

CLASE TEÓRICA DEL 02/07/01

Tipos de especificación

Lenguaje normalizado

No es el lenguaje formal, sino el normal. Lo que existe en este lenguaje es la norma IEEE 830. Lo que tiene esta norma es una lista de control para asegurar que todos los temas estén especificados. Esta lista de control, por ejemplo en lo que son las especificaciones o necesidades funcionales, prevé que se debe contemplar la performance, o sea tiempo, precisión, volumen.

¿Qué quiere decir asegurar la performance? Que yo debo asegurar como desarrollador del producto que con esta máquina yo voy a tener tal performance con la aplicación que voy a hacer. No es que "ey necesito más máquina", porque así cualquiera arregla problemas de performance. Es lo mismo que yo quiera hacer desaparecer Oro Verde y pongo dos bombas atómicas y desaparece, pero no es performance digamos.

- ★ ¿Cuáles son las restricciones físicas en el diseño? ¿El lenguaje que se va a utilizar?
- ★ En el mantenimiento ¿cómo va a ser el diagnóstico? Si tiene autodiagnóstico, o sea si me asegura que no haya habido algún financiamiento de los programas

El autodiagnóstico va más allá de las pruebas comunes. Se hacen todas las pruebas comunes, pero además asegura que si se hizo algún cambio durante la programación, ese cambio es válido. No es que escribió un error por pretexto.

- ★ Si hay utilitario para la prueba que permita repetirla y simule entradas y salidas.
- ★ ¿Cuál es la capacidad de crecimiento que tiene el sistema?. El crecimiento se puede limitar por algunos problemas de diseño o por los propios lenguajes que no tienen escalabilidad.
- ★ ¿Qué fiabilidad se le da? Acá se está hablando de la resistencia a errores. Cuántos errores admite por cada mil sentencias fuentes o eventos que generen sentencias fuentes de código entregado.

El tiempo promedio entre dos fallas. Yo puedo tener una resistencia a errores muy baja pero si se me dan todas juntas me matan. O sea tienen que darse circunstancias especiales para que se presente el problema, por ejemplo pudo darse un caso que nadie probó porque jamás pensó que podría darse, pero aparece a los 8 meses, pero no que aparezcan todos juntos porque si empiezan a aparecer las excepciones todas juntos es porque está mal programado.

- ★ ¿Cómo es la recuperación? Enviciamiento de un archivo.
- ★ Facilidad de uso.
- ★ Ambiente de seguridad
- ★ ¿Qué se debe controlar en el diseño? Si en el diseño es muy importante controlar que de todo lo especificado en la parte anterior se hayan producido distintos módulos, una facilidad de prueba independiente por cada módulo, porque después se acoplan.
- ★ Testabilidad.
- ★ Reutilización.

En la fase de programación se debe controlar la coherencia entre el programa y el diseño, normas de programación, no trabajar con alta complejidad estructural. Presencia de invariantes, si se hace una cosa verdadera siempre sea verdadera. En la fase prueba se proponen cuatro tipos:

Pruebas estructurales: son pruebas de control de cobertura. Atiende a los aspectos de control de cobertura. Los lotes de prueba cumplan con toda la secuencia del programa (conjunto ordenado de instrucciones que se repiten el mismo número de veces en el mismo lugar del programa) condicionados por alguna bifurcación de algún lado, o sea que por lo menos cubra todo el programa. Es imposible que cubra todos los contenidos de la secuencia, por las combinaciones que puede haber, pero sí por lo menos que haga el control por primer elemento de un conjunto de secuencias, si el elemento se repite más de n veces, y por el último elemento. Hay un programa de Mc Cabe que te dice que cobertura no

está cubierta del programa. Vos le das el fuente y te dice que cobertura no está cubierta y además te filtra el lote de pruebas. Te dice que cosas están de más, probadas inútilmente, que es un vicio.

Pruebas funcionales: se ven las entradas y las salidas, no se ve internamente que es lo que hace.

Pruebas comparativas: hay dos ambientes de test que trabajan por separado, en forma distinta, porque un solo ambiente de test puede tener la misma inclinación a un mismo error, no dialogan entre ellos, solo después para comparar resultados.

Prueba de inspección: observar el código, o sea cómo está escrito, no para ver la calidad de la programación (eso es aparte) sino para ver si no presenta omisiones, errores. La única forma de detectar este tipo de cosa es viendo el código. Por ejemplo en el caso de que quiera en un programa que se maneja con password, permitir a un usuario "x" que lo use sin restricciones, entonces la única forma de detectar esto es viendo el código.

Prueba de stress: no está contenida en la IEEE830. Poner mucho volumen de información para ver en que momento se comienza a fatigar el software o hardware, o sea no empieza a responder como uno quiere.

La ISO 9004 habla de cómo hacer la oficina de tareas, el grupo que gestione la calidad durante el desarrollo. El grupo de calidad pertenece a la empresa que desarrolla, y su deber es cubrir las necesidades internas de su empresa pero también las necesidades de los clientes que fueron acordadas en el contrato de calidad. ¿Cuáles son los riesgos con este tipo de trabajo? Es decir que la empresa que desarrolla forme un grupo de calidad. Por ejemplo perder imagen, usar mayores recursos de la empresa de desarrollo, entonces le sale más caro. Para el cliente el riesgo es que tenga insatisfacción porque le están ocultando cosas, pérdida de confianza.

Característica	Para la empresa	Para el cliente
Costos Error en los costos	Error en los estudios, diseño. Material que necesita comprar. Eliminar cosas de la producción porque está mal construida, entonces hay que construirla de nuevo, eso es tiempo, es costo.	Tiempo de inactividad, pérdida de tiempo. Porque él participa del grupo de calidad, es la figura del grupo de calidad
Beneficio	Mayores ventas, mayor mercado	Menor costo, mayor confianza

¿Cuáles son las responsabilidades de la dirección de la empresa? Fijar política de calidad, objetivos de calidad, sistema de calidad conforme a esta norma y a lo que se contrató. También dice cómo debe trabajar el grupo de calidad. Y no tiene otras cosas más importantes que estas tareas. La norma ISO 9004 fija como debe trabajar el grupo de calidad.

Métricas de software (para tener calidad)

a) de complejidad:

- ✓ Cantidad de líneas de código: es una métrica fácilmente administrable porque yo elimino todas las sentencias que tienen asterisco y ya tengo el software. La crítica es que no me dice nada acerca de controlar la persona, nosotros no controlamos a la persona sino los procesos. Entonces yo sé que un sistema de contabilidad bancaria son parecidos en todos lados, cambian algunas cosas especiales, pero si yo sé que el sistema para el banco "N1" posee 10 millones de líneas de código y cobraron 5 mil dólares, para otro banco parecido del mismo nivel ocuparon 8 millones 500 mil líneas y dos mil dólares no es lo mismo para eso no sirve. Me sirve en la masa pero no me sirve en forma particular, para controlar a una persona. Acá no se controlan personas. La ventaja que tiene es que es de fácil localización.
- ✓ Cantidad de variables referenciadas en el proyecto.
- ✓ Cantidad de funciones.
- ✓ Cantidad de módulos.

- ✓ Complejidad ciclométrica de McCabe: es una medida que lo que mide es el nivel de anidamiento promedio que tiene un programa. O sea la cantidad de lógicas que debe tener el programador en un momento determinado. Se trata de no hacer programas muy complejos, tratar de separarlo en módulos. Porque a mayor nivel de complejidad voy a tener más tiempo de desarrollo.

b) de mantenibilidad:

- ✓ Cantidad de comentarios.
- ✓ Tamaño promedio de los módulos (que no superen las 250 sentencias).
- ✓ Cantidad de secuencias.
- ✓ Cantidad de K-NOTS: k-not es programación ni estructurada ni orientada a objetos, desestructurada. Cada vez que desestructurás algo tenés un k-not. Dificulta su seguimiento.

Uno puede observar a través de esto no solamente la parte de los requerimientos, las funcionalidades, sino también los cambios de los requerimientos:

- ¿En qué etapa se dieron esos cambios durante el desarrollo?.
- ¿Cuántos defecto esperaba encontrar y cuántos encontré?.
- ¿Cuál es el costo de cada defecto de acuerdo a la etapa en que lo encontré?.
- ¿Cuál es el costo de las pruebas? A ver si me justifica el costo de los defectos.
- ¿Cuál es la tasa de corrección por defecto?.
- ¿Qué puedo medir? Horas hombre, recursos utilizado, gastos de viaje, etc.
- Cantidad de defectos encontrados a través de inspecciones, prueba unitarias, pruebas de integración, prueba de sistema.

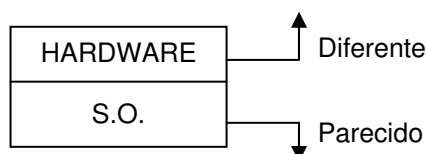
Aclaración: Efecto disfunción: una disfunción es obtener un resultado opuesto al que interesaba. Hay que tener cuidado con lo que se mide, y que la gente tenga conciencia de que no se los mide a ellos sino al proceso. Los defectos encontrados en forma temprana predicen los que se encontrarán más tarde. Anticipar problemas. No es que no vamos a tener problemas. Cuanto más problemas anticipemos más orden tenemos.

Sistemas abiertos

En la tecnología informática hay una amplia gama de llegar a una aplicación estandarizada a tener un chip estandarizado. Entonces se sistematizó en capas al sistema abierto:

- 1) Plataforma de hardware: considera que son abiertos tanto los Risc como los Mips, Power Pc de IBM, Alpha de Compaq, el Intel, considera a todos los procesadores desde distinto punto de vista como abiertos, no hay uno solo, en bus pasa lo mismo, en vídeo pasa lo mismo.
- 2) Sistemas Operativos: pasa algo parecido. Lo que está más o menos normalizado es las Shell (interfaz con las aplicaciones). POSIX: sistema operativo portable para ambiente de computación.

En el año 93 se requirió la estandarización de los sistemas abiertos, entonces se armaron los grupos POSIX, se armaron 15 grupos de trabajo. El grupo que más respondió fue el 1003.1, el 1003.0 es el ambiente que son sistemas operativos, el 1003.1 estableció como deben programarse las aplicaciones portables para que puedan ser compatibles con la norma POSIX. A partir de ahí apareció un solo Unix, después apareció WNT, W2000. De todas maneras no se administran todos iguales. O sea puedo tener algo muy parecido en la segunda capa y diferente en la primera.



- 3) Capa de redes: no está todo unificado, si bien casi todo se trabaja con TCP/IP, sigue habiendo otro tipo de redes.
- 4) Capa de herramientas de desarrollo.
- 5) Capa de Base de Datos.
- 6) Aplicaciones.
- 7) Integración gráfica

Criterios de apertura

- Interfaz absolutamente disponible. Yo pueda tener interfaces sin ningún problema.
- Las normas estén claras y bien escritas.
- La interfaz debe estar legalmente limpia.
- Las múltiples aplicaciones deben ser públicas y estar disponibles para muchas compañías.
- Disponibilidad de una implementación de referencia, a bajo costo.
- El control de calidad y de compatibilidad debe hacerse en forma independiente.
- Libertad de elección y no libertad a causa de la elección.
- Los estándares los fija el mercado: por ejemplo como ADA y DOS.

Objetivos de los sistemas abiertos

- Migrabilidad: pasar a diferentes ambientes sin migrar problemas.
- Compatibilidad: tanto de datos como de ejecutables.
- Escalabilidad: que sea independiente del volumen de datos.
- Conectividad: que sea necesaria para la interoperabilidad (capacidad de interactuar en línea con otras aplicaciones).
- Internalización: que pueda trabajar con diferentes códigos. Uno de los estándares que se utiliza hoy es el UNICODE (código de 16 bits).
- Seguridad: que esté homologado a un nivel de seguridad.

Características de los sistemas interoperables

- Que exista la red.
- Interconexión heterogénea.
- Cumplimiento de estándares.
- Flexibilidad.
- Control.
- Administración, porque la interoperabilidad es algo que se construye.

Seguridad

Existe la oficina nacional de seguridad en computación donde califican niveles de seguridad. Por ejemplo, nivel D es mínima protección (seguridad sin llave en una PC, o la que tenés en una PC pero dejas la llave puesta, tenés la password nada más); nivel C1 es seguridad direccional; nivel C2 es seguridad por control de acceso; nivel B1 es seguridad por niveles, se implantan fácilmente las bases de datos, a este archivo lo accede tal usuario si cumple con tal cosa solamente, puedo tener parte del archivo oculto. Hay que tener en cuenta que todo programa tiene errores. Hasta los de seguridad.

¿Qué características de arquitectura deben tener estos sistemas?

- Tecnología de micro-kernel.
- Tiempo real: exista sincronía entre lo realizado y efectivamente ejecutado. Sincronía entre almacenamiento y proceso.

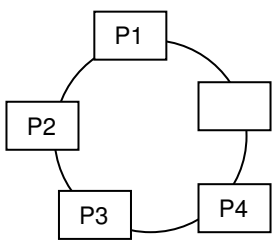
- Multiprocesamiento: puedo hacer ejecuciones en paralelo en tiempo real. Tener un equipo al lado que trabaje en paralelo en tiempo real. Si se me cae uno sigue el otro, esto significa NO-STOP en tecnología, tecnología de no caída. Se cae uno el otro lo levante automáticamente.
- Inquedición dinámica: la vinculación entre los módulos se haga en el momento de convocarlos, cada vez que se convocan. Si yo modifico un modulo, y este modulo está vinculado a otros módulos, la vinculación entre ellos me la hace en tiempo de ejecución, no al momento de compilar.
- Múltiples canales de comunicación para el mismo proceso. Esto es lo que se llama ahora Query paralelo. O sea el tipo detecta que yo puedo hacer este Query, y este Query y son independientes uno de otro, los hace en paralelo y sigue el programa.
- Sistemas operativos orientados a objetos: hoy todos los sistemas están orientados a objetos.

Selección del software base

Gestión de recursos, ¿cómo se hace esto?. Recurso CPU, o sea memoria, tiempo de procesador y acceso a disco. Esta trilogía es el corazón de la performance. Se debe medir eficacia, facilidad de uso, seguridad.

¿Cómo se comparten los recursos? ¿Cuáles son los criterios que se presentan o facilidades del sistema operativo para compartir recursos?:

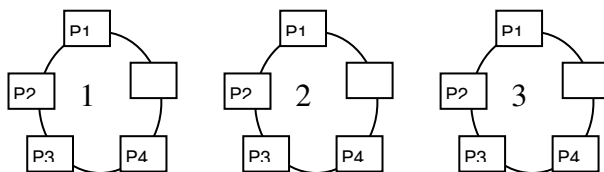
- **Time-Slice:** es una de las más simples. Significa cacho de tiempo. Esto es la forma más simple de compartir un recurso, el tema es ¿cómo se manejan los WAIT?, los waits son tiempos de espera de la CPU, cuando la CPU para poder hacer un trabajo está esperando una respuesta. Hay dos tipos, dependiendo del programa que corre:



- **WAIT largo:** es el más común. Es cuando el tipo está esperando un programa o una respuesta del operador. Pide datos y está esperando una respuesta. Es largo porque esto demora un segundo a que el tipo reaccione. Y normalmente los slice de tiempo son alrededor de 200-300 milisegundos. Todos los sistemas operativos multiusuarios ante los wait largos pasan tiempo
- **WAIT corto:** es cuando está pidiendo alimento a disco. Acá estoy teniendo 10-12 milisegundos.

Este sistema da poca administración en el sentido de que si hay algo prioritario qué hago. Meto todo y lo dejo a ese solo. Desperdicio recurso. Una alternativa es manejar prioridades, y los que manejan son anillos de prioridades.

No solo tiene el anillo de prioridad 1, de prioridad 2, de prioridad 3, el tema es que siempre que tenga trabajo en un anillo de mayor prioridad se hace el trabajo de ese anillo. Entonces puede haber gente esperando tener respuesta. Pero es bueno para salir de problemas puntuales.



Administración dinámica: para manejar los tres elementos, CPU, RAM y acceso a disco, que son las tres cosas que necesito para conectar mi producción, se creó el concepto de unidad de servicio. Una unidad de servicio es un poco de servicio de CPU, o un poco de servicio de memoria RAM o un poco de servicio de acceso a disco. Equilibrando estos tres factores yo obtengo un resultado de aprovechamiento equilibrado de mi sistema. Si yo veo por ejemplo que me estoy quedando corto de CPU, y yo me doy cuenta que todos los programas no pueden ejecutarse correctamente porque le falta CPU, yo castigo el uso de CPU, para que el uso intensivo de CPU sea en el horario menos concurrido de gente, y puedo generar grupos de performance.

Grupo de performance	Prioridad	Desde	Hasta
1	1	0	500

	2	501	4000
	3	4001	No
2	2	0	450
	4	451	fin

Se maneja en forma dinámica a través de una pantalla, con una tabla interna del sistema operativo. Además se pueden fijar horarios. Parte del concepto supermercado, la caja expres. Aquél que consume pocos productos del supermercado es atendido más rápido que aquel que viene con 4 canastos. Acá ocurre lo mismo, por ejemplo si yo quiero saber una dirección y tengo el documento no es lo mismo que quiera saber cuantos alumnos están en condiciones de rendir una materia en los próximos dos años, esto puede tardar unos segundos más.

El otro tema importante es como maneja los spool. Existen dos tipos de spool uno de entrada y uno de salida (éste último está vinculado a la impresión). No es lo mismo que tenga un archivo spool y lo mande completo a la impresora a que yo pueda segmentar ese archivo en tantas páginas y mandarlas individualmente entonces puedo disponer del spool para otros archivos.

El otro tema es el ambiente de desarrollo que soporta el sistema operativo. No todos los ambientes de desarrollo son compatibles con todos los sistemas operativos. Si estoy limitado por la aplicación porque ya la tengo, debo buscar un sistema operativo que trabaje en ese ambiente de software, por ejemplo si uso VBasic necesitaría W2000.

Otro tema es la facilidad de diagnóstico: un elemento que tenemos en sistemas grandes como Unix es poder tener login, no de BD sino de SO. El login lo que hace es registrar todo lo que hago con el sistema operativo. Registro todo el diálogo cronológico. Esto sirve para saber que pasó si hubo una interrupción, que fue lo último que habló. Y además me permite hacer un montón de estadísticas de comportamiento, de atención, de despacho, de tiempo muerto del operador. El problema es que el login no se puede llevar en archivos magnéticos pero si se para el sistema no sabes que fue lo último grabado hay que recurrir al papel.

Otra ayuda es el debuggin del sistema: dice por qué módulo ha pasado. Esto es muy importante en ambientes de sistemas operativos orientados a objetos, en donde si tengo 315000 archivos, por qué archivo/objeto pasó en el programa antes de caerse, un debuggin. Sirve para que el proveedor pueda apurarse para hacer el diagnóstico y la reparación. También sirve para ver que cosas son más usadas y entonces fijarlas en memoria.

TRACE: ayuda de rastreo. Esto se trabaja a nivel de microinstrucción.

El disco cada tanto aumenta su velocidad porque aumenta su capacidad de almacenamiento. La ingeniería de desarrollo de los productos hace que tenga que desarrollar una tecnología que permite mejorar la performance sin modificar mi tecnología original (matrices, máquina, etc.).

EI WAIT: es un loop que espera error de disco.

Restart o re arranque caliente; hace 10 años uno prendía una PC y a los 2 segundos daba el prompt. Y hoy demora cerca de 2 minutos. En los equipos grandes como la demora es considerable se ha optado no apagar nunca las máquinas. El síndrome de los lunes primero es que los motores no arrancan cuando se prenden. Segundo si por algún problema se cayó el sistema y por falta de electricidad la memoria RAM se pone a cero binario, re arrancar todo de cero demora aproximadamente 45 minutos. El restart es cada determinado espacio de tiempo grabar la situación de toda la memoria real y virtual en un disco. Existen instrucciones de hardware especiales que graban campos largos, entonces graban por ejemplo 4Ghz, si tenés 3 Ghz de memoria real y 1 de memoria virtual. La ventaja que hay es que cuando se vuelva a caer el sistema lo único que hay que hacer es apretar re arranque. Y en segundos tenés sistema operativo como estaba minutos antes de que se cayera. Y después hay que recuperar BD del último minuto. O sea el login dinámico como le llaman la gente que hace BD un login que está a propósito por estos minutos porque sabe que se trabaja con re arranque. El tema es el siguiente, no hay que tener un solo archivo de checkpoint (punto de chequeo), hay que tener dos discos checkpoint porque si se rompe cuando estas grabando encima perdiste todo. Entonces hay que tener

dos discos con archivos checkpoint. Segundo hay que tenerlo dedicado para eso a los discos porque sino las aplicaciones no trabajan.

Los aspectos Seguridad y Comunicación se verán cuando se dé el tema software de seguridad y software de comunicación respectivamente.

CLASE TEÓRICA DEL 27/08/01

Selección de software de utilidad

El software de utilidad se llama así porque es útil para desarrollar o administrar aplicaciones. En ese aspecto se puede dividir en dos grupos de software de utilidad:

- 1. orientado al desarrollo de aplicaciones
- 2. orientado a la administración y a la instalación o gestión operativa de la instalación

Software de utilidad orientado a la ayuda de computación para ingeniería del software o herramientas CASE

Siempre hubo herramientas de ayuda para el desarrollo de aplicaciones, porque desde un compilador en adelante, cada vez se fueron facilitando más. Lo que pasa es que las herramientas CASE fueron integrando las diferentes herramientas, fueron evolucionando. Hay dos tipos de CASE. Lo que se denomina UPERCASE, que es el soporte de CASE, y está orientado fundamentalmente al análisis, especificaciones y diseño, lo que se conoce comúnmente como herramientas de análisis. Y el CASE de bajo nivel, o CAP (Ayuda de Computación para la Programación), que está orientado a la codificación la prueba y el mantenimiento. Por ejemplo, Power Builder es un producto CAP, de bajo nivel. El Designer, por ejemplo, es un producto orientado al alto nivel. Normalmente uno puede, entre las herramientas, intercambiar información. Como herramientas están Saino, Sapiet (producto israelí), IEF (que es un producto muy bueno de Texas Instruments, que está relacionado al UPER y al CAP). Debe haber aproximadamente 50/60 herramientas CASE conocidas en el mundo, hay mucha oferta. De todas maneras en estos casos hay que buscar la que se use más, la que cumple con la mayor cantidad de estándares. Las mas usadas normalmente son Power Builder, de Sysbase, que tiene una especie de designer que hace el alto nivel y Case Visual Age, de IBM, y las herramientas de Oracle, que son el designer y el...?.. Son las tres más usadas en el mundo. Podemos analizar los tipos de case haciendo un cuadro como el siguiente:

		Integración	
		Superficial	Profunda
Perspectiva	Múltiple	2) Integración de control	4) Integración de control y datos
	Simple	1) Integración de control	3) Integración de control y datos

La forma del cuadro, con 4 cuadrados, lo utiliza el Garner Club para realizar productos, donde coloca en la columna superficial a los emprendedores, y en la columna profunda a los conservadores.

Por ejemplo hace unos 10 años atrás, los conservadores trabajaron con base de datos propietarias, jerárquicas, etc., y los emprendedores con base de datos relacionales. Hoy es distinto, los conservadores utilizan base de datos relacionales y los emprendedores utilizan base de datos orientadas a objetos. Cuando ubican los productos, lo analizan desde diferentes lugares. "Esto si bien tiene integración superficial pero múltiple, lo ubican por este lado", normalmente son los productos que van evolucionando.

La gente conservadora, no es que por ser conservadora es mala, tiene que conservar los clientes, entonces no los puede mover violentamente, por ejemplo están jugando al fútbol y ahora les decimos que jueguen con las manos, que hagan rugby, no puedes cambiarle las reglas del juego, entonces lo van llevando de a poco. La gente emprendedora no puede entrar a competir con los grandes conservadores,

tiene que hacer productos nuevos, a ver si tiene suerte. Para enganchar en el mercado lo que quiere defender, cosas nuevas.

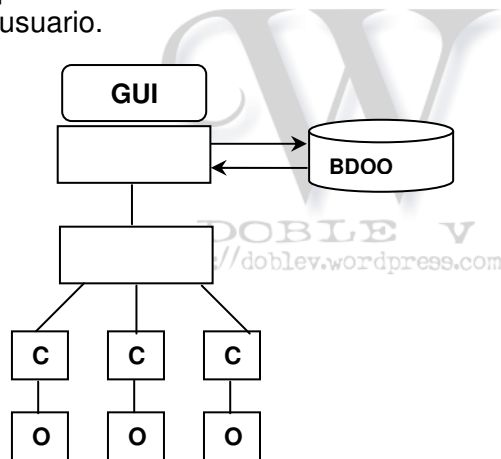
En 1, donde la integración es superficial y la perspectiva simple, los mensajes se pasan entre las herramientas. Cada herramienta es responsable de su propia representación. Esto significa que si tenemos un compilador y queremos hacer un debugger, o sea hacer un seguimiento del programa para ver por donde va, si corre bajo el debugger los mensajes son del debugger y hay que ir al manual del debugger, para entenderlo. Si lo corremos con el compilador, para los errores del compilador hay que ir al manual del compilador. Si bien una herramienta está sobre otra, se pasan los mensajes de la herramienta primaria, porque siempre contesta el compilador, o sea el programa compilado que vos corres.

En 2 la interfaz de usuario muestra datos que pertenecen a más de una herramienta. O sea que hay una interfaz única de usuario, no dos una del compilador y una del debugger. Se puede navegar el código desde cualquier punto, puedo registrar un pedazo. Esa es la ventaja que tengo, yo quiero probar una parte suelta y puedo registrar un pedazo.

En 3 son ambientes de unión de herramientas, o sea son plataformas de integración, se retiene y se comparte el estado del programa con todas las herramientas. Se puede hacer compilación incremental, o sea solamente compila lo que no corro, lo que saco, anulo, para anular es fácil porque hay código de sentencia nula en los assembler, en los lenguajes de máquinas. Para agregar lo que hacen, ponen un punto al final del programa y después vuelven.

En 4 existe un ambiente unificado, y tiene 4 partes:

- a. Un kernell orientado a objetos, que provee la integración de control y de datos. Es el responsable de grabar la información derivada de las diferentes herramientas y la relación que existe entre ellos.
- b. Base de datos orientada a objetos que sirve como depósito permanente de la información sobre el programa.
- c. Uno o más compiladores.
- d. Una interfaz de usuario.



Descripción del gráfico

GUI: Interfaz gráfica de Usuario.

Kernell: núcleo del programa.

BDOO: Base de Datos Orientada a objetos.

El kernel interactúa con una BDOO. Como salida obtengo un fuente, al cual lo paso por un compilador y obtengo como salida..

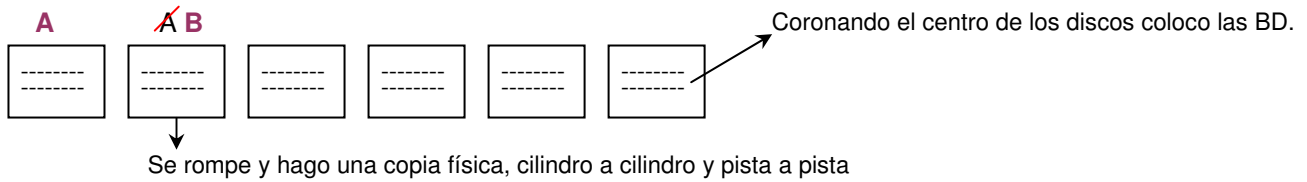
Se puede generar, por ejemplo, Query SQL, con un lenguaje propio de acceso a Oracle, acceder a SQL con Sysbase, se puede sacar JAVA, se puede sacar Cobol, y hay dos tipos de productos que surgen y son interesantes, uno que no te da fuentes, no se ve al fuente, y el argumento es para obligarte a que entres siempre por la herramienta para hacer modificaciones, sino después todas las modificaciones cuando se corre no coinciden con lo que está en la base de datos, entonces se empiezan a hacer modificaciones y salen errores de todo tipo. Hay gente que te dice, yo te doy fuentes por varias razones, primero que vos te puedes ir a cualquier otra máquina, si tenés fuente Cobol te puedes ir a otra máquina, hay otros que dicen yo te dejo un fuente pero está confuso, cosa que lo obligue de alguna manera a seguirlo. Hay otros que te dicen al revés yo te dejo un fuente claro por si vos le querés retocar algo. Da para todo. Algunos te generan algo que corren bajo un runtime propio.

Cuando salieron las famosas herramientas Case, había estadísticas que decían que se reducía en siete veces el tiempo de compilación, que eran absolutamente productivas, el tiempo de poner la aplicación en funcionamiento. Pero no tomaron el tiempo total del ciclo de vida de la aplicación. Cuando yo tengo la aplicación en producción recién tengo invertido el 20% de lo que es el ciclo de vida. En el mejor caso que mejore 5 veces, yo tengo el 20 del 20%, pero me queda el 86% dando vueltas. O sea por eso no se redujo como se pensaba. Y en realidad hay que tener en cuenta, que cuesta mucho tiempo ponerse en régimen para manejar el problema. O sea, digamos, ingenieros de sistemas jóvenes recién recibido, y tiene que estar 6 meses mas o menos para empezar a manejarla, estudiándola. No es una cosa muy fácil de aprender. Le cuesta, y es un desafío, porque si le dicen mira tenés que estudiar esto, si no lo estudias te echan, así que. No son fáciles, ahora cuando están en régimen si son productivas. En general cuando se elige una herramienta para el desarrollo, hay que hacer lo que dice la herramienta, sino hay que cambiarla, porque si a lo que hace la herramienta se lo cambia, hay doble tiempo. Los argentinos fundamentalmente somos muy artesanales, no así los brasileros. Entonces a nosotros no hay pantalla que nos venga bien, hay que cambiar la pantalla, y trabajamos demás. Las rutinas, lo que es el ABM, eso lo hacen bien estas herramientas, lo que no lo tiene hay que programarlo. Hay un software uruguayo, muy bueno, que se llama Genexus y se usa mucho, se vende en EEUU.

Software de utilidad para la gestión operativa de instalación

- 1- Software para la gestión de archivos magnéticos
 - a. Control de archivos externos: CDS, Cartridges. Como ejemplo, en la municipalidad de Sta Fe hay como 20 mil cartridges, que son físicamente iguales. Y si no tenés un software que te diga donde está el cartridge, que tiene adentro el cartridge, y cuando expira el cartridge, no hay forma de llevarlo con fichita. Tengo 20 mil y son todos iguales no hay forma de recordarlos a todos. No solo uno puede entrar por nombre de archivo, y que le diga a dónde está, se puede entrar por usuario, por ejemplo todo lo que es catastro de usuario y le diga que archivos tiene, que archivos van a expirar. Acá los archivos magnéticos, en este tipo de máquina tienen una etiqueta con el nombre del archivo y además tiene la fecha en que se inició el archivo y la fecha en al que expira el archivo (o sea la fecha en al que el archivo puede quedar disponible). Si alguien monta un cartridge para grabación y el archivo no está disponible le avisa al operador, después la decisión que tome el operador es otro tema. Inclusive, el rótulo de la cinta que tiene el nombre del usuario está asociado a un archivo de seguridad para que el que lo abre esté autorizado a hacerlo. Además existen máquinas robots que lo administran automáticamente, buscan y montan el cartridge. Esto es en grandes empresas, en Argentina lo tiene Telecom a este producto. No es un mercado masivo. Lo interesante es que el robot lo administra el sólo, automáticamente. Por ejemplo si alguien quiere correr un programa que utilice dos cartridges, y estos están en otro lugar o edificio, tiene que cancelar ese programa e iniciar otro hasta que tenga los cartridges. Por eso es algo que necesita de buena administración.
 - b. Asignación de espacio en discos magnéticos: o sea si yo tengo un sistema que voy a compartir entre múltiples usuarios, yo voy a tener áreas privadas (base de datos, y áreas de guarda de programas, donde se editan programas) y áreas públicas (la que se necesita momentáneamente, por ejemplo para cargar un fichero).
 - c. Migrar y recuperar ficheros: migrar de un formato de fichero a otro y recuperar información, no sólo hacer backups, sino que también recuperar información que esté defectuosa, hasta donde se pueda recuperar. O sea que estos programas obvian lo que son los controles automáticos de si hay error o no.
 - d. Copias físicas de seguridad: yo tengo un conjunto de discos, y por ejemplo la base de datos que yo mas voy a usar, no me conviene ponerla todo en un disco porque voy a tener un encolamiento muy grande sobre ese disco, entonces me conviene partirla, lógicamente, coronando el centro de los discos que tengo para uso de BD. ¿Por qué coronando los centros de los discos? Porque ahí se queda en situación normal la cabeza lectora/grabadora, entonces el tiempo de búsqueda de pistas se hace mucho más rápido. Lo que pasa es que si se me rompe un disco, yo tengo discontinuada la BD, y si tengo un backup de la BD, tengo que bajar la BD completa. Y si a mi se me rompe ese disco, no voy a tener un disco de backup exclusivo para el sistema, tengo una copia física del disco y lo copio en el disco A. La copia física consiste en copiar cilindro a cilindro, pista a pista. Empieza por el cilindro 0, pista 0, pista 1, pista 2, pista 3 y así hasta llegar a otro cilindro. No me interesa que está copiando, por eso es

copia física. Entonces permite salvar el archivo. Durante la copia si hay usuarios conectados hay que avisarles para que no sigan trabajando momentáneamente.



2- Programa de control de la producción y gestión del trabajo

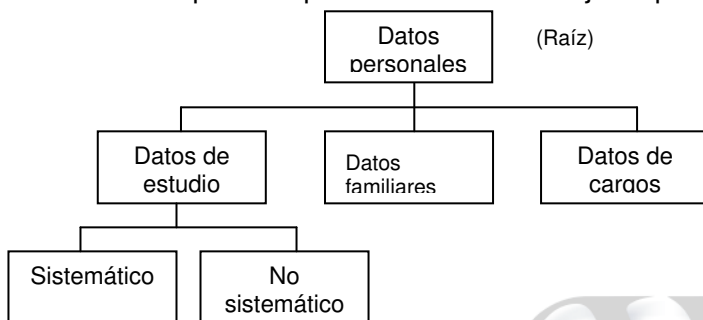
- a. Programas planificadores de la carga de máquina: hay que tener en cuenta que si el tratamiento de ejecución de un programa viene de algún usuario (desarrollador de datos), que no esté precisamente en la misma máquina, y que haya varios usuarios trabajando, no los puedo poner a todos en FIFO, porque si uno quiere programar y hay otro que ejecuta una sentencia de dos horas no va a ser justo. Entonces estos programas me permiten que pueda planificar la carga. Intercepto los comandos de ejecución (API) y digo todo programa que ocupe tanto tiempo de CPU y no ocupa ningún recurso especial, entra a FIFO en esta cola en este momento. El tiempo de CPU se calcula de acuerdo a la potencia de la máquina. Si ocupa más de media hora de CPU, ese programa corre a la noche, salvo que alguna orden contraria lo haga correr a la mañana o a la tarde. Porque a la noche no hay usuarios interactivos, no hay gente consultando BD, es muy baja la que puede haber, entonces está la máquina vacía y todo el trabajo pesado se puede ejecutar durante la noche. Si tengo 4 estaciones de cartridge un programa no puede montar 3 cartridges, que monte no más de dos, tengo que ver si hay cartridges libres, sino no tiene sentido que lo vaya a ejecutar. Además tengo que interceptar el programa ese y decirle que cartridge utiliza para sus procedimientos, para que el programador los tenga al momento de ejecutar su programa, porque sino no hacemos nada con liberar si los cartridges están ocupados.
- b. Programas configuradores: normalmente no se utilizan en las instalaciones, lo utilizan los fabricantes de hardware. El vendedor de marketing es un tipo que tiene la perspectiva de querer ganar siempre. Frente a una licitación o pedido formal de propuesta sabe más o menos como es la competencia, entonces trata de ir con un producto que pueda competir con la competencia, aunque no satisfaga las necesidades del cliente. Después va a buscar en el camino para decirle al cliente porque necesita usarlo. Entonces a las empresas como no les interesa el vendedor, sino salvar su principio de no perder el mercado, entonces tienen programas configuradores, entonces si el vendedor pone en estos programas la información que les piden, y cotiza la máquina que le dice el configurador cumple con las normas de la empresa. Si hay algún problema con el cliente, y el no cumplió con las normas de la empresa, tiene que pasar por caja. Si nadie dice nada se calla la boca.
- c. Programas de ejecución de consolas: interceptan los mensajes de la consola y lo contestan automáticamente. Por ejemplo hay un error de transferencia de archivo, ¿ignoro/cancelo?, cancelo. Después se analizará con el programa de recuperación de archivos que es lo que pasó, donde estuvo el error de transferencia. El archivo no está vencido ¿Qué hago? ¿Ignoro/cancelo? Hay un error recuperable de impresora, me avisa nada más. Todos los warnings que manda los guardas en archivos y después los analizás. Así se alivia al tipo de tareas. Hay programas de control de múltiples servidores, y los manejo a todos desde una sola consola, de ahí viene...entonces puedo tener un servidor para impuestos inmobiliarios, otro para ingresos brutos, otro para sueldos, lo único que hago es administrarlos. No puedo tener una sola consola, por ejemplo destino una para que reciba mensajes del sistema operativo y otra que reciba los warnings.
- d. Programa de control y seguimiento del uso de los recursos de hardware: se conocen como performance tools (herramientas para la performance). Existe un programa residente que hace mediciones a intervalos de tiempo. Un programa que está dentro del sistema operativo y realiza mediciones cada cierto intervalos de tiempo. Mide cada JOP (conjunto de programas) a qué hora ingresó al sistema, a qué hora terminó, qué step tuvo, a qué hora arrancó cada step, cuánto estuvo de tiempo reloj de uso de CPU, cual es la tasa de ocupación de los canales, cual es la tasa de ocupación de cada unidad de disco, cuántas veces leyó físicamente el disco, cuántas páginas de impresión generó, porcentaje de ocupación de CPU. Esta información se puede obtener en estadísticas o monitorearla (tenerla en forma interactiva). Por ejemplo nota

que la máquina está pesada, y el responsable de mejorar la performance se sienta, monitorea y ve, si por ejemplo hay un programa que ocupa mucha CPU y si el programa no hace nada lo baja. Esto permite, por ejemplo, facturar, así sé cuánto consume cada usuario. Cada programa que colocan. La única forma de ser firmes para que la gente sea eficiente es que uno pague. O sea la dirección de informática liquida personal, pero el costo de la dirección de informática es un costo que lo deberían pagar los distintos ministerios para los cuales se brinda el servicio. El que gasta que pague.

Tipo de Base de Datos

- ☆ **Base de datos estructuradas**: que son las BD numéricas, por ejemplo las BD relacionales, las BD jerárquicas, las BD en redes, las BD multidimensionales.
- ☆ **Base de datos semiestructuradas**: como son los GIS (Sistemas de Información Georeferenciales), las BD documentales y las BDOO (BD Orientada a Objetos).
- ☆ **Base de datos no estructuradas**: como son las BD de imágenes.

En orden de aparición primero están las BD jerárquicas.



En éstas la información de se encuentra en forma de árbol invertido. Por ejemplo en el gráfico en la raíz están los datos personales de los empleados, a continuación los datos de estudio, familiares y cargos (que ocupó y/o ocupa). La búsqueda se hace de arriba hacia abajo y de izquierda a derecha.

Lo interesante de las BD, es que en el disco uno define cada registro de BD, cada registro está compuesto por el elemento raíz y los hijos que dependen de él. Los datos se almacenan en intervalos de control (CI). El CI es como si fuera la longitud del registro, donde uno cuando define/organiza la BD le dice que porcentaje de ese CI quiere dejar libre para poner información que pueda aparecer después. Esto es así, porque cuando yo leo el disco, leo una pista completa lo más probable es que los datos vinculados de este segmento va a ir en el mismo CI.

Estas BD se inventaron de esta forma, jerárquica, primero para la industria del armamento, armado de autos, armado de aviones, astilleros, armado de barcos, donde existe el fenómeno para la planificación de la producción, existe el fenómeno de la implosión y la explosión. Esto significa, yo por ejemplo en armadura tengo un subconjunto que estoy armando, y ese subconjunto tiene despiece hasta llegar a la pieza más chiquita, cuando yo voy armando para llegar al motor tuve que pasar por varias partes. Entonces yo quiero hacerme una pregunta como esta: ¿qué productos por el stock que tengo me conviene fabricar? Entonces hago una explosión, tengo que ver todo lo que tengo, y eso lo exploto, lo contemplo, o sea voy a revisar cada modelo de cosas que armo y veo cual me conviene más. Otra pregunta, ¿cuántos autos de tal modelo puedo hacer? Entonces el tipo revisa y me dice todos estos puede hacer. ¿Qué necesitaría comprar? Y necesitaría comprar todo este material y cuesta tanto. En la industria del armado no se trabaja con stock cero, sino no llega nunca. No le puedo decir al fabricante de autopartes, guardame todas las partes y traémelas mañana corriendo. Por eso las BD fueron jerárquicas y ¿por qué se usaron mucho en administración? Se usaron mucho en administración, porque todo lo que es transaccional, donde uno dialoga con un cliente, las BD jerárquicas tenían mucha performance, sobre todo cuando las máquinas eran más lentas. Antes del fenómeno d Internet la BD que más se usaba en EEUU en los bancos era IMS que era una BD jerárquica.

Después cuando cambió la tecnología, que se puede tener clientes flacos, se empezó a tornar.

IBM en la época del '70, era el Bill Gates de los '90, o sea que imponía las condiciones, imponía los estándares. Eso generaba malestar, como pasa ahora, en los ambientes académicos contra aquél que quiere imponer cosas. Entonces se juntó la CODASIL, que es una confederación de usuarios de EEUU, y normalizó sobre BD que no sean propietarias e hizo el modelo de red de BD.

Modelo de Red: los datos están agrupados por tipo de datos, cada dato es un nodo, no están jerárquicos están en una red, todos iguales y se generan los caminos entre ellos y los puntos de entrada (entry-point). En este caso si el programador dice next, next, next, le va trayendo el siguiente registro

asociado con esta raíz, de arriba abajo y de izquierda a derecha. O sea que está el camino hecho, planificado. Y el programador no navega. Esto fue alrededor del año 75. En el 78 CODA, empezó a tirar entre dos tipos de BD y dijo ¿cuál es el problema que tiene esta BD? Esta BD tiene el problema de que las relaciones ya están establecidas. Y cada vez que uno quiere cambiar las relaciones tiene que entrar a todos el programa.

Ejemplo: un director de informática de la municipalidad. Como a una BD jerárquica no se le puede crear un índice secundario, quería crearle un índice secundario a un segmento de la BD de catastro. El segmento era tomo, folio y número de registro propiedad (índice secundario). Pero esto llevaba mucho tiempo ya que había que parar el uso de la BD para poder generarlo, y no era viable de hacerlo en un día porque era necesario que al otro funcionase sí o sí. Se sugirió hacerlo en la semana larga de Pascua, pero el problema era si dejaba de funcionar y el lunes no arrancaba. Es por eso que se dice que “os problemas de administración se hacen serios en grandes volúmenes de información”.

CODA creó las formas normales, la forma normal es la forma simple de canalizar la información, porque a partir de información normalizada las demás vos la podés crear. No hay grupos repetitivos, no hay dependencia, no hay grupo multivalor, no hay redundancia, es la forma más simple de acceder a la información. Entonces no hay riesgo.

Lo que hay que tener en cuenta es que la relación la hacen en memoria real. Y después swapea. Si no tenés índice para acceder a la tabla, entonces carga la tabla completa en memoria y la busca. Y lo que no entra en memoria lo carga al disco, entonces tiene que swapear, o con memoria virtual depende de cómo administre la memoria.

La tabla relacional sirve para todo lo que es proceso transaccional, no proceso analítico. Sirve para todo lo que está orientado a la persona.

Cuando aparecieron los generadores de software para la PC, los tipos tenían como tecnología de última el Proyecto R, proyecto de BD relacional.

Después aparecieron las BD multidimensionales. Recordando primer clase, la información operativa debe elevarse a la gestión, y debe de ser la misma información la que produzca los datos corporativos de los datos transaccionales. William Aymon se dio cuenta que en las reuniones de gerencia en IBM EEUU, cada uno iba con su laptop, para llevar su información. Pero no todos tenían la misma información. Decían que tenían información de bases diferentes a la que tenía otro, ya sea porque no se hacían los cálculos de la misma forma, o porque se habían olvidado algunos datos, o porque veían la misma cosa de forma diferente, tenían diferentes conceptos.

Entonces William le preguntó a los gerentes, por que estás teniendo información vos y no se la pedís a cómputos a eso. A lo que respondió, “y mirá primero nunca tienen tiempo, segundo me dicen bueno te lo vamos a hacer pero para mañana y yo lo necesito ahora. Cuando me lo traen al otro día (si me lo traen) le pido algo más y me dicen por que no me lo dijiste ayer”. Lo que pasa es que el análisis es una tarea escalonada. No es lo mismo que información operativa, vos tenés que saber cuanto te pagaron y que vuelto darles. Los gerentes (de grandes empresas) se cansaron de hacer de intermediario con humanos de menor jerarquía para tener la información. Es por esa razón que trabajan con múltiples alternativas. Entonces Aymon siguió la línea, el esquema relacional.

La normalización es buena pero hay que controlarla, hay que saber normalizar. La información que están teniendo los centros de cómputos de los sistemas tradicionales está muy atomizada. Entonces el tipo saca la Data Work Housing, casa de datos, sería como el supermercado. Antes existía el mercadito, la gente iba y el señor vendedor era quien le proveía los alimentos y demás. La mercadería no se podía quedar y los compradores no se podían quedar dando vueltas ahí dentro si no compraban. Hoy en día ocurre algo casi contrario en los supermercados. La gente va y se sirve sola, puede estar el tiempo que quiera sin que lo echen, lo único que tiene que hacer es pagar al salir y listo. Se tuvo que cambiar la mentalidad de compra.

La idea es tener la información de gestión elaborada, cosa que el usuario lo único que tenga que hacer es ver lo que hay y servirse.

CLASE TEÓRICA DEL 17/09/01

Elección de software de comunicaciones

Una cosa que tal vez es interesante no para rendir sino para conocer: cómo se fue dando la simbiosis informática en comunicaciones; es decir, en la década del 80' no existía sistema comunicaciones, inclusive en materias que yo daba y algún otro profesor que hablaba de sistemas

operativos, enchufábamos el tema comunicaciones en el medio; hoy todo lo que sabemos de comunicaciones es poco y se requiere inclusive una especialización. Entonces, lo que nosotros tenemos en nuestro contexto como ingeniero en aplicaciones digamos, es tener idea de lo que se le puede pedir a las comunicaciones, desde el punto de vista de las aplicaciones, más allá que algunos aspectos ustedes lo tienen que haber profundizado en otras materias. Pero digamos, para que tengan una idea, en la carrera de ingeniería en comunicaciones en Cambridge, de cables tiene 2 materias cuatrimestrales de 8 horas diarias, cable I y cable II; o sea, es muy grande el tema comunicación, pero tenemos que convivir con ello. Mucha gente dice: "...y es todo lo mismo, ejecutar comunicaciones con informática" es falso eso, la prueba está que telefónica puso un autártico de IBM para hacer informática, pero son expertos en comunicaciones no en informática. No quiere decir que los que están adentro no conozcan informática, pero el negocio es diferente, las prestaciones son diferentes, los clientes son diferentes.

Cuando nosotros hablamos de los sistemas teleinformáticos, que hablábamos de los problemas teleinformáticos, habíamos dicho que un sistema teleinformático es cuando uno puede discutir tal cosa por medio de las comunicaciones, o sea, que hay un proceso remoto donde uno introduce datos y recibe respuesta por medio de las comunicaciones, cualquiera sea el canal de comunicación. Una cosa que tengan bien en claro, no confundan teleinformática con interactividad. La teleinformática es cuando yo estoy trabajando con cables de comunicaciones externos, adonde estoy hablando; en cambio lo otro, puedo tener una lan, una red local, o sea, estoy hablando de red de banda ancha o red ciudadana o red mundial. Los sistemas son diferentes, en los dos son sistemas interactivos pero las cosas que hay en el medio son diferentes.

Yo les quiero hablar un poquito de cómo evolucionó este concepto de comunicaciones, y tal vez, convenga que les dé un ejemplo: Tango se ve obligado (hace muchos años), a tener estados consolidados de todo el banco, independientemente de la sucursal en la que está. No sé si me entienden, o sea, no solamente tengo que saber la cartera de plazos fijos que tengo en Paraná para el Banco Nación, sino la cartera de plazos fijos a nivel país, a nivel mundial inclusive. O sea, que el sistema está obligado en algún lugar a concentrarse, por más que yo haga cosas locales. Las computadoras eran muy caras, inclusive las primeras PC eran muy pequeñas en cuanto a capacidad, no tenían disco rígido, costaban 10.000 dólares. Entonces, el gran negocio que se presentó en ese momento, y yo diría que duró hasta el 84'-85', eran empresas que hacían service, ¿qué significaba un service?: "venga y pase". No conozco bien las rutas de Paraná, pero sí las de Santa Fe, OCA (empresa de transporte de documento) salía con listados por distintas rutas: una iba a la ruta 11, otra agarraba la ruta que va a Santiago del estero, otra agarraba la ruta que va a Córdoba, otra agarraba la ruta que va a Rosario e iba repartiendo pedido de resumen de cuenta, listados de saldos, pedidos especiales de algunos totales contables, la contabilidad, es decir, todos los papeles que necesitaba el banco. Y algunos bancos arrancaban, le llegaban a las 6 de la mañana y otros le llegaban a las 8:30, depende el recorrido y la distancia de la base. ¿Qué hacía el banco?: juntaba todos los cheques, con tira de suma que hacía el cajero y todos los depósitos y los pedidos especiales de resúmenes, etc. Además, con esa pericia de caja, cuando uno iba al mostrador a cobrar un cheque, entonces, tenían saldo de cuenta 300 menos 100, restaban y ponían -100 y dejaban 200, firmaban y sellaban y podían pasar a ventanilla a cobrar. Era un traslado de papeles espantoso y peligroso, porque iba el documento original. Entonces, la camioneta llegaba y tenía que empezar a dar la vuelta; se quedaba en el lugar de destino, comía algo, espera y volvía. Volvía hasta el centro de servicio, había que procesar, y a las 3-4 de la mañana arrancaban de vuelta.

¿Cuál fue el gran adelanto tecnológico?: la conexión por comunicaciones que se llamó **Modem to Modem by Transfer**, la transferencia de archivos de modem a modem, o sea, si yo ponía una máquina del otro lado donde partía con los saldos, le grababan al tipo los cheques y los depósitos y ya tenía el saldo actualizado, era un saldo operativo no era un saldo contable; les voy a decir por qué, porque ese saldo después puede mercer algún interés por giro en descubierto, algún débito, etc., era un saldo operable no era un saldo contable, el riesgo que tenían era pagar 10 pesos de más nada más. Entonces, al fin del día ya tenían los movimientos ahí, se llamaban por teléfono a un horario que se pactaba con el centro de cómputos, entonces decían vamos a transferir archivos, bueno, pum! Transferidos, una vez transferidos los archivos se corría un programa de control para verificar que los totales coincidían, ¿por qué?: porque era un compromiso de modem a modem no era un compromiso entre dos aplicaciones. O sea, él me mandaba un archivo en un lugar que yo le decía del disco, si llegaba todo lo que yo quería era problema tuyo controlarlo. O sea, todos conocen esto de Modem to Modem by Transfer, pero desde el punto de vista informático, esto fue un gran avance en este tipo de sistemas, ya no se movían papeles.

Una cosa que se descubrió inmediatamente atrás de Modem to Modem by Transfer fue lo que se llamó **BBS** o Boletín sobre board, boletín sobre cuerpo. Este era un método de comunicación es *eucogra*, ¿qué significa *eucogra*?: que se conectaban dos tipos y dialogaban entre ellos, es una especie cuando vos chateás con uno en Internet, eso se podía hacer. Lo que pasa es que no era tan fácil como con Internet, vos tenías que conocer bien el protocolo del otro, porque en el BBS del comienzo no había una interface común, un método único, cada tipo tenía el suyo. Entonces, lo que había por ahí, había tipos que te hacían de puente porque vos no sabías cómo dialogaba el otro, hacían de pequeña central telefónica digamos.

Posteriormente, aparece el concepto de **e-mail server** o servicio de mail. El servicio de mail trajo varias ventajas sobre lo que era BBS. Primero que trabaja *eucogra*, ¿qué significa *eucogra*?: que vos podés trabajar y mandar el mensaje; no es como el teléfono que está ocupado en el BBS. Segundo que vos podés conectarte para ver si tenés mensajes pendientes cuando vos querés; eso las prestaciones del outlook las conocen todos, pero hay servicios de e-mail como es el lotusnote que son mucho más poderosos que lo que es un simple correo electrónico que todos usamos. Por ejemplo: los correos que están usando a nivel hogareño, no corren bajo Unix y yo puedo tener toda mi plataforma bajo Unix. Algunas ventajas de estos servicios más poderosos son:

- con los hogareños, yo no me entero si el correo fue recibido o no por el otro, yo lo envío y después si quiero asegurarme tengo que hablarle por teléfono o le tenés que mandar otro e-mail “contestáme por favor”. Este tipo de productos permite saber si el correo fue abierto, puede haberlo abierto un agente de él, un secretario digamos, pero lo abrió, ya es responsabilidad del otro.
- Puede ser que yo mande copias invisibles, o sea que yo le diga: “mandame a mi secretaria tal, tal y tal cosa” y no es visible, ella no sabe lo que es.
- Realizar envíos a fecha futura del correo por ejemplo, o sea, con los cumpleaños, que lo saluden 10 días antes a nadie le gusta. Pero también puede ser importante porque puede querer hacer tener un impacto a nivel global.
- Que me notifique cuando el correo no es leído, o sea, si pasaron tantas horas y no lo abrió, no lo leyó al archivo, y bueno, que me notifique.
- Que me abra un nuevo evento para decirme hay correo nuevo sin leer, o sea, cosa que yo no tenga que ir a ver siempre a la puerta a ver si me dejaron correo abajo.
- Bueno, encriptación, lo tienen prácticamente todos ahora.
- Control de contraseña, correo y lugar también se puede poner fácilmente.
- Recursos de precedencia del correo, esto también es algo muy interesante. Yo quiero que el correo le llegue a él, después a él y después a él. El control de precedencia es importante porque uno le puede hacer lo siguiente: una vez que el proyecto lo recibió él, cuando él lo libera, le puede hacer comentarios al margen pero lo retira él. Y él cuando lo vea lo retira él, porque a su vez yo quiero que primero lo lea la parte general, la técnica y la jurídica antes de que me llegue la contestación; para no enviar en paralelo y que lleguen 3 cosas distintas sino que jurídico se entere después lo que opinó un técnico, entonces puedo opinar si esto es válido.

Bueno, ¿por qué les digo todo esto? Porque a mí no me interesa enseñarle la importancia del correo como comunicación entre particulares, porque eso lo conocen mejor que yo, sino la importancia del correo en la gestión de la organización, ¿qué significa esto? Nosotros en las organizaciones teníamos dos formas clásicas de comunicarnos:

- 1) *Tele a tele*, o sea, hablábamos por teléfono o golpeábamos la puerta, permiso y entrábamos a hablar.
- 2) Mandando una nota, ya sea un memorandum o algo el estilo, dejarlo por un medio escrito.

La nota suena formal, y la otra tiene un problema, que si bien vos tenés un problema y querés plantearlo, a la persona se lo llevás puede estar con otro problema. Entonces, como vos le estás pidiendo una respuesta on-line digamos, interactiva, mucha gente tiene que contestarte algo y a lo mejor no te contesta lo mejor, y además el estado se siente molesto porque le dice: “disculpe que lo moleste” porque vos no sabés lo que estaba haciendo, ya lo molestaste y no queda registrado después todo este tipo de cosas informales. Entonces, queda registrado las notas y que vos me dijiste yo te dije. Entonces, un día el presidente de IBM Argentina (el anterior al que estuvo en el conflicto nación), se dio cuenta de esto y dijo: “a partir de mañana yo no recibo a nadie, quiero recibir todo por correo electrónico”, el correo electrónico era lo que usaba IBM para comunicarse empleados entre sí, para cosas no oficiales, pero no era un medio de gestión. A partir de ese momento, como el presidente no

recibía a nadie, todos lo gerentes de primera, segunda, tercera, cuarta sucursal, todos tenían correo. ¿Cuál es la ventaja? Que vos contestás cuando vos querés, o sea que vos podés pensar la respuesta, podés contestar con un registro, la persona que envió el correo ya sabe que lo mandó, se entera cuando lo abrió y cambian las relaciones de información dentro de las empresas porque vos tenés todo lo que te enviaron, todo lo que vos contestaste.

Moraleja: por ejemplo un gerente de IBM que tiene que hablar con todo el mundo, se la pasaba: la primera hora de la mañana leyendo el correo que le mandaban durante el día y a la tarde las últimas horas se la pasaba contestando, pero no hablaba por teléfono; entonces no estaba que lo llamaban a cada rato, él sabía que tenía las horas del medio eran para dedicarse a atender los negocios en su zona y atender a sus empleados si necesitaban alguna cosa en particular. Bueno, es una herramienta muy buena el correo desde ese punto de vista, para organizaciones, fundamentalmente que sean medianas o grandes, eso sí, la única forma de imponer el uso del correo es que se ponga de arriba para abajo.

Otra forma de conexión es **emulación de terminal**, ¿qué significa emulación de terminal? Que yo en una terminal remota, emulo ser otra terminal de otro equipo remoto. O sea, como ven, cada vez se van integrando más los sistemas de comunicaciones con las aplicaciones.

Otra forma de conexión es **control remoto del software**, o sea, que yo puedo lanzar Open de mi sistema desde otra máquina, o sea, no necesito tener un humano en los lugares donde yo voy a prender el equipo. Prendo el equipo en forma remota, controlo el equipo de forma remota. Piensen los costos, si cada sucursal bancaria tiene que tener 3 o 4 operadores que conozcan productos, SO, etc., los costos son espantosos, pero con este tipo de productos entra en una competencia mucho más agresiva que el que no lo hace. Además es mucho más bueno para determinación de problemas, porque vos te haces dueño de la máquina desde un lugar remoto sino tenés que ir al territorio.

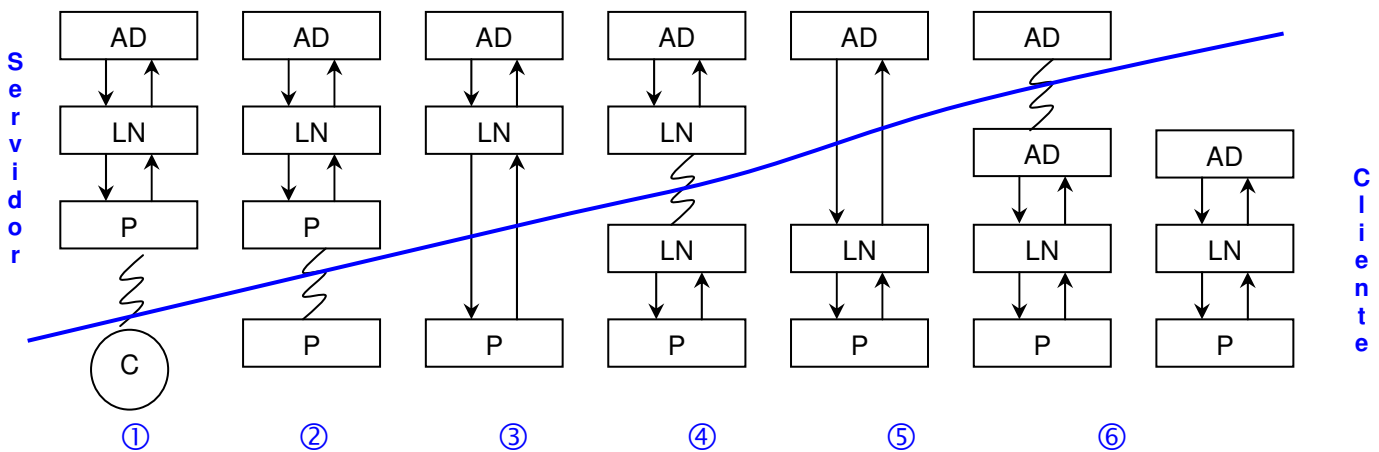
La otra forma de conexión ya más conocida por ustedes, es el modelo que se denomina **modelo cliente/servidor**, esto quiero detenerme un poquito más. Todos hablamos de modelo cliente/servidor pero creería que estamos hablando de un modelo de compromiso, ¿qué significa de compromiso? Que no es un modelo puro. También se denomina modelo de dos capas, ¿qué significa de dos capas? Que hay algo que se hace en el cliente y hay algo que se hace en el servidor. Ahora, qué se hace en el cliente y qué se hace en el servidor, y bueno, depende de qué se haga en el cliente y que en el servidor, van a ser distintos tipos de modelos. Entonces, el Data Group hizo el siguiente grafiquito:

Gráfico del Data Group

AD: Acceso a Datos

LN: Lógica del Negocio

P: Presentación



- ① Estructura mainframe (terminales Bobas)
- ② Presentación Distribuída
- ③ Presentación Remota
- ④ Modelo de función distribuída o lógica distribuída
- ⑤ Administración de datos remota
- ⑥ Modelo de Base de Datos Distribuído
- ⑦ Modelo de PC

- 2) Si el acceso a datos se hace en forma remota, la lógica del negocio se hace en forma remota, la presentación se hace en forma remota y la presentación se hace en forma local (de la línea azul hacia abajo es cliente, la línea azul es una línea de comunicación imaginaria), estamos en un modelo de presentación distribuída porque solamente distribuyo la presentación, hay una parte en el servidor y una parte en el cliente, el resto está todo en el servidor.
- 3) Si yo tengo administración de datos en el servidor, lógica del negocio en el servidor, y presentación en el cliente, lo que estoy teniendo es un modelo de presentación remota.
- 4) Si yo tengo administración de datos en el servidor, lógica del negocio en el servidor, lógica del negocio en el cliente y presentación en el cliente, yo tengo un modelo de *unshock* distribuída o de lógica distribuída.
- 5) Si yo tengo administración de datos en el servidor, lógica del negocio en el cliente y presentación en el cliente, yo tengo un concepto de administración de datos remota.
- 6) Si yo tengo administración de datos en servidor, administración de datos en el cliente, lógica del negocio en el cliente y presentación en el cliente, yo tengo un modelo de base de datos distribuída.

Como ustedes pueden ver, no hay un único modelo que me represente la distribución. Para hacer algunos comentarios: el consorcio Gui-Intel se unieron rápidamente a este esquema desde el comienzo. En el comienzo tenían clientes gordos, servidores flacos, eso le convenía a Intel porque Intel tenía el mercado cliente fundamentalmente ganado y le convenía a Microsoft porque vendía mucho software de ellos. Ustedes por las dudas monten Windows XP que son versiones que corren por la izquierda digamos, pero que son versiones definitivas, comerciales y la máquina que tengan le va a quedar lenta, la máquina que tengan van a notar que les va a quedar mucho más lenta.

Este modelo de cliente gordo, por ejemplo Visual Basic, VB tiene todas las dlls, tiene todos sus programas, toda la lógica la conocen y en el servidor tendrán HSQL Server.

Este modelo, es económico, es potable en organizaciones no tan informales, o sea cuando yo tengo todo acá, porque tienen dos problemas graves: en primer lugar, la distribución del soft, si yo hago un cambio porque estoy haciendo algo mal ¿cómo hago para distribuirlo a toda la provincia? Es un despiole. El segundo problema que tengo es la seguridad ¿cómo sé yo que están ejecutando el programa que yo quiero?.

...no es un misterio, entonces el tipo ve cómo es el comportamiento del programa, y puede ver donde hay una constante de interés o una tabla local y la cambia para un caso, y después la vuelve a cambiar de vuelta. Es decir, no da la seguridad, por eso señores, en los bancos nunca puede haber este tipo de modelo cliente/servidor. Por esa razón se usaba la tecnología Mainframe, no porque le gustara a IBM o Microsoft, era seguridad, nada más que eso: seguridad y administración.

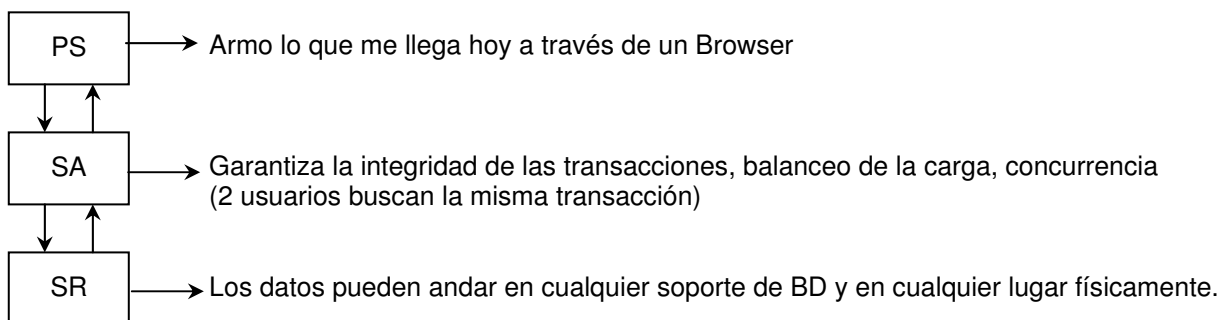
Yo digo que la revolución ésta se dio en el 95', ¿por qué? Porque se dieron algunos fenómenos:

- 1) La desregulación de las comunicaciones en EEUU. Trajo como consecuencia una desregulación también en el mundo. Esto ¿qué significa? Que cualquiera que tenga un ducto, por ese ducto puede pasar ondas, puede comercializarse. Entonces, yo puedo pasar información por las redes telefónicas, lo podés pasar por el caño de agua incluso si se comprometen en que no va a alterar la pureza del agua. Esos son unas cosas muy importantes porque ya el ancho de banda no era controlado por dos o tres en el mundo sino que podía quedar libre a cualquiera. Entonces los anchos de banda se hicieron cada vez más baratos, a tal punto que nosotros (estas son cifras reales), cada dos años cuantificamos los anchos de banda y bajamos los precios totales de comunicaciones y agrandamos los puntos de comunicaciones. Hace 3 años, 256 KB con Rosario costaba alrededor de \$12000 por mes, hoy está costando \$10000 los 512 KB, o sea, cambia totalmente la historia. Entonces las aplicaciones, ya no es tan importante adónde están las máquinas.
- 2) El cambio en lo que fue de la red mic a la www. El hecho de crear una imagen en el explorador amigable, una interface amigable para todos los productos, provocó una estandarización de facto de los buscadores. O sea, no existe persona de cierta cultura dignamos, que no sepa Internet. Esto ¿qué hizo? Que yo estandarice la interfaz de usuario. O sea, no tengo que hacer una interface para cada aplicación, todo el mundo que tenga una PC, le pone un buscador y ya tiene una computadora para Internet.

- 3) Este intento llamado Java o donde permite que yo programe en un lugar y ejecute en cualquier lugar. ¿Qué significa esto? Que si yo tengo una transacción muy pesada, se la mando a ejecutar al jefe, que espere él con su máquina. O sea, ahora tengo todas las ventajas que me daba el mainframe, más todas las ventajas que me da la tecnología visual.

Entonces, se armó una guerra entre cliente delgado y cliente gordo. En general la guerra la está ganando el cliente delgado en el sentido que hubo que sacar un montón de productos para que un par de problemas queden minimizados, para simular que se está trabajando con tecnología web pero no es tecnología web. En este momento, las tecnologías que utilizan estas aplicaciones son: los que hicieron adaptaciones en su viejo método de dos capas simulando tres capas; y los que ya están haciendo algo nuevo en tres capas.

El último modelo que tenemos es lo que se denomina OLTP o Proceso Transaccional On-Line o de 3 capas, que va por la cuarta SOLAP (Proceso Analítico On-Line). En el modelo OLTP existen: un software de presentación (PS); un servidor de aplicación (SA) y un servidor de recursos, datos (SR).



Otra cosa que es interesante comentarles, es que hay distintos servidores de aplicaciones: hay algunos que son simples lenguajes de programación como el ServerSQL, Oracle, Visual Java; hay otros que son productos que te permiten trabajar además de que es un pseudolenguaje, te permiten trabajar distintos tipos de administradores de base de datos; entonces, por ejemplo: a un motor de Base de Datos como es el **GMS**, una base de datos jerárquica que se usa muchísimo, es grande, se interrumpe la pantalla, simula en qué pantalla está acá y te deja el archivo abierto, entonces vos puedes estar trabajando con archivos de base de datos viejas y nuevas, entonces te permite migrar. Es decir, te permiten hacer desarrollos rápidos de aplicaciones y con diversas herramientas.

Un tema a tener en cuenta: ¿qué es la transacción? una transacción es una relación entre el mundo real y la base de datos. ¿Qué atributos debe reunir una Base de Datos?

1. **Atomicidad**: los cambios realizados por una transacción a la base de datos, deberán poder ser abortados y las modificaciones desechas.
2. **Consistencia**: si una transacción es buena, todas sus modificaciones deben tener lugar, si una falla se debe poder volver para atrás, que no es lo mismo que lo anterior. ¿Qué es lo que estoy queriendo decir con esto? Si yo tengo que grabar 7 bases de datos distintas y en una no puedo grabar porque está desconectado el equipo y ya lógicamente actualicé las 6 anteriores, tengo que volver todo para atrás. No es que lo aborté, vino una interrupción externa.
3. **Aislamiento**: cada transacción debe ser independiente de las demás, no debe esperar datos de otra transacción.
4. **Durabilidad**: una vez que se completó la transacción, debe perdurar aunque se hallan presentado fallas antes o después de ésta.

Bueno, acá hay una serie de protocolos que garantizan la conversación, el protocolo XA, pero no tiene sentido entrar en detalle en eso. Tenemos ODBC como base de datos para hacer Open. No sé si ustedes conocen **UPCOMIT** (*operador* de fase de compromiso), esto es lo que hace que se pueda cumplir con la característica de la transacción. Nosotros en realidad, cuando escribimos un programa cumplimos con una lógica secuencial, por ejemplo ¿cómo garantizo la consistencia? Entonces se hace una doble fase de compromiso. La primera fase de compromiso, “que averigüe si puede cumplir con todas esas entradas/salidas de datos y todos sus motores; entonces el tipo, le averigua y le dice “sí señor, podemos” y el tipo dice “hágalo” y cuando el tipo le dice “sí señor lo hice” es donde está el segundo compromiso. Esto significa que el servidor de aplicaciones lo abre, y lo ejecuta de una manera totalmente distinta a como vos lo escribiste. No es como un programita en Cobol o en Visual Basic que

yo sé que acá grabe y acá grabé, ¡minga grabaste! Todo junto, hiciste todo junto, no hay nadie que te controle si grabaste o no lo grabaste; o sea, es un analizador de programas en realidad, ejecuta otro código distinto del que vos escribiste pero con la lógica que vos escribiste. Por eso, cuando se habla de base de datos, hay que saber a qué *tipo* se está refiriendo, no es una base de datos fija.

¿Cuáles son los puntos clave en performance de este modelo ODTP?

- El control de prioridad: lo hace el Server. Esto es muy importante porque yo puedo tener cosas que son prioritarias a ejecutar en distintos horarios y otras que no son prioritarias.
- Control de proceso: por lo que decíamos antes, esto de concurrencia, o sea, para que no me cree un proceso por cada cosa, que me controle la concurrencia.
- Integridad de datos

Uno de los problemas que se presentan cuando hay muchas transacciones y muchos usuarios, es que necesitas un monitor de teleconsenso pero no te alcanza con los elementos de control que te da la propia base de datos o el propio sistema operativo.

Los plazos de comunicaciones no hablamos porque ustedes ya más o menos conocen. Pero sobre el software de comunicaciones, habíamos dicho que teníamos software de base (S.O.) y el software de comunicaciones puede estar dentro del S.O. o fuera. Internet obligó a poner TCP/IP adentro de los S.O., de todos los S.O. Lo primero que hay que ver, si lo que nosotros tenemos como prestaciones del software de comunicaciones en el S.O. es suficiente o no. Esto pasa también en otros aspectos, por ejemplo: a vos el S.O. te ofrece seguridad, ¿pero es suficiente?: hay muchos productos de seguridad que convergen con la seguridad nativa. Eso hay que tenerlo en cuenta, si con las prestaciones que te da el S.O. a vos te alcanza o hay que ver otro tipo de software.

Hay dos tipos de software de sistema de comunicaciones:

- 1) programas de control de comunicaciones: estos residen en el procesador de comunicaciones. Yo puedo tener un *promen* de comunicaciones, o sea, un procesador está *atachado* a mi máquina, a mi CPU, a mi servidor por un cable y hace toda la gestión de comunicaciones. ¿Qué significa esto?:
 - hace el control de la red;
 - gestiona la transmisión, o sea, les pone los caracteres de control;
 - verifica que la trama de bits fraccionada vaya hasta los otros puntos y venga, haya sido recibida;
 - maneja lo que se llama "usuarios físicos con los usuarios lógicos", tienen tablas los usuarios físicos con los usuarios lógicos. El programa de aplicación atiende un usuario lógico pero después yo tengo que mandárselo a un usuario físico, tengo que pasar por este puerto y llegar hasta la persona que está trabajando en concepción del Uruguay, en la oficina de rentas, en el segundo piso que se llama Ana, ahí tengo que llegar. En mi programa de aplicación yo tengo un usuario lógico, ese tipo se llama Juan Pérez, donde está físicamente puede cambiar inclusive pero se maneja a través de tablas. El que hace la gestión para mandar el mensaje al usuario físico es el monitor, es el procesador de comunicaciones que provee los métodos de acceso, ¿qué es lo que hace?. Que si yo tengo un procesador de comunicaciones libero de esa tarea al S.O., porque el S.O. tiene que atender otras cosas para adentro también, es decir, todas las transacciones que estoy procesando.

2) programas de aplicación: son los programas que gestionan a las transacciones.

Hay dos tipos de monitor de teleproceso: el que trabaja Per to Per, o sea, que vos lo podés interrumpir y el tipo puede ver quien interrumpe, o sea, vos usuario lo podés interrumpir. O el que trabaja, que es más seguro (esta es otra de las razones por la cual se sigue trabajando con Mainframes), que polea, o sea, yo veo si vos me estás llamando, yo no veo quien me llama pero lo puedo llamar, o sea, yo soy el que llama no vos. Poleo a ver quien me está llamando, entonces yo puedo identificar a ver quien me está llamando a ver si es mío o no; en per to per eso no se ve, vos recibiste el correo electrónico, y si depende de qué tipo de correo electrónico recibiste podés tener el virus en la primer página, con el XML dinámico eso es posible. Entonces, vos estás asegurando que yo te llamo a vos cuando yo quiero te voy a contestar, selecciono mayor velocidad de transmisión cuando veo que el canal puede trabajar a más velocidad, si ve que está teniendo errores a esa velocidad, la bajo; o sea, está continuamente monitoreando lo que está pasando en el teleproceso.

Y después tenemos monitores de teleproceso que residen en la CPU. En una PC, para darle una comparación, el método de acceso se lo da la plaqueta de comunicaciones. El S.O. le dice al monitor de teleproceso: "resolveme esta transacción", te la pasa y el tipo habla con el procesador de comunicaciones; cuando recibe la respuesta la recibe él y la pasa directamente al programa en cuestión, no pasa por el S.O.; o sea, libera todo lo que es el servicio de comunicaciones al S.O. y lo deja que haga simplemente el servicio de datos y comunicaciones con el motor de base de datos, etc.

Comunicaciones es un problema cuando comenzás a tener más de 10 terminales, cuando empezás a tener 200.000 transacciones por día digamos en 8 hs. de trabajo, empezás a tener problemas de este tipo.

Los módulos que tiene el monitor de teleproceso hacen:

- la gestión de tareas, o sea, le dice al S.O. que recursos necesita.
- gestión de la memoria: si vos le asignas memoria real y virtual y el tipo la administra como mejor le parezca.
- Control de mensajes, o sea que pasa si no me están llegando los mensajes a tiempo.
- Es el cargador de programas de las transacciones, tiene todas las tablas de sistema, temporales, de programas, de archivo, etc. Y lo más importante es que permite tener estadísticas en tiempo diferido de todo lo que pasó en el día digamos, o un monitoreo on-line de todo lo que está pasando en cada transacción, por ejemplo: cuál es el tiempo promedio, esta transacción desde cuándo entró, si entró en algún otro lado, que cantidad de caracteres estoy transmitiendo por horario, etc. Nosotros en Sta. Fe tenemos 2 procesadores de comunicaciones pero por el tema backups, cada uno admite 212 líneas, pero cada línea no es una terminal, cada línea es un clúster y cada clúster admite varios terminales.

Selección de sistemas llave en mano

Sistemas llave en mano digamos, es una sucesión intermedia entre administrar uno la aplicación y la compra del software, es comprar software aplicativo por un lado y software de base por otro.

Es una buena solución para aquel que no quiere tener riesgos, riesgo de tener múltiples proveedores por ejemplo, cuando falla uno me cae todo el negocio. Es más barato cuando uno empieza a contar todas las cosas sueltas, pero a la larga es un producto más caro. ¿Qué ventajas tiene el que lo compra?: que minimiza riesgos, si no me das esto andando y funcionando no lo pago. Otra ventaja que minimiza es el retraso tecnológico, si yo compró hardware y me falla porque se demora como siempre el software aplicativo, cuando lo quiera usar puede que ese hardware ya esté obsoleto, o sea, hubiese sido mucho más barato si lo compraba hoy.

Hay que tener en cuenta algo muy importante, a los proveedores les interesa mucho vender sistemas llave en mano, porque yo prácticamente te conozco el negocio, no para ir a contárselo a otro, sino que a partir de ahí yo tengo un cliente que lo gano por lo menos si yo me porto bien, para toda la vida. ¿Porqué?: porque si yo voy a hacer un cambio mañana, va a estar en mejores condiciones quién me lo hizo, o hacer una reingeniería. Si yo le compré el hardware a esa persona y quiero comprar una ampliación, se la voy a comprar a él, él me va a conseguir mejor precio inclusive, ¿porqué?: porque no tiene que hacer mucho gasto de venta porque ya nos conocemos.

El tema es que cuando uno le define un sistema llave en mano, el problema es describir el alcance desde el comienzo. Si cuando yo estoy trabajando con programadores en la provincia por ejemplo y viene una empresa de corrientes, me voy hago los requerimientos con el usuario, bueno, hacelo así, después lo hago, lo programo, lo pruebo, hago la prueba de sistema, estamos todos de acuerdo, se lo muestro al usuario. Entonces, me dice el usuario: "no che, pero esto no, me entendiste mal, disculpame, ¿porqué no me lo haces de esta manera?" y el usuario te cambia la cosa. Entonces, vos volves con la carpeta a la empresa y decís: "no pero el usuario no entendió un pepino, me cambió todo, ¿yo qué hago?", y bueno "a ver qué te pidió. Y bueno vamos a hacerselo". Pero ese no es el tema, sabés cuál es el tema, que cobran los dos a fin de mes el sueldo el que tuvo que hacerlo de vuelta y el que pidió que hagan el cambio. Y vos me decís yo soy una empresa y vos me encargás que te haga el modelo siena tal y cuando yo te lo llevo me decís no quiero un gol ahora, esperá un cachito, vos me pediste un siena, si ahora querés un gol pagámelo yo tengo que darle de comer a mi gente, esa es la diferencia. Entonces hay que tener muy claro los requerimientos, que nos cambia totalmente el paradigma sobre la descripción del profesional, no solamente a los informáticos, sino también a la sadia que tiene que decir qué quiere. Entonces, es muy importante determinar:

- los objetivos a alcanzar,
- a dónde quiero llegar,
- establecer las condiciones técnicas de las posibles soluciones,
- bajo qué entorno tecnológico se quiere trabajar , si no lo pongo yo me lo pone el otro. Entonces yo digo yo quiero que me desarrollen en Sybase (por decir algo), Power Builder, con el Application Server de oracle pero el motor de base de datos puede ser Sybase, quiero que me presente un esquema de módulos, un esquema de símbolos, de secuencia de teclas, etc.

Aquí en los sistemas en llave en mano, hay un problema muy importante, que es el tema de decidir quién hace el mantenimiento. En organizaciones muy pequeñas (no el estado), uno puede dejar que el mantenimiento lo haga el proveedor, pero vamos a suponer que es la contaduría de la provincia, quiero un sistema financiero de llave en mano, el mantenimiento yo no lo puedo fijar cuál va a ser costo porque no sé lo que me pueden pedir de cambio. Si es un cambio simple, un vicio oculto, lo pago yo como proveedor pero si vienen y me dicen "mirá cambió la ley financiera y cambió todo esto" yo no puedo saber cuál va a ser el impacto, tal vez tengo que cambiar 1000 líneas o 100.000 líneas. Entonces, si hacés compra llave en mano, asegurate de que la transferencia tecnológica sea fuerte. Entonces tenés que aclarar bien bajo qué condiciones va a hacerse la transferencia tecnológica, la transferencia tecnológica no es darte 10 manuales, 1 cd u 11 discos, un código fuente. La transferencia tecnológica es insertarte en la tecnología de desarrollo y que cumpla las normas de desarrollo que vos hiciste, hacer talleres para que te digan no solamente qué hicieron sino como lo hicieron y porqué; hacer cursos formales; presentar la documentación y aprobarle la documentación. Si vas a trabajar con orientación a objetos tenés que decirle que documentación querés, a qué nivel de detalle. Es muy importante el proceso de selección y evaluación de las ofertas, o sea, vos no podés elegir solamente por el precio, al contrario uno se encuentra que los que tienen precio más barato son las que se encuentran en situaciones más precarias, ¿porqué situaciones más precarias?: porque no tiene que cuidar un nombre porque tiene poco nombre, menos estructura que mantener, menos riesgos que salvar.

Otro tema muy importante es cómo fijar la migración y la implementación. La tecnología de migración es muy importante porque en los sistemas legales heredados, uno buscaba al máximo las facilidades técnicas porque era muy caro, entonces sobre un campo si podía ir esto y/o esto, le ponía esto o esto, eso va contra las normas de normalización pero por ejemplo: nosotros teníamos en lo que era el impuesto inmobiliario, el dominio o sea el número que le da el registro de propiedad cuando uno registra la propiedad, pero en algunos casos la propiedad tenía n° pero no tenía dominio porque salió una ley de expropiación, fue expropiada por la provincia para hacer un camino para lo que sea pero no tiene n° de dominio todavía porque eso demora en registrarse en otra parte. Entonces le poníamos ahí el n° de ley, entonces cuando yo quiero tener el campo estructurado no puedo perder lo que tenía de antes, tengo también que preservar lo malo, lo mismo que pasa con los domicilios yo puedo tener calle, n°, código postal con los 6 dígitos pero ¿y lo que tengo?: tengo que tenerlo tal cual está; entonces, hasta que tener muy en claro cuál va a ser esa estrategia. En los sistemas teleinformáticos yo no puedo decir mañana salgo con el nuevo tengo que hacerlo por nodos. O sea, primero hacer una prueba piloto, después probarlo en dos o tres nodos, y después hacer el despliegue general porque ese despliegue general demora días si no hay problemas porque tengo que ir a 40, 50, 100 lugares para poder hacerlo. Entonces, ¿qué hago mientras tanto para tener la información computarizada?: hay cosas que salen por el sistema viejo y hay cosas que salen por el nuevo. Además, hay ciertas particularidades, que yo tengo que seguir atendiendo al contribuyente en cualquier lugar que vaya como lo estoy atendiendo ahora; o sea, vos sos contribuyente de reconquista del impuesto inmobiliario pero tu casa en Bs. As., vos te vas a una oficina en Bs. As. y vos podés hacer la liquidación de impuestos para hacer la consulta. Entonces, ese servicio se tiene que respetar, vos no le podés decir: "estoy cambiando el sistema".

Otra cosa muy importante a tener en cuenta, es que la velocidad de implementación que pueden tener las empresas es muy diferente a la velocidad de asimilación que puede tener el comprador. O sea, yo tengo una solución para este problema y mado 400 tipos en un gol, que se cuelen por la ventana y me metan la solución, pero yo tengo que tragarla. Si bien están las especificaciones dadas, hay que validarlas a esas especificaciones porque pueden haber cambiado algo. El proveedor tiene dos momentos desde el punto de vista económico: el primero ganar la licitación con una ganancia razonable; y el segundo momento es maximizar esas ganancias razonables, o sea, minimizar lo que ya previó gastar. Entonces, hay que tener mucho cuidado de no aflojar para poder defender lo que uno

está pidiendo; entonces, hay que tener buenas herramientas al nivel de contrato para que esto pueda cumplirse.

Una de las cosas que les comentaba al principio es que uno no puede seleccionar sólo por el precio. El precio tiene que ser una de las variables. Cuando uno compra un comodín (algo estandarizado) selecciona por el precio; si ustedes quieren comprar un Siena rojo, modelo tal, el que me lo venda más barato es el mejor. Pero cuando ustedes están comprando un software aplicativo con una solución general de capacitación, etc., el precio tiene que moverse en una franja que tiene que ser un factor que pueda moverse por abajo o por arriba, pero es muy importante la calidad de lo que nos están vendiendo porque no hay dos software iguales, no hay dos formas de hacer las cosas iguales. Entonces, uno dice "un porcentaje lo pongo por el precio y es razonable, otro lo pongo por la solución técnica". El tema es poder valorar las cosas que están adentro, las cosas que le presento, porque entonces ya está comprando por lo técnico y lo técnico cuánto vale y tanto. ¿Qué cosas voy a criticarle a lo técnico?, ¿qué cosas hay que ver?:

- La descripción del software.
- Capacidad de crecimiento (horizonte).
- La descripción del software de aplicación que ofrecen, hoy el software de aplicación que uno tiene que comprar en este tipo de sistemas tiene que crecer en software sustento a que uno no tenga que ver el fuente, porque además si me muestra el fuente me lo va a mostrar en C, y si me muestran 5000 líneas en C no lo va a entender ni mambrú. A los productores también les conviene tener un fuente lo suficientemente parametrizable para que mi secreto industrial quede en Alemania o Francia o en el lugar de origen donde sea y a los demás me parametriza.
- ¿Cómo va a ser la capacitación?: la capacitación es muy importante sobre todo cuando estamos en el interior. Cuando vos estás en el interior en tu trabajo, no podés hacerle hacer un curso y trabajar porque o no va a aprender nada en el curso o no va a trabajar. Entonces, es muy importante saber que no duren más de 4 horas diarias, que el lugar de dictado quede perfectamente fijado, la duración en horas, los contenidos, el perfil del docente, cuántos alumnos pueden concurrir por curso, cuál es el perfil necesario para hacer el curso. Además hay que ver qué otras actividades se van a hacer para capacitar, como por ejemplo los talleres. Yo hablé de un tipo de talleres, pero hay dos tipos de talleres: uno es para transferencia de tecnología (qué hicieron, cómo lo hicieron y porqué lo hicieron con esa tecnología); y otro tipo de talleres son para *esgrimir* otro tipo de controversias técnicas, la gente piensa una cosa y el programador otra, vamos a juntarnos y vamos a ver cuáles son las diferencias; siempre aparecen divergencias, esto es peor que los matrimonios porque en los matrimonios por lo menos uno siente cariño por lo menos al principio, pero acá tenés un interés comercial y se puede ir diluyendo con el tiempo, cuando empiezan a aparecer los problemas.

¿Cuál va a ser el servicio técnico si no está fijado?: nosotros normalmente cuando yo trabajo este tipo de tareas, le pongo que el tiempo de respuesta de atención tiene que estar en menos de 2 horas. Cuando pasa tanto tiempo dependiendo del día y del horario que es porque no es lo mismo que me quede sin sistema un lunes a la mañana que un sábado a la tarde, se le empieza a poner multas, porque ¿qué pasa?: yo pago el servicio técnico para tener el servicio. Además la multa es algo muy bueno no para empobrecer al proveedor sino porque la supervisión del servicio técnico cuando ve que hay multa, no acá no nos estamos portando bien, prometimos una cosa que no anda, a vos te vienen a pedir referencias y vos le decís: "mirá, hay multas acá". Tampoco le pones multa alta por dos razones: primero que ese proveedor si es un proveedor serio, la tasa de riesgo que va a asumir, te va a subir el costo del software; y en segundo lugar, que si el proveedor no es previsor, lo fundis, entonces no ganás nada poniendole multas altas sino que las multas deben ser bajas, que quede registrado efectivamente que no cumplió. Para que se den una idea, el límite de la multa mensual no puede ser mayor a una vez, dos veces lo que cuesta el costo mensual del servicio.

Es muy importante cómo piensa hacer la migración y la implementación que ya lo hablamos.

Otra cosa importante a tener en cuenta al momento de comprar sistemas llave en mano, es tener referencias de otras instalaciones.

En todos los productos que le estén entregando, ver el nivel de estandarización que tenga, el problema de no tener un sistema estandarizado es que no encuentro el personal capacitado.

Un tema muy importante es cómo va a ser el esquema de administración, ya de acceso, backups, controles.

Hay que ver bien el tema costos, que no haya costos ocultos (eso lo vamos a ver después), o sea, que los costos sean absolutamente claros. Yo lo que trato siempre es que una vez que se tiene el puntaje técnico, hacer un solo precio, la forma de financiación fijarla uno y no que cada uno te la fije sino tenés que poner a contadores que trabajan para ver cuál es más barata, que calculen ellos, así no tengo costos financieros, tengo un solo precio, así nadie me puede decir usted hizo mal el análisis financiero.

Los sistemas llave en mano se utilizan mucho en los mercados verticales. Estos mercados verticales son los que funcionan prácticamente en todo el mundo igual; por ejemplo: los hoteles, todos los hoteles funcionan igual ya sean grandes o chicos, lo que cambia es la cantidad de transacciones pero la información operativa es esencialmente la misma. Otro ejemplo son los sanatorios, los sanatorios son hoteles con información por sala, pero también, te cargan remedios, te cargan adicionales, te cobran por el televisor, te cobran por el aire acondicionado, si el acompañante quiere pensión o no quiere pensión; es más caro que un hotel y más incómodo, esa es la diferencia.

El tema personal, me refiero a administración de personal, no es tan vertical porque no es lo mismo como administra el estado al personal a como administran en otro lado. En el estado vos podés tener 3 empleos, podés tener tu rogancia, ¿qués es eso?: tu reemplazo, hay reemplazante de reemplazante de reemplazante, tienen como 7. Eso es imposible imaginárselo en una empresa, le echan una patada al primero y chau, viene el segundo.

Por último, tengan en cuenta que cuando uno está comprando un software aplicativo, sea llave en mano o no, tiene que ver el método de análisis de los cuatro frentes que le dicen, los cuatro pro:

- Un pro que es el proveedor por la dependencia que genera.
- Las características técnicas del producto, porque por más que sea IBM que me lo venda que es la empresa que más vende pero si el producto es malo para qué lo quiero.
- El precio de venta, es otro factor a tener en cuenta.
- Los servicios y prestaciones que ofrece el prestador.

CLASE TEÓRICA DEL 24/09/01

Selección de software de BD

Vamos a detallar las BD relacionales porque son las más comercializadas a nivel de administración. ¿Cuáles son los Problemas que tenemos con aquellos dataset que no están bajo una BD?

1. Redundancia de datos.
2. Dependencia del programa al dataset. O sea yo modifico el dataset, con cualquier cosa puedo modificar todos los programas que yo uso, no a la codificación sino al dataset.
3. Obtener algunos campos significa leer el registro completo.
4. Los accesos son poco flexibles.
5. Existen dificultades para establecer relaciones entre datos de aplicaciones diferentes. O sea el mantenimiento de múltiples índices secundarios es dificultoso. Por ahí hay algún programita que se olvida de actualizar algo, lo perdemos.

La solución para esto fue la tecnología de Base de Datos. En realidad al origen empezó a llamarse de Banco de Datos, y después los norteamericanos le cambiaron el nombre por Base de Datos. La idea de banco de datos es porque el banco de dinero es un depósito de datos bien administrado, donde uno deja el dinero y espera recuperarlo, de acuerdo a las condiciones del depósito.

Una Base de Datos es un conjunto de datos no redundantes, compartidos por una o más aplicaciones que están lógicamente relacionados entre sí.

En algunas industrias, como las farmacológicas, es obligatorio por ejemplo. Un determinado laboratorio fabrica un determinado medicamento, se lo da a la droguería, la droguería lo revende a la farmacia y la farmacia lo revende a uno. Si un componente, por ejemplo algún medicamento, tiene alguna falla, el laboratorio y el fabricante tiene la obligación de decir que partidas se hicieron con esos medicamentos de ese proveedor. Vos no podés mezclar, como el azúcar que compras en dos proveedores distintos y los mezclás, tenés que tener el stock por cada proveedor, que proveedor provocó esa remesa y a que droguería le vendiste la partida. Y la droguería debe dar la información a qué farmacia le vendió. Primero para retirarlo y segundo para ver si se puede detectar quienes fueron los compradores. O sea que requiere vincular proveedores con producción con ventas. Debe estar todo relacionado, y esto es imposible de llevarlo hoy con archivos no base de datos. No es que sea imposible, pero es muy dificultoso.

Ventajas de las BD

- Independencia de los datos de los programas. Cuando modifico una tabla solamente debo modificar los programas que van a usar esas modificaciones. Hay que tener cuidado con esto, y los datos que uno agregue los debe agregar al final de la t-upla porque hay un puntero relativo al final de cada campo, no puedo insertar un dato en el medio. Si lo inserto en el medio todo lo que va de ahí para la derecha se modifica el programa porque cambian las direcciones relativas. Tiene un puntero de dirección cero, de 0 a 4, longitud 5.
- Coherencia en los resultados, porque el dato está una sola vez. No quiere decir que esté bien, por lo menos es coherente.
- Integridad, porque el acceso a los datos se hace a través de un gestor.
- Confidencialidad, porque tienen incorporada seguridad.
- Mayor disponibilidad, porque los datos son compartidos.

Elementos de un sistema administrador de BD

- 1- Base de Datos, o sea los datos en sí.
- 2- Gestor de la BD, o sea el motor, el software. Si uno compra Oracle el motor Oracle es el gestor.
- 3- Transacciones.
- 4- Diccionario de datos.
- 5- El administrador de la BD, éste es un humano. Siempre dijimos que algo que es compartido tiene que administrarse.

¿Qué características generales tendríamos que tener en cuenta para la selección de a BD?

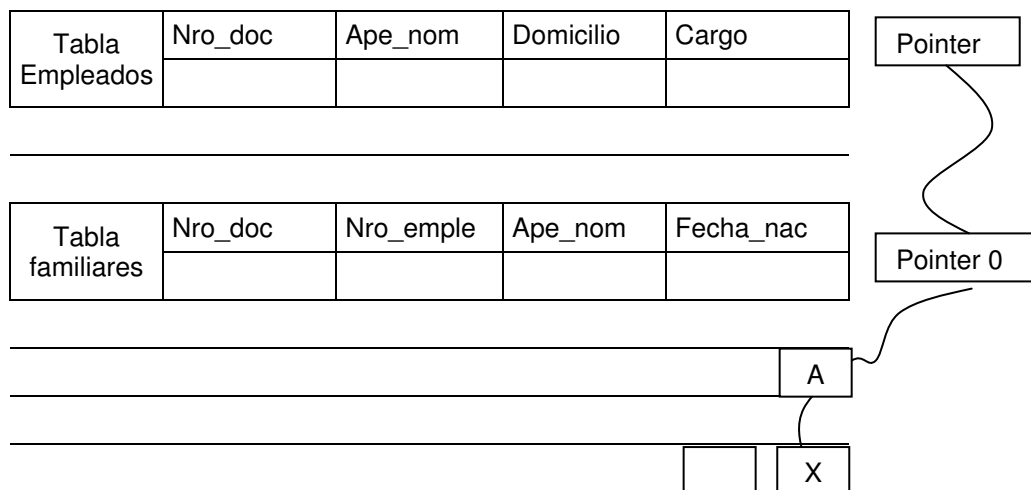
- 1- Mecanismos de acceso y recuperación de la información. Por ejemplo si yo tengo un número de claves correlativo, registro de datos de la persona número de documento, yo puedo usar como clave de búsqueda el número de documento pero no a través de un índice, sino número de documento por longitud del registro-1, sé donde tengo que posicionarme dentro de un área, dentro de un espacio de direcciones, o sea no necesito para buscar un índice ir al área de datos. Hay motores que cuando la clave no es continua te permiten compactar, técnica de compactación de claves. O sea si vos tenés un número de 10 cifras de claves, pero en tu población como máximo van a llegar a ser como 100 mil, toma el número de 10 cifras de la clave lo divide por 10000, el resultado va a ser un número que va a estar entre 0 y 99999. Lo que pasa es que puede tener sinónimos, o sea que dos cosas se dirigen a lo mismo. Entonces ¿cómo trata a los sinónimos? Normalmente se generan áreas de sinónimos, porque solamente en esos casos los busco en forma directa. En realidad lo que tengo es un índice que me dice tal clave está en tal lugar.

Otra forma de búsqueda, por ejemplo, (esto lo tiene Oracle, lo tiene el DB2 de IBM) es guardar índices de los caminos más usados. Es como si fuera el paginado de memoria, que las páginas más usadas tienden a quedarse en memoria central. Si vos le das espacio grande, obvio va a guardar todos los índices, pero si tienen poco espacio va a liberar los índices de los caminos menos usados, para poner caminos nuevos. Hay motores que te aprovechan caminos previos, o sea si vos querés buscar por Oro Verde, por padrón femenino, y tengo el índice por Oro Verde pero no por padrón femenino, primero filtra todo lo de Oro Verde y de ahí todos los que son mujeres. En vez, hay otros motores que no, porque no tienen el camino relacionado por ciudad y sexo busca completo.

Las BD relacionales hacen la relación en memoria real, y cuando tiene que buscar en toda la base de dato el tipo te carga la base de dato en la memoria real completa y la va mandando a swap en caso de que no le alcance la memoria real, lo swapea. Entonces está leyendo del disco y volviendo. Para hacerla performante (que tenga mucha performance) hay que tener mucha memoria real. Cuando uno tiene 1000 registros no hay problemas, porque con las maquinas de ahora de 128 Mb los tiene a los 1000 registros en memoria. Pero cuando hay más de 1 millón de registros en varias tablas y uno quiere relacionar 5 o 6 tablas, empieza demorar minutos en la mejor máquina que se pueda, porque está swapeando, mandando al disco y buscando hasta que arma todas las tablas.

- 2- Gestión de almacenamiento físico: no está guardada como tabla sino como mejor le convenga al tipo. Por ejemplo en las bases de datos jerárquicas decíamos que eran lo más performante con respecto a ir a una unidad de información, por ejemplo a un empleado, a un cliente de banco, porque la información tiende a estar junta. Yo puedo tener las siguientes tablas, empleados y

familiares. Puedo tener índices por el número de documento, por las 4 primeras letras del



apellido. Pongo un pointer en donde está mi primer registro de familia, y de ahí salta al siguiente miembro de la familia, y de éste al último miembro. No voy a usar un índice secundario para recorrerlo, sino que directamente a través del pointer directamente averiguo donde está mi primer familia. Y en realidad si el tipo ve que yo la estoy usando mucho no a nivel de tablas, como si fuera una red, sino que la estoy usando mucho a nivel jerárquico, me lo almacena juntos. Entonces conceptualmente veo tablas, pero físicamente nada que ver con tablas. Eso para poder hacer exploración a nivel de tablas y exploración a nivel de registro lógico, de todos los datos pertenecientes a un empleado que está en distintas tablas lógicas.

- 3- ¿Cómo es el mecanismo de declisión de datos?
- 4- ¿Cómo es el nivel de seguridad? Acá se puede conseguir sensibilidad a nivel de t-upla, o sea a nivel de contenido de segmento. O sea yo le autorizo a alguien a que solamente pueda modificar aquellos campos, por ejemplo aquellos cargos que estén entre tal rango de valores, no puede modificar otros cargos. Se llama sensibilidad del tipo B1 a nivel de segmento. Ver como trata los Dead-locks. Los Dead-Locks se producen cuando un programa pide un registro A (RA) y por ver el RA pide ver el registro B (RB) y por otro lado un programa 2 pide ver el RB, y por ver el RB accede al RA. Entonces el programa 1 está esperando que se libere el RB y el programa 2 está esperando que se libere el RA. En la práctica le decimos se colgó la máquina. Lo que pasa en realidad, para el tratamiento de los dead-locks lo que hace es que un tercer programa recorre una tabla y ve los permisos que están protegidos y esperando los registros. Cuando encuentra que están cruzados, vuelve para atrás las transacciones en curso que pueden ser abortadas, las aborta y las relanza con diferentes tiempos. Nosotros en la práctica en realidad hacemos eso, apagamos la máquina y la prendemos. Eso también pasa cuando tenemos archivos batch que se están procesando con procesos interactivos, como cuando uno trabaja en batch puede procesar muchos registros por segundo, sobre todo si los tiene agrupados en bloques contiguos, y en proceso on-line yo puedo tener un registro varios segundos viéndolo (porque está el humano) puede cursarse este tipo de cosas.
- 5- Mecanismo para garantizarme la integridad, o sea cómo gestiono la recuperación. Hay un fichero log, de login, si uno lo tiene que archivar este archivo, es recomendable que esté en soporte que se pueda extraer, en una cinta o un cartridge o un CD que pueda ser extraído, donde guarda como está la situación de la BD, antes de la modificación y después de la modificación. Además guarda desde donde vino la comunicación, quien la hizo, a que hora la hizo, cuanto duró (segundos, milisegundos). Hay un archivo que me permite hacer copias down, copias imagen de la BD, me copia tal cual están grabadas. Entonces eso me permite hacer backup. Producida una irrupción del servicio, una vez que empecé a activarlo, la irrupción puede ser por Software Crash o por Hardware Crash. Entonces una vez que pude poner el S.O. montado de nuevo, puedo levantar mi ultima copia down y tomo los logins que he usado a partir del último backup, de la última copia down. O sea tomo mis archivos de login por un programa que me dan ellos, que lo que hace es poner la BD bien, al momento del corte. Normalmente se trata de hacer un down todos los días, es aconsejable por las noches hacer el backup de la BD. Entonces uno tiene un login diario. Cuando las BD son muy grandes es muy costoso hacer un

backup, porque no puede alcanzar el tiempo de la noche entonces se puede hacer lo que se llama backup incremental, solo backup lo que cambió, entonces la información que se guarda es menos y los fines de semana hago un backup completo. Siempre conviene tener 2 o 3 archivos de login, que permite que yo lo tenga grabado varias veces, no conviene tenerlo muchas veces porque me carga mucho la transacción, conviene tenerla uno en un carrete que puede ser extraído y otro en un disco, por las dudas se pueda dañar el carrete. Existen dos tipos de recuperaciones o login. Una es que uno toma la BD como dueña cuando actualiza con la cinta de login, entonces nadie puede trabajar. Y otra forma es que uno puede seguir trabajando aunque la BD se sigue actualizando. Depende del tipo de sistemas, por ejemplo un sistema de reservas de pasajes lamentablemente no puede vender un pasaje más si quiero ser honrado hasta tanto no termine mi backup. Muchas veces no es que sobre venden pasaje, siguen vendiendo pasaje especulando que devuelva y no llega al avión. Es importante ver las relaciones con el entorno operativo. O sea que soporta la BD, SQL Server solamente corre bajo W2000 o XP, hay que ver siempre el entorno en que corre, la BD ADABAS corre en AIX pero no en un SOLARIS2 de SAM. No corre todo en todos los ambientes.

- 6- Operatividad: necesidad de aprendizaje y de niveles de experiencia. Esto tiene que ver tanto para los desarrolladores como para los administradores. Si bien a nivel desarrollador casi todas las BD son casi iguales, muy parecidas, son muy parecidas cuando utiliza SQL pero son diferentes del punto de vista de desarrollo en cuanto a los procedimientos almacenados, como maneja tablas temporarias, no son todas iguales. El SQL tampoco es todo igual, pero es bastante parecido, lo que sí es diferente es como se administra. Entonces el database administrator si yo tengo 2 o 3 BD usando, tengo que tener especialistas en administración de cada uno de los productos. Otro tema importante es herramientas para la implantación inicial. Por ejemplo yo estoy trabajando bajo Oracle y quiero irme a Sysbase. ¿Qué me ofrece Sysbase para irme de Oracle?
- 7- ¿Qué facilidades tiene para la administración? Por ejemplo ¿cómo se catalogan las transacciones? ¿Cómo es el monitoreo? Nivel de claridad e idioma de los mensajes, esto permite evitar que se demore tiempo cuando la producción está en actividad, sacar adelante errores que se produzcan. Tener claridad de que significan los errores y poder llegar a subsanarlos.
- 8- Herramientas para reorganizar datos. ¿Qué facilidad ofrece para programar transacciones? O sea que producto puedo usar que me facilite la programación de transacciones para no trabajar en lenguaje nativo. ¿Qué recursos son necesarios para utilizar la base, por ejemplo a nivel físico? ¿Qué cantidad de memoria principal se requiere/recomienda por usuario? Siempre no decir lo mínimo sino lo recomendable. ¿Cuáles son las capacidades de CPU recomendada para la cantidad de usuarios que uno va a correr?. ¿Qué requerimientos de recursos humano se necesita? Administrador de BD, system programmer, análisis y programación. Esto es difícil que lo diga el proveedor, hay que preguntárselo al cliente. Una empresa privada tiene que tener 2 y la administración pública 3, uno siempre se enferma. ¿Cuáles son los servicios técnicos del proveedor que ofrece (que capacitación ofrece)? ¿Cuál es la documentación del producto? ¿Qué piensan los clientes de ese proveedor? Nunca vayan a un cliente del proveedor por el proveedor, siempre busquen contacto institucional, a nivel de empresa, de la cámara de empresas, de la facultad, siempre que sea institucional. Nunca hagan un transplante, por ejemplo nosotros estamos a más de 500 Km de Bs. As y por más que haya gente muy bien atendida allá necesariamente no va a ser muy bien atendido a esa distancia.

Hay cosas muy importantes, por ejemplo si vamos a tener un sistema teleinformático ver si encripta, si sale bien afuera. El encriptado hay que hacerlo para datos muy sensibles. También es interesante el tema de la compactación de datos. Si yo mando ASCII, si el dato es numérico con 3 bits tengo un dato de 0 a 10 y si el dato es alfabético y no me interesa mayúscula, minúscula ni coma, yo puedo decir que con 5 bits tengo el alfabeto, cuando hablamos de que los caños de comunicación son en segundos, hablamos del disco de milisegundos y hablamos de memoria real en nanosegundos, estamos hablando en escala, entonces todo lo que puedo mandar en un mensaje más corto sin perder calidad gano tiempo.

Gestión de la configuración o administración de la configuración

Se pretende tener todos los referenciales del proyecto. Un referencial de proyecto es desde un formulario usado en un relevamiento, una definición de algo, un Diagrama Entidad Relación (DER), un Diagrama de Flujo de Datos (DFD), la definición de un campo de una tabla, un código fuente, un código objeto. ¿Y que tengo que tener? Saber cuando se inició, quién lo inició, y si ese software va por etapas, cuándo se terminó, hasta cuando tuvo vigencia (en el caso de que se cambió), desde cuándo tiene vigencia lo nuevo, quién hizo el cambio, por qué lo hizo, como lo hizo, para que lo hizo, es decir tener una memoria completa de todo lo que fue, de como se desarrolla el sistema y de como se opera el sistema, de como se implementa. Existen herramientas para hacer esto, son BDOO, el motor que está atrás, o sea el esqueleto de lo que es la gestión de la configuración para que se pueda utilizar. Hay una tendencia en centros de cómputos medianos a grandes para preverlos, en los centros de cómputos pequeños los referenciales los tienen en la cabeza, si estoy en una zapatería no voy a llevar la gestión de la configuración. En organizaciones medianas y grande es imperativo tener este tipo de producto. Son varias las empresas que ofrecen software para la gestión de la configuración.

El tema es que en la administración de la configuración yo vuelco el diccionario de datos. Tengo que tener una interface para comunicarme con mi herramienta de desarrollo, con mi CASE, para ver que es lo que me ofrece. La gestión de la configuración sirve sólo para decir qué es lo que tengo, dónde y cómo lo tengo, pero no sirve para programar.

Sistemas de Oficina

Existe una fuerte tendencia desde hace mas de 8 años de automatizar una oficina. ¿Qué es una oficina? ¿Qué hace una oficina? Una oficina recibe, confecciona (firma por lo menos, hace un pase) y transmite información. O sea, prepara y confecciona documentos, en la oficina tradicional era dictado con taquigrafía, mecanografía, escritura manual que todavía sigue existiendo. Pero la preparación y confección de documentos fueron reemplazadas con el procesador de textos.

La oficina transmite, o sea distribuye mensajes, teléfonos, télex, correos, cadetes, e-mails. Mucha comunicación, medio de comunicación. Otra cosa que hace es archivar cosas, también consulta y recupera información archivada. Este conjunto de disciplinas las llamaron buróticas, de tratamiento automático de buró (oficina). Hay dos cosas interesantes:

★ **La ventaja del e-mail**, lo vimos con el tema métodos de comunicación.

★ **El archivo electrónico – Gestión electrónica de la oficina**

Uno almacena información en archivos, cajones, carpetas y el cesto. Lo interesante del archivo electrónico es que el documento está una sola vez. El tema es ¿cómo genero documento de forma electrónica? Tengo dos orígenes de documentos, uno el documento que genero yo, que puede ser por medio electrónico, y lo archivo en forma electrónica, la ventaja que tengo que lo puedo imprimir con muy buena resolución, porque lo tengo en ASCII, una hoja A4 en ASCII tiene 2500 caracteres más o menos de almacenamiento, una hoja A4 que yo tenga escaneada y la escaneo en blanco y negro a calidad 100%, calidad fax, que es muy baja, y le aplico un mecanismo de compactación estándar (CCITT del Grupo 4) estoy en 20 Kb. Todo lo que pueda llevar a ASCII gané dos veces. Hay convertidores semi inteligentes que tratan de buscar por todos los medios tratar de traducir, si hay una palabra que no pueden traducir, la dejan marcada para que vos la traduzcas, y ves si lo puedes pasar a ASCII, si ves que hay mucho que no puedes pasar a ASCII, lo puedes indexar y lo mandas a la BD al lugar que corresponda, a la carpeta. El nivel menor de la codificación es la carpeta. Los documentos que vienen de afuera se quieren integrar en esto, para eso se pidió la firma electrónica con la doble certificación de firmas (pública y privada). En EEUU es muy usual esto, el problema es que puede salir muy caro para certificar la firma. Se busca la oficina sin papeles. Toda la cadena comercial desde el proveedor al cliente sin papeles, pasando por el producto. Usando una tecnología que se denomina de Work-Flow. Esto es una técnica del traslado del expediente electrónico, traslado electrónicamente el expediente. El Work-Flow se embebe con transacciones, con procesadores de texto, con carpetas electrónicas (BD documental).

	Work-Flow	
Transacciones	PTX	Antecedentes?
BD		BD

Lo que uno puede hacer es, vamos a suponer que me llegan papales un trámite de afuera del exterior, entonces el work-flow le pone un número de expediente electrónico. Ya sabe si el trámite está

determinado, por ejemplo va a la caja de jubilaciones y es el pedido de una jubilación, de un beneficio, entonces le pone código de hoja de ruta, puede ser 01 o 02, que tenga la tablita por ahí, y automáticamente le dice cual es el nombre del beneficiario, el documento para que lo puedan asociar en la oficina de beneficiarios, para que puedan ver los antecedentes en su base de datos, los antecedentes del futuro jubilado en hoja de servicio electrónica y si la persona que lo recibe quiere hacer un comentario, por ejemplo el trámite no procede porque tiene tal problema lo devuelve a mesa de entrada y el trámite queda ahí, y la persona de la mesa de entrada a lo mejor lo verá del Hard Copy, la copia dura del último mensaje. Pero el papel no circula por adentro, el papel queda en la recepción. Permite que uno pueda hacer cosas en paralelo, o sea que el expediente lo vea jurídica y técnica, y después lo vea el gerente financiero. Esto permite acelerar los trámites.

Toda oficina mediana o grande tiende a usar Work-Flow, y para tener Work-Flow se tiende a que se necesita una PC por cada empleado. El Work-Flow permite por ejemplo, si dos están trabajando juntos, son abogados, y uno se toma vacaciones el otro toma su trabajo automáticamente. El Work-Flow permite si yo tengo un especialista comercial y un especialista civil, todo lo comercial se lo doy al primero y todo lo civil al segundo. Puedo consultar en una oficina quien esta a cargo de un trabajo o puedo consultar quien tiene menos trabajo para hacer. Me permite saber donde está cada trámite sin moverme de mi escritorio, entonces yo sé en donde está cada trámite y desde cuándo está cada trámite, entonces yo puedo hacer que me alcahuetee todo aquello que está más de 3 días en un mismo lugar, le voy a preguntar o mandarle un e-mail para decirle "che, ¿qué te paso?". En EEUU empezaron muy fuerte con esto hace tiempo, ellos eran grandes gastadores de papeles, un abogado de New York decía que gastaba una cantidad de papeles equivalente a 100 toneladas de madera por año. No solamente hizo reducir tiempo y costo, espacio de archivo, hizo reducir un montón de cosas, pero cuando empezaron a integrar la cadena de terceros y darle validez legal a los documentos, tuvieron que conseguir el método de la doble firma electrónica, la doble clave. De todas maneras lo último es lo de la doble clave, si no tengo los procedimientos sistematizados y automatizados. Con el Work-Flow. No tengo la firma electrónica. Entonces no hay que partir de la firma electrónica, hay que terminar en la firma electrónica.

Evaluación del software de aplicación

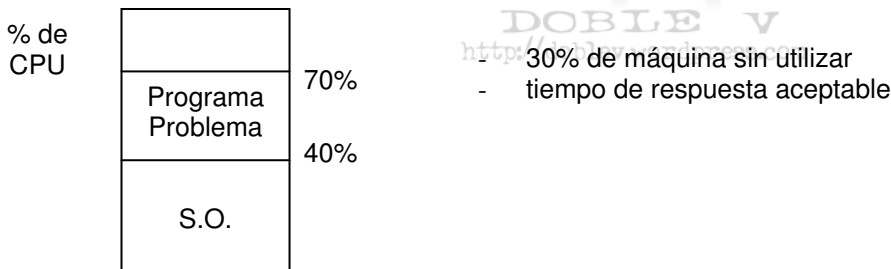
Israel Borovits en su libro "Manejo de operaciones computacionales" propone hacer un cuestionario de evaluación a usuarios del software, un cuestionario de evaluación al vendedor del paquete, un test de Benchmark al paquete de software, y con estas tres cosas hacer una planilla de evaluación.

1. Cuestionario de evaluación al usuario

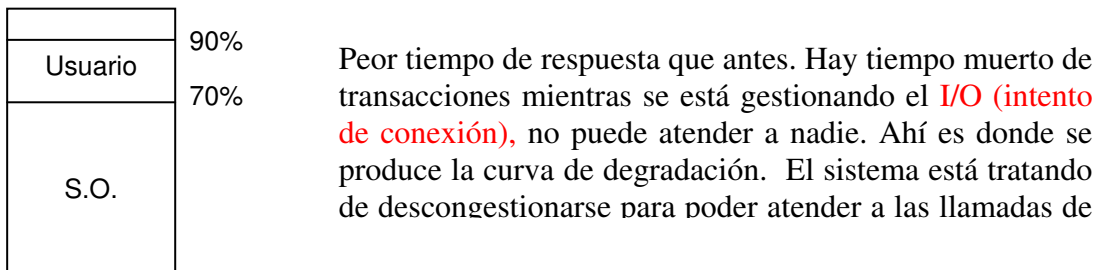
- a - Operación: ¿Es fácil de utilizar? ¿Se requiere mucho conocimiento técnico para usar el mismo? ¿Ha habido rotación de operadores?
- b - Confiabilidad: ¿Qué tiempo lleva de uso? Durante el tiempo de uso, ¿qué cantidad de actualizaciones hubo? ¿Cuáles fueron las causas de las mismas? ¿Cuántos errores fueron detectados? ¿Qué partes presentan posibilidades de error? ¿Qué partes no utiliza? ¿Por qué? ¿Qué errores del usuario pueden provocar la caída del sistema? ¿Cuál es el tratamiento de las recuperaciones? ¿Qué tiempo lleva hacer las mismas? ¿Existe un paquete de diagnóstico para verificar que el sistema funciona correctamente? Acá se usan dos técnicas. Una es como si fuera una prueba de test, uno aprieta un botón y ve si todo está bien, si el alternador corta o no, bueno acá sería que hay un lote de prueba que si los resultados coinciden con el archivo de resultados está todo Ok, nada fue alterado. Esta técnica se complementa con lo que es la casita de auditoría, y ¿qué es? Imagínense en un banco una sucursal virtual, y ¿qué es una sucursal virtual, es una sucursal que a los fines contables no se sumariza, salvo que se pida expresamente, pero uno puede operar como si fuera cualquier sucursal. Entonces vamos a suponer, yo soy el responsable como banco (estoy en la parte de organización y procedimientos) de implementar una modificación del banco central. Entonces llega a la sucursal una disposición del banco central, miro y veo que hay que modificar el software, entonces le doy a la gente de sistemas que me haga el cambio, les digo que es lo que tienen que cambiar. Ahora, ¿cómo sé yo que el cambio está hecho y bien hecho? Porque lo ejecuto en la sucursal de auditoría, por ejemplo si el cambio es el impuesto al cheque, le empiezo a mandar cheques sobre las cuentas ficticias que tengo en la sucursal de auditoría y veo si efectivamente funciona o no, entonces me aseguro que esté bien hecho. Trabajar con un lote de prueba así me aseguro que el software que está corriendo tiene el cambio hecho. ¿Qué respaldo le da el vendedor?

- c - Mantenimiento: ¿es confiable y accesible el vendedor? ¿Con qué frecuencia se requiere el mantenimiento? ¿Con qué frecuencia se requieren cambios al software? Esto es así porque está dado en función de la flexibilidad que tenga, cuando más cambios le pido menos flexible es el software ¿el servicio es competente para resolver el problema? Y acá se refiere a dos cosas, yo puedo tener un problema vinculado al producto que me entregó o al entorno de ese producto, como está administrado, o sea puedo tener un problema vinculado a las funcionalidades, entonces alguien que conozca sobre mis problemáticas, yo quiero un banco que conozca sobre bancos, si yo tengo problemas de administración del entorno ya sea del SO o de la BD que conozca ese entorno de administración, no me sirve de nada un proveedor que conozca una parte y no la otra, por ejemplo los proveedores de tecnología básica conocen todo lo que es el entorno de administración pero no conocen nada de lo que es el problema, por eso generalmente las empresas de tecnología que quieren vender software aplicativo fracasan. Oracle vendiendo software aplicativo fracasa, no le va bien, le cuesta mucho poder hacer funcionar algo, a IBM le pasa lo mismo, porque ellos conocen productos no soluciones para un usuario en particular, vos le comprás el producto a ellos, no yo quiero que me haga esta BD y que sea de esta manera, cuando uno compra una aplicación dice yo quiero un programa que ande de esta manera, no están acostumbrados a hacer cosas a pedido, ellos venden el producto por eso fracasan. En vez la gente que está acostumbrada a hacer desarrollos o aplicaciones va a andar muy bien en el entorno del cliente. ¿Cuál es el tiempo promedio de atención? O sea desde que uno los llama porque tiene un problema, acá hay dos términos, cuál es el tiempo promedio de **adhesión** a partir del cual se me planteó el problema y cuál es el tiempo en que comienza a atender. Una cosa es el tiempo de inicio de la atención y otra cosa es el tiempo promedio de atención, tiempo promedio de reparación de los problemas. Esto está muy vinculado al conocimiento que tengan del producto y a la experiencia que tenga el tipo que está haciendo la ingeniería del cambio. ¿Hay respaldos adecuados? ¿Cuáles son los tiempos de Backup? ¿Qué volúmenes tienen? ¿Existen rutinas o mensajes que inducen al error?
- d - Performance: ¿cuáles son sus volúmenes diarios? ¿Cuánto tiempo lleva su proceso? ¿Cuál es el tamaño de sus archivos? ¿Cuántos usuarios trabajan concurrentemente (el máximo)? ¿Cuántos usuarios pueden trabajar con tiempos de respuestas que usted considera aceptables? ¿Cuántos segundos considera aceptable como tiempo de respuesta? ¿Cómo es la curva de degradación?

Curva de degradación



Si yo sigo poniendo usuarios me voy a encontrar que ocupo hasta el 90%.



Una posible solución en este caso sería, por ejemplo como se hizo en el Banco Galicia, que los programadores trabajasen de noche. A la mañana y a la tarde se empleaba para atención al público, después se corrían los procesos batch para consolidar la información de banco, luego backup y finalmente trabajaban los programadores. Otro recurso heroico sería limitar la cantidad de usuarios, por ejemplo en los horarios picos.

¿Cómo se resolvió el problema de degradación? ¿La capacidad de impresión es la adecuada? ¿Hay **lock out** del sistema cuando funciona la impresora en forma remota, por ejemplo?

- e - Flexibilidad: ¿modificaron el paquete ustedes? ¿Pueden modificarlo? Sino, ¿quién lo puede modificar? ¿Dónde se realizan las modificaciones? ¿Cuánto tiempo llevan los cambios? ¿Se agregaron programas? ¿Quién hizo el agregado? ¿Cómo se mejoró el sistema? (tanto en tiempo, costo, personal) ¿cómo se implementan los cambios? “Los cambios de hoy son los problemas del mañana”. Hay que saber cómo se administran los cambios. Hay dos disciplinas dentro de la informática y la administración de la informática: Administración de problemas y Administración de cambios. Un problema es todo aquel que ve que las cosas no funcionan como cree. La mesa de ayuda es un lugar con dos o tres teléfonos donde uno reporta problemas. La persona encargada registra el problema le da un código, le da prioridad si es que no se puede resolver enseguida. El problema puede ser operativo por ejemplo un cable desconectado. Si esa persona no puede resolver el problema tiene que conocer a quién tiene que derivárselo por la naturaleza del problema que se plantea o quién tiene que resolver en caso de que no sepa a quien derivarlo. Entonces el sistema debe registrar a quien se derivó, hora y fecha, y a quién se dio la solución, la hora y fecha de la misma. El problema puede subir de gravedad, puede ser grave en el origen y volverse más grave o puede ser un problema sencillo en el origen y con el tiempo volverse grave. Entonces estos sistemas de problemas me generan reportes y se los entrego al director para que se entere de los problemas que acontecen porque a él nunca le llegan los problemas.
- f - Instalación: ¿se instaló tal cual fue planificado? ¿Cuánto tiempo lleva y cuánto costó la instalación? ¿El entrenamiento fue adecuado? Hay gente que dice “Siempre la capacitación que te dan es mala”. Una cosa es hacer un curso del producto y otra cosa es hacer un Work Shop del producto y otra cosa es tener una consultoría del producto, y se cotizan distinto. Lo peor que le puede pasar a uno es ser incapacitado, ser un profesional y recibir un curso de un colega. Cuando uno termina un curso y quiere usar la herramienta pide hacer un work shop para llegar a hacer parte de una aplicación real, trabajarla, y llegar a un resultado. Y después cuando a uno le quedan dudas están las consultorías, y consultorías son tipos de primer nivel que conocen el producto en detalle. ¿El apoyo de instalación fue completo y competente? ¿Hubo anomalías en la instalación? ¿Cuáles se presentaron? ¿Qué cambios en el entorno fueron necesarios para instalar el sistema?
- g - Costos: ¿cuáles fueron los gastos imprevistos antes y después de instalar el sistema? ¿El costo es compensatorio? ¿Cuál es el costo de las mejoras? (por ejemplo buscar algunas variables, como horas hombre por ejemplo) ¿hubo gastos adicionales para actualizar el software? ¿Cuánto le costó el software? ¿En que área en relación al costo el sistema le resultó más efectivo? O sea, ¿adónde le sacó más provecho a esto?
- h - Seguridad: ¿los niveles de seguridad de usuario y data-server son adecuados? ¿Pueden correrse transacciones no autorizadas? ¿Los controles satisfacen la auditoría? Acá es importante que la auditoría esté presente al inicio del desarrollo del sistema, porque después si no está al inicio y no quedan rastros para que se pueda seguir una correcta auditoría después no se hace más el cambio ese.
- i - Documentación: ¿es precisa? ¿Es adecuada? ¿Se mantiene actualizada? ¿Por medio de qué sistema se actualiza? (por medio de hojas móviles, se pone en negrita lo que cambio de cada hoja, se hace una fe de erratas)
- j - Varios: ¿por qué compró el sistema? ¿Lo compraría hoy? ¿Qué cambios haría? ¿Qué cambios piensa usted se podrían haber implementado en realidad? ¿Qué aprendió de otros usuarios del sistema?

CLASE TEÓRICA DEL 22/10/01

2. Cuestionario de evaluación del vendedor

- a. **Uso del paquete aplicativo:** relevar la siguiente información: ♦Nombre del paquete. ♦Fecha de la primera utilización. ♦Número de instalaciones que lo usan. ♦Nombre, domicilio y teléfono de los usuarios en el país. ♦Si existen grupos de discusión en Internet (grupos de usuarios que intercambian información en Internet). ♦Cantidad de instalaciones en donde se modificó el paquete?, que es lo mismo que decir ¿A cuántos clientes necesitó modificar el paquete? ♦Se puede realizar una demostración del paquete en nuestras oficinas?. Normalmente si uno tiene que trasladarse no puede trasladar a todos los interesados porque sale muy caro, ir a Bs. As todos los

- interesados, por ejemplo, puede salir muy caro. Hoy es imposible trasladar una demo a un lugar. Desde ese punto de vista es muy interesante ver primero la demo antes de hacer evaluación a usuarios. ♦ ¿que restricciones tiene el producto conforme a sus objetivos? Las restricciones pueden ser sobre plataformas de uso, por ejemplo, corre solamente bajo Windows NT o solamente bajo Solaris. ♦ ¿Cuál es el plazo de entrega?
- b. **Documentación:** ♦ Si tiene Diagrama de Entidad Relación (DER)? ♦ Si tiene DFD? ♦ ¿Cómo se describe el sistema? ♦ ¿Cómo es el flujo de operación del sistema? ¿Cómo es la secuencia? ♦ ¿Qué instrucciones deben cumplir los operadores? ♦ ¿Qué contiene el manual de usuario? ♦ Ejemplos de informes y de salidas por visor. ♦ ¿Cuáles son los procedimientos de operación? ♦ ¿Cuáles son los procedimientos de arranque? ♦ ¿Cuáles son los procedimientos de recomienzo? O sea de reenganche, como hacer si en la mitad se me interrumpe el sistema como tengo que rearrancar el sistema. ♦ Ver si se puede ver la documentación en ese momento, y no que “después te la preparo”. Esto es importante si se entreguen los fuentes para que uno pueda hacer el mantenimiento de esos fuentes o no, porque si no comprendemos cómo y por qué funciona el sistema, independientemente de tener los fuentes. La documentación del sistema debe tenerse sí o sí por más que no tengamos acceso a los fuentes ni podamos tocarlos.
- c. **Modificación del paquete:** ♦ ¿Se puede modificar el paquete para adaptarlo al usuario? ♦ ¿Quién lo puede hacer? ♦ ¿Dónde se va a hacer? ♦ ¿Cuánto tiempo real lleva? ♦ La modificación cancela la garantía? Normalmente sí cancela la garantía. Puede darse lo siguiente que el usuario está en condiciones de hacer el cambio (conoce la tecnología para hacer el cambio) y el producto está en garantía, si hace la modificación sin autorización ni supervisión del que tiene la garantía puede ser que diga el tipo “ustedes me cambiaron tal cosa se corta la garantía”, entonces muchas veces se pacta, sobre todo cuando el software está desarrollado en otro lugar, se pacta hacer los cambios con gente propia bajo el control de ellos. A ellos se les alivia bastante porque muchas veces la gente que trabajó en una aplicación o dejó la empresa o está haciendo otras aplicaciones, no está disponible permanentemente. ♦ ¿Qué mejoras del paquete están planeadas o en proceso? Esto es para ver como está evolucionando el paquete. ♦ ¿Se entregarán mejoras futuras? ¿A qué costo?. Acá a veces la trampa está en la diferencia que se origina entre update y upgrade. Update es un cambio de **servicio**, upgrade es un cambio de tecnología. Por ejemplo hubo grandes upgrades con los sistemas cuando pasaron de trabajar en una sola capa a sistemas de dos capas, cliente servidor. Normalmente los productos que venían en una sola capa quería comprarlos en dos capas venían con mayor costo, prácticamente como si fueran un producto nuevo. No tenía un costo de update, de actualización, tenía precio sobre el producto monocapal. Cuando se paso de 2 capas a 3 capas ocurrió lo mismo. Y ahora que ya se habla de la cuarta capa, que es la capa de aplicaciones web también está pasando lo mismo. Hay que tener muy bien en claro que es lo que se pacta de cambio. Ellos le dicen actualizaciones, pero no es el mismo producto es un nuevo producto por eso tiene un costo distinto.
- d. **Costos:** ♦ ¿Cuáles son las alternativas de comercialización disponibles? Hay muchas alternativas y algunas tienen mas riesgos que otras y algunas son muy provechosas, por ejemplo hay gente que alquila el paquete de acuerdo al uso, como el teléfono digamos, cuando más lo usás al paquete más plata vas a pagar, entonces eso permite tener un entry-point muy barato. Por ejemplo el negocio del banco es prestar plata, los bancos normalmente alquilan los paquetes tanto al hardware como al software lo alquilan, porque la inversión la tiene que hacer el otro, no yo. Prefieren tener más costo económico inicial, porque lo están alquilando, nunca llega a ser de ellos, que hacer una inversión de compra y después queda el problema con uno. Si uno vende la casa que tiene, el problema pasa a ser del otro, ya está cobrada, si se rompió una pared o hay una cañería rota perdieron. Si yo alquilo la casa yo eso lo tengo cubierto. ♦ Si se alquila, ¿Bajo qué condiciones se otorga la licencia? Por ejemplo ¿se otorga por una año renovable, o yo puedo discontinuarla en cualquier momento?. Por ejemplo en IBM cuando uno alquila software, el software se discontinúa a criterio del cliente el día que el cliente diga o al día siguiente que notifique.
- ♦ ¿La documentación está incluida en el precio? La documentación de un software tiene que tener a veces muchos usuarios entonces la documentación no es lo mismo que me lo entreguen en papel que me lo entreguen en CD porque después el costo del papel lo tengo yo.
- ♦ ¿Existen costos adicionales? (horas de soporte, traslado, etc.)

- ◆ ¿Existen costos adicionales por garantía incondicional? Garantía incondicional no significa que van a hacer todos los cambios que yo les pida sino si el sistema anduvo mal por una mala operación mía, pero el sistema de alguna manera lo permitió a esa mala operación y con eso se provocó algún daño, por ejemplo envió un archivo, ellos se comprometen a repararlo.
 - ◆ ¿Cuál es el costo de mantenimiento post-garantía? Acá lo que se hace normalmente es fijar un tope, no va a ser más allá del 18% anual del valor del producto, del 15, del 12 de lo que sea. Acá hay que tener cuidado porque los costos del software tienden a aumentar y los costos del hardware a disminuir. Entonces hay que decir el costo actual del software, porque tiende a aumentar.
 - ◆ ¿Cuál es el costo por instalaciones múltiples? O sea lo voy a usar en un montón de lugares dentro de mi organización.
 - ◆ ¿Cuál es el costo de copias adicionales de la documentación?
 - ◆ ¿Se puede revender o alquilar el paquete a un tercero? Con Microsoft no hay problemas, mientras tenga cada uno su licencia.
 - ◆ ¿Cuál es el costo de modificaciones y mejoras? En todo desarrollo de software hay dos variables, ¿cuál es el costo de la hora de desarrollo?, que incluye costo de estructura, etc., que tiene ese organismo. Y ¿cuál es la cantidad de horas que necesita para poder hacer las mejoras o el cambio? Hoy al costo de desarrollo algunos lo separan en tres: consultoría, que es carísimo, que no es el desarrollo en sí; después análisis y después programación. Pero ya hay una tendencia de juntar estas dos cosas en desarrollo. Con las herramientas CASE, por ejemplo, se hace muy difícil distinguir donde empieza el análisis y donde la programación. Lo que pasa es que a veces uno empieza el desarrollo haciendo programación de bajo nivel hasta que toma experiencia pero nada más que por eso. Es un costo que uno tendría que fijarlo porque después la cantidad de horas, si hay dudas, uno puede recurrir a algún mediador que exija en el contrato, o sea en caso de dudas la UADER **auditará**. Pero la UADER no puede definir los costos que va a tener cada empresa, porque no es lo mismo el costo que puede tener Brumatti, que el costo que puede tener IBM por hora. Lo que es importante es fijar el precio por hora. Después la otra variable es más controlable.
- e. **Operaciones:** ◆ Si no suministran programas fuentes, ¿Quién es el tercero que los puede tener? Si yo comprador tengo una controversia de cualquier tipo con el vendedor, entonces el vendedor corta relaciones, yo no le pago y el corta el servicio, pero yo no me puedo quedar sin el servicio de asistencia del software aplicativo, porque si tengo que hacerle algún cambio hay que hacerlo, independientemente de la controversia que tenga con este señor. Entonces lo que se fija es un tercero que tenga la versión actualizada del software, generalmente una empresa de auditoría, donde en el caso que haya una controversia ese tercero pueda hacer el cambio. Eso es lo que se hace normalmente y a veces se pide que una copia de software además esté en una escribanía registrado, guardado en una escribanía.
- ◆ ¿Qué facilidades se necesitan para correr el paquete? O sea se refiere a hardware y software. ¿Cuáles son las facilidades recomendadas?, porque las mínimas siempre hay que multiplicarlas por dos.
 - ◆ ¿Se puede trasladar el paquete a otros entornos operativos?
 - ◆ ¿Qué nivel técnico recomienda para el personal que va a operar el sistema? ◆ ¿Cuáles son las versiones conocidas del paquete? ¿Qué incorporó cada versión?
 - ◆ ¿Qué tiempo de respuesta tiene documentado para cada transacción? ¿Con qué entorno operativo se corrieron las mismas?
 - ◆ ¿Qué procedimientos off-line son necesarios realizar? Esto significa procedimientos con sistemas que no estén disponibles al usuario final. ◆ ¿Entregan interfaces para la iniciación y/o conversión de archivos?
 - ◆ ¿Qué no se puede eliminar a los efectos de no afectar la finalidad del paquete? O sea, "Ojo, este listado genérenlo igual por más que no lo impriman porque actualiza otro archivo".
 - ◆ ¿Qué funciones cumple cada módulo?
 - ◆ ¿Cuáles son los pasos necesarios para instalar el paquete?
 - ◆ ¿Qué tiempo, cantidad de personal y facilidades son requeridos en cada caso?
 - ◆ Describa cuál va a ser la asistencia técnica que va a brindar en el momento de la instalación.
 - ◆ ¿Se puede brindar entrenamiento in-situ al personal? Esto es muy importante porque la mayoría de las personas que no están acostumbradas al ejercicio de estudiar, por la edad o por lo que sea, ve con el ejemplo. El concepto lo puede tener o no pero cuando uno le da el ejemplo termina

- comprendiéndolo. Entonces lo ejemplos los tiene en su oficina. Por ejemplo este trámite ¿cómo lo hago?
- ◊ ¿Cuándo comienza la garantía? Esto es importante en los sistemas teleinformáticos. Yo tengo que desplegar mi sistema en toda la provincia, y ¿cuando comienza? ¿Cuándo terminó el último nodo o cuando empecé con el primero? Porque ellos van a decir “no, ya lo estuvo usando ya empieza la garantía”, eso hay que tenerlo muy en claro porque pueden ser 3 o 4 meses desde que yo instalé el primer nodo hasta que llegué al último.
 - ◊ ¿Entrega garantía escrita de los niveles y calidades de los servicios provistos?
- f. **Seguridad:** a nivel de la aplicación, también puedo tener a nivel de sistema operativo, a nivel de motor de base de datos o a nivel de hardware.
- ◊ ¿Qué clase de seguridad suministra la aplicación? ◊ ¿Existe un diario de todas las transacciones?
 - ◊ ¿Se conocen los intentos de violación de la seguridad? O sea, si bien no pudo violar la seguridad, lo intentó. Entonces saber de dónde se hizo eso.
 - ◊ ¿Cuál es el esquema de backup propuesto?
- g. **Descripción de la empresa vendedora:** ◊ Datos de localización del vendedor.
- ◊ Datos de localización de los representantes técnicos y de ventas.
 - ◊ Oficina más cercana.
 - ◊ Nombre y apellido del personal de apoyo de la oficina más cercana.
 - ◊ ¿Qué cantidad de empleados tiene en total la corporación? ¿Qué cantidad de empleados tiene en la Argentina? Clasifique esos empleados por función.
 - ◊ ¿Se pueden entregar los tres últimos balances de la Empresa? Si es una sociedad anónima si lo tiene que entregar, porque los balances de las sociedades anónimas son públicos, no sabemos quien es el accionista, por definición es anónimo, por eso son públicos. Lo que pasa es que si no es una sociedad anónima puede tener restricciones para entregar los balances porque uno sabe quién es el dueño, y en ese caso siempre se presenta algún tipo de problema de seguridad propia de la empresa.
 - ◊ ¿Qué otros paquetes de software venden? ¿Qué cantidad de instalaciones los tienen?
 - ◊ ¿Puede el vendedor instalar el paquete en un lugar remoto? Yo puedo tener un paquete pero estoy circunscrito a una zona para poder venderlo, o sea no puedo venderlo en Entre Ríos si estoy vendiendo en Santa Fe, porque Entre Ríos tiene otro vendedor. Entonces ¿puedo reinstalar un paquete que compré en Santa Fe en Entre Ríos? ¿o tengo que arreglar algo? ¿O tengo que arreglarme con el tipo de Entre Ríos que vende?, entonces tengo dos proveedores. Normalmente se hacen arreglos entre ellos
- h. **Otras garantías:** ◊ ¿El vendedor tiene el derecho absoluto sobre el paquete? En caso de que no, que no sea el productor del paquete o que no tenga derecho absoluto porque ya cedió algunos, ¿se puede examinar el contrato que lo autoriza a comercializarlo? Esto puede traer algún tipo de problemas y está vinculado con el siguiente ítem.
- ◊ ¿El vendedor indemnizará al comprador por infracciones cometidas por el uso del paquete? Por ejemplo violar el copyright, derecho de propiedad del paquete, etc. ¿Cuál es el riesgo acá? Por ejemplo yo tengo una empresa importante y necesito un paquete de manejo general de la empresa y consigue un vendedor, de un negocio pequeño que vende PC y dice “yo represento a la empresa de la que Ud. necesita este paquete” y entonces yo instalo el paquete de esa empresa. Si este vendedor no tiene el derecho de propiedad sobre eso o de venta, y no está autorizado por el fabricante a venderlo, el fabricante me a acusar a mi, yo soy el que lo estoy usando. Y yo iré después por repetición contra el que me lo vendió, pero si el que me lo vendió desaparece, porque es tan chiquito que no puede soportar el daño, el daño lo tengo que pagar yo, comprador del paquete. Es delicado. En este tipo de cosas, que antes todo era permitido, hay que tener cada vez más cuidado.
- i. **Contrato:** ◊ ¿Es estándar el contrato?
- ◊ ¿Se puede ver uno?
 - ◊ ¿Son negociables sus términos? Hay una ley de ética en EEUU que dice que no se pueden pagar los precios bajos las mismas condiciones de **comercio**. Todas las empresas norteamericanas, fundamentalmente europeas, tienen un contrato estándar, donde vos lo único que podés hacer es firmarlo. Se presentan situaciones especiales. Algunas empresas dicen que uno puede hacer un

convenio aparte del contrato, local, por ejemplo que se comprometen a dar más capacitación, pero sin tocar el contrato estándar. Es un acuerdo de menor nivel, de menor jerarquía normativa. Como se daban situaciones, que no es lo mismo vender una PC que vender mil PCs, o que la empresa solamente me compra PC a mi, entonces no tengo gasto de venta. Se crea lo que se llama el acuerdo corporativo, acuerdo de conveniencia tecnológica donde sí las condiciones varían porque no son las mismas condiciones de contratación para cada uno de esos acuerdos. Y en esos acuerdos, que se llaman muchas veces alianzas tecnológicas o acuerdos corporativos, se considera que ambas partes se ven beneficiadas con ese acuerdo.

No se puede normalizar no se pueden usar estándares, no se pueden racionalizar criterios de uso, por ejemplo reusabilidad, si no tenemos un entorno tecnológico unificado. Porque si todos los días estamos cambiando nunca vamos a poder hacer estas actividades.

3. **Benchmark del paquete de software**

El Benchmark, estrictamente hablando, es comparar con alguna situación real que se va a encontrar, o sea comparar ese software desde el punto de vista funcional. No es una demo que voy a hacer, sino que quiero comparar situaciones reales, comportamiento real. Hay varios tipos de pruebas que se pueden hacer a ese software. Cuando más pruebas le hagamos, mejor. Hay que hacer:

★ Pruebas funcionales: ver que cumpla con las funciones que yo le doy. O ver hasta donde me las cumple y hasta donde no.

★ Pruebas de tiempo o de stress: son pruebas de performance. No es lo mismo probar lo mismo con 10 usuarios cuando lo vaya a instalar tenga 200. Puede ser que con los 200 los tiempos de respuesta disminuyan de una manera que no lo pueda utilizar.

★ Pruebas de resistencia a errores: poner datos erróneos y comprobar que esos datos me los está detectando como erróneos.

★ Pruebas vinculadas a la seguridad.

La primera prueba y la más importante es la de funcionalidad, porque si no me sirve para que voy a hacer lo demás. Uno tiene que entrar con una prueba de funcionalidad. Lo primero que hay que hacer es una lista de controles, todo lo que hay que controlar con la prueba. Después hacer una lista de todas las transacciones que tiene el sistema. Después preparar datos para probar cada transacción y describir que resultados se esperan, o sea si yo pongo este input en la entrada que datos tengo que sacar. Para cada transacción colocar un porcentaje de frecuencia y una cantidad absoluta que se espera tener. ¿Esto por qué?, por que puede haber una transacción muy pesada pero que se ejecute una vez por mes, entonces no es lo mismo que la ejecute una vez por mes que 10 veces por día, o 100 veces por día. En el caso de 200 usuarios una transacción por usuario, la transacción demora 5 minutos, no me va a alcanzar el día. En cambio si demora 5 minutos una vez por mes no hay problema.

◆ ¿Cómo se diseñan las consultas Ad-Hoc? Éstas son consultas de **query by interface**, consultas que ocurren en ese momento, consultas no programadas.

◆ ¿Cómo es la administración o el mantenimiento de los archivos?

◆ ¿Qué grado de satisfacción tienen los informes?

◆ ¿Cuáles son las rutinas de cierre? (diario, semanal, mensual, etc.)

◆ ¿Cómo son los controles de los rastros para la auditoría? ¿Qué elementos tiene el auditor para ver si se está comportando bien el sistema o no?.

◆ Si se pasa la prueba funcional y la prueba de resistencia a errores hay que hacer la prueba de stress. Para esto hay software de simulación que permiten simular escenarios, y sino hay que meter a todo el mundo a probarlo. La mejor prueba de stress es la que se hace en forma real con la gente. La prueba de stress lo que hace es poner la carga real y anotar tiempos. Normalmente si hay un diario, de seguridad, tiene tiempos, tiempo que arrancó y tiempo en que terminó. Además hay monitores del sistema operativo que me pueden decir cómo está trabajando. Hay monitores de la propia base de datos. Hay monitores de comunicaciones para saber cuánto tiempo una transacción estuvo **in** dentro del sistema y cuanto tiempo estuvo en línea, tráfico. Porque a lo mejor se demora mucho pero puntualmente en algún lugar tengo un problema de línea y no de la transacción en sí.

b- Facilidades de uso: ◆ ¿Cómo son las instrucciones al operador? ¿Son fáciles las tareas que tiene que hacer el operador?

◆ ¿Cómo es el diálogo?

- ◊ ¿Cómo son los formatos de pantalla? Si son consistentes, si son claros y bien distribuidos. ¿Cómo usa el color? Recordar el clásico, verde amarillo, rojo. Y no usarlo al revés.
- ◊ Con respecto al hardware, si el teclado es familiar, si es un teclado estándar de uso en la Argentina.
- ◊ ¿Cómo es la calidad de despliegue del vídeo?
- ◊ ¿Cómo es el manejo de la impresión?
- ◊ ¿Cómo administra los archivos?
- ◊ ¿Cuáles son las condiciones ambientales?

c- ¿Como son los procedimientos de respaldo?

CLASE TEÓRICA DEL 29/10/01

Criterios de evaluación del software

El método para evaluación de software de Israel borovits tiene como base hacer un cuestionario de evaluación al usuario, un cuestionario de evaluación al vendedor y un benchmark al paquete. Con esos tres elementos se puede hacer una matriz de evaluación, como la siguiente:

Criterio	Ponderación	Paquete 1		Paquete 2	
		Valor	Resultado	Valor	Resultado
Funcionalidad	0.05	9	0.45	8	0.40
.....
Total	1				

La sumatoria de los valores de ponderación tiene que ser 1.

Valor va a estar entre 0 y 10.

Resultado = Ponderación * Valor

Son 11 criterios

- 1- Funcionalidad: ¿las transacciones de entrada/salida, archivo, informe contienen todos los datos necesarios? ¿Se efectúan todos los cálculos necesarios? ¿los tipos de consultas Ad-Hoc son los que se desean?
- 2- Capacidad: acá lo que se ve es un poco la escalabilidad, o sea que capacidad hay de niveles de ocurrencia, de cantidad de elementos, de volúmenes de transacciones por tiempo.
- 3- Flexibilidad: que grado de parametrización uno puede tener. Todo lo que no se puede poner por parámetro a un software hay que modificar el fuente, y requiere mucho más trabajo y mucho más riesgo porque hay que probar.
- 4- Uso: se refiere si se tiene capacidad técnica para manejarlo. Si los informes son fáciles de interpretar, o sea no llevan a inequívocos. Si bien uno podría estar en condiciones de manejarlo, si el entrenamiento que ofrece al usuario es adecuado, eso se ve cuando uno consulta usuarios, lo que le informa. Si los usuarios están de acuerdo con el producto, ¿en qué medida?
- 5- Confiabilidad: está dispuesta por la cantidad de errores encontrados después de entregado el software. Si su arquitectura es modular, porque puede aislar el problema. Está vinculado a qué partes pueden fallar si otras fallan. Si el usuario puede cometer errores en forma involuntaria que produzcan la caída del sistema. O sea el sistema no sea autocontenido en su seguridad. ¿Cuál es la capacidad de recuperación?
- 6- Seguridad: si hay backups que sean correctos. Si tiene manera de prevenir que por on-line se traigan datos incorrectos. Para esto hay software que lo hace. Se detecta que si vos desde una estación de trabajo estás haciendo cosas rutinarias, repetitivas podés estar robando los archivos. ¿Se posee un procedimiento estándar de prueba para verificar que funciona correctamente? O sea que yo pueda decir siempre que este archivo dé resultados correctos, resultado correcto es controlar la salida de ese archivo con otro archivo que ya lo tengo previamente hecho, nadie alteró nada. Es decir los resultados son correctos. Entonces uno corre ese archivo como cuando corre un scan para ver si hay algo. Otra forma que es recomendable utilizar, lamentablemente no se utiliza mucho y esto es debido a que la gente de auditoría no participa en el diseño del sistema entonces no deja huellas para después asegurarse el control, es lo que se llama la sucursal de auditoría. ¿Qué es la sucursal de auditoría? Se genera una

sucursal virtual, ficticia donde puede aparecer o no a solicitud del que tiene la entrada a la sucursal de auditoría, la password para entrar a la sucursal de auditoría, con información consolidada de todas las sucursales o tenerla aparte, y eso permite que uno pueda hacer movimientos con el software que está corriendo. Vamos a suponer que soy un auditor de un banco y llega una directiva del banco central que me obliga a modificar procedimientos, por ejemplo cuando salió el impuesto al cheque, entonces la gente de organización y métodos da directivas a todos los gerentes de las sucursales de cómo se va a aplicar eso, y da directivas a la parte de informática para que corrija todos los programas. En un momento dado la parte de informática le dice está ok. El auditor, en la forma tradicional, tiene un lote de pruebas y prueba que lo haga correctamente, a esos programas. Ahora, ¿qué pasa con el programa que está corriendo? Porque ese es un lote de prueba que pasó por otro lado. Pero no sé si es el programa que está corriendo operativamente todo los días. Entonces yo me garantizo que el procedimiento esté hecho porque hago movimientos en la sucursal de auditoría. En la sucursal de auditoría veo que los resultados sean correctos. Entonces me aseguro que no me alteran el software que está corriendo. Que la auditoría se haga efectivamente. Cuantas veces ha pasado que uno corrige un documento y lo corrige en la versión anterior, por esas magias que tienen uno para operar y el Office para guardar las cosas, y uno dice pero si yo lo corregí, pero acá no está corregido. Entonces a veces después que está aprobado todo se olvidan e instalan la versión vieja del programa o una inversión intermedia, una versión que estuvo en prueba y que no fue la definitiva.

- 7- Performance: trata de determinar la curva de degradación en ultima instancia. El tiempo de respuesta es el elemento más importante de performance. Uno tiene que hacer en este caso varios tipos de prueba para verificar. La mejor prueba de performance es la prueba de stress, de poner a carga máxima. Las bases de datos relacionales arman su relación en memoria, los lotes de prueba chicos no sirven. No es lo mismo probar con base de datos grandes que con base de datos pequeñas. No es lo mismo probar con muchas terminales y remotas que tengan que pasar por varios lugares para poder llegar a probar todo en forma local, los consumos son diferentes.
- 8- Mantenimiento: ¿se entregan los fuentes? ¿Qué nivel y calidad de mantenimiento suministra el vendedor? ¿la garantía es escrita? ¿El producto es transportable a otras plataformas? A veces no es que el productor sea tonto y no se de cuenta de hacer el producto transportable, sino que hace un acuerdo corporativo con el productor de un sistema operativo. Entonces si vos corres esto solamente bajo lo que es Microsoft, yo te doy todas estas ventajas y a tus clientes. Entonces no es que no sepa trabajar en ODBC o cualquier herramienta que permita comunicar distintos tipos de base de datos, distintos tipos de sistemas operativos, sino que lo tiene por convenio que no lo puede hacer, porque le conviene por otro lado. Si el mantenimiento está a cargo del comprador ¿se conocen el lenguaje y las técnicas de uso? Esto es muy importante porque por ahí en organizaciones que están acostumbradas a trabajar en forma estructurada el productor le hace algo en Java u orientado a objetos y no es solamente conocer Java, tiene que conocer y manejar la tecnología orientada a objetos.
- 9- Propiedad: ¿que clase de derechos adquiere? Si el producto es a medida ¿el vendedor puede venderlo a terceros? ¿pagará regalías? ¿Hay restricciones de copiado? Si el vendedor abandona el negocio, ¿el comprador tendrá derecho completo de propiedad por entrega de fuentes?
- 10- Minimización de costos operativos y de mantenimiento: ¿cuánto cobra por el mantenimiento? ¿cuál es el costo por transacción por el uso de este producto? Este es un dato muy interesante, porque aunque el producto sea caro por el número de transacciones que voy a hacer resulta barato el costo. ¿cuánto cuesta el costo hora de trabajo por los cambios a costa del comprador? Si yo le pido una mejora yo tengo dos variables en todo trabajo: la cantidad de horas y el costo de hora. Yo fijo el costo hora, la cantidad de horas no porque no se lo que le voy a pedir. ¿Por que es importante poner una de las dos variables? Porque entonces va a quedar como incógnita la cantidad de horas, y para la cantidad de horas uno puede ir a una institución para que evalúen si les parece correcto o no esa cantidad de horas. Lo que no puede hacer ninguna institución es evaluar los costos, porque si viene IBM y me dice mi hora es de \$200, que le voy a decir yo. ¿Cuál será el período de vida útil, el horizonte que va a tener el software?.
- 11- Minimización de costos de compra e instalación: ¿qué costos iniciales existen además de los básicos? Transporte, alojamiento del personal, etc. ¿cuál es el costo de la conversión? De un

viejo sistema a uno nuevo. El sistema puede salir barato pero los gastos de conversión pueden ser muy altos.



Entonces ¿como funciona la matriz? Yo pongo por ejemplo funcionalidad. La ponderación vamos a suponer que le dé 0.05. La nota que le puse (valor) entre 0 y 10, 9. y para el resultado, $9 * 0.05$ da 0.45 para el resultado. Y así se van poniendo los 11 criterios. La sumatoria de la ponderación tiene que dar 1. ¿Para que son importantes estas cosas?. Si bien los conceptos tienen un grado de subjetividad, los resultados son numéricos y son objetivos. Cuando se rinde una materia como historia es relativo porque claro hay que saber que fue lo mas importante que hizo San Martín pero no se puede saber toda la vida de San Martín, es imposible, habría que haberlo seguido, ver como comía, como dormía. Entonces es relativo la ponderación que le puede dar el profesor a la respuesta que le da uno en un examen de universidad. Es muy subjetivo. Y el tipo te da una nota después. De todas maneras no hay que olvidar que hay una comisión, un grupo que hace la evaluación, no es solamente gente de sistemas. Otro dato importante además que uno tenga un número es que no se maneja solamente por la interface. Porque todas las interfaces para vender son buenas. Entonces es natural que la interface sea grafica, hay que ver que lo que hay detrás sea bueno también.

Pruebas de rendimiento o técnicas de medición de performance

- a- **Cálculos de tiempo**: es una medida muy simple donde se estima la potencia de cálculos del equipo según los MHz que tenga el equipo y los ciclos de máquinas que utilizan las instrucciones. Cuando las máquinas son Risc los MHz se multiplican por 1,5 para hacerlos compatibles con las máquinas Cisc, las máquinas de Intel son las más conocidas. Para poder hacer esto correctamente lo que hay que ver es que ponderación de instrucciones uno tiene. O sea, ¿que significa esto?, por ejemplo ¿cuantas instrucciones de bifurcación promedio Ud. hace? Hay cómputos que están establecidos en tablas para todo lo que es por ejemplo para todo lo que es información administrativa o bancaria. Entonces dice cuántas bifurcaciones hace, cuántos movimientos de datos hace, o sea cuando mueve un dato de un lugar a otro. Entonces sabiendo cuantos ciclos de CPU hacen esto puedo determinar cual es mas rápido. Pero no es totalmente así. No es una medida buena. Pero a pesar de no ser una medida buena convengamos que tengo una máquina de 800 MHz, todos estamos usando esa medida. y hay que ver como son los anchos de paso, no es lo mismo 800 MHz a 64 bits que a 32, los búferes que está teniendo, la memoria caché, etc.
- b- **MIX**: estima la potencia tomando en forma ponderada la frecuencia y el tiempo de ejecución de las instrucciones. Estos MIX es lo que se conoce como MIPS (Millones de Instrucciones Por Segundos). La empresa de Benchmark que audita los MIX hace los siguiente, va a grandes clientes de una empresa determinada, le pide autorización y le deja un programa residente que lo que hace es que cada vez que se ejecuta un programa la instrucción o código de operación que se ejecuta se graba. Entonces yo se al cabo de un mes y para esa empresa planifico por código de operación, cual es la frecuencia del código de operación que utiliza esa empresa. Haciéndolo con un montón de empresas puedo decir que el promedio de las 200 grandes empresas de administración dentro del rubro que no sea científico, cálculo puro digamos, utilizan tal MIX de instrucciones por ejemplo. Entonces cada vez que una empresa lance un procesador compatible con el código de instrucciones que yo tengo de un MIX, porque no es lo mismo para IBM que para Intel que para Cisc o para máquinas Risc, es distinto, yo corro si la empresa quiere que yo le avale su performance corro el test y le digo esta máquina tiene tantos MIX, tantos MIX de ejecución.
- c- **Programas Kernell**: son programas tipos para medir el tiempo de ejecución sin considerar la entrada/salida. **Remito** tener datos sobre cambios de velocidad de algoritmo, funcionamiento de registros y direccionamiento. O sea mide la relación CPU y memoria. Este tipo de programas cuando yo tengo que hacer un Benchmark, una prueba de rendimiento al CPU, corro este tipo de programas. Otro programa es para medir la entrada/salida. La E/S se mide haciendo muchas operaciones sobre disco, o sea leyendo y grabando disco, leyendo y grabando disco, por ejemplo primero en forma secuencial y después del centro a la periferia. Y lo único que esto graba es la hora que toma el sistema, que está haciendo cada operación, para ver cual es el tiempo que está demorando. Primero corro un programa en memoria disco, stand-alone, solo, después un programa que mueve memoria, y después 5 programas que mueven el disco, o sea que tenga mucha E/S, después 5 programas que muevan memoria, que obligue ir a memoria virtual que no trabaje en memoria real, y después pongo los 10 programas, 5 veces el programa que utiliza disco y 5 veces el programa que utiliza memoria.

- d- **Modelos analíticos:** son interpretaciones matemáticas en realidad. Se basan en la teoría de colas. En los sistemas multiusuarios tenemos colas. Hay mucha gente que está demandando el mismo recurso. Se usan para dos cosas, para obtener datos de rendimientos esperados o para ver que debo cambiar de mis elementos para poder tener la máquina en equilibrio.
- e- **Benchmark:** Ismael Borovits lo define como conjunto de programas que se corren juntos como representación de la carga de trabajo esperada. Para Joslin, una demostración de validación Benchmark consiste en una corrida testigo de un grupo de programas representativos del usuario en un sistema ofertado por el vendedor para validar el tiempo de ejecución que aquel anuncia, considerando las habilidades del hardware y software para administrar la carga de trabajo en el orden y en los lenguajes en los cuales es probable que se procesen los programas reales. O sea lo que le dice es "bueno esta bien, ustedes los kernell lo ven, validan las versiones que les dice el vendedor", pero la carga de trabajo que yo espero para esta máquina es la adecuada, porque puede haber una carga típica particular del trabajo que no puede ser, por más que sea la más rápida para mover disco esa máquina y para mover memoria, no se adecua a la carga que yo tengo. El problema de estos programas cuando uno está en una organización estatal es que te pueden decir como usar un test de control, en vez en los kernell tiene la ventaja que vos le das el programa al tipo, los fuentes, y después le podes llevar el ejecutable, pero el tipo puede ver el programa, puede ver que no lo engañan, se lo haces lacrar, cuando vos los invitás a una reunión previa a todos los proveedores les hacés lacrar el programa, el programa acá está podes llevarte una copia y yo me llevo una copia. Esa es la única diferencia pero en la empresa privada el Benchmark anda muy bien como prueba.

Propósitos del Benchmark:

Eliminar la posibilidad de una compra por debajo del nivel necesario, o sea que la máquina sea suficiente, no tenga que estar ampliándola después de instalarla.

Probar la configuración para ver si es capaz de ejecutar la carga de trabajo en el tiempo esperado.

La forma de trabajar para armar el Benchmark es la siguiente:

1. Seleccionar una mezcla de jobs representativos de nuestra carga. Jobs son el conjunto de programas que se corren juntos, todo el batch que voy a tener.
 2. Presentación de esta carga de trabajo a los vendedores, para que vean que es lo que le vamos a probar.
 3. Que el proveedor diga el entorno exacto en que se va a realizar la prueba y en su caso el factor de conexión. ¿Esto que significa? Vamos a suponer SUN acaba de anunciar el E15000, el manual cuesta un millón de dólares. Ellos tienen el E10000 que es el modelo que estaba en uso hasta la semana pasada, ellos presentan una licitación que abre mañana para el E15000, yo quiero hacer una prueba de este tipo, entonces yo no quiero un E15000 en la Argentina, cuesta una fangote traerlo, hay que ver si los fabricantes le dan prioridad en la primer salida a la Argentina. Pero ellos pueden ofrecer la prueba en el E10000 y decirle el factor de corrección que Ud. tiene es 2, es esperar todo ese formato en la otra maquina. O viceversa tiene una máquina más grande y me da un factor de corrección que lo reduce. Porque a veces las configuraciones cuando uno se va del ambiente PC no hay muchas veces todo el material disponible en el país como para poder hacer la prueba, de lo que está anunciado.
 4. Hacer el Benchmark y tomar algunas mediciones:
 - a. Número de registros de entradas de cada tipo que hubo.
 - b. Número de registros de salidas que hubo.
 - c. Tiempo de respuesta.
 - d. Tiempo de CPU. Si bien el tiempo de respuesta es el tiempo Start-Stop, el tiempo desde que yo le di el enter hasta que me devolvió la respuesta, pueda saber cual es el tiempo de CPU que son nanosegundos.
 5. Evaluar los resultados.
- f- **Job Accounting:** es un registro de actividad del sistema que permite realizar la contabilidad del mismo (accounting). Pero a su vez ver su aprovechamiento o su uso. Job Accounting junto con los programas monitores son los que permiten ver las actividades, revisar el sistema. Pero además como vos tenés el total de uso de la maquina vos podes repartir costos. No sale nunca de ese circuito, está consumiendo recurso. Entonces vos monitoreas y ves que una actividad solamente está consumiendo CPU. Entonces vos hablas por teléfono al tipo y le decís "che que

programa es ese que tenés acá que lo único que hace es consumir CPU no hace entrada y no hace salidas". Entonces probablemente te dice "matalo", pero uno habla por las dudas de que no haya en algún momento que ocupar mucha CPU, por algún minuto por una cosa mal organizada. Pero además como vos tenés el total de la actividad que estuvo en la máquina vos podés repartir costos. Todo lo que no se ve, no se lleva a costos la gente no lo entiende. Cuando vos tenés que pagar algo la gente lo ve.

Evaluación de performance

En los centros de datos medianos y grandes hay una tarea permanente, que se hace en forma diaria, de evaluación de performance. Esto permite, la tarea de evaluación de performance, tomar conocimiento suficiente de lo que esta ocurriendo diariamente en el centro de datos, permite tomar las medidas correctivas en forma diaria, y permite el planeamiento de las capacidades, o sea hasta donde soy capaz de cumplir con lo que me están pidiendo. Me permite conocer hasta donde se esta cumpliendo un crecimiento vegetativo como es y me permite planificar las ampliaciones o cambios de tecnologías. La AFIP tiene en maquina central una inversión de alrededor de 10 millones de dólares, que no es una cosa que uno puede decir cambio de máquina y compro una mejor. La utilización de las máquinas depende de varias razones, depende de cómo está configurado el hardware y el software y depende de cómo es organizada o las características que tienen las aplicaciones que usa. Para poder hacer esta tarea es necesario tomar a través de monitores mediciones de lo que está ocurriendo en el sistema. El propósito de medir, cuando yo pongo un programa que mide, toma y guarda datos....yo tengo que considerar que esa inversión sea más económica que lo que voy a ahorrar modificando los parámetros de la máquina, optimizando mucho la máquina.

Matriz de evaluación de performance (análisis del apunte)

Celda de manejo de tiempo	Sistema (Entrada)	Operaciones (Proceso)	Aplicaciones (Salida)
Supervisor (0-6 meses)	Análisis de Utilización	Carga de trabajos	Monitores de programas
Operacional (6 meses-1 año)	Utilización análisis de productividad	Análisis del nivel de servicios	Perfil de la carga de trabajos
Táctico (1-2 años)	Distribución de costos	Perfil de la carga de trabajo	Planeamiento de recursos
Estratégico (2-5 años)	Planeamiento de recursos	Análisis de configuración	Distribución de costos

Si tomamos los nombres de las tres últimas columnas, sistema, operaciones y aplicaciones, se ve que hay actividades, a nivel sistema que pueden considerarse de entrada. O sea estas tres están vinculadas a como yo puedo tener al arrancar las operaciones diarias del sistema, las características con que yo arranco el sistema. A nivel de proceso tengo la carga de trabajos y a nivel de salida las aplicaciones. Este es un artículo norteamericano, ellos miden todo, y además que ellos no planifican para dentro de dos horas como nos acostumbra a hacer a los argentinos, sino que ellos planifican a corto, mediano, tercer plazo y cuarto plazo (esto sería largo plazo). Para ellos el corto plazo sería 6 meses, 6 meses para nosotros es una eternidad, no sabemos donde vamos a estar, ellos trabajan a nivel operacional, certifican un cambio de rutina a nivel operativo, lo trabajan a 1 año, entre 6 meses y un año de plazo. Lo que es el planeamiento táctico lo ven entre 1 y 2 años. Y el planeamiento estratégico lo ven entre 2 y 5 años. Esta es la forma de planificar de ellos. Entonces dentro de la matriz tenemos las diversas actividades que hay.

Si la vemos en el corto plazo, supervisor de 0 a 6 meses, a nivel de sistemas es el análisis de utilización, por ejemplo estoy ocupando la CPU en un 80%, un 50%, estoy ocupando los canales un 20%, tal disco lo estoy ocupando el 30%, tal disco el 50%, tal el 70% y que estoy optimizando en cada lugar, si es bueno o malo eso. Normalmente el objetivo de performance más claro que hay en lo que son máquinas grandes, porque ahora vuelven los mainframes, no con arquitectura libres, pero vuelven los mainframes, por ejemplo el E15000 es un semi mainframe, el E20000 va a ser un mainframe. Entonces en ese tipo de sistemas la idea es llevar las CPU al 100% de utilización. Esto es teoría de cola. Para poder llegar al 100% todos los recursos que me conducen a la CPU deben estar ocupados menos del 30. O sea el 30 es la medida óptima. Porque sino la CPU queda ociosa esperando que algún recurso sea liberado. Si está ocupado más del 30% hay que hacer un estudio para ver por que se está

reutilizando, ¿hay reintentos por errores?, el módem ¿se acomoda automáticamente bajando la velocidad en el caso de que esté haciendo reintentos? ¿no se acomoda en esa función? La otra cosa que puede pasar es que esté ocupada el 30% y esté trabajando, esta **haciendo trafico de registros**. Entonces entramos al nivel de operaciones. ¿Cuál es la carga de trabajos? ¿Qué estamos haciendo? Por ejemplo puede ser que estemos haciendo mucho servicio de FTP con muchos usuarios simultáneamente y archivos largos, entonces estamos haciendo mucho servicio de transferencia de archivos y la líneas se comen, entonces el tipo que quiera hacer transacciones y quiera hacer una apertura no puede. Le demora mucho tiempo. No hay que olvidar que los tiempos de Internet, Internet clásica, abierta con muchos dibujitos, son impracticables para una actividad productiva. Si una empresa tiene que esperar todo ese tiempo para trabajar ¿cuántos puestos tiene que tener?, un banco por ejemplo, si cada vez que tiene que pagar un cheque tiene que esperar que le venga la **parte** se va a pasar 10 minutos para pagar un cheque. No son tiempos aceptables. Entonces hay que ver muy bien que es lo que se manda, los gráficos cuestan mucho. Y a nivel de aplicaciones cuando veo que la carga de trabajo es mucha hay que ver los programas, que hacen los programas, monitoreo de los programas. En el monitoreo de los programas puedo tener un montón de indicios, por ejemplo ocupa mucha CPU, hace mucha E/S, me están transfiriendo muchos archivos, el monitoreo además me permite detectar cancelaciones, por ejemplo por que cancelo tal programa, o sea un programa puede hacer terminación normal o terminación anormal. Si la terminación es anormal la causa puede ser Machine check (error de máquina), puede ser Program check (error de programación), puede ser que ha sido abortado, alguien lo abortó. Además permite ver por qué programas en producción son cancelados, o el program check, están mal probados.

A nivel operacional uno puede, a nivel de entrada, uno puede hacer el análisis de productividad. Es bueno o malo el rendimiento que voy a tener de acá a 1 o 2 años. Es importante tener en cuenta que hay tres elementos de un sistema que tienen que estar en equilibrio, la memoria el disco y la CPU. Cuando una máquina empieza a escasear, primero es porque tiene problemas de disco, está esperando que le llegue la información. O tiene un problema de memoria, está paginando mucho, está haciendo mucho swap. O tiene un problema de CPU, tiene todo libre para ejecutar pero la CPU está al 101%. En este tipo de programas 101% significa que está al 100% pero tiene carga de trabajo y no puede contestar. Lo tiene en cola, 102 tiene más cola, o sea si está al 100% es óptimo porque la está usando a full pero si está más del 100% dice ojo tengo muchas cosas en la cola y no las puedo ejecutar, no es que estoy bien. Y normalmente cuando empieza a ir al 101, 102 o 103% empiezo a tener menos producción. Entonces análisis de productividad no hay. a nivel de operaciones a nivel de servicios, algo que se fija el usuario es el centro de datos. Armó un contrato de servicios con el usuario, en el cual yo digo que tipo de servicios le voy a dar, cual es el tiempo de respuesta, cuales son los volúmenes en las horas pico, cual es el trafico normal que voy a tener por transacción promedio. Entonces yo veo que estoy cumpliendo con el nivel de servicio que contraté con el usuario, y si el usuario se está sobrepasando porque me está consumiendo más, yo tengo que hablar con él para decirle que no está cumpliendo con lo que me prometió si me incrementa muchos los consumos yo no puedo brindarle el tiempo de respuesta pactado. Acá se da vuelta la bocha porque no es culpa de informática como siempre dicen. Y a nivel aplicaciones el perfil de la carga de trabajos. Esto está vinculado a que yo no puedo poner una carga de trabajo que compita con los mismos recursos. O sea yo tengo programas que utilizan mucho disco y los discos son los mismos no los puedo hacer correr juntos porque los discos son los mismos, programas batch. Y si son programas interactivos, estoy trabajando en forma interactiva tengo que separar los recursos, separar las bases de datos, ponerla en otro lugar, no puedo poner en el mismo disco porque están los dos compitiendo por lo mismo.

A nivel táctico. A nivel sistema es distribución de costos, ¿cuánto me sale por hoja impresa? ¿cuánto me sale por transacciones?. A nivel operaciones perfil de la carga de trabajos. Y a nivel aplicaciones planeamiento de recursos. ¿qué es planeamiento de recursos?, para hacer esta aplicación ¿qué recursos necesito? Esto está sistematizado, pero por ejemplo para nosotros hacer una aplicación de impuestos estamos planificando cuanto necesitamos para dentro de un año o dos porque ahí vamos a tener la aplicación, no la vamos a tener ahora. O sea estas actividades se hacen, yo ya las tengo sistematizadas.

A nivel estratégico, a nivel sistema el planeamiento de recursos completo ya, a nivel global. Si yo se que mi subsistema de impresión tiene tanto tiempo de vida, la cantidad de hojas que imprimen, yo puedo saber cuándo se me agota eso, cuándo tengo que salir a comprar. El ciclo de adquisición de compra de estos elementos no son de 4 horas como cuando uno compra una impresora para la casa. Los ciclos de adquisición son de 6 meses, un año o un año y medio. Depende de cómo sea la gestión

de compra. Una gestión de compra de un organismo internacional, desde que uno inicia el proyecto hasta que no cuando lo termina sino hasta que lo adjudica pueden pasar 4 años. No son ciclos cortos, uno tiene que imaginarse escenarios futuros. A nivel de operaciones el análisis de configuración o sea qué configuración voy a necesitar, y a nivel de aplicaciones la distribución de costos. La distribución de costos es lo más importante, un centro de datos debe de tener 0 costos, es un centro de servicios no puede tener costos, los que tienen que pagar son los que lo consumen. Entonces por ejemplo yo tengo una máquina y se comparte con la oficina de personal, con la contaduría, con 5 o 6 oficinas distintas. En la primera distribución de costos yo digo la oficina de personal me gasta el 10% de la máquina le cobro tantos pesos, la oficina de contaduría tantos, y las demás lo mismo. Pero en esta distribución de costos que hacemos al final la dirección de personal de la provincia es la que está gastando tanta plata. Y la dirección de personal de la provincia está gastando a su vez por cuenta y orden del ministerio de salud tanta plata, la única forma de hacer racionalización presupuestaria en informática es que cada uno pague lo que consume. Es muy difícil que se tome conciencia de esta distribución de costos.

