mantenimiento. El código estructurado incluye las tres estructuras básicas que se listan a continuación:

- Secuencia, sentencias que se ejecutan una tras de otra.
- Selección, un IF-THEN-ELSE permite escoger entre dos acciones. También se puede utilizar la estructura CASE, la diferencia radica en que se utiliza para acciones que tengan más de dos caminos.
- Iteración, Un DO-WHILE hace que se ejecuten determinadas acciones mientras una condición es verdadera.

Existen otras estructuras como el REPEAT-UNTIL que son consideradas como una derivación de las básicas.

Documentar el código

Incluir comentarios dentro del código sobre los siguientes puntos:

- Función
- Parámetros de entrada y salida
- Tablas
- Switches
- Terminación normal y anormal
- Códigos de retorno
- Notas sobre lógica especial
- Modificaciones

Las modificaciones tienen el registro de todos los cambios que se han hecho al código, la fecha y el nombre del ingeniero que lo realizó.

Cuando se realiza un cambio se actualiza toda la documentación referente al código y la de usuario.

Utilizar las librerías de pruebas

Estas librerías tienen datos prueba que son utilizables por todos los ingenieros, y se encuentran en el ambiente de desarrollo.

Construir pantallas y reportes

Todas las pantallas y reportes que fueron incluidos en el diseño de negocios se construyen con la herramienta que se seleccionó. Algunas veces dichas pantallas y reportes se definen internamente en el programa y en otras se utilizan generadores externos y se incluyen por medio de copys dentro de los programas.

Pruebas de escritorio

Una vez que ya se codificó el programa, se lleva a cabo una prueba de escritorio para verificar que se cubren todas las funciones.

Con esta prueba se ayuda a entender de mejor manera el código y a eliminar errores o a completarlo por funciones faltantes. El beneficio principal de las pruebas de escritorio es el ahorro en tiempo de proceso de computadora. Una vez que el programa se comienza a ejecutar, las pruebas de unidad requieren debuggers para corregir los errores y las correcciones requieren de teclear nuevo código, compilaciones, migración y pruebas. Las pruebas de escritorio permiten depurar el código sin la utilización de recursos computacionales.

No hay un método específico para ejecutar las pruebas de escritorio, pero algunas recomendaciones son seguir manualmente las líneas de código del programa con el apoyo de un listado y anotar los datos y los cambios que sufren a medida que avanzan por el código.

Compilar el código fuente

Una vez que ya se codificó el programa y ya se llevaron a cabo las pruebas de escritorio, se compila. Es importante tener todas las librerías necesarias, es decir, contar con las trayectorias en donde se pueden encontrar.

Solucionar todos los errores de compilación

Por general en las primeras compilaciones se producen errores debido a fallas en la sintaxis del lenguaje de programación. Estos errores se corrigen.

Llevar a cabo Walkthrough

Una vez que ya se tiene el programa compilado sin errores se determina si es necesario llevar a cabo un walkthrough. Un walkthrough es una revisión de escritorio del programa por otros elementos del equipo de trabajo.

Ligar el programa

Si el programa llama a otro, se ligan juntos. El programa que liga, combina diferentes programas en uno solo. Todas las ligas dinámicas y estáticas que tenga el código son resueltas en este proceso.

Revisar el código fuente

Se pueden utilizar pruebas unitarias para corregir errores de lógica o de transformación de datos en el código. La programación es un proceso iterativo, el código fuente es revisado y manipulado con base en los resultados de pruebas unitarias y/o walkthroughs hasta que los casos de prueba son cubiertos satisfactoriamente.

Desarrollar CL's para pruebas

Para el caso de sistemas que requieren CL o JCL (Job-Control-Language) se desarrollan indicando los archivos que se utilizarán y con que opciones se ejecutará el programa. Se consideran condiciones de no existencia de archivos, ABENDS (abnormal-end), límite de espacio por archivos, etc. .

Desarrollar CL's

Se desarrollan los CL o JCL necesarios para el sistema que operará en producción, en algunas ocasiones se pueden utilizar con modificaciones mínimas los utilizados en las pruebas. Entre las funciones que se suelen incluir están:

I/O de archivos

Secuencia de ejecución de procesos

Asignación de valores iniciales a parámetros

Establecer opciones de procesamiento tales como prompts, displays o mensajes

Ejecución de funciones jerárquicas

Funciones de terminación o salida

Compilar programas en batch

Utilerías de la base de datos

Migración de archivos

Funciones de impresión

Procedimientos de respaldo

Reportes de la base de datos, incluyendo el diccionario de datos

4.3 Pruebas Unitarias

Las pruebas unitarias son la prueba preliminar a la de integración.

El objetivo que se persigue con estas pruebas es verificar que todo el código de un programa sea utilizable y que mantenga la integridad de los datos. Esto se logra probando pequeñas partes del código a la vez, hasta completar la totalidad.

Las razones por las cuales se deben llevar a cabo estas pruebas son:

- Incrementar la flexibilidad de la secuencia de las pruebas unitarias
- Relativa facilidad para probar algo más pequeño
- Posibilidad de probar varias unidades al mismo tiempo

Durante las pruebas unitarias se desarrollan: planes de prueba individuales, casos prueba y datos prueba. Las pruebas unitarias validan la función que desempeña contra los requerimientos y también se verifica que la estructura del código es correcta y eficiente. El objetivo básico en estas pruebas es encontrar defectos y corregirlos.

No hay un orden específico para ejecutar las pruebas, por lo general uno tiene la libertad de escoger la secuencia en que se llevarán a cabo.

Este proceso de pruebas termina con la revisión o una prueba adicional por parte de algún compañero y la calidad de la prueba es responsabilidad del ingeniero.

Desarrollar plan de pruebas unitarias

Este plan describe la manera en que se llevará a cabo el proceso de pruebas. Se incluye el alcance, recursos necesarios y el calendario de las actividades. Un plan puede servir para varias pruebas pero cada unidad se debe probar por separado.

El alcance de las pruebas se limita de acuerdo al tiempo y presupuesto que se tenga disponible, además se consideran los siguientes puntos:

Evitar probar el hardware y software de un proveedor. No generar casos prueba para verificar el sistema operativo, el hardware o software de un proveedor externo tales como compiladores o utilerías.

Eliminar pruebas no necesarias como defectos de I/O, overflow u otros eventos que son inusuales.

Enfatizar las pruebas en cuestiones vitales para el sistema tales como paths, algoritmos, integridad de datos, etc. y no en los colores de la pantalla o formatos.

Probar rutinas comunes, subrutinas y copys. Por lo general estos elementos se prueban en una sola ocasión y no requieren que se vuelvan a probar.

Si se tiene un diseño modular, las pruebas también deben serlo.

Determinar el criterio de validación de las pruebas

Por lo general el criterio de validación de una prueba es que la ejecución termine normalmente y sin errores, pero se pueden dar casos en los que no es así.

Utilizar herramientas automatizadas

En caso que se cuente con herramientas de debugging se deben utilizar. Actualmente hay en el mercado algunas que ofrecen facilidades de seguir el código paso a paso e inclusive mostrar y alterar el valor de las variables.

Casos de prueba

Es evidente que la efectividad de las pruebas dependerá de la variedad de casos que sean procesados por el programa. Un caso de prueba contiene información detallada de una prueba específica. Además se incluyen las entradas o supuestos, procedimientos de ejecución y los resultados esperados.

La guía es la siguiente:

Diseñar los casos de prueba con base en los requerimientos de usuario, es decir, si fué requerido que se deban manejar X cantidad de movimientos diferentes para una transacción, debemos asegurarnos que los casos prueba tengan los X movimientos distintos.

Diseñar los casos prueba con base en la lógica del programa, en otras palabras se ejecutan al menos una vez todas las líneas de código, tanto en secuencias como en decisiones simples o múltiples.

Datos prueba

Cada caso de prueba tiene los datos que se utilizarán. El encontrar estos datos puede llegar a representar un esfuerzo considerable, principalmente cuando se requieren grandes volúmenes de información. Es recomendable respaldar esta información ya que se puede reutilizar cuando se tenga que repetir alguna prueba.

Probar las unidades

Las pruebas se pueden ejecutar de manera informal siempre y cuando no se pase a producción el programa. Ningún programa deberá pasarse al ambiente de producción hasta que cubra todas las pruebas formalmente y sean validadas con los criterios establecidos. Los resultados son documentados para utilizarse en futuras modificaciones al código. Los pasos son los siguientes:

Ejecutar la prueba

Ejecutar todos los casos de prueba según el plan. Los resultados obtenidos se comparan con los resultados esperados, las diferencias se documentan. Las fallas se analizan y corrigen, utilizando si es necesario, alguna herramienta de debugging.

Monitorear la prueba

Existen varias herramientas para depurar las aplicaciones. Particularmente las que permiten interrumpir la ejecución del programa en cualquier momento. Hay algunas que permiten cambiar el valor de variables para verificar el comportamiento del programa o para permitir completar un ciclo de proceso que se ha interrumpido debido a un error.

La versatilidad de las herramientas depende del lenguaje de programación utilizado, cada proveedor tiene la suya y la tendencia es que sean mejores y den mayores facilidades.

4.4 Desarrollo de Documentación

Se documentan todos los procedimientos a seguir por el operador y los usuarios.

Toda la documentación que se elaboró durante las fases de Diseño Técnico y de Negocios se utiliza para elaborar los manuales. Algunas consideraciones que se deben tener cuando se elabore la documentación son: estructura, contenido, formato, estilo y el hardware y software necesarios.

En general, el objetivo es proporcionar la mayor información sobre el sistema desde el punto de vista de la operación, distribución de salidas, solución de problemas, inventario del sistema, funciones, etc.

Desarrollo del manual de usuario

El propósito del manual de usuario es explicar como utilizar el sistema. El manual contiene cada uno de los pasos a seguir para utilizar cada uno de los elementos del sistema.

Se define claramente el propósito del sistema y las relaciones entre éste, los usuarios y los operadores. Si el proyecto es la modificación de un subsistema ya existente es necesario actualizar la documentación para reflejar las modificaciones. Si se trata de un nuevo sistema y no existe documentación, se desarrolla.

Se utilizan los lay-out de las pantallas, reportes y formas creadas en el diseño de negocios para incluir ejemplos. Actualmente se cuenta con herramientas para obtener dichas

pantallas directamente de los programas ejecutables. Un manual cuenta con la siguiente información:

La primera sección del manual incluye lo siguiente:

- **Resumen General.** Se incluye una descripción de alto nivel del sistema y sus funciones, también se especifica cuáles son On-line o Batch.
- **Resumen del Sistema.** Este resumen tiene más detalle que el general, se enfoca a los componentes funcionales del sistema. La descripción de las funciones diarias del negocio ayudan a adquirir un conocimiento básico de éste, así como del propósito de las funciones desarrolladas. Se incluye en un párrafo por separado los objetivos principales del negocio.
- **Describir las funciones.** Cada proceso esta compuesto de funciones. Se incluye la descripción de cada una, además del detalle de cómo se incorporan en el sistema.
- **Diagramas de Flujo.** Incluir los diagramas de flujo del sistema, tanto de las funciones On-Line como de las Batch. Es un resumen del sistema en una sola página utilizando los símbolos convencionales de diagramación.

En la segunda sección se pueden incluir los procedimientos

- **Resumen de procedimientos de log-on y log-off.** Describir paso a paso como firmarse en el sistema y como salir.
- **Instrucciones del lenguaje de usuario final.** Si se utilizan lenguajes "amigables" para el desarrollo del sistema, se incluyen instrucciones especiales para su uso.
- **Funcionamiento general de las pantallas.** Se muestra el flujo de proceso normal utilizado en las pantallas.
- **Procedimiento de ejecución de comandos.** Si el sistema tiene línea de comandos, se debe mostrar donde se localiza o como puede ser llamada.
- **Descripción del ingreso de datos en pantallas.** Se describe como se posiciona en la pantalla el cursor. Se incluyen las instrucciones para responder a los mensajes de validación.
- **Manejo de mensajes de error.** Incluir una serie de pasos a seguir por el usuario en caso de errores en la utilización de la pantalla. Esto incluye los diferentes tipos de errores que maneje la pantalla.

- Manejo del teclado. Descripción del teclado, localización de caracteres, teclas de función, etc., según se este utilizando. No olvidar los distintos tipos que pueden existir, incluir pc's emulando.
- Definición de las teclas de función. Se definen todas las teclas de función válidas para los programas o en su defecto donde se pueden visualizar.
- Descripción de funciones on-line. Breve descripción de las funciones que se tienen disponibles en la pantalla, como ayuda, impresión de pantallas, etc.

La tercera sección describe las funciones on-line

- Funcionamiento de las pantallas. Se describe el funcionamiento de todos los programas on-line y batch del sistema. También se incluye el lay-out, validaciones de campos, salidas, mensajes y teclas de función. Las pantallas se agrupan de tal manera que de sentido su utilización, por ejemplo, por módulo, por orden alfabético, por subsistema, etc. .
- Lista de programas. Una lista de todos los programas con su clave y una breve descripción de su función.

Desarrollo del manual de operación

El manual debe dar una visión general de todo el sistema, incluyendo archivos transmitidos, fuente de los archivos (lugar), fecha y hora en que están disponibles, interfaz con otros sistemas o sites, responsabilidades contractuales, procesos críticos y épocas críticas en el año.

También se documentan los parámetros necesarios para la ejecución de cada proceso, esto incluye procesos diarios, semanales, mensuales, anuales, etc. La documentación muestra las salidas esperadas para cada proceso.

Los operadores no deben tener autoridad para actualizar procedimientos de ejecución pero pueden restablecer un proceso para hacer ajustes durante las corridas.

Debe contener los procedimientos para ejecutar los procesos del sistema, calendarios de ejecución y las salidas de éste con su distribución. También debe incluir una guía para casos de contingencia.

Es importante que describa los componentes del sistema y los pasos a seguir cuando uno de estos no realiza lo esperado.

El manual de operación de una aplicación tiene que contener lo siguiente:

- Diagrama de flujo del sistema

- Archivos transmitidos. Descripción de todos los archivos que llegan o que salen, de donde vienen y a donde se envían. La frecuencia de uso, importancia de cada archivo.
- Responsabilidades contractuales. Documentar cuales son las responsabilidades, usuarios con capacidad de autorización de ejecución de procesos.
- Lista de procesos críticos. Procesos críticos para el negocio, se incluyen respaldos de información.
- Calendarización de procesos. Indicar los tipos de procesos que se ejecutan, por ejemplo, diario, semanal, mensual, anual, etc. .
- Descripción del proceso. Secuencia de ejecución de procesos. El manual se separa por tipo de calendarización.
- Plan de contingencia. Qué hacer en caso de falla en procesos críticos.
- Distribución de salidas. Los reportes o salidas en general, se envían a usuarios específicos.
- Inicio/Reinicio del sistema y procesos especiales. Condiciones por procesos especiales, puntos de control en inicio/reinicio del sistema.

Actualización de documentación

Se cuidan los dos puntos más importantes de la documentación de un sistema: el mantenimiento de la actual y el desarrollo de la nueva. Cuando se lleven a cabo adecuaciones o modificaciones al sistema se actualizan los manuales y las guías.

La actualización considera cuando menos el nivel de información que se tenga, es decir, la actualización deberá considerar toda información afectada dentro del manual actual.

En las nuevas aplicaciones o módulos se desarrolla la misma documentación para mantener el nivel de información que existe sobre los sistemas. Por otro lado, toda la documentación generada por el proyecto en sí, como plan de trabajo, requerimientos, diagramas de flujo de datos, etc., deberá incluirse en la documentación actual.

4.5 Prueba de Componentes No-Programa

Esta actividad se refiere a la validación de componentes del sistema que no son sujetos a pruebas unitarias.

El objetivo de esta prueba es verificar que estos componentes cumplan con los requerimientos y/o que sean utilizables.

Algunos ejemplos de estos componentes son:

- Procedimientos manuales
- Distribución de salidas

- Manual de Operación
- Manual de Usuario
- Comunicaciones
- Respaldos y recuperación
- Seguridad

Estas pruebas se llevan a cabo de la misma manera que una prueba unitaria de programa, es decir, se elabora un plan de prueba, se elaboran los casos prueba y se lleva a cabo la prueba.

En el caso de los procedimientos manuales, se ejecutan para verificar si no esta fuera de consideración algún paso. En general podemos agrupar estas pruebas como se indica a continuación:

- **Documentación.** Para probar la documentación se utiliza personal no familiar con el sistema para que verifique la operación del sistema con los manuales o los estándares con los programas, etc.
- **Comunicaciones.** Muchos equipos de comunicación cuentan con ellos mismos con procesos de prueba de comunicación. Se incluye dentro de la prueba el equipo de respaldo. Dentro de estas pruebas también se incluye la distribución de impresiones, es decir, los reportes son enrutados a los diferentes dispositivos de la empresa.

5 PRUEBAS AL SISTEMA

El concepto de prueba aplica en todas las fases de la metodología SLC. Tradicionalmente las pruebas incluyen únicamente pruebas al sistema y de aceptación de usuario, ya que las pruebas unitarias de los programas y las de componentes no-programa se llevan a cabo en la fase de construcción.

La fase de pruebas es la última oportunidad para asegurarnos que el cliente va a recibir el sistema que comprometimos. Las pruebas son tan importantes que se consideran como un proyecto a parte o un subproyecto. Al igual que en otras etapas del desarrollo del sistema, se requieren especificaciones, documentación y posiblemente programas para generar datos o verificar los resultados de las pruebas. Se comienzan las actividades tan pronto como se definan los criterios de aceptación.

La mayoría de los problemas que se encuentran se pudieron originar durante fases anteriores debido a especificaciones incompletas o inexactas, planes de prueba unitarios incompletos, mala codificación, etc. Detectar esas fallas o defectos es el objetivo único de esta fase.

Algunas técnicas de prueba son las siguientes:

- Paralelos: requiere que funcionen al mismo tiempo el sistema anterior y el nuevo. La información es alimentada en ambos sistemas o en algunos casos se pueden utilizar programas de conversión. Las salidas de ambos sistemas se comparan y se identifican las diferencias. Este tipo de pruebas se utilizan en grandes sistemas que pueden afectar las funciones vitales del cliente. Algunas ventajas de los paralelos son :
 - Todos los problemas del sistema son descubiertos y corregidos sin afectar el proceso del negocio del cliente
 - Si se requiere cambiar físicamente el equipo de la nueva aplicación, el proceso puede seguir normalmente
 - Se puede probar la efectividad del entrenamiento y la documentación

Algunas desventajas son:

- Es costoso operar ambos sistemas. Por lo general se requiere de apoyo adicional y el cliente no cuenta con el

- La comparación de resultados entre ambos sistemas puede ser difícil ya que los sistemas pueden tener características distintas
- Probablemente no se cuente con tiempo para operar ambos sistemas
- Modelo de oficina: Las pruebas se llevan a cabo en un ambiente que esta simulando producción. Se cuenta con datos reales de tal manera que se puedan probar todas las funciones del sistema. El volumen de información será tan grande como se acuerde con el usuario. Algunas ventajas son:
 - Los problemas que se reportan pueden ser repetibles más fácilmente
 - Las correcciones pueden ser aplicadas sin destruir archivos de producción
 - Las pruebas en paralelo son prácticas en un escala más pequeña

Para este proyecto se utilizó el modelo de oficina, ya que no se cuenta con un sistema actualmente.

5.1 Organización

Dentro de la organización de esta fase se determinan una serie de actividades a desarrollar. En proyectos grandes es recomendable separar las actividades y asignarlas a equipos de trabajo. Algunas actividades pueden ser:

- Control de librerías y migración
- Control de cambios
- Aseguramiento de calidad
- Administración de datos de prueba
- Control de pruebas
- Estándares
- Criterios de aceptación
- Herramientas y recursos

Mantener control de cambios

Es crucial documentar todos los cambios que se lleven a cabo durante las pruebas ya que algunas veces se requiere justificarlos. Al mismo tiempo que se ejecuten cambios se actualiza toda la documentación que se vea afectada.

Es importante seguir los procedimientos de movimiento de programas de acuerdo a las pruebas, ya que un programa que fue aprobado se protege en contra de versiones anteriores. Estos procedimientos incluyen el tener librerías separadas de acuerdo a los

distintos grados de aprobación de un producto. Para este proyecto se tuvo cuidado de manejar distintos subdirectorios para controlar las versiones del sistema.

Entre más cercana sea la prueba al ambiente de producción real se es más estricto con los programas. Aunque un programa pase una prueba se consideran puntos adicionales tales como: ¿la versión de programa es la correcta?, ¿los copys y los include son los actualizados?, ¿están cubiertos todos los estándares de documentación?, ¿se eliminaron los trace, display o debug de los programas? (cuando aplique).

Se mantiene el control de los cambios del sistema, los costos y los planes de trabajo. Los cambios pueden ser atacados según se detecten, pero cuando alguno pueda afectar el costo, fechas compromiso, calidad o exactitud de las salidas del sistema se debe conseguir una aprobación por parte del usuario.

5.2 Prueba del Sistema

Las pruebas del sistema sirven para verificar que el sistema cubra los requerimientos y objetivos de negocio del cliente. Dichas pruebas se ejecutan en las condiciones más cercanas a lo que será el ambiente de producción.

Otras áreas que se prueban son la intervención del usuario y del operador (si aplica), tiempos de proceso y el almacenamiento.

A medida que avanzan las pruebas se incrementa la complejidad y la rigidez que se tenga al evaluar los resultados. Si hay programas que tienen comunicación entre sí, ambos deben tener el mismo avance en las pruebas.

El control de cambios debe ser muy estricto, se registra la falla, se corrige y se aprueba.

Desarrollar Plan de Pruebas

Se elabora un plan que contemple una prueba de todo el sistema aún cuando en alguna de las fases no se obtengan los resultados esperados, en proyectos grandes, los elementos de equipo de trabajo tienen sus actividades asignadas de tal manera que las pruebas terminen al mismo tiempo.

Para elaborar el plan de pruebas se utiliza como insumo, toda la documentación con la que se cuenta tal como: componentes, funciones, descripción de procesos y la estructura del sistema.

A continuación se mencionan los pasos sugeridos para elaborar un plan de pruebas:

- Definir los objetivos de la prueba. Cada sistema se concibe como la solución a un problema de negocios. El costo de un sistema esta sustentado con el costo

del problema. Por lo anterior, los objetivos deben ser perfectamente claros y entendibles para todos los participantes del proyecto. Los objetivos deben ser medibles, para que con base en estos podamos evaluar el desempeño del sistema.

Los objetivos del sistema están definidos en los "criterios de solución", en este documento se especifica que el sistema es una solución a un problema de negocios. Entre más específicos sean los objetivos será más fácil desarrollar las pruebas que demuestren que el sistema cubre lo esperado. Por lo anterior, los objetivos deben ser cantidades discretas como "10% de mejora en el tiempo de proceso".

Se obtienen la mayor cantidad de datos para comparar como lo permita el cliente. La mejor medida de la efectividad de un sistema es probar con situaciones reales y comparando con las salidas actuales. Si el sistema no cubre con alguno de los objetivos se pueden hacer mejoras para lograrlo. La mayoría de los diseños tienen áreas susceptibles de mejora como desempeño, reducción de costo de operación o funcionalidad.

Muchos objetivos pueden probarse sin una prueba tan extensa como: exactitud, facilidad de uso, seguridad, redundancia, etc. Si el cliente va a juzgar el sistema con estándares o requerimientos que no fueron considerados en el diseño, el intentar corregir esta situación puede resultar un desastre.

- Obtener recursos. Si las pruebas se llevan a cabo en un centro de datos que no tiene soporte para actividades de desarrollo, se verifican accesos remotos y disponibilidad del equipo antes de comenzar las actividades.

Para este proyecto se verificó el espacio disponible en disco.

- Seleccionar el método. Se selecciona el método de acuerdo al sistema a probar. Los requerimientos del sistema indican las pruebas a incluir en el plan. A continuación se mencionan distintos tipos de pruebas:
 - Uso del sistema. Se intenta encontrar las dificultades que existan entre el usuario y el sistema. A pesar que el sistema sea técnicamente correcto, las funciones pueden parecer fáciles de usar o ser más prácticas que las de la aplicación actual. En caso que no lo sean, el usuario percibe lo anterior casi como si fuera una falla del sistema e impondrá esos requerimientos "implícitos". Para evitar esta situación se recomienda evaluar algún prototipo con el usuario.
 - Instalación. Cuando ya se cuente con un ambiente para las pruebas, ya se puede comenzar a probar programas, procedimientos y el mismo hardware, todo para determinar si el ambiente de producción puede ser

utilizado y probado. Si la instalación no es fácil y toma mucho tiempo, se puede repetir este proceso para mejorarlo. Se verifica que todos los datos fueron cargados correctamente.

- Disponibilidad y mantenimiento. El fácil mantenimiento del sistema es una de las mayores preocupaciones que se tienen durante el desarrollo de éste y se hace mención de ello durante todas las fases de la metodología. El código estructurado es sólo una parte de lo necesario para poder dar mantenimiento al sistema. La disponibilidad que brinda el sistema es constantemente evaluada mediante el seguimiento de su proceso normal. Otros factores que afectan la disponibilidad son la documentación y el entrenamiento; el no molestar a usuarios cuando se llevan a cabo cambios es sólo una visión parcial de la disponibilidad.
- Compatibilidad. Este tipo de pruebas aplican cuando se convierte un sistema ya existente en uno que va a funcionar en distinto hardware y sistema operativo o al instalar nuevas mejoras o módulos. Los cambios que sufre un sistema por este tipo de conversión aplican por lo general a todos sus componentes, por lo que se requiere probar todo el sistema. Algunas veces se puede comparar los resultados del sistema que está en producción y el nuevo.
- Tolerancia de volumen. Las pruebas de volumen se llevan a cabo para verificar que el sistema puede aceptar y almacenar grandes cantidades de datos. Estas pruebas dan una doble verificación, la primera es la administración de recursos, demostrando la habilidad del sistema para conseguir recursos y para determinar la necesidad total de almacenamiento. La otra es que, se pueden obtener datos estadísticos sobre el desempeño del sistema, que no se pueden obtener en casos de prueba.
- Costo del desempeño. Los gastos de operación del sistema comienzan cuando el sistema es aceptado por el cliente. Cuando el cliente este interesado en dicho costo se puede obtener de la siguiente manera: durante las pruebas de volumen se puede saber el espacio ocupado en disco, uso del CPU y redes, así como otros gastos como papel, formas, etc. Si los costos son altos puede ser signo que el sistema tenga un diseño pobre, si es posible aún, se debe corregir. Con los datos anteriores es posible obtener un presupuesto.
- I/O. Cuando hay actividad excesiva de I/O es posible que el programa y la lógica del sistema necesitan reevaluarse.
- Nuevo sistema en una integración múltiple. Una integración múltiple de sistemas es insertar un nuevo sistema dentro de un ambiente en

donde ya existen otros trabajando. Las pruebas se enfocan hacia el desempeño del nuevo sistema y en la interacción de todos los sistemas juntos. Las interfaces de todos los sistemas deben ser compatibles y el nuevo sistema no debe causar problemas. Los administradores de los otros sistemas están enterados de la instalación para que estén al pendiente de cualquier situación anormal durante las pruebas.

- Desempeño a presión. Este tipo de pruebas tienen que ver con el ingreso de grandes cantidades de información en un periodo específico de tiempo. Este tipo de pruebas valida la capacidad del sistema para conseguir y compartir recursos. Cuando la presión no es factor no se evalúa la capacidad del sistema para manejar colas, calendarización, priorización, manejo de interrupciones, tiempo de atención, acceso a recursos y contención.

También hay varias categorías de volumen a considerar:

- Carga típica
- Carga máxima prevista (ej.: a fin de año hay más información que al principio)
- Capacidad prometida (usualmente está estipulada por contrato)
- Cargas progresivamente más grandes hasta que falle el sistema

El nivel en donde ocurre la falla es importante para el desarrollador y para el cliente ya que determinan las necesidades de recursos y la vida del sistema. Por otro lado, las pruebas de Carga Máxima Prevista ayudan a verificar el comportamiento del sistema en el sentido de dar notificación adecuada acerca de la falla y proporcionar una recuperación adecuada.

Una prueba de Presión requiere un buen número de terminales y personal para operarlas, mientras que una Carga Típica puede resultar más fácil.

El tiempo de respuesta es un criterio que usualmente está incluido en el contrato. El tener tiempo lento aumenta las posibilidades de error de los operadores y los usuarios ya que pierden la atención de la actividad que están realizando. Las pruebas de tiempo de respuesta solo son válidas cuando se ejecutan con las de Presión.

También la eficiencia del I/O afecta el tiempo de respuesta. Este es un parámetro que generalmente puede ser controlado por medio de hardware, que generalmente es más costoso.

Para este proyecto se probó el uso del sistema. Se consideró esta prueba porque no se trata de ninguna integración, no funcionará en un ambiente multi-usuario y porque según la especificación de la base de datos a utilizar, el límite para el volumen de datos estará limitado primeramente por el espacio disponible en disco.

Desarrollo de casos prueba

Cada caso prueba sirve para validar un requerimiento y cubrir determinados objetivos. Un caso de prueba incluye la entrada definida para alcanzar un resultado determinado, una serie de pasos a ejecutar, los resultados esperados y el criterio que debe cubrir para considerarlo válido.

Los casos prueba verifican que todas las funciones del sistema se desarrollan de acuerdo con los requerimientos establecidos y con el diseño. Los defectos son encontrados y corregidos.

La guía general es la siguiente:

- Definir la prueba.- Identificar los objetivos a cubrir en cada caso prueba. Se incluyen sus componentes y funcionalidad. La compatibilidad entre los diferentes componentes se verifica en las pruebas de integración.
- Identificar entradas.- Las entradas de cada prueba pueden tener distintas formas: datos transmitidos por líneas de comunicación desde otros sites, librerías que residen en otro ambiente o la captura desde diferentes terminales. Lo importante es documentar los diferentes tipos que se utilizan para poder reproducir las entradas si es necesario.
- Documentar detalladamente los procedimientos de ejecución.- Obtener un documento que describa paso a paso como se ejecuta la prueba, incluye entradas, salidas, procesos manuales y automáticos.

En el proyecto se definieron los objetivos según los requerimientos de alto nivel. Las entradas de la prueba son datos capturados directamente por el usuario final.

Generar datos prueba

Los datos prueba son un muestra de los utilizados en producción, el utilizar una muestra de este tipo ayuda a confirmar la eficacia del sistema.

Cuando se van a generar datos se requiere del diagrama entidad-relación para asegurar que están incluidas todas las relaciones.

A continuación se especifica una guía general:

- Generar o adquirir entradas.- Se recomienda utilizar datos de producción. Para pruebas de volumen se necesitarán bastantes. También es recomendable incluir datos capturados.

- Plan de respaldos y recuperación.- Los procedimientos de recuperación de desastre se deben incluir en la prueba.

Prueba del Sistema

Probar el sistema es demostrar que todas las funciones del sistema cubren todos los objetivos definidos en los planes de prueba.

Un sistema se prueba a través de dos procesos, pruebas de integración y pruebas globales. La combinación de programas probados y componentes no-programa ya probados dan como resultado las funciones integradas. En las pruebas globales se prueban todas las funciones al mismo tiempo.

Las pruebas de integración pueden tener muchos beneficios cuando se ejecutan correctamente, además pueden reducir el tiempo a invertir en la actividad. Algunas ventajas de las pruebas de integración son:

- Cuando las unidades de prueba ya están disponibles por grupos, se pueden considerar como un pseudo sistema ya probado. Este método permite que se resuelvan muchos problemas de interfaz e interacción antes de la prueba global.
- El control de las pruebas es más enfocado a una parte del sistema. Monitorear un grupo pequeño de programas es más fácil que un sistema completo.
- La depuración es más simple. La fuente de posibles problemas se limita a un grupo funcional.
- Los grupos que son integrados más rápido pueden producir datos de prueba con calidad de producción para subsecuentes pruebas de otros programas. Por lo tanto, las pruebas de integración deben seguir el mismo curso que el flujo del sistema.
- La integración se convierte en un ejercicio de los procedimientos de control de cambios y validación.
- Se puede invitar a los usuarios para revisar los resultados de las pruebas, aportando prontas noticias en áreas problema. Cualquier grupo de programas sirve para mostrar características del diseño global tales como colores de las pantallas, estructura de los menús, defectos en los reportes o en los encabezados.

Las pruebas de integración son la combinación de unidades de programa en unidades lógicas para comprobar que fueron cubiertos los requerimientos. Un grupo lógico puede ser un conjunto de programas requeridos para llevar a cabo una función, un ciclo de proceso o corridas de programas con frecuencia particular (diaria, mensual o anual).

Durante las pruebas de integración se da seguimiento de cada prueba individual, en proyectos grandes, por lo general, se comparten recursos de tal manera que se puedan ejecutar varias al mismo tiempo.

Una prueba global es la interacción de todas las funciones combinadas en un sistema completo. Las pruebas globales comprenden varios tipos de pruebas, obviamente no todos los tipos de pruebas aplican a cada sistema. Las pruebas funcionales son las únicas que aplican para todos. La siguiente lista de tipos de prueba puede o no aplicar:

- Disponibilidad del sistema
- Instalación
- Mantenibilidad
- Compatibilidad
- Respaldo y recuperación
- Desempeño (volumen, I/O y presión)
- Seguridad
- Conversión
- Interfases

Ahora se muestra una serie de pasos recomendados para ejecutar las pruebas:

- Integrar grupos de programas relacionados.- Durante una prueba de integración, los componentes tienen alguna conexión lógica con otros. Un grupo debe estar relacionado con una función, tal como "ingresar un ajuste al inventario". Para procesar un ajuste al inventario debe pasar la "validación de ajustes", posteriormente puede pasar al módulo de "afectación de ajustes". Con lo anterior podemos ver que tres módulos forman un grupo de prueba de integración y cuando sean aprobados, estará lista una función útil para el cliente. Cada función se verifica independientemente.

Posteriormente, los grupos que fueron integrados satisfactoriamente se combinan en grupos más grandes de integración. En algunas ocasiones una aplicación completa puede ser probada como un grupo de integración. Este enfoque en las pruebas lo podemos considerar como bottom-up.

La recomendación sobre la manera de dar secuencia a las pruebas depende del sistema en sí, pero se identifican las siguientes secuencias de integración:

- Flujo normal del sistema
 - Práctica
 - Ciclo del sistema
- Revisar los resultados de las pruebas de integración.- Verificar que todos los programas y componentes no-programa que fueron probados se encuentren en las librerías o subdirectorios correctos.

- Ejecutar los casos prueba.- Ejecutar la prueba según los grupos que se formaron hasta completar todos los casos que fueron considerados para la ejecución de la misma.
- Simular ambiente de producción.- El sistema se prueba bajo las condiciones más realistas de operación. Los archivos deben contener volúmenes reales de información. La prueba del sistema se ejecuta acorde al plan.
- Evaluar los resultados de la prueba.- Asegurarse que todos los planes de prueba sean preparados, junto con ellos están los datos correspondientes para cada caso. También se verifica que todos los procedimientos estén completos y estén listos para su ejecución. Asegurarse que el producto cubra los requerimientos.

5.3 Pruebas de Aceptación de Usuario

Son la demostración al usuario que sus requerimientos están cubiertos con el sistema que tenemos y que está listo para instalarse.

El ciclo de pruebas tiene tres niveles: pruebas unitarias, pruebas del sistema y pruebas de aceptación de usuario. Las pruebas unitarias se preocupan de probar programas o módulos individualmente. Las pruebas del sistema validan el correcto funcionamiento de las interfaces internas y externas y aseguran que todo el sistema funciona bien. Las pruebas de aceptación de usuario son el nivel final del ciclo de pruebas.

Las pruebas de aceptación de usuario son generalmente ejecutadas por el usuario. Esta prueba confirma que el sistema en general opera conforme a los requerimientos y que satisface las necesidades de negocio del cliente. En algunos casos el usuario puede pedir que el desarrollador ejecute las pruebas mientras este observa y evalúa.

Estas pruebas, por lo general, son las que toman más tiempo, a pesar que el sistema ya está probado en sí. No se llevan a cabo si no se han completado exitosamente las pruebas anteriores.

Plan de pruebas de aceptación de usuario

Estas pruebas incluyen el ejercicio de ejecución de funciones del sistema y prueban o verifican la documentación, el entrenamiento y los procedimientos. Se identifican todos los componentes y requerimientos del sistema para probarlos. El plan incluye resultados predefinidos, junto con los datos que están diseñados para validar los requerimientos.

A continuación se especifica una serie de pasos generales para desarrollar un plan de pruebas de aceptación de usuario:

- Revisar los nuevos requerimientos de negocio y las unidades de trabajo.- Los nuevos requerimientos de negocio contienen todas las características funcionales del sistema. Una unidad de trabajo es un subconjunto funcionalmente independiente del resto del sistema y soporta una función bien definida.
- Definir los entregables al cliente.- Todos los entregables del sistema se listan aunque no participen en la prueba. Se describe cada componente no programa e incluir el criterio de aceptación y el método a utilizar para validarlos. Los entregables incluyen corridas de prueba y los archivos necesarios para generarlos y validarlos (si aplica).
- Validar requerimientos.- La validación de requerimientos es una parte muy importante del plan de pruebas de aceptación de usuario. El plan identifica todos los requerimientos del sistema que se probarán y forman la base de la matriz de validación. La matriz de validación de requerimientos es una referencia cruzada de requerimientos y casos de prueba, creada para asegurar que todos los requerimientos establecidos en las fases de diseño son cubiertos.

Desarrollar casos de prueba de aceptación de usuario

Los casos de prueba son una muestra representativa de los datos actuales de producción. Cada área funcional es verificada durante estas pruebas. Por lo tanto, los casos de prueba son globales. A pesar que el desarrollo de los casos prueba, es responsabilidad de los usuarios, el desarrollador debe estar presente para asegurarse que sean razonables y realistas. Los pasos recomendados a seguir son los siguientes:

- Identificar los requerimientos.- Es recomendable utilizar un enfoque top-down para listar los requerimientos más importantes y posteriormente refinar la lista para incluir más detalles.
- Definir todos los productos.- El usuario describe completamente el criterio de aceptación y la manera en que los productos son validados.
- Desarrollar la estrategia de pruebas.- El usuario debe utilizar una estrategia para las pruebas de aceptación. Para evitar pérdida de tiempo utilizando métodos incorrectos de prueba, el usuario participa en la planeación de la estrategia. Se recomienda probar por subsistema o requerimiento.

Desarrollar datos de prueba de aceptación de usuario

Los datos de prueba se enfocan a simular situaciones reales del negocio y a satisfacer un conjunto dado de casos prueba.

Probar el sistema para aprobación del usuario

La ejecución es el último paso de las pruebas de aceptación de usuario. Los usuarios prueban el sistema con casos prueba ya preparados y comparan los resultados esperados con los arrojados por el sistema. La prueba de aceptación de usuario demuestra que todos los requerimientos de usuario, requerimientos de negocio y objetivos están cubiertos. El proceso de validación está basado en la secuencia y plan establecidos en el plan de pruebas de aceptación de usuario.

Los siguientes son los pasos recomendados:

- Establecer el ambiente de pruebas.- Los pasos necesarios para establecer el ambiente de pruebas son similares a los del establecimiento del ambiente de desarrollo. Dependiendo de la naturaleza del proyecto algunos pasos se ejecutan al inicio cuando antes eran al final o viceversa.

En caso de trabajar en algún centro de datos se deben tener consideraciones especiales sobre la recuperación de datos o cualquier actividad que se desarrolle en éste.

- Involucrar al Cliente.- Este punto es vital, esta puede ser la primera vez que el cliente vea la funcionalidad final del sistema. El hardware y la red de comunicación deben ser similares a los esperados en el ambiente de producción.

Es recomendable no esconder o maquillar los resultados de la prueba, el cliente debe notar cualquier discrepancia entre la funcionalidad del sistema y los nuevos requerimientos de negocio.

- El cliente o usuario determina los datos para probar cada requerimiento.- Los datos son previamente validados para verificar los resultados. Cuando se obtienen los resultados se evalúan y se determina si son correctos o no. Si son correctos se obtiene una aprobación escrita, en caso contrario el cliente valida los datos de prueba y se debe llevar a cabo una más.

Se documenta cada problema y se comunica al cliente que se está corrigiendo el defecto. Dichas correcciones son priorizadas de acuerdo a la importancia de su impacto en la terminación de las pruebas y en el ambiente de producción.

- Revisar los resultados de las pruebas de aceptación de usuario.- Cuando se concluyen las pruebas se revisan los resultados. Al final de esta actividad, el cliente aprueba y acepta el sistema. Es recomendable obtener una firma de lo anterior. Por lo general, si las actividades están dentro de los planes de trabajo, se obtiene la aprobación del usuario de las fases previas y la firma final será solo una formalidad. De otra manera, las pruebas de aceptación de usuario no

descubren fallas muy internas del sistema y es común, que el usuario comience a elaborar una lista de "deseos". Entonces el proceso de revisión se convierte en un foro para resolver esas diferencias.

- **Completar las pruebas de aceptación de usuario.**- El esfuerzo invertido en las pruebas es necesario para demostrar al cliente que el sistema satisface los acuerdos establecidos. El resultado de estas pruebas de aceptación es un acuerdo escrito en donde se especifica que el sistema cubre los requerimientos del cliente.
- **Atender solicitudes de cambios o problemas.**- Los problemas que se descubren en las pruebas se documentan. Este esfuerzo incluye no sólo la documentación del problema en sí, sino la solución, quién encontró el problema y quién lo solucionó. Esta documentación es necesaria para cuando se repare el problema, se pueda regenerar la situación y se verifique que efectivamente fué solucionado.
- **Obtener aprobación del cliente.**- Se debe estar de acuerdo en que el sistema fué completado. Todos las peticiones de servicio, promesas y problemas son documentados, y se obtiene una firma condicional para que el sistema entre en producción. Este punto es muy importante para no alargar demasiado el proyecto.

6 IMPLANTACION

En la fase de implantación se instala el sistema ya terminado en el ambiente de producción y se capacita a los usuarios en su uso. El esfuerzo aplicado en esta fase requiere de un plan detallado.

Después de la implantación del sistema, se monitorea por el (los) desarrollador(es) y el equipo de soporte a la producción (cuando aplica). Después que transcurre un período satisfactorio, se entrega formalmente al equipo de soporte a la producción o cliente y comienzan las actividades de post-implantación.

6.1 Organización

Las tres actividades principales de esta fase son: planeación, instalación y post-instalación.

Desarrollar plan de trabajo

El método o la manera en que se implanta un sistema cae en alguno de los siguientes:

Paralelos. Requiere que el sistema nuevo y viejo funcionen al mismo tiempo. Las salidas de ambos sistemas se comparan y las diferencias son conciliadas. Este método se utiliza bastante en grandes sistemas donde cualquier defecto puede afectar las funciones vitales de negocio del cliente. Las transacciones que manejan estos sistemas pueden ser financieras, como la nómina o pagos de primas de seguro. Algunas ventajas de este método son:

- Cualquier problema que se encuentre se puede corregir sin afectar el proceso normal.
- Si el nuevo sistema necesita detenerse, el cliente puede continuar su proceso sin interrupción alguna.
- La efectividad de la documentación y el entrenamiento puede ser probada en vivo.

Algunas desventajas son:

- Es costoso operar ambos sistemas. Por lo general se requiere personal extra y excede los recursos disponibles. Algunas veces se requiere capturar en ambos

sistemas la misma información para mantenerlos sincronizados. Si se paga tiempo de proceso de máquina, se puede duplicar.

- La comparación de resultados puede ser difícil ya que los sistemas pueden tener diferentes características.
- Puede que no se tenga tiempo para correr ambos sistemas.
- Puede alargarse el tiempo de la implantación.

Por Fases. Se liberan únicamente porciones del sistema de acuerdo a la posibilidad de procesamiento. Por ejemplo, una compañía muy grande puede designar una localidad geográfica o una parte de su organización para que sirva como piloto para probar el nuevo sistema. Otra alternativa es instalar subsistemas de la aplicación. Por ejemplo, el subsistema de contabilidad puede instalarse en la primera fase, cuentas por cobrar en la siguiente y así sucesivamente. Este método se utiliza para aplicaciones medias a grandes que tienen funciones de media importancia a críticas. Algunas ventajas pueden ser:

- Los problemas que tenga el sistema se pueden corregir en la siguiente fase de instalación.
- Es posible conseguir personal más capaz.
- Se pueden hacer mejoras en el sistema.

Las desventajas son:

- No se tiene funcionando todo el sistema al mismo tiempo.
- La implantación en fases puede romper el ciclo de proceso de ambos sistemas, el viejo y el nuevo
- Se requieren interfases temporales entre los sistemas.
- Se requieren conversiones temporales que requieren ser monitoreadas constantemente.
- Las funciones que desempeñará el cliente aumentarán.

Completa. Se cambia de manera casi instantánea del uso del sistema viejo al nuevo. Un día la compañía procesa sus datos en el sistema viejo y al siguiente los procesa en el nuevo. Los usuarios deben estar totalmente comprometidos con el nuevo sistema. Este método se utiliza generalmente en aplicaciones pequeñas a medianas de funciones de baja a mediana importancia. Algunas ventajas son:

- Es el método más barato y produce resultados a corto plazo.
- El cliente recibe toda la funcionalidad del sistema al mismo tiempo.

Hay algunas desventajas como:

- El servicio que nuestro cliente brinda se puede más lento mientras los usuarios toman experiencia en el uso del nuevo sistema.
- Si llegara a ser necesario regresar al sistema viejo, podría haber déficit de personal para reprocesar los datos de acuerdo al sistema anterior.
- El plan de contingencia podría requerir programas adicionales para transferir datos de regreso del sistema nuevo al viejo.
- El cliente puede sentir que existe demasiado riesgo en este método.

Para este proyecto se utilizó la implantación completa debido a que no existía un sistema anterior.

6.2 Planeación de Implantación

Es el proceso de planeación para convertir, instalar y distribuir un sistema nuevo. En esta planeación se considera todo lo que se requiere para arrancar el sistema, correrlo y mantener la consistencia del ambiente de producción. Este proceso incluye tanto la instalación del sistema como la conversión de datos, así como procedimientos manuales utilizados para soportar la operación del mismo. Desde el punto de vista metodológico se tiene toda una serie de actividades a desarrollar, pero dependiendo del proyecto se seleccionan cuáles aplicarán o el detalle requerido para cada una de estas. Para el proyecto no aplicó plan de contingencia ya que no existía ambiente de producción en equipo alguno. Las actividades que fueron incluidas fueron las de planeación y lista de chequeo que son generales, pero que sirven de apoyo para elaborar la adecuada planeación de acuerdo al proyecto.

Desarrollar plan de implantación

El plan de implantación da una guía para instalar el sistema en el ambiente de producción. El objetivo de esta actividad es crear un plan que permita la instalación del sistema con mínimas interrupciones a las operaciones de negocio del cliente.

Los grandes rubros que incluye este plan son: instalación del software y hardware, migración de archivos, carga de datos, conversión de datos y entrenamiento. Los pasos generales son:

- Identificar todas las actividades requeridas.- Incluir toda la información relacionada con tiempos estimados, recursos necesarios, dependencias y otras consideraciones.

El plan debe ser tan detallado como sea posible. Incluir las siguientes actividades:

- Verificar que se cuente con todo el hardware y software necesarios, incluyendo equipo de oficina, se adquiere, envía, instala y se prueban materiales e insumos.
- Estimar los tiempos de corrida de la conversión de datos y de la instalación de programas, permitiendo tiempo para verificar los resultados, hacer cambios menores (si es necesario) y repetir las corridas.
- Crear backups intermedios durante la conversión de datos, para evitar pérdidas de información.
- Tomar en cuenta consideraciones especiales (por ejemplo, la necesidad de dar de baja el sistema para inicializar un dispositivo).
- Probar los programas de conversión, definir el orden en que se deben ejecutar y si es posible, determinar cuáles de éstos pueden ser ejecutados al mismo tiempo.
- Planear la instalación en días u horas en que el usuario no tenga demasiada carga de trabajo, tales como días festivos, fines de semana, etc.
- Identificar todos los archivos que se enviarán y recibirán.
- Identificar el destino de todos los archivos de salida que son creados.
- Analizar y definir los requerimientos de seguridad del site de producción.
- Preparar el ambiente de producción (incluir nombres de dispositivos, usuarios, etc.).
- Crear datos para probar el ambiente de producción.
- Probar el ambiente de producción.
- Migrar componentes que son específicos de la estructura del sistema y/o RDBMS que se utilizará.
- Migrar el diccionario de datos.
- Instalar y probar la base de datos en el ambiente de producción.
- Tener en cuenta los procedimientos propios del site para el cambio a programas.
- Migrar código fuente, utilerías, definición de pantallas, rutinas comunes y archivos de datos.
- Conipilar y ligar todos los módulos (si es necesario), o utilizar los de pruebas de aceptación de usuario.
- Enviar listados de los programas a un grupo especial de revisión.
- Crear e implementar procedimientos de respaldo y recuperación.
- Crear un Plan de Recuperación de Desastre (PRD), para el ambiente de producción.
- Llevar a cabo actividades previas tales como instalar el sistema de seguridad, cargar tablas, cargar la organización del cliente, agregar mensajes de error, etc.
- Probar la aplicación en el ambiente de producción.
- Ejecutar los programas de conversión y carga de datos.
- Examinar los nuevos datos para verificar la calidad de la conversión y carga.
- Establecer el procedimiento de soporte a usuarios.
- Entrenar a los usuarios.

- Llevar a cabo actividades de liberación de recursos tales como borrar usuarios utilizados en el desarrollo, archivos de prueba y utilerías de carga de datos.
- Definir actividades para entregar el sistema.

Cuando se desarrolla el plan se incluye tiempo y recursos requeridos para cada actividad es recomendable incluir un margen de seguridad. Es mejor sobrestimar el esfuerzo de alguna actividad ya que la instalación del sistema es la actividad más visible y que tiene mayor impacto en el proyecto.

- Desarrollar la secuencia de actividades.- Examinar las dependencias de cada actividad. Determinar las actividades que se pueden llevar a cabo al mismo tiempo.
- Asignar responsables de cada actividad.- La persona responsable de una actividad la ejecutará o verificará la terminación de ésta.
- Determinar la fecha de decisión de arranque o no del sistema.- Esta fecha se establece en el punto en que se pueda medir perfectamente la funcionalidad del sistema y para que, en dado caso, se desinstale la nueva aplicación y se reinstale la anterior y se pruebe el sistema sin afectar la operación del cliente.
- Establecer un mecanismo de reportes que permitan determinar y corregir rápidamente desviaciones al plan de trabajo.- Reportes semanales o juntas de trabajo, ayudan a mantener el control de las actividades durante la instalación. Las fechas deben ser observadas de cerca para alcanzar la fecha final.

Desarrollar plan de contingencia

El plan de contingencia se desarrolla para la eventualidad que la implantación falle. Este plan describe como se desinstala el nuevo sistema y se restaura el viejo ambiente de producción. Se podrían requerir distintos planes para cada site. Los pasos son los siguientes:

- Crear una lista de archivos y parámetros que son críticos para la operación del sistema viejo.- Si la instalación del nuevo sistema debe ser retrasada o rechazada, el sistema anterior debe continuar su operación sin ninguna interrupción. El propósito de este paso es determinar los archivos cruciales y los parámetros requeridos para mantener la operación completa del sistema viejo. Si existe un PRD, se puede utilizar como base para el plan de reinstalación.
- Mantener el ambiente de producción previo.- Todo el hardware y software del ambiente previo se mantiene hasta que la instalación exitosa sea confirmada.
- Desarrollar procedimientos para desarrollar la reinstalación.- Si existe un PRD, el plan de contingencia puede ser similar. Estos procedimientos pueden ser manuales y/o automáticos. Se siguen los requerimientos de cada centro de datos.

- Estimar el tiempo de reinstalación.- Pueden existir estimados dentro de PRD, son una buena base para los planes de contingencia. El plan incluye tiempo para probar el ambiente de producción viejo para verificar que el software/hardware, el sistema de seguridad, y todos los programas de la aplicación son ejecutados correctamente.
- Revisar el plan de reinstalación con el cliente.- Cuando se complete el plan se revisa con el cliente. Cualquier preocupación se analiza y menciona en posteriores juntas de revisión. El cliente y el equipo de trabajo se deben sentir cómodos con el plan.

6.3 Instalación

La instalación incluye preparar el ambiente del usuario. La preparación incluye la instalación del hardware y software, la instalación de la aplicación, la conversión de los datos existentes, la carga de los datos nuevos, entrenamiento a usuarios y soporte en el arranque del sistema.

Instalar el hardware/software

La instalación de la plataforma de hardware y software debe concordar con lo especificado en el "Diseño de Arquitectura". Esta actividad podría requerir el involucrar a proveedores, personal adicional, o el personal de soporte del site. Los pasos recomendados son los siguientes:

- Instalar la plataforma de aplicación como fué diseñada en las especificaciones de arquitectura.- Se deben tener algunas consideraciones especiales como alumbrado, tomas de corriente, aire acondicionado, hardware, software de soporte y equipo y muebles de oficina. Podría incluir inclusive reconfigurar el hardware actual.
- Probar el hardware y software para asegurar que la plataforma de la aplicación este operando.- Se prueba todo el hardware, así como la habilidad de los operadores para utilizarlo. Se verifica todo lo relacionado con redes, terminales y líneas de comunicación. El software a probar incluye el sistema operativo, DBMS y herramientas de desarrollo, entre otros. Los procedimientos también se deben probar.

Convertir y/o cargar datos

La conversión de los datos del cliente es esencial para conservar la integridad de la información con la que cuenta. También se cargan datos nuevos que requiere el sistema. Por lo general la conversión se lleva a cabo a través de programas que se utilizarán una única ocasión.

El tiempo que tarde la conversión es muy importante si el cliente esta utilizando algún sistema automatizado. La conversión de datos es generalmente un proceso largo que

puede tardar hasta días, dependiendo del tamaño del sistema. Para hacer más ligero el paso al nuevo sistema es recomendable hacer esta conversión en fines de semana largos, si es posible. Es importante considerar los tiempos con holgura por si se necesitara. Los pasos generales son:

- Poner fecha para la conversión de datos.- Hacer la lista de actividades y las fechas de terminación de cada una de éstas.
- Respaldar los archivos o base de datos del sistema actual.- Se crea un respaldo del sistema viejo y se tiene a la mano por si se requiere.
- Examinar los archivos de entrada que se reciban de fuentes externas.- Si es posible, se verifican las versiones o fechas. Verificar el contenido y asegurarse que los que se envían se reciban correctamente.
- Ejecutar la conversión en la secuencia planeada.- Si la conversión de datos y la carga de estos incluye varios pasos, se hacen respaldos intermedios para minimizar el tiempo requerido para recobrar datos perdidos.
- Verificar el éxito de cada conversión o carga.- Los datos se verifican cuidadosamente antes de poner en marcha el sistema. Si la información es errónea, será difícil operar el sistema. Datos inválidos pueden causar serios retrasos en la instalación y es posible que tengan efectos negativos en la operación diaria del cliente. Los programas de conversión deben verificar que los datos sean correctos, especialmente durante la carga inicial. Si no se verifican los datos se pueden acarrear errores en la operación del sistema.

Específicamente en el proyecto, no aplicó la conversión de datos pero si se elaboró un plan para capturar toda la información que se tenía en papel.

Instalar la aplicación

El objetivo de esta actividad es instalar el nuevo sistema con el mínimo de interrupción a la operación del cliente. El plan debe especificar la fecha en que el sistema necesita estar disponible al usuario. Todo el hardware, software y datos de producción deben estar en su lugar, listos para operar en ambiente de producción real. Los siguientes son los pasos a seguir:

- Respaldar el ambiente de producción del sistema anterior.- Si la instalación del nuevo sistema no es exitosa, será necesario ejecutar el plan de contingencia. En este caso, el ambiente de producción anterior es restaurado para que continúen las operaciones normales del cliente.
- Migrar la nueva aplicación al ambiente de producción.- Verificar que todos los componentes del software han sido instalados como se planeó y esta disponible.

- Compilar y ligar todos los módulos en el nuevo sistema.- Verificar el éxito de este proceso, se crean listados de compilación para esto.
- Hacer respaldos de la aplicación y base de datos.- Cuando se ha instalado el sistema y se ha verificado se hace un respaldo del ambiente de producción.
- Instalar los datos indispensables para propósitos de prueba.- La prueba incluye suficientes datos para verificar la instalación exitosa de cada componente de la nueva aplicación.
- Probar el nuevo ambiente.- Sería deseable ejecutar cada componente del sistema al menos una ocasión antes de considerar la instalación como exitosa. Se debe hacer lo necesario para que el desarrollador se sienta seguro que la aplicación funcionará correctamente.
- Restaurar los archivos y/o base de datos del respaldo anterior.- Cuando se completó la prueba se restaura todo el ambiente que se tenía. Se verifica nuevamente el éxito de esta operación.
- Elaborar un reporte de la instalación.- Documentar la información que se obtuvo de los pasos anteriormente ejecutados.

Proporcionar la capacitación

La capacitación es importante para obtener la satisfacción del cliente. La capacitación incluye la instrucción, documentación y soporte en la operación. La capacitación provee al usuario de conocimientos sobre el sistema para que ejecute su trabajo y se enfocan a enseñar a utilizar eficientemente las funciones que tiene el sistema.

Después de la capacitación, el usuario se debe sentir cómodo de utilizar el nuevo sistema. De alguna manera, el como se sienta el usuario al utilizar el sistema es un termómetro del éxito de la capacitación. Los pasos son los siguientes:

- Elaborar plan de capacitación.- La capacitación se planea teniendo en mente las actividades del usuario. No planear sesiones en horas o fecha pico. Una capacitación bien planeada permite que el usuario tenga toda su atención en esta. Los grupos deben ser pequeños para permitir atención personal. Se debe contar con todo el equipo necesario tal como: proyectores de acetatos, pizarrones, equipo de video, terminales y/o PC's, etc. .

Es probable que los usuarios no se involucraron en el desarrollo del sistema y se sientan excluidos de este. Los usuarios que no están comprometidos con el sistema pueden afectar el desempeño de este, por lo tanto, es recomendable insistir en tener actitud positiva al nuevo sistema. El instructor debe enfatizar el importante rol de

usuario en la implantación. Los instructores identifican específicamente y enfatizan los beneficios que tendrán los usuarios del sistema, tal como ahorro de tiempo, reducción de actividades monótonas, eliminación de papeleo excesivo, facilidad de uso, características de ayuda a usuarios, tiempo de respuesta y disponibilidad.

Se debe incluir un ejercicio completo en la utilización del sistema para verificar la efectividad o comprensión de la capacitación. La meta de la capacitación es dar una visión global del sistema y entrar a detalle en el área en la que el/ella son responsables.

- Practicar las sesiones de capacitación.- Si es posible se practican las sesiones de capacitación para verificar que sea adecuada y que se lleve a cabo en el tiempo estimado.
- Probar el equipo que se utilizará en la capacitación.- Asegurarse que todo el equipo a utilizar funcione adecuadamente. Verificar que se cuenta con los insumos necesarios (hojas, lápiz, marcadores, borrador, etc.).
- Dar la capacitación.- En algunas ocasiones es bueno proporcionar la documentación de usuario. Todas las preguntas del cliente se deben contestar y, si es necesario agendar para una sesión posterior.
- Elaborar un reporte.- Utilizar toda la información que se obtuvo del proceso anterior.

Soporte al Arranque del Sistema

El día en que el sistema esta disponible por primera vez para el usuario, es un punto crítico del proyecto. Todo el nivel de satisfacción del cliente se ve afectado significativamente, por lo que se debe tener una buena primera impresión. La atención a problemas debe ser rápida y efectiva.

Se debe monitorear el sistema para detectar posibles problemas.

6.4 Post Instalación

Esta actividad determina que, una vez instalado el sistema, este funcionando correctamente en el ambiente de producción. La responsabilidad del desarrollador no termina hasta que todos los problemas que se tengan durante el desarrollo hayan sido resueltos. El objetivo es obtener la aceptación del usuario para liberar la responsabilidad del sistema. Cuando se obtiene esta aprobación comienzan las actividades de soporte a la producción.

Monitorear el Sistema

Idealmente, el sistema funciona correctamente y no se detectan defectos después de la instalación. El desarrollador se mantiene al tanto durante un período razonable para verificar que el sistema fué correctamente instalado y que el usuario es autosuficiente para operarlo. El objetivo de monitorear el sistema es mantenerlo disponible para el cliente y optimizar su eficiencia.

Con dos ciclos de ejecución del sistema se puede verificar que esta generando correctamente todas las salidas, y que éstas pueden ser utilizadas por los usuarios. Más adelante durante la operación del sistema, se podría requerir soporte, como en procesos trimestrales o de fin de año.

Los desarrolladores y el personal de soporte a la producción deben estar en el mismo lugar. Cuando menos una persona de soporte debe estar presente para contestar las preguntas de los usuarios. El personal de soporte documenta todas sus actividades para que pueda ser apoyado por el personal de desarrollo. De esta manera, poco a poco, se pasa la responsabilidad del sistema a soporte a la producción. Hay una serie de pasos generales que son:

- Revisar las salidas del sistema.- Se recolecta una muestra de todas las salidas del sistema, para verificar su exactitud y compararlas contra los requerimientos del cliente. Cualquier problema que tenga el sistema se debe analizar y elaborar una Solicitud de Servicio del Cliente (SSC).
- Analizar el espacio en disco, uso de CPU e I/O.- A medida que este funcionando el sistema, se analiza el uso de los recursos mencionados con el objetivo de determinar si están dentro de lo que se tenía previsto.
- Analizar el desempeño del sistema.- Se debe tomar nota de los tiempos de proceso y compararlos con los requeridos por el cliente.
- Analizar violaciones de seguridad.- Si se está registrando en el sistema de seguridad accesos indebidos, analizar las posibles causas de éstos.

Revisar la Implantación

Las revisiones del cliente son un proceso que se da continuamente durante toda la fase de implantación. Es recomendable documentar cada paso de la implantación respecto a su cumplimiento y exactitud. Se elabora un reporte general que es revisado por el cliente, lo aprueba y acepta el sistema. El cliente concluye la revisión de la instalación del sistema y se firma la aceptación formal.

Después que se completó la instalación, es bueno revisar y reflexionar sobre los logros, los problemas y elaborar conclusiones. Esta actividad es muy utilizada ya que ayuda a perfeccionar la técnica y a lograr la satisfacción del cliente y la personal.

Entregar el sistema al cliente

La entrega del sistema debe ser casi transparente para el usuario, es recomendable involucrar a personal que se encargará del mantenimiento durante las fases de Construcción e Implantación. En caso que no se pueda hacer esto, tardará más tiempo la entrega ya que la información sigue siendo la misma, pero se requiere capacitar en la operación y mantenimiento del sistema. Las actividades son las siguientes:

- Liberar recursos.- Por lo general, siempre quedan archivos y documentación utilizados durante la implantación que ya no son necesarios. Después que ya se entregó formalmente el sistema se respalda esta información y se libera el espacio. Se decide si la información se quedará permanentemente o sólo será por un tiempo determinado, esto dependerá de la naturaleza del sistema y de los acuerdos contractuales de soporte y mantenimiento.
- Actualizar la documentación.- Antes de entregar formalmente el sistema se debe asegurar que la documentación está completa y actualizada. La documentación a entregar es: manual de usuario y operación, especificaciones de programas (si hay), listados de programas, planes de prueba y casos, resultados de pruebas y material de entrenamiento.

7 CONCLUSIONES

Hoy en día se requiere de profesionistas especializados que puedan resolver problemas, otorgar ventajas competitivas a las organizaciones, mejorar su productividad y rediseñar los procesos de negocios aprovechando las oportunidades que brindan las tecnologías de información. Esta actividad es cada vez más importante en el entorno que vive nuestro país.

Las nuevas tecnologías de computación han modificado substancialmente las posibilidades para almacenar, acceder, manejar y compartir información en las organizaciones. Gracias al uso de estas nuevas tecnologías, múltiples organizaciones han podido darle un mayor valor agregado a sus productos y/o servicios y modificar sus procesos administrativos y de producción para hacerlos más ágiles, flexibles, rentables, innovativos y orientados al cliente.

Dentro de las tecnologías de información se encuentran una gran variedad de elementos dentro de los cuales hay metodologías y herramientas auxiliares para el desarrollo de sistemas.

Analizando las metodologías puedo decir que el trabajo de todas las personas que estudiaron desde los inicios de la computación la manera de organizar esta actividad de tal manera que sea repetible y mensurable está ahora unificado en metodologías, que ayudan al desarrollo de sistemas. Sin hacer menos al principal actor dentro de esta actividad que es el ingeniero de sistemas, podemos decir que la utilización de metodología ayuda a obtener mejores resultados.

Todas las metodologías tienen una serie de pasos o actividades a seguir, normalmente es una lista bastante grande que al parecer puede alargar mucho más de lo que esperamos cualquier proyecto. No se debe considerar que son una receta de cocina con la que siempre saldrán bien las cosas, se debe seleccionar de acuerdo a un criterio, las actividades que se desarrollarán en el proyecto. En que se puede basar dicho criterio, puede ser de acuerdo a lo que pague el cliente, al tiempo, los recursos, el proyecto en sí mismo. El conocimiento en la industria del cliente ayuda bastante para saber que actividades se tienen que reforzar.

Una de las claves para asegurar el éxito de proyectos de sistemas, es involucrar al cliente en el proceso. Las formas de participación pueden ser mediante entrevistas, examinando documentos o productos elaborados por el cliente, estudiando flujos de trabajo, por nombrar algunas. Aún cuando involucrar al cliente es crítico, este trabajo de tesis no tiene referenciado cada actividad de interacción con éste. Se tiene que utilizar el juicio personal para determinar si el cliente debe ser involucrado.

Mi experiencia profesional me dice que la metodología es más fácil de asimilar cuando se aplica a situaciones reales. Por mencionar un ejemplo, es realmente difícil obtener un diseño de pantalla o reporte aprobado cuando nunca se ha trabajado con un usuario, esta práctica se me facilitó hasta que la utilice normalmente día a día. Creo que la metodología que aprendí en la facultad no difiere conceptualmente hablando de la que presento en este trabajo de tesis. La principal diferencia es que la que utilicé es un modelo modificado de acuerdo con experiencias reales de una empresa cuyo giro es la integración y desarrollo de sistemas.

El objetivo de cualquier metodología es normar una serie de actividades para poder llegar a un resultado esperado. En ese sentido, la utilización de éstas es indispensable para el desarrollo de sistemas. Algunos beneficios son:

- Reducir el riesgo de falla en el proyecto, ya que la metodología está probada
- Consistencia, que mejora la eficiencia del mantenimiento y las mejoras
- Acuerdos entre el cliente y el equipo de trabajo sobre responsabilidades y entregables (productos)
- Productividad y calidad a través del uso de herramientas compatibles y conjuntos de habilidades

La metodología es bastante completa ya que incluye todas las actividades necesarias para que un sistema se pueda seleccionar, diseñar, construir, implantar y mantener. Las fases son:

Definición.- Se delinea una oportunidad o problema y se elabora un plan para aprovechar o solucionarlo.

Análisis.- Se definen los requerimientos del sistema para resolver el problema o para que el cliente aproveche la oportunidad

Diseño de negocios.- Se determina junto con el cliente la forma en que se cubrirán aquellas partes del sistema que afectarán los procedimientos dentro de la empresa.

Diseño técnico.- Se determina la forma técnica en que se desarrollará/instalará el sistema.

Construcción.- El diseño se convierte en procedimientos utilizables, programas y documentación.

Pruebas.- Se verifica junto con el cliente que el sistema cubre los requerimientos definidos en el análisis.

Implantación.- Se instala el nuevo sistema y se verifica que este listo para funcionar y que el usuario está listo para utilizarlo.

Soporte a la producción.- Se opera y controla el sistema y si es apropiado, se modifica o se canalizan los problemas o solicitudes de mejoras a equipos de trabajo que repetirán las fases de la metodología como sea necesario.

A medida que avanza la tecnología de información salen al mercado nuevas herramientas de desarrollo de sistemas y con ello siguen evolucionando las metodologías. Hoy en día se comienza a hablar de prototipos de rápida construcción y optimización, mediante la

utilización de lo que podríamos llamar iteraciones con el resto de las fases de la metodología.

Las herramientas auxiliares para el desarrollo de sistemas ayudan a aplicar de manera eficiente y efectiva técnicas de ingeniería de información. Esta ayuda computarizada es llamada comúnmente "CASE".

Analizando las características de la estación de análisis del IEW, tiene dentro de su funcionamiento la aplicación de la técnica de obtención de diagramas entidad-relación que Peter Chen escribió alrededor de 1976. También incluye el modelaje de procesos tal y como lo describieron Chris Gane y Tom DeMarco en 1970.

Puedo asegurar que estas herramientas auxiliares son útiles en la medida en que se cumplan las siguientes premisas: se tenga definido perfectamente bien la función que desempeñará dentro de un proyecto y que los ingenieros la sepan utilizar.

Si una herramienta se utilizará como el elemento que generará los documentos de análisis, diseño o los programas fuente, debe ser considerada como vital para el proyecto. Si por otro lado únicamente se utilizará como generador de documentación técnica para efectos de cumplir con ciertos requisitos contractuales, el papel en el proyecto es totalmente secundario. Por otro lado, puede resultar un fracaso el comprometer en un proyecto el uso de una herramienta cuando no se tiene dominio en la utilización de esta. Por lo general en proyectos grandes se debe contar con un líder técnico que sea quien guíe al resto del equipo que tiene conocimientos generales. Como en cualquier herramienta hay un proceso de aprendizaje, a medida que se conoce más mediante su utilización, se puede aprovechar de mejor manera el potencial de ésta.

Respecto a la automatización de funciones en la industria quiero puntualizar su papel como elemento de cambio en todos los niveles de la organización de una empresa. El contar con sistemas con bajo mantenimiento diseñados para cubrir necesidades específicas ayuda a disminuir costos dentro de la operación. Como sabemos el mundo esta en un proceso de integración donde el comercio tiene un papel preponderante. Los países que ofrezcan mejores productos al precio más bajo serán los que tendrán la ventaja en el desarrollo interno y global. Actualmente es difícil imaginar cualquier empresa sin el soporte de sistema de computación, por lo que se requiere que ofrezcan gran beneficio a bajo costo.

La empresa que actualmente tiene funcionado el sistema desarrollado en este trabajo, ha experimentado cambios que la ayudarán a afrontar de otra manera sus problemas de mejora de procesos en cualquier aspecto. Como sabemos la mayoría de las empresas de nuestro país se consideran de pequeño a mediano tamaño. Casi todas estas son negocios familiares, que en la mayoría de los casos no tienen un sentido muy claro sobre las mejoras en procesos de producción, aumento de calidad automatización de funciones, entre otras. Se debe provocar el cambio para que estas empresas den el salto a la

modernidad y tengan opciones reales de crecimiento para poder competir en este creciente mercado mundial.

Finalmente, en todo este trabajo se hizo especial énfasis en la relación con el cliente, en el conocimiento en la industria en la cual vamos a trabajar y el aspecto de negocio. Es muy importante tener claro que actualmente más que antes, cualquier proyecto debe tener fundamentación costo-beneficio. Algunos beneficios se pueden obtener al utilizar la nueva tecnología.

Desde mi punto de vista, para poder aprovechar al máximo las nuevas tecnologías es necesario tener conocimiento conceptual y práctico de éstas y por otro lado entender el funcionamiento de los diferentes aspectos y/o componentes que conforman a la empresa. Sin embargo, he podido apreciar cierto divorcio entre las áreas administrativas y las responsables de desarrollar aplicaciones de tecnologías de información. Creo que hace falta crear un "puente" entre esas áreas para poder aprovechar al máximo por un lado los conceptos y técnicas de negocios y por otro la aplicación práctica de dicha tecnología.

A TERMINOLOGIA Y SIMBOLOGIA PARA DIAGRAMAS ENTIDAD RELACION

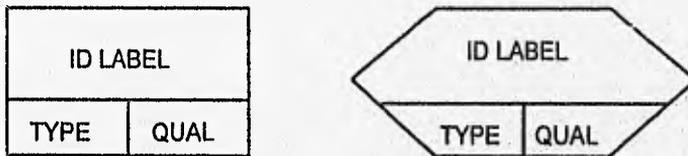
En este apéndice se detalla la terminología y simbología a usar para el desarrollo de diagramas entidad-relación.

DEFINICIONES

Como primer paso definiremos entidad, relación, cardinalidad, participación y restricciones de borrado, que son conceptos importantes para el desarrollo de los diagramas.

ENTIDAD.- Son los objetos del negocio sobre los cuales la empresa quiere mantener información, generalmente se expresa como un sustantivo.

Una entidad se representa como un rectángulo o hexágono nombrado en el diagrama de entidad relación, como se muestra en el siguiente diagrama.



Cada entidad se compone de cuatro componentes que describen la entidad. Estos componentes son:

- Identificador.- Nombre interno para uso en CASE
- Etiqueta de la entidad.- Descripción
- Tipo de entidad
- Calificador

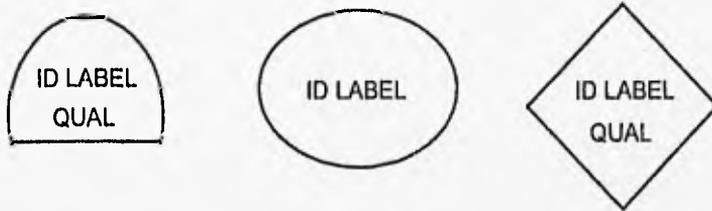
En IEW sólo es posible representar la entidad como un rectángulo nombrado.

RELACION.- Define como se relaciona una entidad con otra. Estas relaciones ayudan a definir las reglas de negocio que gobiernan los datos.

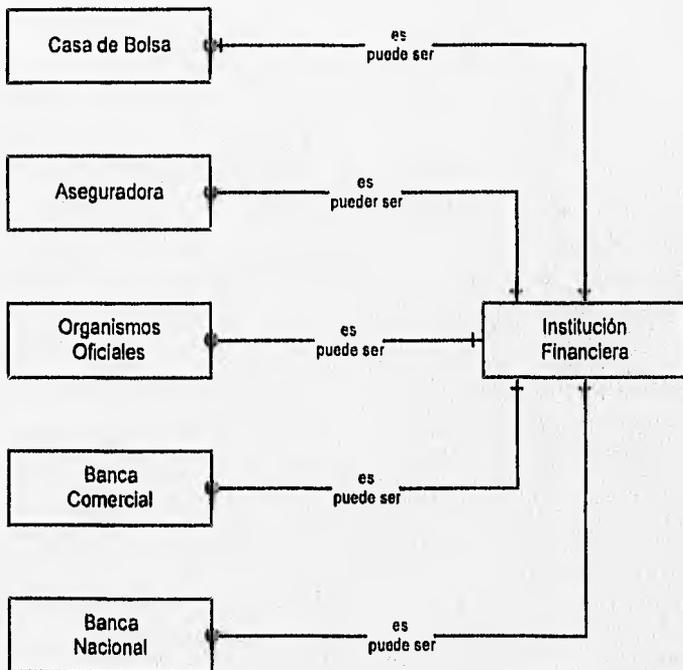
Una relación se expresa con un diamante, un círculo o un semicírculo extendido en un diagrama de entidad relación.

La relación se une con las entidades que relaciona a través de arcos o líneas que van de la relación a la entidad.

En el siguiente diagrama se muestran los símbolos utilizados para representar relaciones .



En IEW la relación no se representa en un símbolo, sino que se usa para nombrar el arco entre una entidad y la otra.



CARDINALIDAD.- Este concepto se usa para indicar el número máximo de entidades que participan en una relación. Se especifica con una punta o dos puntas de flecha

en el arco que conecta la entidad con la relación. Una punta limita la entidad opuesta a participar en sólo una relación, dos puntas representan que cada entidad opuesta puede participar múltiples veces en la relación.

En IEW esto se representa al indicar el número máximo de ocurrencias de la entidad relación.

La cardinalidad se puede especificar en la punta que llega al símbolo de la relación (entidad fuerte), o en la punta que llega a la entidad (entidad destino), en este último caso se dice que la entidad mantiene una relación de dependencia con las entidades cuyos arcos van de la entidad a la relación (*entidad primaria*), y que no puede existir por sí sola si no existen las otras entidades.

En IEW las entidades primarias se representan con el tipo de entidad fundamental y las entidades con relación de dependencia se representan con el símbolo de entidad asociativa.

Una entidad primaria tiene las siguientes características:

- La llave debe consistir en un solo elemento de datos
- No debe contener "inteligencia" o información escondida (una porción de la llave tiene significado por sí misma). Esto permite reducir la volatilidad de la llave para entidades primarias
- Puede ser asignada por el sistema

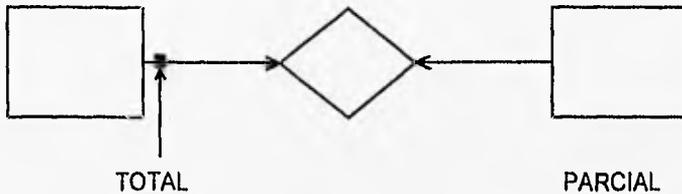
Un caso especial son lo que se llaman entidades primarias menores, que son aquellas entidades primarias que cumplen con alguna de las siguientes condiciones:

- Sólo 2 elementos de datos
- Pocas ocurrencias
- Pocas actualizaciones

PARTICIPACION.- Este concepto indica la extensión en la cual una entidad fuente debe participar en la relación.

TOTAL.- Establece que para cada ocurrencia de la entidad fuente, debe participar al menos una vez en la relación (cardinalidad mínima 1).

PARCIAL.- Sólo algunas de las entidades fuente tienen que participar en la relación (cardinalidad mínima de 1).



Sólo se especifica en los arcos fuente

Se considera que entidades destino no tienen participación total en la relación que los crea.

En el diagrama la participación total se indica como un punto en la base del arco que va de la entidad a la relación.

En IEW se indica especificando el número mínimo de ocurrencias de la entidad en la relación.

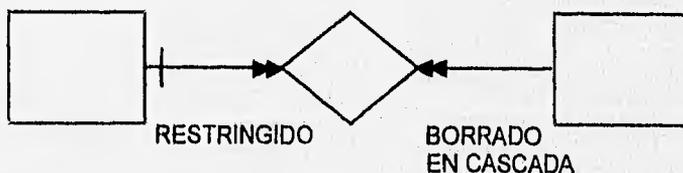
ELIMINACION (baja).- Estas reglas se aplican a relaciones regulares que crean y no crean entidades asociativas.

RESTRINGIDA.- Cuando la entidad fuente no puede ser borrada si participa en la ocurrencia de una relación.

CASCADA.- Si una ocurrencia de la entidad fuente se borra, entonces las ocurrencias de la relación en que participa la entidad fuente serán borradas.

La eliminación restringida se representa cruzando el arco con una pequeña línea.

En IEW no hay manera de representar este concepto, por lo cual esta condición se especificará en los comentarios.



CLASIFICACION DE ENTIDADES Y RELACIONES

Las relaciones y entidades pueden clasificarse en los siguientes tipos:

a) Relaciones de Agregación

Se llama así a relaciones que generan concurrencias de información de mayor nivel que la involucrada en entidades participantes. Es decir, proporcionan información adicional que no estaría disponible si no establece la relación entre entidades.

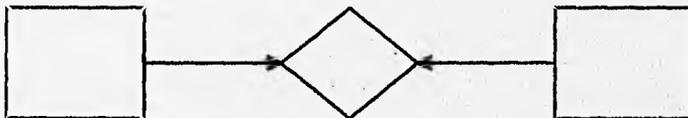
a.1) Relaciones Regulares

Las relaciones regulares se clasifican de acuerdo al número de entidades que participan en la relación.

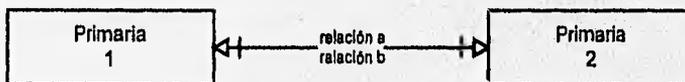
a.1.1) Binaria.- Tiene 2 entidades fuente participando.

La cardinalidad puede ser 1:1, 1:M, M:M

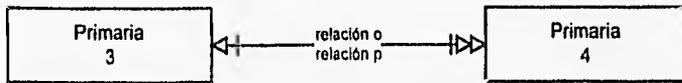
Este es el tipo de relación más común en el diagrama entidad relación



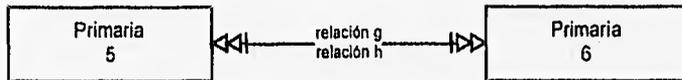
En el IEW se representan de la siguiente manera:



Relación uno a uno

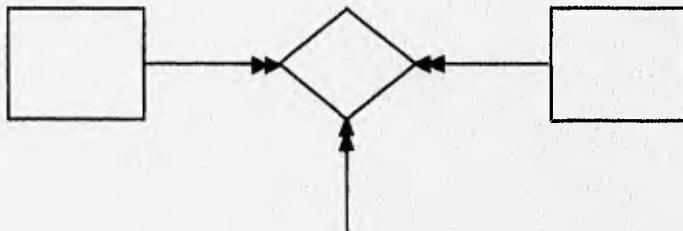


Relación uno a muchos

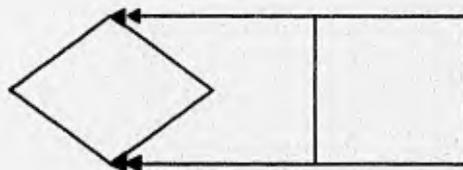


Relación muchos a muchos

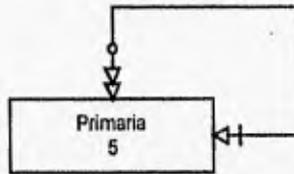
a.1.2) N'aria.- Contiene 3 ó más entidades fuente participantes.
La cardinalidad es siempre M:M:M



a.1.3) Recursiva.- Entidad que se relaciona consigo misma.



En IEW se representa de manera similar:



Relación Recursiva

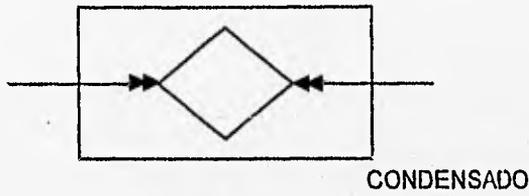
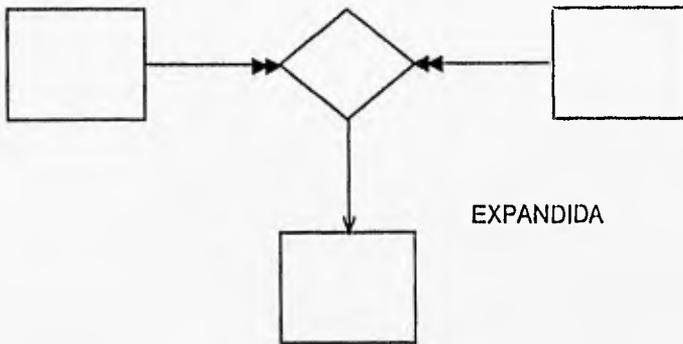
Entidad Asociativa.- formada por la asociación de 2 ó más entidades fuente en una relación regular.

Depende de la ocurrencia de todas las entidades fuente en la relación.

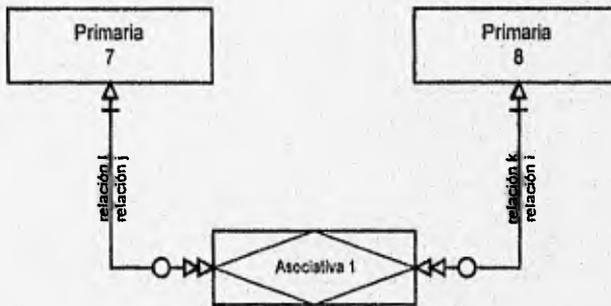
Las entidades asociativas se pueden calificar como complejas cuando su llave se forma por una combinación de las llaves de las entidades participantes mas algún identificador que las haga únicas. Cuando el identificador que las hace únicas, es una fecha, se calificará como histórica.

Un calificador es realmente un atributo o elemento de datos.

Las entidades asociativas son creadas en relaciones binarias, especialmente cuando la cardinalidad es M:M .



La representación en IEW es la siguiente:



Entidad Asociativa

Para la asignación de llaves en relaciones binarias, se seguirán las siguientes reglas:

En relaciones binarias de cardinalidad 1:1 una de las entidades hereda el foreign Key de la otra entidad.

En relaciones binarias 1:M poner la llave primarias de cardinalidad 1 en la entidad de cardinalidad M, aunque en este caso es recomendable mejor el crear una nueva entidad asociativa, ya que este diseño es más flexible a cambios de cardinalidad.

En relaciones binarias M:M siempre es recomendable el crear una nueva entidad asociativa.

a.2) Relación característica

En este caso participan sólo 2 entidades, una entidad actúa como entidad fuente y la otra entidad o *entidad característica* depende de la existencia de una ocurrencia en la entidad fuente.

Se crea una relación característica, cuando todas las condiciones siguientes son verdaderas:

- Tenemos un grupo repetitivo de atributos que describen la entidad fuente, y si
- El grupo repetitivo de atributos no tiene significado, si la entidad fuente es borrada
- La relación entre la entidad fuente y el grupo repetitivo de atributos es 1:M en términos de cardinalidad

En la siguiente figura se muestra la representación de las relaciones características en el SLC.



En IEW la representación es la siguiente:



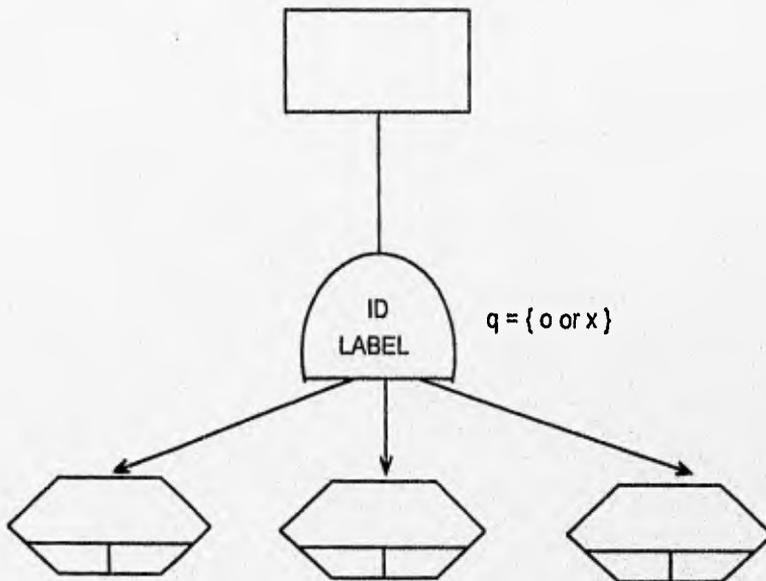
Relación Característica

b) Relaciones de Generalización

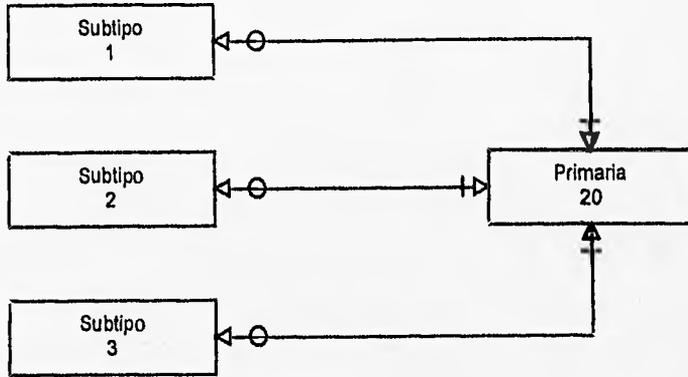
Una relación de generalización representa la distinción entre clases o roles de entidades. Existen 2 tipos de relaciones de generalización.

b.1) Subtipo.- La entidad principal se divide en 1 ó más sub-entidades, llamadas entidades subtipo.

Subconjunto o rol de la entidad fuente. Es una entidad dependiente de la existencia de la fuente (el arco punteado a la entidad subtipo implica la existencia de dependencia)

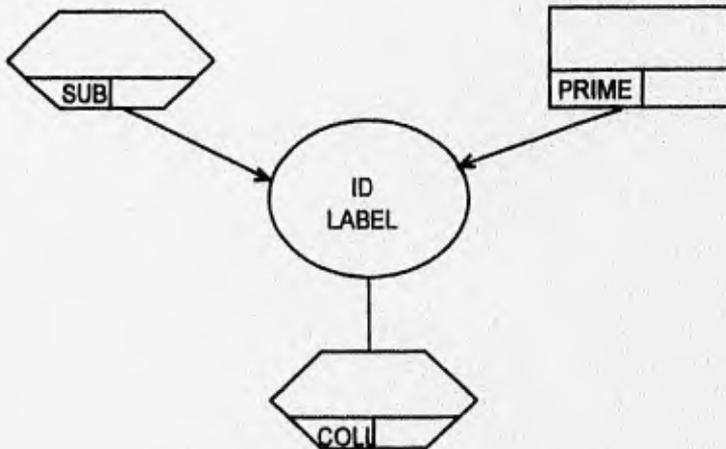


En IEW el subtipo se representa así:



Subtipo

b.2) Colección.- Representa la colección de ocurrencias de diferentes entidades fuente en una sola entidad destino.
Supraconjunto parcial o total de las entidades fuente, que depende de la existencia de esas entidades.



GUIAS PARA ASIGNACION DE ATRIBUTOS A ENTIDADES

A continuación se especifican algunas reglas a seguir para la asignación de atributos a las entidades y para su normalización:

Un elemento de datos debe asignarse a una entidad, sólo si está definido para todas las ocurrencias de la entidad. Si no existe una relación de subtipo y la entidad debe ser categorizada.

Elementos de datos no llave deben asignarse solo a una entidad.

Normalización

- **Remover grupos repetitivos**
- **Remover dependencias parciales en la llave primaria**
- **Remover dependencias en datos no llave primaria**

9 BIBLIOGRAFIA

Software Engineering
Roger S. Pressman
Second Edition

Systems Life Cycle
Quick Reference
Electronic Data Systems Corporation

Information Engineering & Application
Development Using Knowledgeware's Case Tool Set
Rick Napier
Prentice Hall 1991

SLC Standars and Procedures Manual
EDS de México - Border Operations
February 1992

SLC Development and Maintenance Guide
EDS Corporation
Version 1.0

SLC Managers Guide
EDS Corporation
Version 1.0

Project Management: StartUp and Planning
EDS Corporation
March 1994

Manual de Redacción e Investigación Documental
Susana Gonzalez Reyna
Trillas 3ra Edición