Capítulo XIX

Diseño de Sistemas

Diseño de sistemas

Tabla de contenido

Diseño de sistemas

1.- ¿Qué es diseño?

El diccionario de la real academia española incluye diferentes acepciones para la palabra diseño, entre ellas destacamos las siguientes cuatro:

- Traza o delineación de un edificio o de una figura.
- Proyecto, plan en diseño urbanístico
- Concepción original de un objeto u obra destinados a la producción en serie en la industria de diseño gráfico, de modas o de producción.
- Descripción o bosquejo verbal de algo.

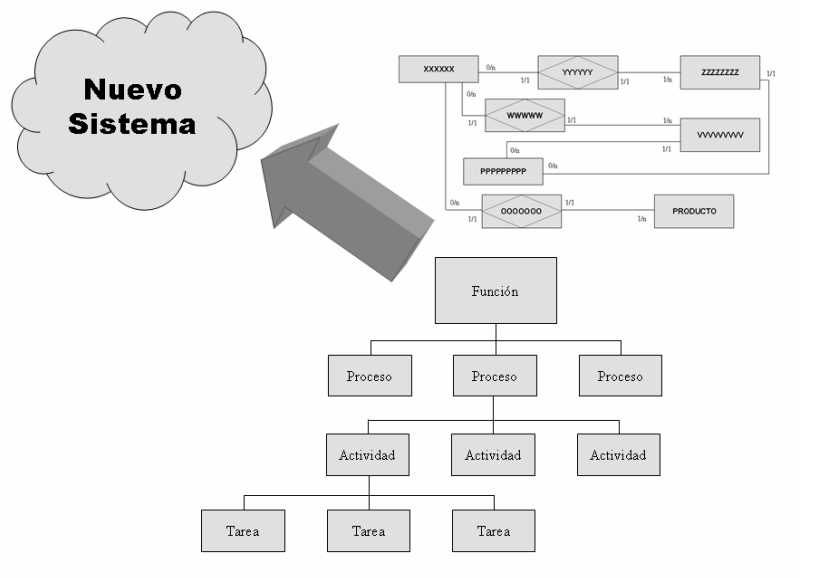
Para nosotros, en sistemas, el significado de diseño incluye dos grandes aspectos:

1. Delinear la forma cómo se implementarán los procesos y la información de un área funcional; esto es, cómo funcionarán esos procesos.
2. Delinear los componentes -tanto componentes de software como procedimientos- que harán posible que los procesos funcionen en la forma prevista.

Adicionalmente, debemos añadir que las actividades de diseño de sistemas tienen dos grandes alcances: el diseño general o diseño de la arquitectura del sistema y el diseño detallado o especificación de los componentes del sistema. En otras palabras, el diseño de sistemas tiene dos tiempos perfectamente diferenciados, el diseño del sistema como conjunto y el diseño de cada una de las piezas que lo conformarán.

Es importante señalar que el diseño general gobierna o establece los lineamientos sobre los cuales se realiza el diseño detallado. Si bien es cierto que existe una tendencia natural a subestimar la importancia del diseño general, no es menos cierto que un buen diseño general es un paso fundamental hacia el desarrollo de un sistema de alta calidad y un

elemento que favorecerá significativamente la productividad de las actividades de desarrollo y, una vez instalado el sistema, de las actividades de mantenimiento.



2.- Diseño de datos

Es muy importante que se tenga claro que las tareas de diseño de datos no se realizan separadamente de las de diseño de procesos; todo lo contrario, ambos grupos de tareas conforman una sola unidad. Sin embargo, por razones didácticas, resulta conveniente presentarlos como temas separados.

Tal como arriba señalábamos, la palabra diseño la aplicaremos en el sentido de: “definir la forma cómo funcionará algo o cómo ese algo será utilizado, estableciendo la forma cómo sus componentes deben organizarse para que funcionen en la forma deseada”. Así pues, diseño de datos es el proceso mediante el cual el ingeniero de sistemas define la manera en que los datos serán utilizados, traduciendo el modelo conceptual de datos a un modelo de utilización de los datos del sistema; el cual, a su vez, será el punto de partida para realizar el diseño físico de las bases de datos que deberán dar servicio a las funciones del sistema.

2.1.- Definición del modelo de utilización de los datos

El método para traducir el modelo conceptual de datos a modelo de utilización de los datos, que a continuación se describe, puede ser

aplicado para cualquier ambiente, tanto archivos convencionales, como de bases de datos. Sin embargo, como cada ambiente tiene sus propias peculiaridades, será necesario que en cada caso se hagan las distinciones que correspondan.

Los pasos a seguir para transformar el modelo conceptual de datos en modelo de utilización de los datos son los siguientes:

1. Definir, para cada entidad representada en el modelo conceptual de datos, una tabla -registro- cuya clave primaria será el atributo identificador de esa entidad.
2. Definir, para cada entidad asociativa representada en el modelo conceptual de datos, una tabla -registro- cuya clave primaria será la concatenación de las claves de las entidades asociadas; cada una de estas claves, además, se identificará como clave foránea.
3. Definir, para cada entidad característica representada en el modelo conceptual de datos, una tabla -registro- en la cual la clave de la entidad caracterizada formará parte de su clave. Este campo se identificará, también, como clave foránea.

   Como parte de la clave primaria de esta tabla así definida, será necesario incluir algún campo que permita individualizar las ocurrencias de la entidad característica.
4. Para cada entidad de datos, representar cada atributo como un dato dentro de la tabla que la representa.
5. Las asociaciones designativas -una a muchas- se representarán de la siguiente forma: la clave de la entidad que designa pasará a ser un campo más de la tabla que representa la entidad designada; este campo se definirá como clave foránea.
6. Revisar las tablas definidas, con el fin de asegurar que ninguna de ellas viole las reglas de normalización:
   - Validar si, con el conjunto de tablas definidas, puede decirse que “se ha definido un lugar para cada cosa y se ha colocado cada cosa en su lugar”.
   - Verificar si las tablas definidas “contienen datos que describen características de toda su clave primaria y sólo de toda su clave primaria”; es decir, las únicas formas de dependencia funcional que existen en cada una de las tablas definidas son de los datos hacia su clave primaria.
   - Será de especial importancia que se verifique si en alguna tabla con clave compuesta existe algún dato que

dependa funcionalmente de parte de la clave o si esa dependencia ocurre a través de otro campo.

7. Revisar en forma muy meticulosa las "reglas del negocio" que imponen reglas de integridad para el manejo de los datos, las cuales, o bien deberán ser construidas en los programas de aplicación, o bien deberán ser establecidas para el manejador de bases de datos cuando se defina la base de datos:
    - Regla del valor nulo: las claves primarias no pueden tener valor nulo.
    - Regla de unicidad de la clave primaria: cada clave primaria debe ser única.
    - Regla de inserción o de integridad referencial: si la frecuencia mínima de una relación designativa es uno, ello implica que la entidad designada no puede existir sin estar asociada a una ocurrencia válida de la entidad que la designa; por ejemplo, no puede existir un empleado sin departamento o una factura sin cliente. Debe destacarse que no se trata sólo de que "el registro de empleado tenga un valor en el campo de departamento", se trata también de que ese número de departamento exista en la tabla de departamentos, es decir, que sea un código válido.
    - Regla de eliminación: no puede eliminarse una ocurrencia de una tabla si su clave primaria existe en alguna ocurrencia de otra tabla como clave foránea; por ejemplo, no puede eliminarse el registro de un cliente si aún tiene facturas pendientes o no puede eliminarse el registro de un departamento al cual están asignados uno o más empleados.
    - Regla de actualización: no puede cambiarse la clave primaria de una ocurrencia de una tabla si su valor existe en alguna ocurrencia de otra tabla como clave foránea; por ejemplo, no puede modificarse el número asignado a un cliente si aún tiene facturas pendientes bajo el número viejo. En general, siempre se evitará cambiar claves, obligando a la aplicación a eliminar primero para después reinsertar los registros afectados por el cambio.

Existen casos en que, por las características de las aplicaciones, al eliminar una ocurrencia de una tabla o modificar su clave se requiere que

se eliminen o actualicen automáticamente todas las ocurrencias de aquellas tablas que tengan dicha clave como clave foránea. Algunos manejadores de bases de datos ofrecen facilidades para “actualizar o eliminar en cascada”; sin embargo, sólo en casos plenamente justificados debe adoptarse esa modalidad.

Las “reglas del negocio” debe construirse dentro del sistema, bien sea utilizando las facilidades que brinde el manejador de bases de datos o creando las rutinas de validación necesarias en los programas de aplicación. De no tomarse las previsiones de diseño necesarias, a la larga, la base de datos contendrá información inexacta -los reportes de empleados por departamento no serán correctos, los listados de cuentas por cobrar no reflejarán la deuda real de cada cliente, etc.-.

Especialmente, en el caso de registros con claves foráneas que representen asociaciones cuya frecuencia mínima es uno, las reglas de inserción se deben construir estableciendo que, para aceptar su inserción, tales claves foráneas no pueden tener valor nulo y su valor debe existir en la tabla básica en la que son clave primaria.

2.2.- Refinar el modelo de utilización de los datos

Siguiendo el método descrito en el punto anterior, se desarrollará el modelo básico de utilización de los datos; lo denominamos básico, porque este modelo aún no incluye todos los requerimientos de acceso a los datos, los cuales se irán identificando a medida que se desarrolle el modelo de funcionamiento del nuevo sistema.

Por ejemplo, las estructuras definidas en el modelo básico de utilización de los datos, no incluyen las consideraciones acerca de transacciones u operaciones que se realizarán sobre la base de datos. Las estructuras del modelo básico sólo presentan el estatus de las entidades y asociaciones, pero sabemos que ese estatus cambia continuamente -los datos contenidos en los diferentes registros de la base de datos sufren cambios- como resultado de la ejecución de transacciones. Por tal razón, el modelo de utilización de los datos deberá complementarse con las estructuras que almacenarán la evidencia de tales cambios a la base de datos: inserciones, cambios y eliminaciones.

Es conveniente destacar que, en algunos casos, no es necesario definir un registro para cada transacción; es frecuente que para algunos registros de la base de datos no se requiera mantener una cronología de sus cambios. Como regla general, eso es cierto para transacciones que no modifiquen el patrimonio de la empresa, como es el caso de la creación

del registro para un nuevo producto, la modificación del nombre y dirección de un cliente, etc.

En resumen, a medida que se avance en las tareas de diseño del sistema, el modelo básico de utilización de los datos se complementará con la definición de los accesos secundarios y de los registros de transacción necesarios, a medida que se vaya definiendo y refinando el modelo funcional de procesos.

2.3.- Volúmenes y volatilidad

Dado que el modelo de utilización de los datos es la materia prima fundamental para el diseño de la base de datos, será muy importante que, a medida que se vaya refinando el modelo de utilización, también se registren meticulosamente tanto los volúmenes como la volatilidad de cada registro.

El volumen asociado con un registro debe entenderse como la cantidad promedio de ocurrencias que se almacenarán para ese registro.

El concepto de volatilidad refleja cuán estables son los registros dentro de su archivo; es decir, hay eliminaciones o inserciones muy frecuentes -como sería el caso de los registros de pedidos pendientes de clientes- o, por el contrario, son registros muy estables que permanecen por años en el archivo -como sería el registro de clientes-. Normalmente, la volatilidad de un registro se expresa a través del tiempo promedio de vida de ese tipo de registro -período de tiempo que, en promedio, transcurre desde que una ocurrencia de ese registro ingresa al archivo hasta que la misma se elimina del archivo que lo contiene-.

3.- Diseño de procesos

Diseñar los procesos que conforman un sistema significa definir la forma en que cada una de sus funciones será ejecutada y los componentes que deben ser desarrollados para que el sistema pueda funcionar en la forma deseada.

3.1.- Diseño general o diseño de la arquitectura

En el pasado era costumbre llevar a cabo una fase de diseño general del sistema a construir; el objetivo central de esa etapa era producir una especificación funcional del sistema. Este documento contenía normalmente una visión global del nuevo sistema, de sus procesos -tanto manuales como mecanizados-, de los diálogos del operador con su estación de trabajo, de los reportes más importantes, de los archivos y de las bases de datos. Junto con el diseño general, también, se determinaban las necesidades de hardware y software para el nuevo sistema.

En una época en la que el desarrollo de sistemas no podía ofrecer al usuario final tantas alternativas, ni encontraba ambientes tan sofisticados, era suficiente definir el sistema en esos términos.

Los sistemas que se desarrollan hoy día requieren de una definición general más perfecta, por lo que se prefiere hablar de diseño arquitectónico, para significar que, no sólo se definen los componentes del nuevo sistema, sino también todo el ambiente operacional en el que este sistema desenvolverá sus funciones.

3.2.- Diseño de la arquitectura del nuevo sistema

El diseño de un nuevo sistema es una actividad compleja que incluye elementos de ingeniería -búsqueda de resultados prácticos-, elementos de ciencia -aplicación de teorías tomadas de las ciencias matemáticas, estadísticas, administrativas, de la computación, etc.- y elementos de arte -habilidad personal y buen gusto-.

La tarea de diseñar la arquitectura de un sistema no es una tarea fácil que pueda ser descrita en pasos o procedimientos simples; todo lo contrario, es una tarea retadora, en la cual el ingeniero de sistemas, haciendo uso de sus conocimientos, su experiencia y creatividad, combinará una y otra vez una serie de ingredientes hasta obtener la aprobación del diseño arquitectónico del nuevo sistema.

El equipo de desarrollo, después de haber definido con precisión lo qué hará el nuevo sistema, deberá diseñarlo, es decir, deberá definir dos cosas fundamentales:

- Cómo será el nuevo sistema “desde el computador hacia afuera”: Cómo será utilizado el nuevo sistema, de qué forma afectará la vida de sus usuarios, cuál es su “personalidad”, cómo será percibido por la comunidad usuaria.
- Cómo será el nuevo sistema “dentro del computador”: Cómo serán los programas que conforman el nuevo sistema, en dónde estarán alojados sus datos, de qué forma se integrará cada componente de software.

Los ingredientes que el equipo de trabajo combinará en su camino hacia la definición de la arquitectura del nuevo sistema son: los modelos conceptuales y de utilización -o funcionamiento- de datos y procesos, los volúmenes de transacciones, los perfiles de usuarios -cantidad y ubicación-, los estándares de la instalación y el hardware/software que estará -o deberá estar- disponible para utilizar el sistema.

Utilizando el modelo conceptual del sistema se definirá un modelo de funcionamiento, es decir, un modelo que represente la forma en que el sistema será utilizado por el usuario final -o la comunidad de usuarios-, con el fin de establecer las actividades que deberán ser llevadas a cabo manual o mecanizadamente; centralmente o en forma distribuida; bajo una modalidad interactiva -diálogo usuario con su estación de trabajo-, interactiva vía web, haciendo uso de dispositivos móviles -hand held o lap top computers- o por lotes -procesos masivos que se ejecutan con cierta periodicidad, diaria, semanal o mensualmente-.

Este modelo de funcionamiento o de utilización -junto con los volúmenes, la cantidad de usuarios y su distribución geográfica- apuntará hacia necesidades de hardware y software que, a su vez, si resultaran muy costosas o no coincidieran con los recursos disponibles -o planificados- o representaran riesgos muy altos, señalarán la necesidad de revisar el modelo de funcionamiento para buscar otras alternativas.

Para analizar y evaluar cada alternativa, el ingeniero de sistemas debe echar mano de cualquier técnica o herramienta conocida, con el fin de simplificar su tarea. Es posible que para poder determinar el costo de desarrollo de una alternativa determinada deba preparar formatos de pantalla, diseño de reportes, diagramas de estructura o flujogramas e, incluso, prototipos para hacer consideraciones de bastante detalle. La timidez y la estrechez de criterio son los peores aliados que un ingeniero de sistemas puede tener en esta etapa, mientras que la agresividad y la creatividad serán su mejor arma.

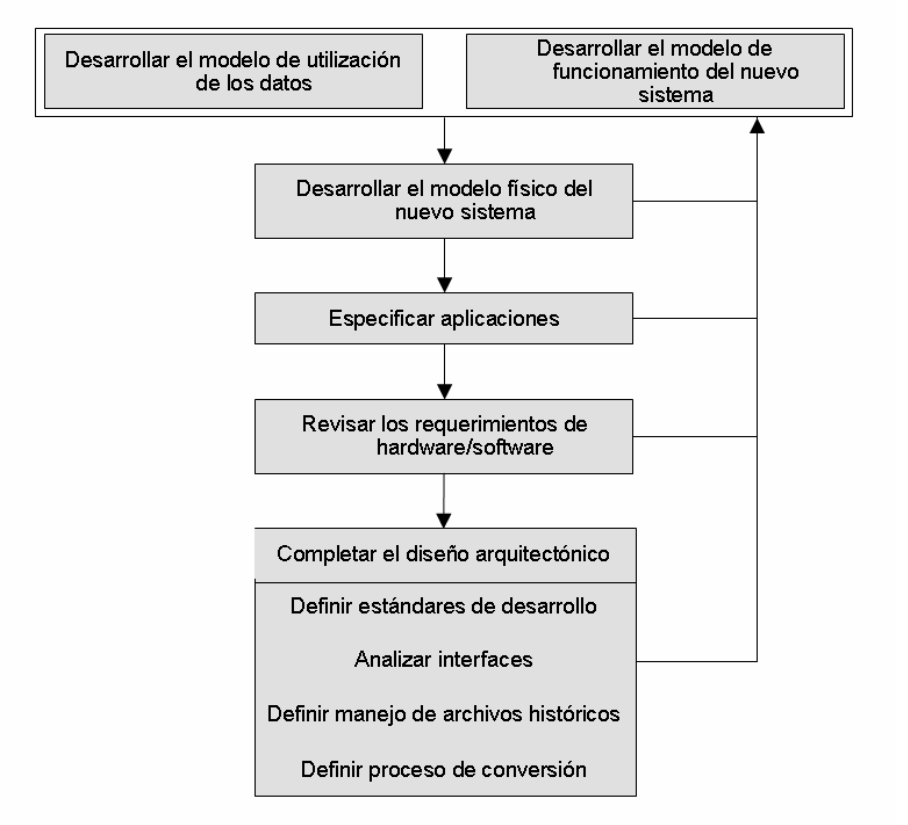
Una vez identificada la forma en que el sistema deberá funcionar, se completará el diseño arquitectónico con una definición detallada de la base de datos física, de la estructura de procesos manuales, por lotes e interactivos y de los estándares que se adoptarán tanto para el uso de las diferentes facilidades que ofrece el hardware y el software, como para el diseño detallado y construcción de cada componente.

4.- El proceso de diseño arquitectónico

A pesar de que, como ya afirmáramos, no es posible crear una “receta” para diseñar la arquitectura del nuevo sistema, en los puntos que siguen queremos señalar un grupo de tareas específicas que, en forma iterativa, el equipo de diseño tendrá que llevar a cabo, ellas son:

1. Desarrollar el modelo de utilización de los datos.
2. Desarrollar el modelo de funcionamiento del nuevo sistema
3. Desarrollar el modelo físico del nuevo sistema

4. Especificar aplicaciones
5. Revisar los requerimientos de hardware/software
6. Completar el diseño arquitectónico.
7. Definir los estándares de desarrollo.
8. Analizar interfaces.
9. Definir archivos históricos.
10. Definir el proceso de conversión.



4.1.- Desarrollar el modelo de utilización de los datos

Tal como se explica en la sección de diseño de datos, el modelo de utilización de los datos se desarrollará en dos grandes pasos: primero, se representarán los registros que conformarán la base de datos en un diagrama de estructura de datos y, a medida que vaya refinándose el diseño del nuevo sistema, se irá ajustando el modelo básico para ir incluyendo los requerimientos de acceso que vayan siendo identificados.

Debe tenerse presente que la refinación del modelo de utilización de los datos incluye, no sólo los requerimientos de acceso, sino también los registros de transacción, los volúmenes y la volatilidad de cada uno de los registros.

4.2.- Desarrollar el modelo de funcionamiento del nuevo sistema

Puede afirmarse que la definición del modelo de funcionamiento del nuevo sistema constituye la descripción de cómo "lucirá" el nuevo sistema "visto desde el exterior del computador". Esta visión de lo que el sistema hará, más la forma cómo será utilizado, constituye la materia prima necesaria para definir la arquitectura física que el nuevo sistema deberá tener.

Aunque ello no esté establecido como regla en ningún texto, el equipo de trabajo analista-usuario, a medida que ha ido avanzando en las tareas del proyecto, ha ido dando vuelo a su imaginación y, en sus mentes, han ido construyendo el nuevo sistema. Ellos han ido ideando de qué manera cada usuario final realizará cada tarea, cómo el computador aceptará cada transacción y cómo la información será presentada en las pantallas y reportes. Este nuevo sistema, que ya no es algo desconocido para ellos, está parcialmente hecho, sólo que no puede ser utilizado porque está en sus mentes, no es aún algo físico.

Nos gusta comparar este fenómeno con una situación presentada en la célebre película Amadeus, que trata sobre la vida del famoso compositor Wolfgang Amadeus Mozart. Hay una escena en ella en la que al protagonista se le exige que muestre la obra que ha prometido componer y éste responde afirmando que la música ya está lista, sólo falta escribirla: "La música está aquí, en mi cabeza; lo único que falta por hacer es ponerla en papel". Lo mismo les ocurre a los ingenieros de sistemas: en este punto del proyecto, el sistema está listo; lo único que falta es escribir sus especificaciones y sus programas. Pues bien, el modelo de funcionamiento constituye el primer intento de poner esa visión del nuevo sistema en "blanco y negro".

Desarrollar el modelo de funcionamiento del nuevo sistema consiste, básicamente, en definir el comportamiento o la personalidad que tendrá el nuevo sistema. Para ello, el ingeniero de sistemas revisa cada cuadro o "caja" del modelo conceptual de procesos, desarrollado en la etapa de análisis, y responde a la pregunta ¿cómo va a "correr" o a operar esta función? En esta tarea, para poder ordenar sus ideas, deberá visualizar, imaginar y establecer:

- Quién es el usuario final.

- Cuál es su nivel de preparación y familiaridad con sistemas computarizados.
- Dónde se encuentra ubicado ese usuario final.
- Qué tareas realizadas por ese usuario final están representadas en esa “caja”.
- Cuál es la forma más simple de realizar esas tareas.
- Cómo debe ayudarlo el nuevo sistema en la ejecución de esas tareas.
- Qué problemas debe resolverle.
- Qué relación tiene ese usuario con los datos -es propietario exclusivo, está autorizado a actualizarlos o sólo puede consultarlos-.
- Qué otras tareas realiza el mismo usuario y cómo se relacionan.
- Cómo se relaciona con otros usuarios.
- Cómo se comunica con esos otros usuarios.
- Con qué otros sistemas está relacionado.
- Cómo se comunica con esos otros sistemas.
- De qué forma el nuevo sistema podría facilitarle la comunicación con esos otros usuarios y sistemas.
- Con qué frecuencia el usuario estará realizando la actividad cuya “caja” se analiza.
- Cuáles son las limitaciones de tiempo que existen para realizar esa tarea.
- Cuando el computador falle, de qué alternativas dispondrá el usuario para seguir realizando sus tareas.
- Qué controles deben establecerse, para garantizar la integridad de los datos manejados por el usuario.

Para visualizar la forma en que cada “caja” -o grupo de ellas- tomará vida, son de gran utilidad los esquemas o paradigmas que, aunque no constituyan una descripción rigurosa de un proceso, ilustran en forma general el proceso que se estudia y sirven como marco de referencia para que la imaginación no divague y se concentre en ideas concretas.

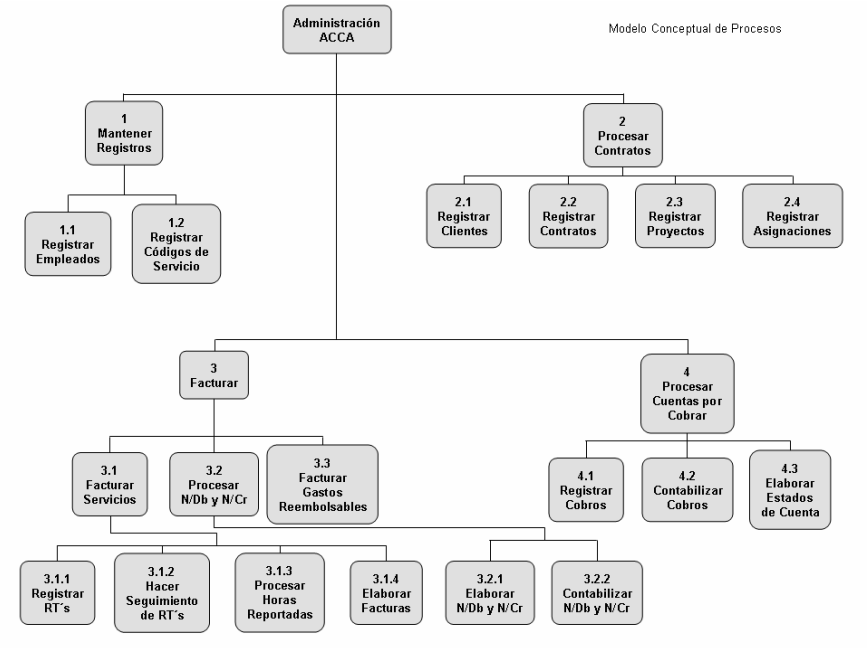
Será fundamental que se registren todos los aspectos que se visualicen acerca del funcionamiento del nuevo sistema dibujando DFD´s de trabajo para hacerlos más explícitos, preparando especificaciones de use case -o

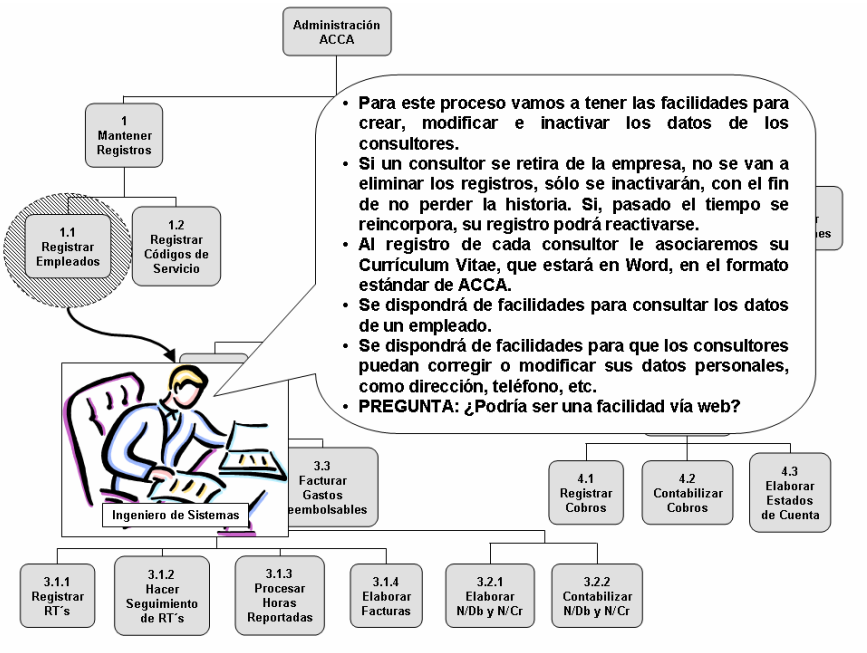
descripciones detalladas de proceso- en forma similar a la que muestran los ejemplos y, como ya señaláramos, haciendo uso de esquemas o paradigmas que ilustren el comportamiento de cada una de las piezas del sistema.

La revisión detallada de cada una de las “cajas” del modelo conceptual de procesos y las consideraciones que se hagan en relación con el funcionamiento o utilización pueden llevar al ingeniero de sistemas a introducir modificaciones en ese modelo, fraccionando una “caja” en dos o más o reuniendo varias en una sola. El criterio para realizar una agrupación o algún fraccionamiento será, principalmente, el uso de información y el tipo de servicio -vía web, interactivo, por lotes diarios, por lotes semanales, etc.- que será requerido por la tarea que se diseña y el conjunto de consideraciones hechas en torno a controles y autorización de uso.

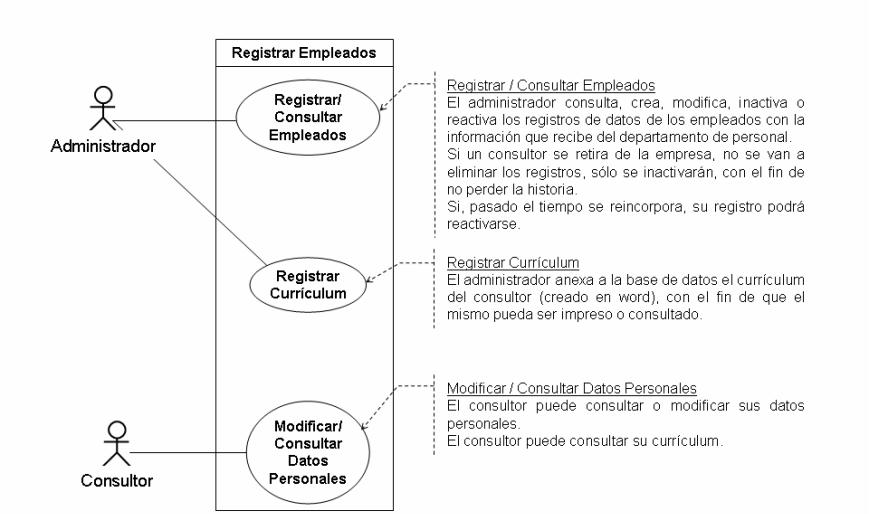
El diseño del nuevo sistema requiere que el ingeniero de sistemas realice repetidas “acrobacias mentales”, pasando del detalle al conjunto, para volver nuevamente al detalle sin perder de vista el conjunto. No es una tarea fácil, como ya hemos dicho varias veces, pero es importante que el ingeniero de sistemas estudie y diseñe cada pieza individual sin que el sistema sea una mera agrupación de funciones, sino que sea un sistema integrado, en el cual un mismo dato sólo se registra una vez, en el cual no se espera que el usuario haga nada que el computador pueda hacer, al que no entrarán datos que no hayan sido validados con meticulosidad extrema, en el que ninguna persona que no tenga la debida autorización podrá hacer uso de las facilidades del sistema, al que será fácil modificar o ampliar sus funciones, etc. Todo esto significa que, una vez “completado” el modelo de funcionamiento y cuando se esté desarrollando el modelo físico del nuevo sistema, muy probablemente se tomarán decisiones o se “descubrirá” una mejor forma de hacer algo y se vea la necesidad de volver un paso atrás, para hacer alguna modificación para mejorar los modelos de datos y de funcionamiento.

Cuando se indica que el funcionamiento de las unidades de diseño debe “ser especificado en detalle”, es necesario tener claro hasta qué nivel de detalle debe especificarse. La respuesta a esa pregunta es: hasta el punto en que se logre producir una descripción satisfactoria y completa de las funciones que se realizarán.

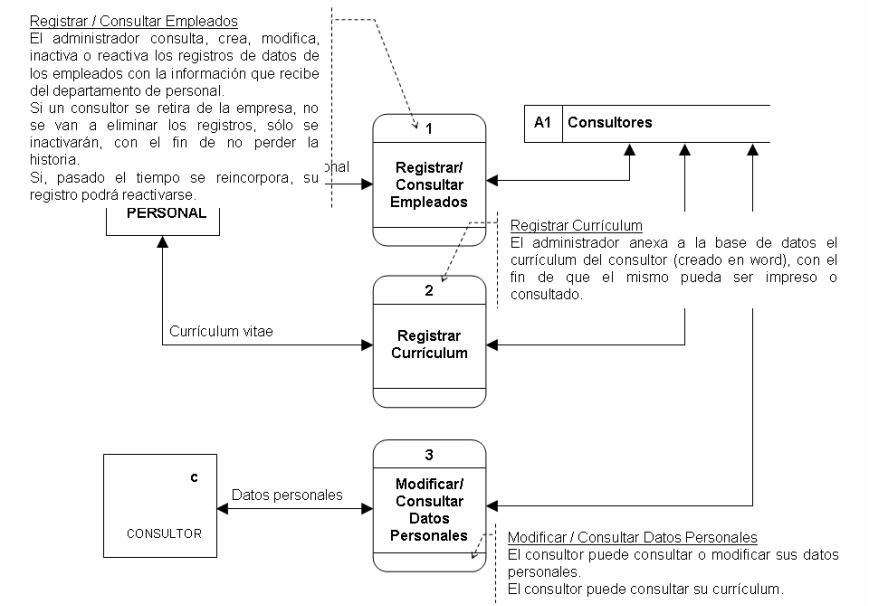

Modelo Conceptual de Procesos
Administración ACCA
1 Mantener Registros
2 Procesar Contratos
1.1 Registrar Empleados
1.2 Registrar Códigos de Servicio
2.1 Registrar Clientes
2.2 Registrar Contratos
2.3 Registrar Proyectos
2.4 Registrar Asignaciones
3 Facturar
4 Procesar Cuentas por Cobrar
3.1 Facturar Servicios
3.2 Procesar N/Db y N/Cr
3.3 Facturar Gastos Reembolsables
4.1 Registrar Cobros
4.2 Contabilizar Cobros
4.3 Elaborar Estados de Cuenta
3.1.1 Registrar RT´s
3.1.2 Hacer Seguimiento de RT´s
3.1.3 Procesar Horas Reportadas
3.1.4 Elaborar Facturas
3.2.1 Elaborar N/Db y N/Cr
3.2.2 Contabilizar N/Db y N/Cr


Administración ACCA
1 Mantener Registros
1.1 Registrar Empleados
1.2 Registrar Códigos de Servicio
• Para este proceso vamos a tener las facilidades para crear, modificar e inactivar los datos de los consultores.
• Si un consultor se retira de la empresa, no se van a eliminar los registros, sólo se inactivarán, con el fin de no perder la historia. Si, pasado el tiempo se reincorpora, su registro podrá reactivarse.
• Al registro de cada consultor le asociaremos su Currículum Vitae, que estará en Word, en el formato estándar de ACCA.
• Se dispondrá de facilidades para consultar los datos de un empleado.
• Se dispondrá de facilidades para que los consultores puedan corregir o modificar sus datos personales, como dirección, teléfono, etc.
• PREGUNTA: ¿Podría ser una facilidad vía web?
Ingeniero de Sistemas
3.3 Facturar Gastos eembolsables
4.1 Registrar Cobros
4.2 Contabilizar Cobros
4.3 Elaborar Estados de Cuenta
3.1.1 Registrar RT´s
3.1.2 Hacer Seguimiento de RT´s
3.1.3 Procesar Horas Reportadas
3.1.4 Elaborar Facturas
3.2.1 Elaborar N/Db y N/Cr
3.2.2 Contabilizar N/Db y N/Cr

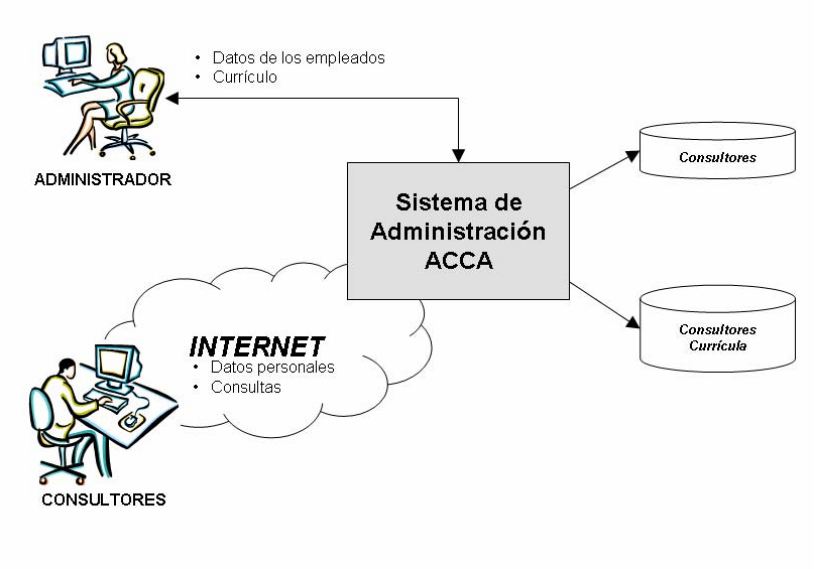
Expresando el funcionamiento con Use Cases



Expresando el funcionamiento con DFD´s



Expresando el funcionamiento con Paradigmas



4.3.- Validar el modelo de funcionamiento del nuevo sistema

Una vez integrado el modelo de funcionamiento del nuevo sistema es necesario validarlo, haciendo diferentes revisiones y presentaciones a diferentes niveles de usuarios.

También es importante validarlo con respecto a la matriz de entidades vs. Procesos, de tal manera que pueda verificarse en qué puntos del sistema se crean, se utilizan, se actualizan y se eliminan las ocurrencias de cada entidad del modelo de datos.

Es posible que para alguna entidad, al preguntar ¿dónde se crea?, la respuesta nos señale otro sistema, como podría ocurrir con una tabla de departamentos; en esos casos, deberán crearse los puntos de alerta, con el fin de que al revisar o diseñar las interfaces con otros sistemas, todo quede perfectamente especificado y coordinado.

También es posible que para alguna entidad, al preguntar ¿dónde se elimina?, la respuesta sea nunca; en esos casos habrá que preguntarse si su volumen señala la conveniencia de crear archivos históricos, con el fin de evitar que las bases de datos que estarán en línea crezcan sin control y, a la larga, puedan afectar los tiempos de respuesta del sistema.

4.4.- Desarrollar el modelo físico del nuevo sistema

En este paso, el equipo de trabajo comienza a perfilar cuáles deberán ser los componentes de software a ser construidos e integrados para poder brindar al usuario final todas las facilidades que hasta este momento han sido identificadas.

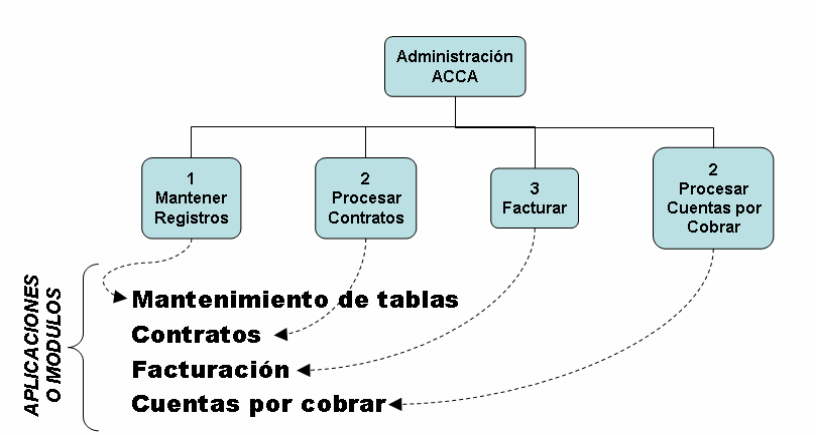
Tal como hemos venido insistiendo, el diseño del nuevo sistema es un proceso iterativo, en el cual el ingeniero de sistemas va "tallando" la arquitectura con base en: el modelo de funcionamiento, los volúmenes de transacciones, la cantidad y perfil de los usuarios, los niveles de calidad del sistema, los estándares de la instalación y el hardware/software que estará -o deberá estar- disponible para ejecutar el sistema.

Sobre el modelo de funcionamiento del nuevo sistema se han identificado las actividades y tareas que deberán ser llevadas a cabo manual o mecanizadamente, centralmente o en forma distribuida, interactiva o por lotes -diariamente, semanalmente, mensualmente, trimestralmente, etc.-. Esta identificación, junto con los volúmenes, la cantidad de usuarios y su distribución geográfica, apuntará hacia necesidades de hardware y software, las que, a su vez, si resultan muy costosas o de alto riesgo o no coinciden con los recursos disponibles -o planificados-, señalarán la necesidad de revisar el modelo de funcionamiento para evaluar otras alternativas.

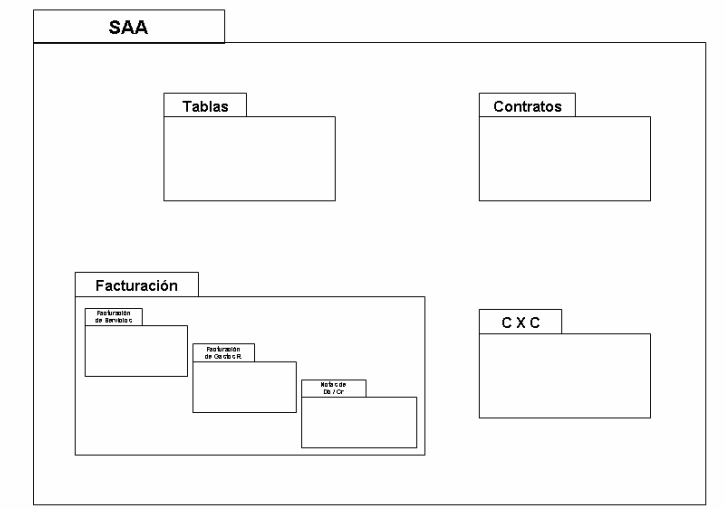
En forma paralela y muy similar, el diseñador de bases de datos trabajará con el equipo analista-usuario para definir la arquitectura de datos que satisfaga los requerimientos del sistema.

Para realizar el diseño físico del nuevo sistema también es necesario ver, simultáneamente, el todo y las partes. Es necesario analizar, como ya dijéramos, cada una de las actividades representadas en el modelo de funcionamiento del nuevo sistema y visualizar la forma en que serán llevadas a cabo dentro del contexto del nuevo sistema, para, así, poder definir los componentes de software que deberán ser creados.

El ingeniero de sistemas comenzará por identificar las aplicaciones que compondrán el sistema. Ello lo hará, en principio, de acuerdo con la separación de procesos que se haya establecido en el modelo de conceptual del sistema. Dentro de cada aplicación, a su vez, se identificarán las unidades de diseño, lo cual se hará estableciendo los conjuntos de tareas que componen una misma actividad -actualización de registro de clientes, elaboración de facturas, entradas al inventario, etc.- y que recibirán un mismo tipo de servicio -manual, interactivo o por lotes diario, por lotes semanal, etc.-.



Estos componentes, así identificados, constituirán las unidades de diseño en las que el ingeniero de sistemas se concentrará durante la etapa de diseño detallado, para especificar los módulos y programas que llevarán a cabo las tareas asignadas a cada aplicación y procedimiento. Es posible que, en algunos casos, sea necesario fraccionar un poco más esas unidades de diseño en base a la experiencia o a la intuición o a la creencia que pueda tener el ingeniero de sistemas de que cada una de esas actividades será una actividad separada para el computador. Una vez que se fijan las unidades de diseño básicas, las complicaciones que se encuentren en la práctica serán la mejor guía para decidir si la unidad de diseño debe ser fraccionada o no. Ajustar su definición no debiera causar mayores traumas.



Es importante destacar que cada procedimiento o unidad de diseño puede incluir varios componentes, es decir, una sola unidad de diseño puede dar lugar a una o varias unidades de construcción o programas.

La resultante de este proceso se irá plasmando en diagramas de estructura que muestren, desde arriba hacia abajo, los componentes del nuevo sistema.

La definición de la arquitectura para el nuevo sistema se completará con la revisión del plan de hardware/software y la red de proceso -ubicación de los usuarios y terminales-, con el fin de tomar las acciones necesarias para ajustar dicho plan o ajustar los requerimientos del sistema.

4.5.- Completar el diseño arquitectónico

Una vez que el diseño básico del sistema ha sido validado y la directiva del proyecto le haya dado su aprobación, se completará el diseño arquitectónico definiendo:

- Los lineamientos de desarrollo o nivel de calidad.
- Los estándares que se adoptarán para el uso de las diferentes facilidades que ofrece el hardware/software.
- Los estándares que se adoptarán para el diseño detallado y construcción de cada componente.
- Las interfaces con otros sistemas.
- El proceso de conversión.

4.5.1.- Los lineamientos de diseño

Al definir el diseño de un sistema en la forma señalada en los pasos anteriores, se habrán identificado, actividad por actividad, las características y necesidades de servicio que el nuevo sistema deberá satisfacer. Estas características, vistas como conjunto, constituyen su nivel de calidad o, dicho en otros términos, el conjunto de características que éste deberá reunir, con el fin de brindar a la comunidad de usuarios una herramienta que, durante muchos años, la ayude a cumplir con las tareas del negocio en forma eficaz, fiel y oportuna. Este conjunto de características podemos denominarlos el nivel de calidad o los lineamientos de desarrollo del sistema.

Un sistema de calidad no se produce por accidente; todo lo contrario, es producto de un trabajo consciente, orientado hacia la excelencia; por tal razón, la definición de las características de funcionamiento de un sistema no estará completa sin una especificación de su nivel de calidad o de los lineamientos que el desarrollador deberá cumplir.

Una definición de los lineamientos de desarrollo para el sistema debe expresarse en términos de los siguientes factores:

1. **Desempeño**

   El desempeño de un sistema se expresa a través de tres elementos básicos, el tiempo de respuesta para componentes interactivos, el throughput -capacidad de proceso por unidad de tiempo- para los componentes por lotes y por la oportunidad en que -o periodicidad con que- un determinado resultado debe ser producido, como puede ser el caso de un reporte contable requerido a final de mes o un listado de producción diaria que debe estar en el escritorio del gerente todos los días a las 7 a.m.

2. **Facilidad de operación**

   Es una medida del esfuerzo requerido para operar un sistema, preparar sus datos de entrada e interpretar los resultados que produce. Normalmente, esta cualidad se asocia con la "amigabilidad" del sistema.

3. **Facilidad de entrenamiento**

   Es una medida del esfuerzo requerido para aprender a operar y a interpretar los resultados que produce un sistema.

4. **Integración**

   Es la capacidad que tiene un sistema para integrarse con los restantes sistemas de la empresa, intercambiando y compartiendo datos con otras aplicaciones o usuarios.

5. **Disponibilidad**

   Es una medida de la facilidad con que el usuario puede tener acceso a las aplicaciones y la información procesada por ellas. Normalmente, la disponibilidad de un sistema se asocia con los niveles de servicio o con la continuidad del servicio prestado por el sistema.

6. **Seguridad**

   Es el grado de dificultad que encuentran las personas no autorizadas al tratar de tener acceso a las funciones, los componentes o los datos de un sistema.

7. **Integridad**

   Es el grado de dificultad establecido para evitar que algún evento o persona, autorizada o no, pueda dañar, en forma

voluntaria o accidental, los componentes o la información manejada por el sistema.

8. **Mantenibilidad**

   Es una medida del esfuerzo requerido para localizar y corregir la fuente de error en un sistema.

9. **Flexibilidad**

   Es una medida del esfuerzo requerido para mejorar o ampliar las funciones que realizan los componentes de un sistema.

10. **Facilidad de prueba**

    Es una medida del esfuerzo requerido para probar un sistema y, de esa forma, asegurar que está libre de errores y cumple con sus especificaciones.

11. **Reusabilidad**

    Es la capacidad de adaptación de los componentes de un sistema para ser utilizados en otras aplicaciones.

12. **Portabilidad**

    Es una medida del esfuerzo requerido para transferir un sistema, desde una plataforma de hardware o software a otra.

Al fijar los lineamientos de desarrollo para un sistema es necesario establecer un balance entre cada uno de ellos, ya que si, por ejemplo, se desea un producto de alta generalidad, indudablemente éste sistema será de mayor complejidad para el usuario, lo cual podría colidir con los objetivos que se fijen en relación con su facilidad de operación. Análogamente, el objetivo generalidad puede colidir con los factores costo y desempeño, pues un producto de alta generalidad será más difícil de desarrollar y probablemente consumirá más recursos en su ejecución.

Cada lineamiento debe fijarse en relación con los restantes; proposiciones como “maximizar la mantenibilidad del sistema” suenan bien, pero no significan nada si no están establecidas las prioridades para cada objetivo, de tal forma que, en un momento dado, sea posible identificar el objetivo que debe prevalecer sobre otros que resulten conflictivos.

4.5.2.- Los estándares de desarrollo

Junto con los lineamientos de diseño es fundamental enfatizar la importancia de definir los estándares de desarrollo que deberán ser seguidos durante la construcción del nuevo sistema, ya que la adopción de este tipo de guías permitirá garantizar la calidad del producto final y, a

la vez, simplificar las tareas de diseño detallado, construcción, prueba y mantenimiento.

Los programas desarrollados bajo estándares bien establecidos pasarán a ser componentes de un todo homogéneo, cuya operación es uniforme y cuyo comportamiento es predecible. Los beneficios de la inversión de tiempo y esfuerzo que se haga en definir los estándares para diseñar y construir componentes se aprovecharán cada vez que se desarrolle un módulo, ya que, si se siguen los estándares, el producto será fácil de entender, por lo que resultará fácil incorporarle cambios o corregirle problemas.

Dentro de la categoría de estándares incluimos todas las normas y reglas que deben guiar al ingeniero de sistemas en la organización de diálogos en línea y procesos por lotes, así como también la forma en que deberán ser utilizadas todas las facilidades de hardware y software disponibles.

Entre los estándares para el uso de hardware/software deben señalarse:

- Uso de las estaciones de trabajo -teclas de control de programa, intensidad, colores, alarmas-.
- Inicio de procesos por lotes por parte del usuario, qué tipo de trabajos serán iniciados directamente por el usuario y cuáles serán iniciados bajo el control de algún programa “scheduler” o bajo el control de la unidad de operaciones.

Entre los estándares de diseño y construcción deben señalarse:

- Estándares de programación.
- Utilización de lenguajes de programación.
- Estructura de los diálogos en línea -menús, submenús, etc.-.
- Estructura de los procesos por lotes.
- Nomenclatura.
- Formatos de reporte y pantalla -encabezados, localización de los mensajes de error, colores, etc.-.
- Formato de botones.
- Mensajes de confirmación después de actualizar en línea cualquier registro de la base de datos.
- Rastros de auditoría para actualización por grupos, por lotes o interactiva.
- Ayudas para el usuario.

Muchos de esos estándares estarán definidos para la instalación pero, en caso de que no hayan sido preestablecidos, deberán especificarse para el sistema con el fin de que éste se construya de manera uniforme y no de acuerdo con los diferentes estilos de los diferentes analistas/programadores que participen en la construcción del mismo, evitándose así que cuando el sistema entre en producción sea una babel de procedimientos, sino un conjunto homogéneo de éstos.

4.5.3.- Las interfaces

El diseño arquitectónico de un sistema no está completo si no existe una definición adecuada de las interfaces del nuevo sistema con los restantes sistemas de la empresa. En otras palabras, cómo se integrará el nuevo sistema a la arquitectura de sistemas de la empresa.

Una de las fallas más comunes que se observa en muchos proyectos es la superficialidad con que se analizan y construyen las interfaces con otros sistemas. Superficialidad en términos del conocimiento del para qué se prepara un determinado output -qué proceso se hará con él, cuál es su significado, cuál es su objetivo- o del por qué un determinado input que se recibirá en el sistema viene en la forma que viene -qué proceso lo creó, cuál es su significado, cuál es su objetivo-.

Normalmente, el análisis de interfaces se limita a investigar los formatos de los registros que se intercambian y, al hacer esto, el ingeniero de sistemas pierde la oportunidad de que los sistemas funcionen en forma realmente integrada, de que se utilicen facilidades que ofrecen los sistemas destinatario o emisor y de que no se dupliquen funciones de uno a otro sistema.

El análisis de interfaces debe ser extensivo y ello sólo se logra invirtiendo el tiempo necesario para conocer las facilidades y características de los sistemas que se interconectarán con el nuevo sistema. Así pues, este análisis de interfaces debe ser realizado en forma rigurosa, con el fin de que quede muy claro:

- Qué pasará con las salidas preparadas por el nuevo sistema, a qué procesos serán sometidos, cuál es el objetivo de cada uno de esos procesos.
- Cómo funcionan los sistemas receptores, qué opciones y facilidades ofrecen, en qué puntos puede entrar la información que preparará el nuevo sistema.
- Qué ocurrió con el input que el nuevo sistema recibirá de otros sistemas, cuál es el objetivo de esos sistemas.

- Cómo funciona cada sistema emisor, qué opciones y facilidades ofrece, desde qué puntos el nuevo sistema podría tomar la información requerida.

Dependiendo de la calidad y actualidad de la documentación de los sistemas de interfaz, el análisis de éstos será más o menos difícil; en cualquier caso deben prepararse pequeños modelos de funcionamiento -DFD, DA o Use Case- que expresen las tareas de interfaz y que muestren el futuro -o pasado- de los flujos que el nuevo sistema dirigirá -o recibirá- hacia -o desde- los sistemas de interfaz.

El objetivo principal del análisis de las interfaces debe ser: comprender el sistema de interfaz casi tan bien como se comprende el sistema objeto de estudio, con el fin de que ambos ensamblen perfectamente. No se debe tener miedo a "perder el tiempo" en estudiar algo que no corresponde al "objetivo del estudio", ya que una buena comprensión de las interfaces permitirá utilizar eficazmente los recursos que los sistemas instalados ofrecen o, por el contrario, poner de manifiesto deficiencias que tengan, revelando así las fuentes de problemas potenciales y la carga de mantenimiento que ellas ocultan.

4.5.4.- El proceso de conversión

Al igual que las interfaces, los procesos de conversión normalmente se dejan a un lado hasta último momento, siendo ésta la causa de que, en muchos casos, no puedan cumplirse las fechas de instalación planificadas.

El proceso de conversión debe incluir todas las tareas que deben ser realizadas para que el nuevo sistema pueda ser instalado sin traumas de ninguna especie, encajando perfectamente en la arquitectura actual de sistemas y procedimientos. Esto es, el proceso de conversión incluye tanto la carga de las nuevas bases de datos como la incorporación de los ajustes necesarios a los sistemas existentes, de tal forma que tales sistemas funcionen en forma integrada con el nuevo sistema.

Una de las tareas más importantes dentro del proceso de conversión es la conversión de los datos existentes -manuales o automatizados- a la base de datos del nuevo sistema. En la mayoría de los casos, si bien los datos se encuentran disponibles, se hallan en diferentes formatos, con diferentes nombres y dispersos en un sinnúmero de archivos, formularios o documentos. El proceso de conversión tiene como misión capturarlos, validarlos, depurarlos, organizarlos y almacenarlos en la nueva base de datos, de tal forma que esta nueva base de datos esté actualizada y depurada para el momento en que el nuevo sistema comience a operar.

Al analizar y definir los procesos de conversión debe ponerse especial énfasis en las equivalencias de códigos -sistema actual/sistema nuevo-, en la calidad de los datos del sistema actual y en la forma cómo se recopilarán aquellos datos que el nuevo sistema requiere y que, en el sistema actual, no son manejados. Por tal razón, la conversión de datos requerirá de una cantidad de programas que deben ser definidos, desarrollados y, muchos de ellos, probados con la misma rigurosidad que se requiere para los programas del nuevo sistema.

El correcto funcionamiento del nuevo sistema dependerá tanto de un buen diseño, como de un buen proceso de conversión, por lo que el plan de conversión debe ser el complemento del diseño arquitectónico para que su instalación sea exitosa.

4.5.5.- El subsistema de seguridad

La definición de la arquitectura del nuevo sistema también debe contener la definición de la forma cómo se manejarán los aspectos de seguridad:

- Niveles de acceso, es decir, niveles de usuario y tipo de autorización que se le otorgará a cada nivel para utilizar la información y las facilidades del sistema.
- Detalle de cada uno de los componentes de las funciones de:
  - Control de acceso al sistema
  - Control de acceso a un determinado componente o módulo
  - Control de acceso a la base de datos

Es muy probable que el esquema de seguridad que deberá utilizarse esté preestablecido por las facilidades del sistema operativo en uso, los paquetes de software de seguridad instalados y el propio manejador de bases de datos. En este caso, la definición de los aspectos de seguridad para el nuevo sistema se centrará en la definición de los perfiles de usuarios y el tipo de autorización que se le otorgará a cada perfil.

En caso de no contar con las facilidades arriba citadas, será necesario definir, para después desarrollar, los módulos que mantendrán el directorio de usuarios del sistema y sus perfiles, los módulos de seguridad que regularán el acceso al sistema y los módulos que deberán incluirse dentro de los diferentes programas, con el fin de asegurar que sólo usuarios debidamente autorizados podrán tener acceso a los diferentes componentes del sistema.

4.5.6.- El subsistema de administración

Dentro de muchos sistemas existe una serie de funciones que están reservadas al usuario denominado Administrador del Sistema. Entre estas funciones, normalmente se incluyen la actualización de tablas, parámetros y mensajes del sistema y la ejecución de procedimientos para crear archivos especiales de respaldo -diferentes a los backup normales y a los que se almacenan en lugares alejados de la instalación, con el propósito de reiniciar operaciones después de algún desastre, como incendio o terremoto-.

En muchas instalaciones, la misma persona que administra las facilidades del sistema también se encarga de manejar y administrar las funciones de seguridad del sistema. Sin embargo, buscando siempre simplificar la concepción de los componentes del nuevo sistema, no deben entremezclarse o confundirse las funciones de administración del sistema con las de seguridad.

Entre los parámetros del sistema se incluyen, generalmente, las fechas de cierre mensual y anual, listas de distribución de reportes, valores que pueden tomar ciertas variables -por ejemplo, tabla de conversión de monedas- y selección de reportes a ser producidos.

Dentro de las funciones del administrador del sistema también pueden estar los procedimientos de creación de archivos históricos a final de año, así como también todas aquellas funciones de protección y mantenimiento de los datos que no estén dentro de las funciones normales del programador de mantenimiento.

Todas las facilidades que serán creadas para el administrador del sistema deberán ser definidas y especificadas como parte del diseño arquitectónico, dentro de lo que se denomina el subsistema de administración.

5.- Estrategia de desarrollo e implantación

5.1.- Concepto de versión

El concepto de versión, referido al desarrollo de un sistema de información, se basa en la idea de que un sistema puede ser construido e instalado gradualmente en lugar de hacerlo todo de una sola vez. Se aplica en forma muy similar al concepto de “release” utilizado por los proveedores de software, pues cada nueva versión de un sistema incorpora un grupo adicional o mejorado de funciones.

La idea central en el desarrollo de un sistema por versiones es instalar, lo más pronto posible, las funciones y facilidades más importantes para el

usuario, de tal manera que pueda obtener los beneficios del nuevo sistema sin tener que esperar a que todo el sistema esté construido. Las funciones que constituyen el resto del sistema se irán incorporando paulatinamente a medida que se desarrollen nuevas versiones, hasta que, con la última versión, se hayan incorporado todas.

5.2.- Desarrollo de sistemas por versiones

Una vez que se ha definido la arquitectura de sistemas para un área del negocio, la siguiente fase –análisis y diseño general- se cumple para cada uno de los sistemas que componen esa arquitectura. Análogamente, una vez que se ha diseñado la arquitectura de un sistema, las restantes fases -diseño detallado, construcción y pruebas e implementación- se repiten tantas veces como versiones se desee o se pueda desarrollar.

El desarrollo de sistemas por versiones y su solapamiento permiten optimizar el uso del personal y, a la vez, satisfacer más rápidamente las necesidades prioritarias del usuario.

La administración de versiones permite, además, manejar convenientemente los cambios que a través del proyecto el usuario va requiriendo, ya que cada nueva versión puede incorporar los cambios que hayan sido planteados por los usuarios, una vez que han adquirido experiencia en la operación de la versión instalada.

Es importante señalar que el desarrollo por versiones no sólo permite que los beneficios del nuevo sistema puedan obtenerse a más corto plazo que si el sistema se instalara en una sola pieza, sino que también permite manejar eficientemente la complejidad total del proyecto, al concentrar la atención de sus participantes en un reducido número de funciones.

Es necesario, sin embargo, tener presente que el desarrollo por versiones presenta dos problemas:

- Aumenta el riesgo de tener un producto heterogéneo. Se hace necesario, por lo tanto, vigilar muy de cerca que cada versión encaje dentro del concepto arquitectónico del sistema.
- Crea, en muchos casos, la necesidad de desarrollar módulos provisionales de interfaz con el sistema actual. Dado que el nuevo sistema sólo se instala parcialmente, es posible que deban conservarse algunos módulos del sistema vigente, por lo que pueden requerirse módulos provisionales que permitan intercambiar datos entre ambas partes.

5.3.- Mantenimiento por versiones

La implantación de sistemas por versiones permite manejar la necesidad de cambios en forma planificada, ya que los diferentes cambios solicitados por el usuario pueden irse acumulando con el fin de incluirlos en una versión posterior. Un criterio similar puede aplicarse cuando el sistema ya ha sido totalmente implantado y todos aquellos cambios que no sean de emergencia pueden reunirse para una próxima versión.

Es mucho más eficiente incorporar de una sola vez todo un conjunto de cambios en un sistema que irlos incluyendo individualmente, a medida que van siendo solicitados. Adicionalmente, es muy probable que una nueva versión pueda ser probada mucho más a fondo, por lo que su nivel de calidad será mayor que la de un sistema con varias modificaciones puntuales.

Una nueva versión implica un solo proceso de diseño y actualización de la documentación, por lo que los modelos, tanto de datos como de procesos, serán actualizados de una sola vez; mientras que aplicando los métodos tradicionales de mantenimiento puntual, este proceso debe repetirse tantas veces como cambios se requieran, con la consiguiente pérdida de tiempo y esfuerzo.

En resumen, podemos afirmar que el mantenimiento por versiones favorece la eficiencia del personal técnico y fortalece la calidad de los sistemas instalados.

6.- Planificación de versiones

Una vez definida la arquitectura de un sistema, la atención pasa a centrarse en la forma cómo será desarrollado e implementado, para lo cual se establece el plan de versiones y se definen grupos de trabajo.

La división en grupos de trabajo, para proyectos medianos o grandes, se hace con el fin de mejorar las comunicaciones y coordinar mejor las diferentes tareas de construcción e implantación.

Es muy común que el criterio para definir versiones y grupos de trabajo se base en las características de los módulos que componen el sistema: entrada de datos, reportes, etc. Resulta tan obvio y están tan perfectamente delimitados estos módulos, que resulta difícil no caer en la tentación de subdividir grupos de acuerdo con ese criterio. Sin embargo, estas divisiones o fraccionamiento de versiones y grupos de trabajo a menudo causan que los equipos tropiecen unos con otros durante el desarrollo de la versión.

Una división establecida sobre un criterio físico, normalmente crea muchas interdependencias entre los subproyectos, ya que un mismo proceso estará siendo atacado por varios grupos de trabajo a la vez, lo cual genera la necesidad de que un grupo termine algo, antes de que otro pueda realizar su trabajo. Esto se hace especialmente crítico cuando comienzan a realizarse las pruebas de integración.

Si en lugar de fraccionar los proyectos de acuerdo a un criterio físico -por lotes, en línea, etc.- se utiliza un criterio conceptual -entrada de materiales, salidas, transferencias, etc.-, es muy probable que existan menos interdependencias y que cada grupo concentre su atención en un solo problema, todo lo cual evitará interferencias y contribuirá a preservar la integridad conceptual del sistema.

6.1.- Necesidad de una visión global

Es indiscutible que, al subdividir un sistema en versiones, aumenta el riesgo de heterogeneidad, ya que, ahora, una misma aplicación podría quedar construida bajo múltiples estilos. Esto impone la necesidad de especificar con mayor detalle el diseño arquitectónico y de velar por su integridad con mayor celo.

Sabemos que la forma más eficaz de trabajar en la construcción de un sistema consiste en dividir el sistema en versiones. Sabemos también que este múltiple fraccionamiento implica riesgos grandes para la homogeneidad del producto final, lo cual impone la necesidad de que, al desarrollar un sistema por versiones, exista una vigilancia estrecha sobre la integridad conceptual y arquitectónica del sistema. Debe estar claro, pues, que en un sistema de mediana envergadura cada tarea que se complete en cada grupo de trabajo debe ser evaluada por un responsable de la calidad del sistema, encargado de preservar dicha integridad. Esa tarea puede recaer en el diseñador del sistema o en los roles que conforma lo que en el libro de *Gestión de Proyectos de Tecnología de Información* hemos denominado el grupo de revisión.

Es una práctica común -y muy sana- vigilar la ejecución de los planes de trabajo y seguir de cerca el cumplimiento de presupuestos y fechas. Sin embargo, tan importantes como el dinero y los recursos invertidos en desarrollar un sistema son la calidad y el concepto del nuevo sistema, por lo que es totalmente razonable que éstos también sean vigilados y seguidos de cerca. Por eso, en un proyecto de mediana importancia se hace indispensable cumplir ejercicios de control de calidad.

6.2.- Elaboración de un Plan de Versiones

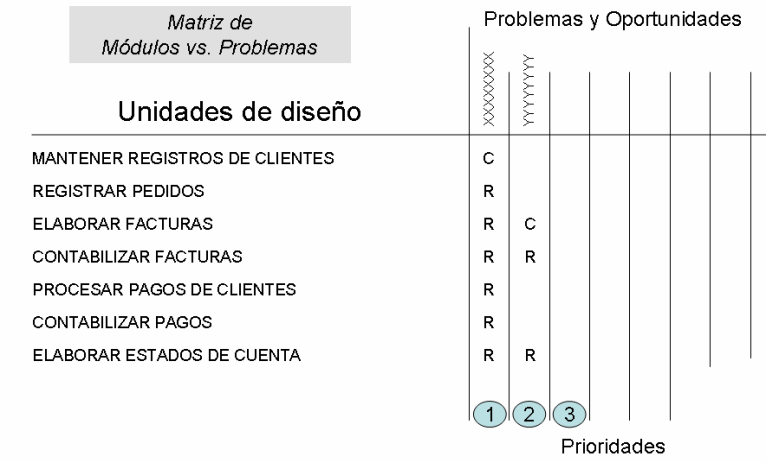
El desarrollo de un plan de versiones debe realizarse en dos pasos: primero, establecer una estrategia de versiones basada en las prioridades del negocio, y, después, ajustar esa estrategia, de tal forma que puedan entregarse resultados a la función en plazos lo más cortos posibles.

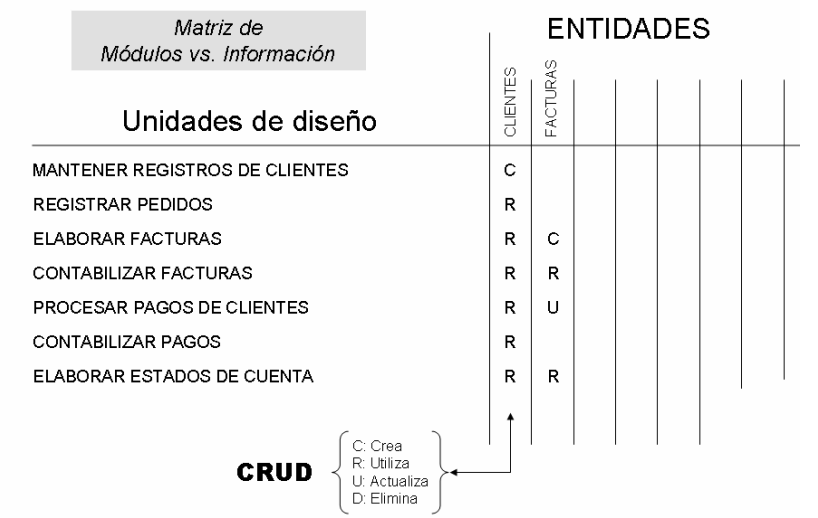
La elaboración de un plan de versiones requiere poner en perspectiva los beneficios que se esperan derivar del proyecto -problemas y oportunidades de mejora-, junto con todo el diseño arquitectónico.

Los pasos que deben darse para elaborar un plan de versiones son:

- Interrelacionar problemas y oportunidades con unidades de diseño; ello permitirá establecer las prioridades de desarrollo que, desde el punto de vista del negocio, deben darse a cada unidad de diseño.
- Interrelacionar las unidades de diseño con las estructuras de datos; ello permitirá establecer las prioridades técnicas de desarrollo.
- Contrastar las prioridades del negocio y las prioridades técnicas, con el fin de hacer los ajustes necesarios y establecer una única lista de prioridades de desarrollo.

Al realizar el análisis de las prioridades técnicas, se buscará identificar cuáles son las unidades de diseño que actualizan -crear, actualizar, eliminar- registros requeridos por otras unidades de diseño. Por ejemplo, la aplicación Facturación requiere que el Directorio de Clientes esté instalado, ya que, de lo contrario, no será posible imprimir los encabezados de las facturas; así pues, la aplicación Directorio de Clientes debe desarrollarse antes que la impresión de facturas.





Si las prioridades del negocio no establecen algo diferente, el orden de desarrollo debe ser: primero, Directorio de Clientes y, después, Facturación. Sin embargo, es muy probable que las prioridades del negocio señalen la necesidad de implantar primero Facturación. También, es muy probable que, para los efectos de Facturación, no sea necesario desarrollar toda la aplicación de Directorio de Clientes, sino únicamente su unidad de diseño -Actualización de Registros de Clientes-. Eso permitirá establecer la estrategia de versiones que puede trazarse para estas dos aplicaciones, con el fin de satisfacer, simultáneamente, las prioridades técnicas y las del negocio.

La estrategia de versiones puede llevarse más lejos aún. Supongamos que, dentro de nuestro ejemplo, para el funcionamiento de Facturación sólo es necesario contar con las funciones de Añadir un Nuevo Cliente y Modificar Cliente. Ello permitiría postergar para otra versión la función de Eliminar un Cliente. El resultado en este caso sería: fraccionar una unidad de diseño en dos o más unidades de diseño y dar a cada una de ellas la prioridad necesaria.

Así pues, al contrastar las dos listas de prioridades, se tomará, para cada Unidad de Diseño, una de tres alternativas:

- Adoptar la prioridad del negocio
- Adoptar la prioridad técnica

- Fraccionar una unidad de diseño, con el fin de agrupar las tareas de mayor prioridad en una unidad de diseño y las restantes tareas en una o más unidades de diseño.

Es importante evitar repeticiones innecesarias de trabajo, ya que ello impacta negativamente la productividad. Cuando mencionamos "fraccionar una unidad de diseño", no deseamos implicar que deban reemplazarse las especificaciones de esa unidad por dos nuevas especificaciones. Lo que quiere decir es que a las especificaciones actuales de esa unidad de diseño se le debe añadir una nota, en la que se especifique cuáles son las tareas que se incluirán en una versión y cuáles se incluirán en otras versiones.

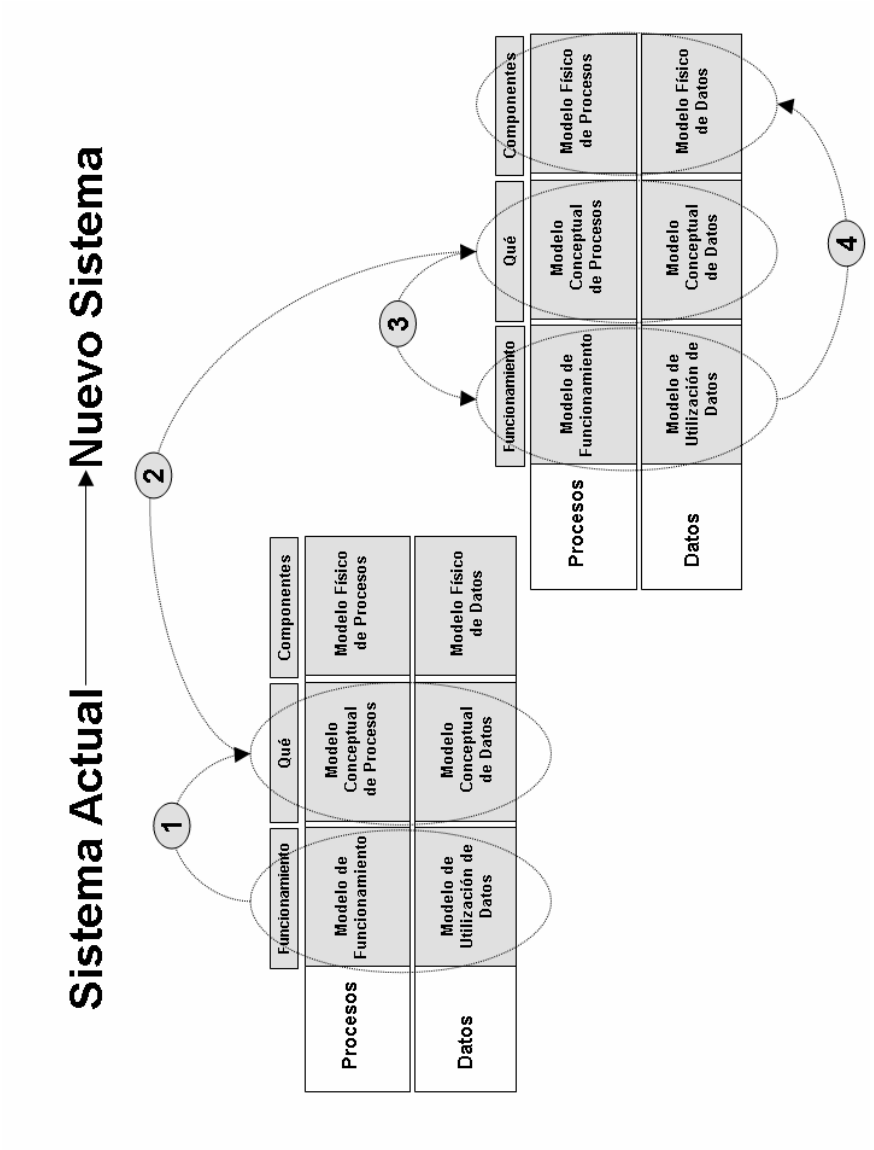
La siguiente consideración que debe hacerse, una vez definidas las prioridades para cada unidad de diseño, es la criticidad que, desde el punto de vista de seguridad, tiene el sistema, con el fin de asignar una prioridad al subsistema de seguridad. Esto es, si se trata de un sistema que procesa información muy sensitiva para la empresa -nómina, cuentas por cobrar, cuentas por pagar-, el subsistema de seguridad tiene alta prioridad y por lo tanto debe formar parte de la primera versión. Sin embargo, si se trata de una aplicación de menor criticidad, una decisión práctica puede ser postergar para otra versión el desarrollo de los módulos de seguridad.

Los componentes del subsistema de seguridad guardan relación con los componentes del sistema; o sea que es muy probable que la primera versión no requiera todo el subsistema de seguridad. Lo mismo puede ocurrir con las subsiguientes versiones. Por lo tanto, será de gran utilidad establecer los requisitos de seguridad de cada unidad de diseño y, en base a ese criterio, establecer la estrategia de desarrollo para el subsistema de seguridad. Este enfoque permitirá fraccionar en versiones el subsistema de seguridad, de tal forma que los componentes más necesarios sean desarrollados en primer lugar y, posteriormente, los módulos complementarios.

Análoga consideración debe hacerse con el subsistema de administración. En este caso, el desarrollo de la mayoría de las facilidades que se dan al Administrador del Sistema pueden ser incluidas en versiones futuras; sin embargo, eso acarreará la necesidad de utilizar programas de utilidad general para la actualización de tablas, mensajes, ayudas y parámetros, por lo que, si se adoptara una decisión de esa índole, deberán tomarse las previsiones necesarias para que el Administrador del Sistema reciba el apoyo necesario del personal de informática hasta tanto no se automaticen sus tareas en la forma

requerida, pues debe recordarse que el uso de estos programas utilitarios normalmente no resulta simple.

Al igual que el subsistema de seguridad, los módulos de administración guardan relación con los componentes del sistema, por lo que es muy probable que la primera versión no requiera todo el subsistema de administración. Por lo tanto, será de gran utilidad establecer los requisitos de administración de cada unidad de diseño y, en base a ese criterio, establecer la estrategia de desarrollo para el subsistema de administración. Este enfoque permitirá fraccionar en versiones el subsistema de administración, de tal forma que los componentes más necesarios sean desarrollados en primer lugar y, posteriormente, los módulos complementarios.


Sistema Actual
Nuevo Sistema
1
2
3
4
Procesos
Datos
Funcionamiento
Qué
Componentes
Modelo de Funcionamiento
Modelo Conceptual de Procesos
Modelo Físico de Procesos
Modelo de Utilización de Datos
Modelo Conceptual de Datos
Modelo Físico de Datos