

Diseño informático de un Tribunal de Cuentas Públicas

*José Miguel Rincón, Catedrático de Informática de la Universidad del País Vasco/EHU
M^a Gloria Areitio, Profesora Titular de Informática de la UPV/EHU*

Un Tribunal de Cuentas Públicas (TCP) es un sistema de gestión económica y social cuya actividad -circuitos gestores, flujos de funciones, metodología organizativa y acciones de sus agentes operativos- está impregnada de una carga de información que supone un altísimo porcentaje de esa actividad. Como en muchos sistemas pertenecientes al sector terciario, la información es la *materia de obra* más importante -casi única podríamos decir- de todas las que se pueden manejar.

Toda materia de obra *física* -materiales, energía, dinero y factor humano- de cualquier Sistema de Gestión (SG) es acompañada por materia de obra *informativa* que sirve de amalgama para todas las anteriores, especialmente con vistas al adecuado control de sus actividades. Un SG, como un TCP, es un "sistema mixto físico e informacional que puede ser indistintamente informado e informante". En el caso de un TCP la carga de información es tal que convierte al sistema en pre-

ferentemente informacional y tiene a la información como materia de obra casi única (Rincón, 1985). Un tribunal de cuentas es, por lo tanto, una expresión muy fiel del paradigma de los sistemas informacionales.

La conclusión que puede sacarse es inmediata: *informatizarse o morir* (¡valga el énfasis *revolucionario!*). Por esta razón, una de las primeras preocupaciones de los responsables directivos y ejecutivos de un TCP es diseñar, desarrollar, implementar y explotar un buen sistema informático, dentro de sus sistemas de información (SI). Las características del SI de un Tribunal de Cuentas Públicas es tan específico que casi podría decirse que la gestión sigue a la informática. ⁽¹⁾

(1) Tomemos esta frase como una figura retórica para enfatizar la necesidad de la informatización. La Informática, como todo el mundo sabe, no es más que un instrumento que exige la presencia de un universo de actividades políticas, económicas, sociales, e industriales para poder vivir. Únicamente hemos querido resaltar que los agentes activos de un TCP se deben mover siempre en un "espacio informático" muy amplio y muy denso.

Si admitimos esa gran presencia de la información y la especificidad de los flujos informacionales y de los diseños, desarrollos programáticos, y actos de producción informáticos podemos preguntarnos cuáles pueden ser las *pautas* que encuadran y rigen la existencia y desarrollo del Sistema de Información de un TCP. Y también cuáles pueden ser las *normas* para llevar a cabo el diseño, construcción e implementación de su Sistema de Información Automatizada (SIA). Podemos examinar los puntos más importantes que constituyen el esquema básico del sistema de gestión y del sistema de información de un TCP:

* En primer lugar hay que decir que los entornos del sistema son en un 90% grandes productores y consumidores de informática. Un TCP, a diferencia -por ejemplo- de un instituto de estadística, es preferentemente un consumidor de información formalizada informáticamente: bases de datos, ficheros de directorios, ficheros de transición y ficheros complementarios. La información

automatizada que producen es *de contraste* con la existente en los espacios auditables y no es de diseminación expansiva, ni extensiva para un amplio espectro de población. Por el contrario, las conclusiones políticas y sociales sí se expanden por un amplio espectro social; pero en general no se replican informáticamente sobre otros órganos o individuos.

* Los *proveedores de información* informatizada -entornos de entrada- son parte del propio ente general que los tribunales de cuentas tienen que auditar; consiguientemente sus relaciones tienen que ser fluidas, aunque el SIA de los TCP debe acoplar sus capturas a los formatos y cadencias de sus proveedores (González, 1995). Todo ello revierte en una mayor necesidad de control y de reducción paramétrica de las tablas, registros y atributos componentes de las bases de datos. En términos informáticos

podemos decir que es necesario un desarrollo específico y complejo de las tareas de captura, de control inmediato de los errores y de la formalización de un nivel alto de fiabilidad de una información que a partir del momento de su captura tiene que marchar en forma autónoma y producir resultados contrastables con sus orígenes.

* Los *consumidores de información* informatizada por el TCP -entornos de salida- tienen que estar dispuestos a contrastar su actividad gestonaria con los resultados obtenidos y manejados por los auditores. Esta situación puede ser objeto de controversia -tanto en la auditoría financiera y contable, como en la de gestión- y exigirá muchas veces de los auditores la necesidad de ampliar su campo de investigación aumentando su capacidad informacional con nuevas peticiones a las bases de datos del TCP.

* El esquema de producción informática dentro del tribunal de cuentas públicas se basa en la construcción -a partir de las informaciones de entrada- de un banco de datos global (formado por una o más bases de datos también de ámbito global). Este es el aspecto **CORPORATIVO**. Por otra parte, los agentes activos de las áreas gestonarias del TCP podrán obtener, de esas BD's corporativas, la información necesaria para construir su documentación y las bases de datos transaccionales para llevar a cabo su actividad auditable. Es el aspecto **DEPARTAMENTAL**.

* Si trasladamos el esquema funcional expuesto a un modelo informático podemos llegar a la conclusión de que el núcleo de las aplicaciones informáticas reside en la creación y mantenimiento de una BD corporativa nutrida con los datos capturados desde el entorno, y una serie de operaciones en cascada que pongan esa información al alcance de los auditores.

* Los usuarios finales del sistema informático son fundamentalmente los auditores que trabajan fuera del ámbito del sistema de información del propio TCP; se asemejan a los profesionales especialistas de cualquier empresa de auditoría, de consulting o de distribución. Se precisa la presencia activa de los auditores en el centro, organismo o institución auditada, con la especial particularidad que puede ocurrir que los auditores del TCP necesitan adaptar, corregir o ampliar sus documentos y anotaciones de partida. Precisan disponer de información que sea inmediatamente informatizable y de herramientas que permitan tratamientos primarios o la construcción de consultas discontinuas y frecuentemente inestables.

* Los usuarios del sistema informático tienen que replicar su información sobre una base de datos de tipo *histórico*, pero también de ámbito corporativo. Esta información será fundamental para la redacción final de los informes de



auditoría de los responsables de las áreas y para la dirección del TCP. Su carácter histórico hace de esta BD una herramienta de gran valor para la fijación de objetivos que constituyan la finalidad del sistema en etapas sucesivas. ⁽²⁾

Como conclusión podemos decir que un TCP es un sistema de gestión que maneja una gran carga de información organizada esencialmente en un Sistema de Información (SI) en forma de dos grandes bases de datos corporativas: una que almacena los datos de la contabilidad presupuestaria de las áreas -subsistemas- que deben ser auditadas, y otra que recoge los datos históricos de la actividad de auditoría. El diseño más racional del Sistema de Información Automatizada (SIA) o Sistema Informático debe seguir este esquema y tener en cuenta en la etapa de factibilidad las pautas y normas que han sido resumidas en los siete puntos anteriores.

Diseño del SIA en un TCP

El diseño del sistema informático aconseja informatizar el sistema de gestión de un tribunal de cuentas públicas reuniendo las características funcionales de este sistema que hemos expuesto con la metodología del modelo informático, también indicada, y formalizando las funciones de proceso de la información en aplicaciones que mantengan los siguientes principios:

- * Diseño de una estructura flexible de la BD corporativa y determinación de los modos de mantenimiento de sus datos.
- * Captura de los datos básicos desde los entornos mediante soportes directamente accesibles por el Sistema Informático (SIA).

(2) Incluso puede servir para cambiar las peticiones de información de acceso a los entornos suministradores de información primaria y, por lo tanto, para mejorar la captura de los datos.

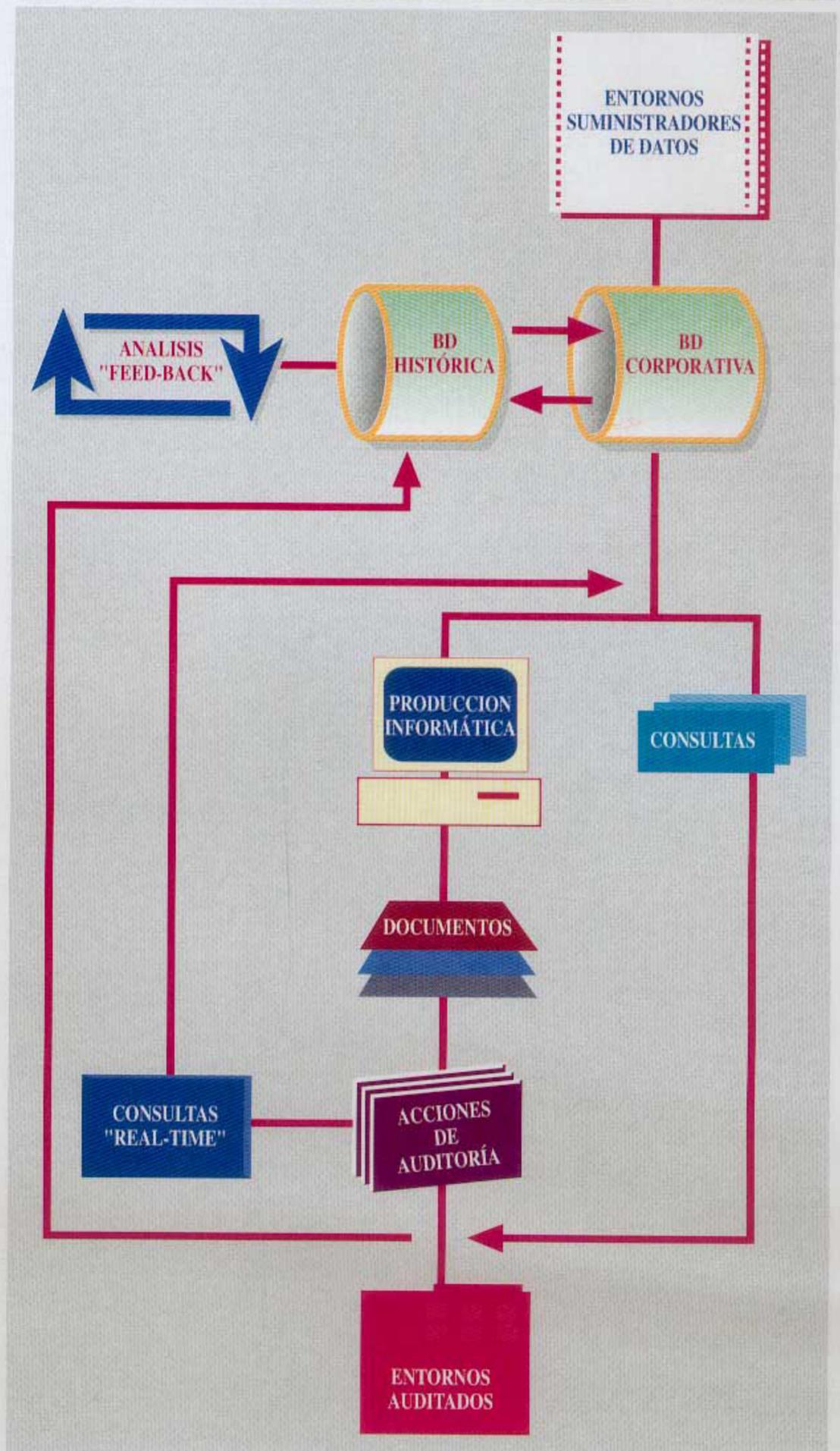
* Desarrollo de las funciones de producción de información informatizada basadas en la explotación de la BD.

* Utilización de recursos informáticos distribuidos y orientados a su explotación por los usuarios.

* Establecimiento de una línea abierta de consultas a la BD corporativa por parte de los usuarios.

* Diseño de la estructura de la BD corporativa que debe recoger la *traza histórica* de la actividad auditora del tribunal de cuentas.

FIGURA 1



✦ Desarrollo de las funciones de análisis del comportamiento de la actividad gestiona-ria del TCP basadas en la explotación de la BD corporativa de datos históricos.

La forma natural -casi evidente- de informatizar un sistema de información, como el que hemos descrito, es establecer un modelo distribuido *orientado hacia los usuarios* que fundamentalmente se basará en la obtención, para cada usuario o grupo coherente de usuarios, de una *imagen* de su parcela de información situada en la BD corporativa de datos suministrados por los entornos.

Cuatro grandes grupos de funcionalidades pueden ser contemplados desde el punto de vista del diseño del SIA; y un conjunto más numeroso de procesos aplicativos tendrán que formalizar esas funcionalidades en el momento de la construcción del SIA. De una manera resumida y meramente descriptiva estas son las funcionalidades:

□ **Diseño, extensión y mantenimiento de la BD corporativa de datos.** Como cualquier analista y responsable del sistema informático podrá confirmar esta es la funcionalidad más complicada, delicada y más sujeta a entropías externas de todas las que deben existir en el diseño y construcción del SIA. El gran volumen de datos, la complejidad de los entornos y la necesidad de decidir con gran precisión el grado de desagregación de los *items* funcionales originan la complicación de este aplicativo. Si cualquier aplicación es sencillamente un subsistema, ésta en el extremo de su desagregación debe ser considerada como una *black box* cuyos estímulos acabamos de ver y cuyas respuestas también deben ser cuidadas para fijar un método lo más abierto posible -pero riguroso y controlado- para servir datos a los usuarios.

□ **Explotación de la BD y a la producción de datos para los usuarios.** En un modo básico se

podrá disponer de las siguientes formas de resultados para los auditores:

✦ Listados estables de información procedentes de formularios e informes que relacionen, filtren, depuren, reduzcan paramétrica-mente y editen los datos contenidos en las tablas o ficheros de la BD.

✦ Documentos de trabajo editados en estado de proceso de textos o de métodos de autoedición con los índices procedentes del tratamiento de los datos correspondientes a la parcela auditada por cada usuario o grupo de usuarios.

✦ Vectores y submatrices (*vistas*) de la BD en estado de ficheros de trabajo y tablas de utilización inmediata implementadas en los recursos propios de los usuarios auditores.

✦ Los ficheros y tablas anteriores deberán tener sus datos en formatos adecuados para poder hacer exportaciones a *software* productivo u ofimático: SGBD's, hojas de cálculo, analizadores gráficos, paquetes estadísticos, procesadores de texto y cualquier otro producto que pueda servir para el análisis interactivo de la información e incluso para su presentación automática.

✦ Consultas interactivas por parte de los usuarios -auditores y agentes activos del sistema- sobre la BD corporativa para completar en forma inmediata su documentación y para realizar tareas de control en *tiempo real* en los entornos auditados.

✦ Dotación a los usuarios del *software* de consulta adecuado para realizar las indicadas en el punto anterior; un lenguaje SQL, o quizá la posibilidad de utilizar ventanas QBE, pueden ser soluciones agradecidas por los usuarios.

✦ Disposición en los recursos periféricos del *hard* y *soft* nece-

sarios para poder establecer comunicaciones con el *host* que sustente la BD corporativa y dotar de rapidez, flexibilidad y seguridad a las consultas interactivas.

□ **Diseño, extensión y mantenimiento de la Base de Datos corporativa con los datos históricos** resultantes de los actos de auditoría en los centros que se controlan por cada auditor o grupo de auditores. Sin entrar en demasiados detalles pensamos que habría que tener en cuenta:

✦ La información reglada y normativa que los auditores hayan deducido de sus actos de gestión.

✦ Las notas documentales sobre carencias, dudas, advertencias y en general items de carácter cualitativo y documental que vayan a utilizarse en la confección de los informes.

✦ Las consultas no estables que hayan podido producirse -posiblemente únicamente una referencia- para su análisis por el sistema informático y su eventual incorporación al conjunto de las consultas estables.

Debemos reconocer que la formalización y programación de alguna de estas cosas no es tarea fácil, pero creemos que vale la pena pensar en ellas.

□ **El análisis automático de la BD histórica** es una funcionalidad típica de manejo de información generada para el subsistema de dirección (en general para la cúspide decisoria). No sólo se trata de facilitar la redacción de los informes finales del TCP, sino también el estudio del comportamiento del organismo y deducir acciones *feed-back* que mejoren la gestión.

En la Figura 1 se esquematiza el diseño del sistema de información de un tribunal de cuentas públicas considerando las funcionalidades o grupos de tratamiento de la información que hemos expuesto.



HERRAMIENTAS DE AUDITORIA

El modelo de informatización examinado es canónico y probablemente ha sido implementado en muchos TCP, aunque admite algunas variantes, y también es preciso decir que no es el único que puede ponerse en marcha.

Reingeniería de diseño de un SI

Sea cual sea el modelo informático que se haya analizado, diseñado y construido, y cuyas funcionalidades estén en explotación si han transcurrido 4/5 años desde su inicio posiblemente habrá aumentado la entropía en el uso del sistema lo cual puede llevar a una falta de adecuación entre los resultados y los objetivos que constituyen la finalidad del sistema de gestión.

La cosa se agrava porque la distancia entre los rendimientos de las plataformas hard de hace 5 años y las actuales es enorme tanto en precio, como en rendimiento. Añadamos a esto la distancia entre el *software* de una fecha y otra. Y rematemos el tema fijándonos en los cambios profundos que ha habido últimamente en la propia configuración, diseño, estructura y uso de los sistemas informáticos y del propio sistema de información.

La situación puede significar que el simple mantenimiento correctivo e incluso el preventivo -siempre difícil de llevar a cabo *por falta de tiempo*- no sea capaz de adaptar el SIA existente a la realidad cambiante.

Se produce entonces la necesidad de una *reingeniería*, que si los cambios son muy profundos puede ir desde el propio diseño del SI hasta el *software* aplicativo (el SAP) o incluso el técnico (el SO).

El término de *reingeniería* supone, en su acepción más genérica, el replanteamiento global del SIA, en un momento dado, aprovechando el historial informacional y procedural hasta ese momento vigente para adecuarlo a las más recientes

expectativas del sistema de gestión y los avances del hard y del soft (Currid, 1994).

Tres pueden ser las razones que pueden llevar a una organización a acometer un plan de reingeniería de su sistema de información y de su sistema informático:

- Adecuación del SIA actual a nuevas demandas informacionales - internas y/o externas- del sistema.
- Mal funcionamiento del SIA. La fiabilidad del sistema informático se base en el hecho de que debe ajustarse a sus especificaciones, con independencia de las entradas, nunca permitirá su corrupción (entropía), debe reaccionar de una manera significativa y útil en situaciones inesperadas y no debe dejar de avanzar dentro de unos límites correctos de coste.
- Adecuación de sus recursos *hard*, *soft* y factor humano a las nuevas expectativas del avance tecnológico. Esta situación debe ser contemplada a la luz de la posible baja de la productividad y consiguiente pérdida de eficacia del sistema.

Aunque el concepto de reingeniería -sobre todo si se aplica el término al diseño del sistema de información- es muy reciente, ya Lehman en 1980 exponía una serie de principios o *leyes evolutivas* del ciclo de vida de un SIA, algunos de los cuales pueden ser reconocidos como elementos que incidan en el planteamiento de una reingeniería (Lehman, 1980).

Dice, por ejemplo, que un SIA que se explota y mantiene en un ambiente cambiante debe alterarse o será cada vez menos útil. Se basa en el razonamiento de que cuando un sistema *hard/soft* se construye para asistir a las necesidades de un entorno, éste queda modificado por su presencia de forma que los usuarios cambiarán su comportamiento a medida que se familiaricen con él y obligarán a su redefinición.

Asimismo, un SIA que paulatinamente se vaya adaptando a las nue-

vas situaciones, sin cambiar su estructura en profundidad, se va haciendo cada vez más complejo y difícil de manejar.

Durante el tiempo de vida del SIA, aunque se produzcan cambios de mantenimiento en sus recursos y se reajusten los servicios del *software* aplicativo, la metodología organizativa permanece estable y el *gap* entre el par *hard/soft* y el esquema del sistema de información se agranda.

Parece que podemos concluir que después de un tiempo de funcionamiento del sistema informático de un gran sistema de información, como puede ser un TCP, es necesario plantearse un proceso de reingeniería de los sistemas y del *software* aplicativo. Ello no implica que todo lo realizado deba ser abandonado, salvo en algún caso muy excepcional.

El problema puede residir en saber qué modelo escoger dentro del espacio tecnológico existente y de las metodologías sistémicas en el momento de realizar los rediseños.

Arquitectura cliente/servidor

Actualmente con el desarrollo fulgurante de la Microinformática y el uso extenso de sistemas y metodologías de construcción de *software* muy avanzadas y muy amigables para los usuarios, se están diseñando sistemas informáticos basados en la llamada *arquitectura cliente/servidor* que entrelaza soluciones *soft* adaptadas a un cierto tipo de *hard*. Al mismo tiempo las formas y técnicas de análisis de los sistemas, los cambios en la gestión económica y la amplitud de soluciones de las tecnologías de la información hacen que el diseño de los sistemas también haya cambiado preparando el terreno a las nuevas metodologías.

Esencialmente la *arquitectura cliente/servidor* es una forma de tratamiento de la información que divi-

de cada aplicación del SIA en funciones soportadas por plataformas hard interconectadas, formando una base de microordenadores y estaciones de trabajo que disponen de un *interface gráfico* (Windows95, NT, X-Windows, Wrap, ...), que se reparten la explotación de las BD's, el proceso de las aplicaciones y el uso de esas aplicaciones.

El diseño de una arquitectura de aplicaciones cliente/servidor se apoya en el uso de cinco productos y metodologías:

+ Los *Graphical Users Interfaces* (GUI) que permiten a un usuario interactuar con el ordenador por medio de la señalización, selección, activación y traslado (arrastre) de símbolos gráficos (*iconos*) que representan a los objetos ficheros, tablas, BD's, programas y comandos de acción que permiten el procesamiento y organización de la información. Las acciones sobre los iconos se hacen con un *ratón*. Los GUI's significan mejor comprensión de las acciones a llevar a cabo por parte del usuario, mayor productividad y un aumento de su comodidad y, consiguientemente, un mayor grado de satisfacción.

+ Potentes *sistemas operativos* que facilitan el manejo de los *interfaces gráficos*, de las operaciones multitarea y de las comunicaciones LAN y WAN.

+ Utilización de altos niveles de conectividad por medio del uso de los *Applications Programming Interfaces* (API's) que permiten separar la administración y el servicio de aplicaciones comunes a muchos usuarios, de su explotación personalizada por esos mismos usuarios. Básicamente las API's son *conectores* entre el servicio a los usuarios y el rastreo, consulta y explotación de las BD's utilizadas comunitariamente por esos usuarios.

+ *Servidores* de datos y transacciones formados por un ordenador que maneja una *cara* de las API's y sirve las aplicaciones a los ordenadores de los usuarios (clientes) a través de los GUI's. El servidor puede ser el *almacén* (*warehouse*) de las BD's para lo que dispondrá de un SGBD suficientemente potente y de un sistema de manejo de *queries*; o puede ser un elemento intermedio entre el puesto de trabajo activo o cliente y el ordenador en el que se implementan las BD's (*host*).

+ La *Tecnología Orientada a Objetos* (OOT) es la forma de programar -y de analizar las funciones- natural y eficiente para poder realizar los desarrollos de los procesos *ad hoc* o *standard* necesarios para construir las aplicaciones.

La arquitectura cliente/servidor⁽³⁾ puede ser de varios tipos:

★ Servidor de BD's y cliente de aplicaciones: sólo exige respuestas de las BD's.

★ Servidor de transacciones y cliente de aplicaciones: el cliente obtiene los registros debidamente procesados desde la BD.

★ *Peer-to-peer* (*de igual a igual*): es un doble enganche cliente/servidor/cliente.

★ Presentación distribuida: maquillaje de las aplicaciones servidas desde un *host* con una arquitectura convencional.

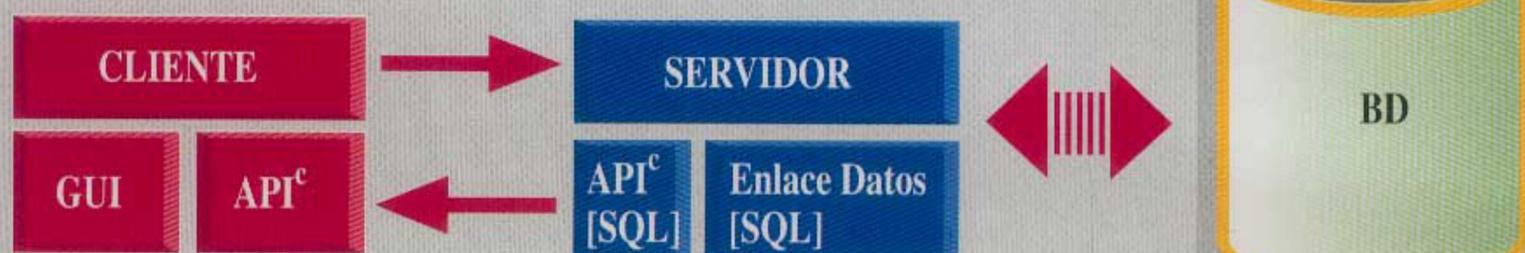
★ Presentación y lógica distribuida: es similar al anterior, pero con maquillaje de ciertos procesos de tratamiento servidos por el *host* y retocados en el cliente.

De todos esos tipos el más elaborado y rentable es el denominado comúnmente *Transaction Server Model* (TSM) y también, en ciertas circunstancias, el *Peer-to-Peer*. El primero de los expuestos es un simple manejador de bases de datos corporativas con clientes trabajando en grupo y usando un *software* adecuado (*Groupware*). Los dos últimos

(3) Recordemos que se trata de un conjunto global hard/soft, que forman un par indisoluble.

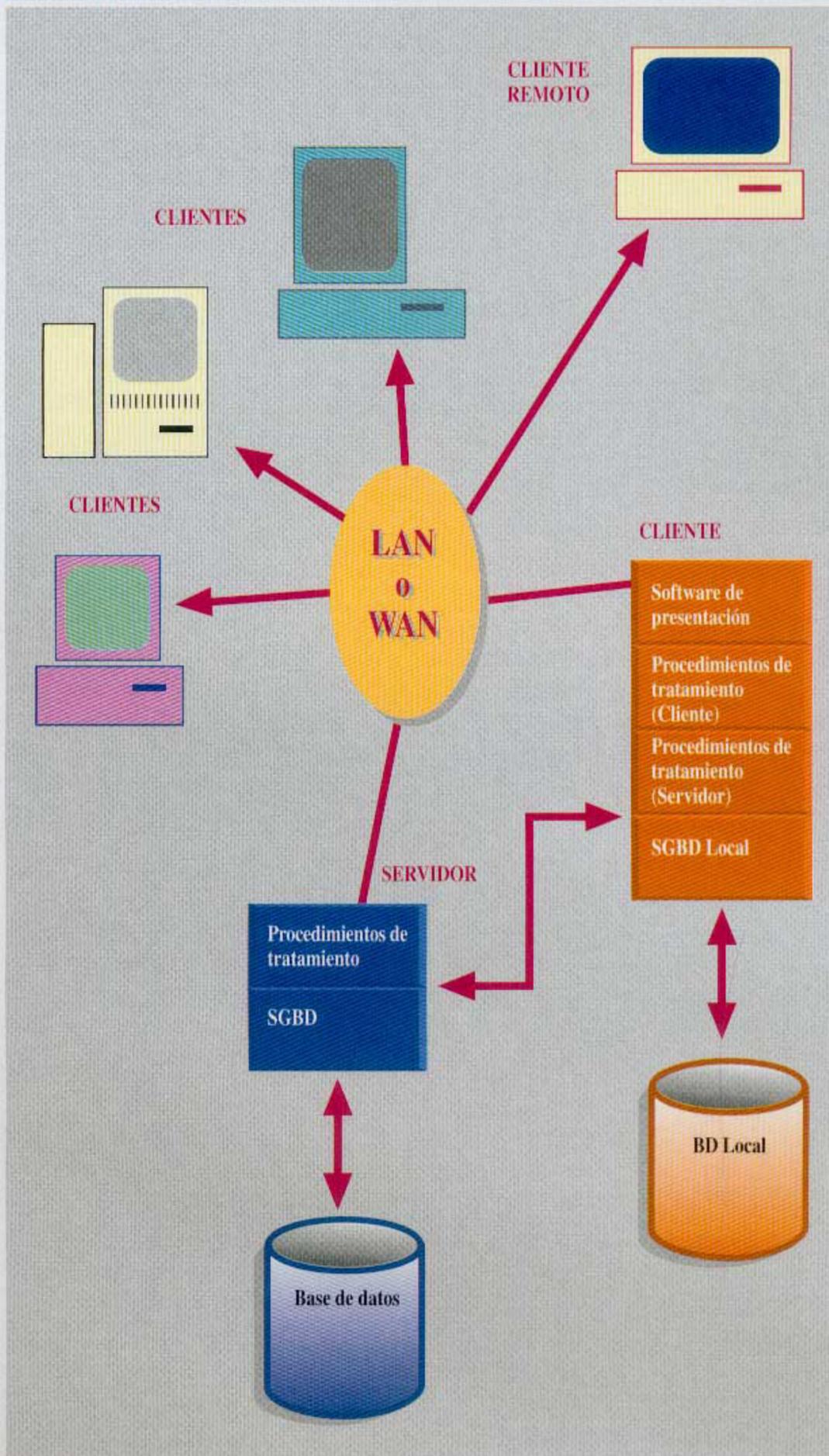
FIGURA 2

Petición de transacción



Retorno de respuesta

FIGURA 3



son aprovechamientos de arquitecturas convencionales soportadas sobre un *host*.

El modelo cliente/servidor de transacciones, no sólo divide la función entre adquisición de datos y procesos, sino que también rompe

la lógica de la aplicación entre el cliente y el servidor. Una transacción es una *unidad lógica de trabajo* que el *cliente* demanda y el *servidor* pasa a realizar. Este completa las operaciones necesarias para que el otro muestre resultados para el usuario, que no se entera de esas

operaciones transaccionales, ni de la forma *back-ground* de las interrogaciones y demandas de datos a las BD.

Las transacciones se *heredan* por todas las aplicaciones existentes en el Sistema Informático. Cada cliente maneja y soluciona diversas aplicaciones que pueden tener transacciones comunes.

El servidor *maquilla* (es decir, hace presentaciones cómodas y siempre muy parecidos entres sí) los resultados de las transacciones y hace las presentaciones de los datos de todo tipo -texto, números, gráficos, ...- de la forma pedida por los usuarios en sus puestos clientes. Para hacer esto el servidor utiliza las API's y los enlaces gráficos del usuario (los GUI's). Por ejemplo, el servidor puede *recordar* que un puesto-cliente quiere los formularios de salida en un formato determinado y siempre *maquilla* los datos para presentarlos así. El usuario trabaja muy cómodamente, con todo tipo de información, y la eficacia de sus trabajo aumenta (ver Figura 2).

El empleo de la tecnología orientada a objetos (OOT) hace extremadamente rentable el diseño del *software* para construir las API's y montar las transacciones. Es muy útil y eficaz el concepto de *herencia* que permite que muchas acciones puedan utilizarse con la misma estructura en diferentes contextos de programación y explotación de aplicaciones, funcionalidades y unidades de uso ⁽⁴⁾.

La ventaja de utilizar el modelo de servidor de transacciones (y eventualmente el *peer-to-peer*) sobre otros tipos, es que en las aplicaciones cuando hay cualquier proceso es el servidor quien devuelve los registros de las BD's debidamente procesados, y al cliente sólo le resta editarlos y pasarlos al sistema de

(4) Es lo que ocurre en todas las aplicaciones standard de los paquetes integrados tipo "office"; acciones como copiar, cortar, pegar, eliminar, buscar y otras muchas son idénticas y se manejan de la misma manera en diversos productos. Se facilita así la familiaridad del usuario con el software aplicativo.

recuperación (pantalla, impresora,...). El cliente no se preocupa más que de hacer peticiones y esperar los resultados. El tráfico de información es notablemente menor, la gestión de transacciones -cálculo y tratamiento en general- reside en el servidor, y la gestión de las presentaciones (que es más débil y más standard) se hace en el cliente. La BD puede estar en el servidor (si el Sistema de Información no es muy grande) o en un *host* corporativo, si se trata de un gran SI.

En la Figura 3 puede verse un esquema de una arquitectura cliente/servidor. Los recursos de los clientes pueden ser de cualquier tipo, lo mismo puede decirse con respecto al servidor; esto hace que la solución informática sea muy versátil y no constriña la libertad de la organización con respecto a las adquisiciones de recursos.

Bibliografía

- AREITIO,G. (1995) - *Auditoría de un SIA. Consideraciones de calidad* IX Reunión ASEPELT, vol. 5. Santiago de Compostela.
- CASHIN,J. (1993) - *Client/Server technology* COMPUTER TECHNOLOGY RESEARCH. Charleston (USA)
- CURRID,CH. et alter (1994) - *Computig strategies for reengineering your organization* PRIMA PUBLISHING. Rocklin
- GONZÁLEZ,L. (1995) - *El tratamiento informático de los presupuestos en AUDITORIA PUBLICA*, nº 1-03.95. OCEX. Vitoria-Gasteiz
- HAMMER,M. y CHAMPY,J. (1994) - *Reingeniería de la empresa* PARRAMÓN. Barcelona
- HARVEY,D.A. (1992) - *Downsizing Media* en BYTE, may 92. New York
- KAVANAGH,P. (1995) - *Downsizing for client/server applications* AP PROFESSIONAL. Boston
- LEHMAN,M.M. (1980) - *Programs, Life Cycles and the Laws of Software evolution* in Proc. IEEE, nr 68(9). New York
- MADERS,H. y CLET,E. (1995) - *Le management d'un projet* LES ÉDITIONS D'ORGANISATION. París
- RINCÓN,J.M. (1985) - *Diseño e informatización de sistemas de información económica* Sº EDITORIAL UPV/EHU. Bilbao

Conclusión

Un problema general en una organización informatizada y apoyada en una o dos grandes Bases de Datos corporativas es la posible degradación de la información contenida en ellas cuando se reparte entre los usuarios, especialmente si hay que hacer transacciones operacionales con los datos. La integridad de los datos está en juego y la seguridad de la información también. Después de un tiempo de actividad es casi seguro que surja la necesidad de una reingeniería (Hammer, 1994).

Un TCP, si ha seguido las pautas de diseño indicadas anteriormente, es un sistema que posiblemente puede estar en una situación de reingeniería como la que acabamos de citar. Por otra parte, su sistema informático es un candidato muy cualificado para hacer un rediseño y enfocar una metodología cliente/servidor. Hay una gran BD corporativa, generalmente situada en un *host* potente y de buena capacidad de respuesta. El servicio informático es experto en la depuración de tablas, ficheros y vectores de bases de datos y maneja con eficacia la información superabundante de sus

entornos. Los usuarios, no sólo están acostumbrados a crear sus propios entornos de actividad, sino que trabajan con recursos propios y exigen autonomía para utilizar sus propias herramientas de *software* aplicativo.

Si la experiencia acumulada es suficiente consideramos a un TCP como un candidato idóneo para hacer una reingeniería hacia el modelo cliente/servidor. Es preciso, no obstante, que el SIA haya llegado a una situación adecuada de *rightsizing*; es decir, que exista la conciencia clara de que aunque la base (el *background*) del sistema sea un gran ordenador -un *mainframe* o un *cluster* de miniordenadores potentes- el espacio transaccional debe estar entregado a la microinformática. Esto requiere una reingeniería de diseño.

El sistema de información de un TCP tiene que tener en cuenta que los grandes sistemas son muy vulnerables ante los cambios solicitados por los entornos y ante las exigencias de los usuarios -auditores y agentes activos- demandadores de una informática personalizada y adaptada a sus necesidades. ■