

◆ **Materia:**

Sistemas de Información III

◆ **Profesores:**

- Eduardo Iugman
- Mónica Vera

◆ **Regularidad:**

- 3 trabajos prácticos
- 80% de asistencia
- Un examen final.

DOBLE V  
<http://doblev.wordpress.com>

◆ **Objetivos de la materia:**

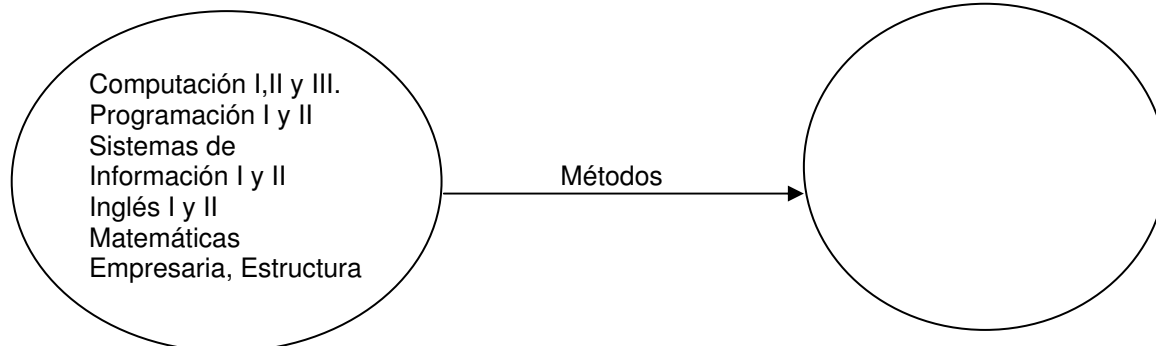
- Administrar recursos informáticos.
- Organizar y conducir centros de información.

**CLASE TEÓRICA DEL 09/04/01**

**Introducción**

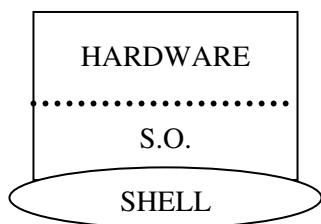
Dominio tecnológico

Dominio de la aplicación



Para pasar del dominio tecnológico al dominio de la información se utilizan diversos métodos. Uno de ellos consiste en dejar de lado la entrevista como herramienta de recolección de datos. La realidad está, solo hay que extraerla no inventarla (inventa el usuario o nosotros). Este método hace hincapié en la observación.

- ¿Cómo seleccionar elementos de hardware?
- ¿Cómo seleccionar elementos de software? (como S.O., aplicaciones y demás)

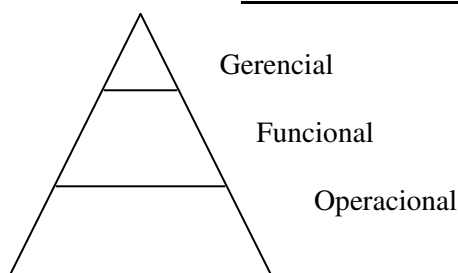


La relación con el SHELL siempre es la misma entre los diferentes S.O., lo que sí varía es la interrelación Hardware ↔ S.O.

**Evolución de los Sistemas de Información**

Teniendo en cuenta el tipo de información para tomar la decisión puede presentarse una jerarquía de la siguiente forma:

**Sistema de Información Operacional**



Operacional → Información → Dato  
 Datos simples: permiten operar.

Los primeros sistemas operacionales se pensaron para resolver problemas repetitivos. Por ejemplo un sistema de facturación, de liquidación de sueldos.

Control operacional/transaccional: si el programa falla se puede arreglar, detectar donde falla.

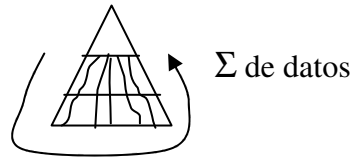
Se justificaba el uso de los sistemas operacionales por su bajo costo. Pero existían islas entre las áreas involucradas, no se comunicaban entre sí. Por otro lado la existencia de islas acotaba el problema.

**Sistema de información Funcional o Integrado**

Cubren un área vertical de la empresa. Brindan información para decisiones intermedias, permitiendo reducir costo. Responden al esquema de la organización. Los datos están integrados por sistemas, no existe redundancia de datos.

Los sistemas en administración se planifican de arriba hacia abajo y se implementan de abajo hacia arriba.

**Sistema de Información Gerencial**



Usan tecnología para incrementar los negocios y mejorarlos. La información está en función de los datos. La gerencia posiciona la empresa en el mercado.

Enfoque interfuncional: no solo generar datos internos, sino también recibir datos externos.

Los datos deben tener suficiente solidez para su uso. Si el grupo de poder de la organización no se capacita no conoce métodos científicos va a hacer más difícil la tarea.

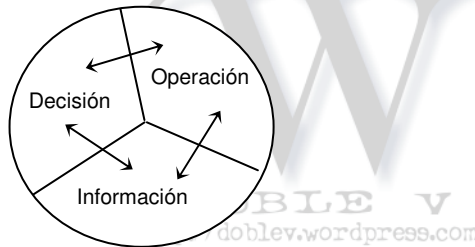


Como ejemplo de un buen uso de decisión gerencial podemos citar a Wal-Mart. La gerencia analizó durante 8 años las líneas de los tickets para identificar cuales eran los productos de mayor consumición en los EEUU. Vieron que los consumidores jóvenes adquirirían bastante pañales y cerveza. Entonces redistribuyeron estos productos en los extremos del supermercado de manera que tuvieran que pasar por varios productos antes de llegar al que buscaban. De esa manera se aseguraban que en el recorrido cargaran un mayor número de productos que el que tenían previsto comprar.

♦ **ERP:** proceso de recursos empresariales. Tener programas diferentes a los de la competencia. Patentamiento a nivel de ideas.

♦ **Sectores de la empresa**

En una empresa podemos encontrar tres sectores muy interrelacionados.



**Operación:** produce bienes y servicios.

**Decisión:** fija los objetivos, verificando que éstos se cumplan y verifica la interacción entre los que intervienen en los objetivos.

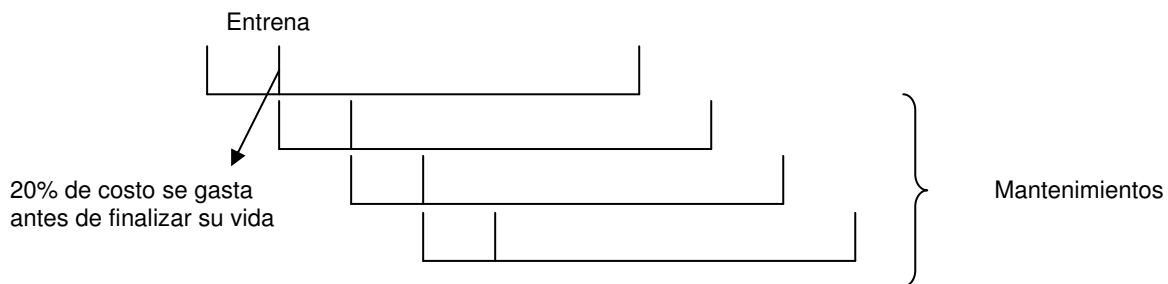
La existencia de los sectores decisión y operación es fundamental para el desarrollo de la empresa.

**Información:** antes no existía como sector de la empresa sino que estaba inmerso en los otros dos. Este sector depende de las necesidades, demanda. Su estructura depende de los demás sectores. En las empresas orientadas a negocios el gerente de información tiene una remuneración superior que el de venta.

♦ **Mantenimiento:** es un término mal utilizado. Actualmente a esta etapa se la denomina *Ingeniería de Cambio*.

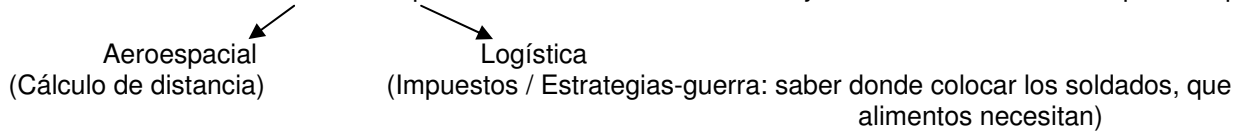
**Ciclo de vida del software**

Es parecido a los humanos. Al principio aparecen problemas, se van estabilizando, y después cuando no da más hay que cambiar todo. El sistema en algún lado muere.

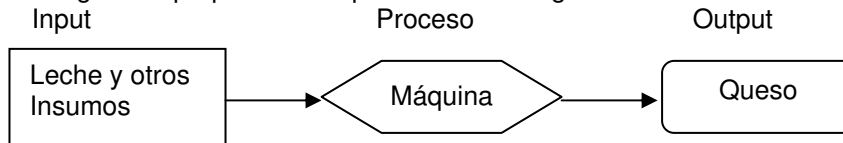


Al cabo de 2 años el encargado del sistema ocupó el 80% en mantenimientos. Cada persona cada tres años necesita el doble de líneas de código para hacer lo mismo que antes. Porque sube la demanda, se requiere mayor segmentación de la información, mayor segmentación del mercado. Por ejemplo por edades, por sexo, por clase económica hasta llegar a una persona exclusivamente.

En un inicio no existían especialistas en la materia. El trabajo se orientaba hacia dos tipos de aplicaciones:



Buscaron ingenieros para hacer este tipo de tareas. El ingeniero partía de lo físico. Por ejemplo fabricar queso. El ingeniero propuso un esquema como el siguiente:



Cuando al ingeniero se le solicitó la información de las facturas de gastos, y los desechos de los lácteos para reutilizarlo como insumo de otro producto, este dijo que como no se lo habían pedido no había previsto guardarlos.

La empresa tiene como Input objetos que son siempre los mismos. Lo que cambia es la visión de estos. Si un atributo de un objeto es carente de propiedad deberá preverse en el diseño de datos.

♦ **Elicitación:** no le puedo echar la culpa al usuario por un dato no entregado. Fallé al no solicitar el dato. El hombre es heurístico por naturaleza, no busca en todas partes sino en los lugares predeterminados. Debe tratar de buscar las situaciones menos previsible de ocurrencia. Por esto no se puede pedir todo el usuario, esto es un vicio oculto.

♦ **Cambios perfectivos:** cada 3 años.

♦ **Cambios mandatorios:** imposición de Cavallo para el uso del cheque. Se derivan de problemas de regulación de leyes.

♦ **Información centralizada**

Pierre Lhermitté, autor de "Cómo conducir un proyecto informático", es el impulsor de una metodología aprobada por el gobierno francés para presentar proyectos de informática.

Después de la segunda guerra mundial quedó poca población joven pensante. Reconstrucción de la población. Trabajaron en sistemas de inteligencia centralizada. En Francia generaron buenos sistemas de comunicación que sirvieron para implementar sistemas de inteligencia centralizada.

Sistemas fuertemente centralizados → Proyectos rigurosos.

El esquema metodológico de Pierre cubre toda la gama. La metodología es similar a la utilizada en Sistemas de Información I y II.

No se aprueba un proyecto tecnológico hasta no ver su rentabilidad después de 8 meses.

El ciclo para adquirir tecnología es mayor al ciclo de vida de la tecnología misma.

♦ **Metodología:**

- Guía para la acción.
- Serie de herramientas que se usan para resolver un problema. Métodos.

El uso de una metodología no es una panacea, no todo se resuelve. El problema de la metodología está en su implementación.

"La teoría es experiencia acumulada" (Einstein)

La metodología trata de cubrirnos de riesgos computacionales como:

- a. Existe dependencia con la empresa que provee servicios. No es fácil independizarse. No puedo tener compromiso con más de una empresa.
- b. inadecuación entre el organismo y la aplicación. El objetivo de la empresa es convencer que la organización se adecua al software.

Los costos de sistemas empresariales son elevados.

## CLASE PRACTICA DEL 16/04/01

### Metodología APX – Metodología de Evolución y Planeamiento de un Sistema Informático

Objetivos:

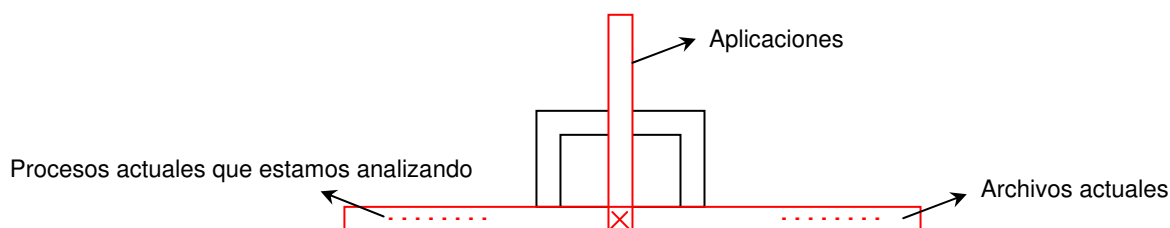
1. Definir la situación actual
2. configurar la situación deseada
3. Indicar el camino a seguir (para que la situación deseada se alcance).

Se logra:

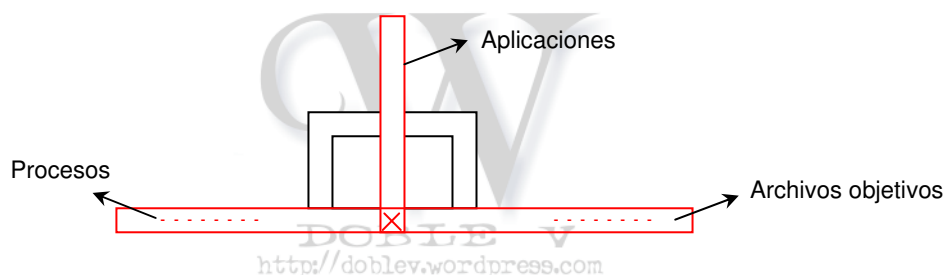
1. Identificar sus necesidades de información y programas relacionados.
2. Definir aplicaciones.
3. Definir archivos.
4. Definir planes de desarrollo.

El método APX está basado en el proceso que maneja las aplicaciones y los archivos.

Se parte de una matriz actual



Y se llega a una matriz objetivo



Se divide en tres fases

- Fase 0: Preparación (análisis de la situación de la empresa, cuáles son los problemas, cuáles son los procesos, etc.)
- Fase 1: Evaluación y Planeamiento.
- Fase 2: Validación.

Nosotros vamos a llegar hasta la fase 1.

Esta metodología se puede aplicar modularmente.

### **Fase 0 – Preparación**

1. Definición del ámbito APX (en qué áreas vamos a trabajar)
2. Formación de un grupo de trabajo (hasta 10 personas).
3. Definición de los procesos y sus necesidades de información (toda la información queda plasmada en las tablas APX).

## Definición de los procesos – APX1

<b>Área:</b>	<b>Responsable:</b>
<b>Proceso:</b>	<b>Fecha:</b>
<b>Hoja N°:</b>	
<b>Descripción del proceso:</b>	
<b>Subproceso (actividades):</b> _____ _____	

## Definición de las necesidades – APX2

<b>Área:</b>	<b>Responsable:</b>		
<b>Proceso:</b>	<b>Fecha:</b>		
<b>Hoja N°:</b>			
<b>Descripción del proceso:</b>			
<b>Información que genera</b>	<b>Enviada para</b>	<b>Frecuencia</b>	<b>Observación</b>

## APX3

<b>Área:</b>	<b>Responsable:</b>		
<b>Proceso:</b>	<b>Fecha:</b>		
<b>Hoja N°:</b>			
<b>Descripción del proceso:</b>			
<b>Información que recibe</b>	<b>Origen</b>	<b>Frecuencia</b>	<b>Observación</b>

## APX4

<b>Área:</b>	<b>Responsable:</b>		
<b>Proceso:</b>	<b>Fecha:</b>		
<b>Hoja N°:</b>			
<b>Descripción del proceso:</b>			
<b>Información necesaria</b>	<b>Posible origen</b>	<b>Frecuencia</b>	<b>Observación</b>

El APX1, APX2 y APX3 reflejan la situación real.

El APX4 refleja la necesidad de información → información deseada.

Relevamiento, aplicaciones y archivos actuales

<b>Área:</b>	<b>Responsable:</b>
<b>Proceso:</b>	<b>Fecha:</b>
<b>Hoja N°:</b>	
<b>Descripción del proceso:</b>	
<b>Aplicación usuarios</b>	(en desarrollo o ya implementados)
<b>Clase de datos</b>	<b>Fuente</b>

Conclusión de la Fase 0

Se produce junto a la determinación del objetivo.

Preparación de las prestaciones

- 1- Misión área
- 2- Objetivo del área a corto y mediano plazo
- 3- Procesos actuales ejecutados o no
- 4- Problemas y dificultades emprendidos
- 5- Necesidad de información
- 6- Expectativas relacionadas al proceso de datos.

Etapas del ciclo de vida de cada recursoEjemplo

Etapa de ciclo de vida general	Procesos del área finanzas	Procesos del área personal
Planeamiento	Presupuesto financiero	Planificación de ingreso Carrera
Entrada	Facturación / cobranzas	Reclutamiento / Selección
Desarrollo	Inversiones	Entrenamiento / Promoción
Uso / Distribución	Plan Operativo	Designación
Salida	Pagos	Renuncias / Muerte
Registro	Contabilidad	Registro personal
Control	Control Financiero	Control personal o evaluación de desempeño

<u>Área</u>	<u>Recurso</u>
1. Personal	Recursos Humanos
2. Finanzas	Dinero
3. Patrimonio	Bienes Físicos
4. Marketing	Mercado (cliente)
5. Legal	Conocimientos jurídicos
6. Producción	Stock / materia prima
7. Concepto	Información

Ejemplo: área o departamento Ventas, proceso clientes

APX1: ¿qué actividades realiza ese proceso?

APX2: ¿qué información genera (remito, factura, etc.)? ¿Se envía a las distintas áreas de la empresa?

APX3: ¿cuáles son los datos que se reciben?

APX4: reflejar lo que la persona nos pide (listado de clientes, nuevos productos)

APX5: si no tiene ningún equipo, ¿qué archivos se llevan manualmente?. Relevar cuántos equipos tienen.

## CLASE TEÓRICA DEL 23/04/01

Habíamos quedado la última clase, en la introducción de un tema, que es una metodología para conducir un proyecto informático.

Fundamentalmente lo que vimos antes de irnos, es porqué debemos usar una metodología. Entonces, vamos a empezar por la primer etapa.

### ♦ Primer etapa: “Estudio del proyecto”

Les quiero comentar que cuando a uno le cae un proyecto, es porque ya hay un diagnóstico previo, es porque ya el proyecto es viable. El estudio del proyecto, si ustedes pudieron hacer APX, vieron que para hacer APX hay que reunir determinadas condiciones, es decir, si la gente que tiene que tomar decisiones no participa hay que hacer otro tipo de análisis. Siempre hay que tener un anteproyecto, un estudio preliminar.

a) **Estudio preliminar:** se inicia con un contacto. ¿Qué es un contacto? que alguien convoca a un gerente de informática o a la gente que revisa la informática, y le dice: “mirá queremos desarrollar un proyecto de tales características”. ¿Qué puede ser de tales características? y puede ser “queremos cambiar todos los servicios”, estamos en un servicio habilitado por nosotros y queremos hacer al revés, o a la inversa. Puede ser que el proyecto sea menor, por ejemplo: seleccionar un motor de base de datos, “vos que me aconsejas”. Muchas veces, para esto se utiliza personal externo; cuando hay conflictos internos, se busca alguien externo para que asesore. O como pasó con los supermercados hace como 10 años cuando se popularizó el código de barras, todos los supermercados tuvieron que cambiar sus Input.

b) **Se presenta un anteproyecto:** esto es algo que voy a hacer yo, no es más que un APX. Pero si no podemos hacer un APX, lo mínimo que tenemos que tener para poder llegar a alguna propuesta, es un examen somero de las existencias, qué equipos tienen, cuáles son los déficit de las aplicaciones que ellos observan, cuáles son las necesidades que quieren cubrir que no están satisfechas. Y a partir de ahí, uno ya sí puede tener ideas de alternativas de solución. Nunca un proyecto técnico tiene una sola solución; lo que pasa que el límite es el costo de todo proyecto. Por lo tanto, se arman sugerencias sobre alternativas de solución; oportunidad que se puede realizar; ventajas y desventajas que se tiene en cada una con relación a las otras; qué facilidad técnica de poder utilizar los recursos locales, locales puede ser la empresa o la ciudad; y cuál es el orden de inversión en cada alternativa. Con esto se arma un informe escrito; se presenta a quién lo convocó; se lo explica someramente, qué características tiene el informe; y lo dejan que lo mastique ¿no cierto?. Yo les comento, para corralón Paraná demoré 1 mes en hacerlo. Puede ser que te llamen de nuevo o que no te llamen más; por ejemplo: los dueños de pinturerías Daneri, después de ver el informe decidieron comprarse auto. Lo que puede pasar es que le digan, vamos a hacer ésta solución; normalmente uno no le presenta una solución pura, normalmente uno le presenta una mezcla de dos o tres alternativas. Una vez hecho el estudio preliminar y aprobada la inversión, se comienza a definir el sistema.

c) **Definición del sistema:** cuando se define un sistema, lo primero que hay que fijar es el objetivo, decir qué voy a hacer y cómo. El qué voy a hacer implica hacer un relevamiento que no solo entra las aplicaciones en sí, sino la tecnología que se va a emplear, las metodologías de trabajo que se van a utilizar. El problema mayor de qué, en general es que hay que atender mucho a la técnica (como le decíamos antes) “relevamiento”, ahora se le dice “*licitación*”, técnica de poder rescatar la realidad. O sea, no necesariamente lo que te dicen cuando vos le preguntas a la persona que estás entrevistando, es lo que sucede. Puede que te diga lo que se le ocurre en el momento, cosas que él considera que se están haciendo mal pero que él considera que se deben hacer bien. Muchas veces en algunos lados tuvimos que poner gente de las oficinas a laburar, porque ven un montón de cosas que no se ven, que no te las cuentan por miña, por obvia. Además por lo general somos ambiguos para hablar.

Entonces lo que hacemos acá, tener un claro planteo del problema, análisis de existencia, poder hacer un censo en el sentido de tener la universalidad de las necesidades, no descartar cosas a priori, a veces el valor agregado puede ayudarnos a hacer otras cosas. Por ejemplo: ustedes quieren hacer un curso acá en Paraná con personas que vienen de buenos aires a darlo, le van a cobrar hasta 15 personas tanta plata, no te van a cobrar por uno. Entonces si vos tenés por 3, por 5 el curso, como valor agregado pueden invitar a una persona de lujo que le interese, y aunque no sea igual el costo del curso, ustedes están ganando igual dinero; es decir, no hay que descartar nada a priori. Si por ejemplo ustedes tienen que hacer un censo de establecimiento o de hogares, pedirles el documento, con eso tienen el cuil; si la casa está alquilada, preguntarle con qué inmobiliaria para ir a reclamarle a la inmobiliaria.

d) **Hacer un proyecto de definición:** después uno va definiendo en forma sucesiva, afinando lo que uno quiere digamos.

e) **Determinar las funciones a cumplir y la organización que se va a tener:** cómo va a funcionar la organización. Normalmente estos cambios sistémicos así de todos los servicios de la organización, implican cambios en la organización.

f) **Definir el sistema:** con ese % de definición se logra la afinación definitiva, se logra la descripción definitiva, está definido. Como tarea concomitante, nada más que por hacerlo metodológica lo digo fácil, hay que llevar de definición del sistema, es decir, documentar. La documentación, normalmente en organizaciones

importantes de informática, si no cumplís con hacerla concomitante te vas del grupo de trabajo. Hay un chico Osorio, que está haciendo la tesis en tecnológica, está trabajando en una empresa de comunicaciones de buenos aires, y ellos tienen para probar 5 tipos, con 5 pruebas distintas con documentación diferente, 5 no 1. Porque el riesgo de que salga mal. No piensen que la forma en que ustedes trabajan acá en los prácticos, es la forma en que se trabaja a nivel industrial; en las empresas chiquititas sí es más o menos a la que te criaste, pero hay normas internacionales de cómo se prueba. Uno de los libritos más interesantes es "El arte de probar el software" de Meier, éste trabajó para la IEEE haciendo normas sobre software. Lo que es importante distinguir, que no es lo mismo la tarea intelectual de definir que la tarea intelectual de documentar, es decir, si vos definís una porquería, vas a documentar una porquería. ¿Qué tiene que tener este documento como mínimo?

- **los objetivos retenidos**, o sea el límite del sistema, es decir, qué cosas contiene el sistema. Ustedes fíjense por ejemplo un sistema de ventas, de facturación; pero no es lo mismo que me llegue en forma electrónica el pedido de cotización, que yo lo contemple en forma electrónica al presupuesto a que me llegue en un papel y lo mando en una carta. No es lo mismo que yo contemple el pedido en forma electrónica y que yo le mando el pedido y la factura en forma electrónica y en papel, y él me paga automáticamente en una cuenta corriente. Entonces, hay que saber qué contiene el sistema, porque sino parece que todos los sistemas son iguales, y no lo son. Por ejemplo, en el corralón Paraná, compraron un sistema tipo tango de gestión y pensaron que con eso resolvían todos los problemas y no fue así porque por ejemplo el sistema tenía solo 91 tipos de formas de pagos y ellos ya tenían 120, entre otras cosas.
- **Cuáles son las opciones básicas seleccionadas:** una opción básica por ejemplo es el servidor de procesos, dónde está el servidor de aplicaciones. Una opción básica es tenerlo en la empresa y tratar de tener algunas grandes aplicaciones con servidores dedicados.

Otra opción básica es tener servidores con una muy buena red de comunicaciones, en distintas oficinas. Cuál es la ventaja y desventaja de cada una: la ventaja pareciera ser la disponibilidad porque lo tengo en cada oficina y la desventaja suena ser que si uno apaga la máquina y yo necesito entrar no se puede entrar, por supuesto, hay que tenerlo fuertemente comunicado.

En realidad al que le interesa donde está el servidor es al informático, al no informático le interesan los datos, la información oportunamente. En los años 80', como la comunicación era muy mala, la única opción era tenerlo cerca de la máquina.

Otra opción técnica, es juntarse un grupo de usuarios y compartir servidores. ¿Qué ventaja tiene esto? primero que normalmente arman otra empresa de informática, entonces administro informática y no que si yo estoy fabricando caramelos además tengo que administrar informática.

Otra opción es contratar a alguien, el servidor de aplicaciones. Este es uno de los negocios que más está aventándose. Hay empresas de primera línea que hacen servicios de servidor de aplicaciones, por ejemplo: IBM, Oracle, Sybase. ¿Qué ventajas tiene uno como empresario, como usuario tiene un servidor de aplicaciones? que no tiene que tener gente en la administración del servidor. ¿Qué significa administración? Explorer, administrador de base de datos, especialista en comunicaciones, etc. Y eso me lo arregla mi proveedor, o sea, mi servidor de aplicaciones. ¿Y a qué me dedico yo? justamente a usar con desarrolladores de sistemas, con ingenieros en sistemas, con especialistas en sistemas a hacer mejores aplicaciones para hacer mejores cosas. ¿Porqué es posible esto? por la tecnología de intranet e Internet, por la web. Y ¿qué tenés del otro lado? una terminal boba con entorno Windows. El primero que empezó a utilizar html fue Quilmes. Otra ventaja, es que no tienen que pensar en crecimiento, el crecimiento lo piensa el proveedor, se pacta. Hay una tendencia a un nuevo servicio, a que vos puedas tener el servicio por el tiempo de uso; esto qué significa, que a vos antes te comercializaban la base de datos con motor de base de datos, por ejemplo: ayer tuve una licitación y para el mismo producto uno ofreció \$94.000, otro \$135.000 y el otro \$450.000.

Otra opción técnica es para la captura de datos. Con captura de datos me refiero a abrir los archivos, no a mantenerlos. Por ejemplo: quiero digitalizar las actas del registro civil, y acá debe haber actas del siglo. Entonces, van a ser tomos bastante gruesos, manuscrito, etc. Bueno, yo voy a tener 2 sistemas, uno que vaya aquí para adelante y otro que vaya de acá para atrás. Ustedes piensen que a las personas las tenemos vivas, algunas por lo menos 90 y pico de años. Yo digo, "bueno, este es un proyecto técnico que va a estar funcionando de aquí hasta el 2100 más o menos", no va; y tampoco van a quedar los libros. Entonces, yo digo: "bueno, hago el escaneo yo". Estos motores que se pagan, dependen del tamaño, del rango con que se trabaja; no es lo mismo una base de 2 o 3 millones de registros a una tabla que tenga 10.000 registros. En este tema, la selección del software, porque en el hardware me preocupan dos cosas: si voy a trabajar con un sistema seguro o si voy a trabajar con 32 o 64 bits; con 32 bits, uno no puede ir más allá de 4GB de memoria real, con 64 te vas lejos. Y las bases de datos relacionales, hacen la relación a la memoria real, y la memoria real vs. Disco, es comparar nanosegundos con milisegundos, la diferencia es abismal y la memoria es cada vez más barata. Hoy una buena opción, es trabajar con Linux, ya se puede pensar en aplicaciones no críticas sino en aplicaciones que pueden quedarse a pata un rato, es seguro; pero si la aplicación no se

merece continuidad, linux es una buena opción técnica. El 80% de las personas que utilizan el Office, utilizan el Word y de ese 80%, el 90% lo utiliza reemplazando a una buena máquina de escribir. Cuando uno elige una herramienta de este tipo, no solo tiene que ver la herramienta sino que también debe mirar el proveedor porque vos vas a arrastrar el uso de la herramienta por varios años (o pretende arrastrarlo). Por ejemplo: "Tigres" era una base de datos relacional muy buena que hace 6, 7 años estaba en el mismo nivel que Oracle, dBase y el código fuente lo compró Computer Assoc. En 450 millones de dólares, saben qué hizo con ese código fuente, se le sentó encima, simplemente lo sacó del mercado. El mercado de tecnología es un mercado así. Toda la gilada confiaba en las empresas tecnológicas cómo crecían, pero cuando vos tenés que el patrimonio neto de una empresa vale 100 veces más que lo que tiene el átomo que tiene la empresa, o sea que está comprando posibilidad futura, y la posibilidad futura se puede conseguir de un 30, 40% pero no de 50, 60 veces más. Bueno, ustedes fíjense lo que está pasando en EE.UU., el índice tecnológico que agarra el conjunto de todas las empresas tecnológicas de informática, estaba en 4276 el año pasado y ahora está a menos de 2000, o sea, que esa plata se perdió o que nunca se ganó. Fíjense que no van a quedar todas las empresas de tecnologías; ahí hay una cosa curiosa, en EE.UU., la gente confía en sus empresas, los jubilados compran acciones, porque las acciones siempre le están dando más. Esa plata la perdió la gente, no la perdió Bill Gates. Entonces, qué les quiero decir con esto que hay que ver el proveedor, que ya no es una decisión técnica sino comercial, si yo veo que una empresa que tiene un excelente producto viene perdiendo los 3 últimos años acciones y sus competidoras van subiendo, no lo puedo comprar aunque sea el más barato. De qué me sirve comprar un producto si a los 2 años no va a estar más, por eso en este tipo de decisiones (no tanto con el hardware) hay que ver quién es el proveedor

- **Cuáles son las funciones a realizar**, no qué van a hacer las aplicaciones.
- **Qué organización se le va a dar al nuevo sistema.**
- **Cuál es el balance económico:** ¿qué es el balance económico? en la práctica es comparar el futuro con la gestión actual, cuándo voy a obtener beneficios. Estaba viendo en una encuesta, que de las 500 empresas más grandes del mundo de todo tipo desde las empresas financieras hasta grandes comercios minoristas, gastan el 2% de su presupuesto anual en tecnología, perdón de su ingreso anual.

g) **Plan de realización:** se hace una vez que el documento técnico es aprobado, el plan de realización es decir cómo vamos a hacerlo. Ustedes habrán visto técnicas de programación por camino crítico (CPM, PERT), a estas técnicas se les llama "plan director" o "planeamiento estratégico". ¿Cómo hacemos un plan director?

- Hacemos una lista de todas las grandes tareas
- Hacemos las precedencias
- Y lo más importante que debe figurar en un plan director, son medidas administrativas que se deben tomar, por ejemplo cambiar circuitos, liberar espacio.
- Los puntos de decisión durante la realización, un punto de decisión es por ejemplo: elegir un proveedor. Uno dice en el plan le puse 15 días, pero si al proveedor me lo eligen a los 4 meses, me voy a atrasar. Es muy importante que estén los puntos de decisión, dónde tiene que intervenir la autoridad que sino se atrasa y te retiran ellos, no por incumplimiento del camino crítico sino porque una tarea que no es crítica se convierte en crítica, por atraso, atraso de decisión no de elaboración.
- Qué resultados se espera por cada tarea, para poder corroborar el plan.
- Quiénes van a ser los responsables de las distintas tareas.

Hoy con el Project, esto se hace muy fácil.

h) **Estudiar qué materiales (hardware y software) voy a utilizar:** también hay que ver si puedo aprovechar de los sistemas existentes, cuánto tiempo van a convivir los sistemas. Las convivencias tecnológicas son muy difíciles de mantener. Este es un tema importante, porque cuando ustedes empiezan a preguntar por los productos que hay en las distintas empresas o en Internet, los tipos se empiezan a interesar en el posible negocio y entonces, nos dan un asesoramiento más personalizado porque ya ven la posibilidad de un negocio y se van a enterar de cosas que a lo mejor no sabían. No todo lo que creemos que es seguro es realmente seguro. También hay que estudiar cómo se va a desarrollar el software: si se va a comprar hecho o se va a desarrollar. Hoy a muchos softwares que cumplen una gran parte de las necesidades, sobre todo en organizaciones del segmento verticales; por ejemplo si yo estoy en un hospital, ni mamado desarrollo el software completo, porque hay un montón de software de hospitales. No me va a cubrir el 100% de las necesidades, bueno, eso que falta sí lo desarrollo. Y si se va a desarrollar el software, hay que ver si li va a hacer personal interno o externo. Como hay una tendencia general en las empresas en no acrecentar su personal, también la tendencia hace que se busque el desarrollo a nivel externo y el mantenimiento a nivel interno. Otro tema a tener en cuenta acá, es la formación de los recursos humanos intervinientes para la emisión del sistema, no para los usuarios. esta frase creo que hace más que nada a la ética profesional: "los usuarios deben definir el tipo de procesamiento a usar, los especialistas deben intervenir solo para las elecciones y las consecuencias técnicas de las elecciones", o sea, el sistema no es lo que el analista quiere sino el que el usuario quiere. Lo que uno tiene que indicar son las restricciones, ¿qué son las restricciones? una restricción es algo que la técnica del arte le dice que no se puede hacer, por ejemplo: todo dato debe

estar validado con el mismo criterio de validación que debe regir en ese sistema, o sea, que si vos tenés un motor de base de datos, metelo como gatillo al software ese en el motor, cosa que se dispare cada vez que se meta en ese campo. Esas son reglas del arte, no vayamos a que no se puede tener doble contabilidad. Otra cosa es la consecuencia técnica de las elecciones, si el usuario quiere para algo no en pantalla, quiere papel; el papel no es necesario siempre pero lo quiere, y bueno, no va contra la moral, va contra el gasto pero el gasto es de él. por eso, nunca tengan que juzgar a un analista de sistema por un sistema, porque puede ser que el mandato haya sido lo que haya hecho. Ustedes vieron que los arquitectos, si vos los dejás te hacen la casa que se imaginan y por ahí te pone escalones por todos lados y vos sos rengo. Entonces, la casa tiene que ser la que yo quiero, por supuesto que si yo le digo: "quiero un baño con salida a la calle", me tiene que decir no amigo, eso no se puede, está prohibido por el código de edificación. Ahora, si vos querés poner el lavadero en el fondo y la cocina en el frente, te va a decir: "bueno, compre una bicicleta para su mujer para que vaya de fondo a frente, porque funcionalmente la casa debe ser así"; en realidad así es mi casa, pero mi mujer nunca fue a lavar, por eso quería el lavadero bien lejos. Pero no me avisó el arquitecto eso, porque no le conviene, se desvaloriza la casa y sin embargo es uno de los arquitectos más conocidos de Santa Fé, entonces no tiene nada que ver juzgar a un profesional por una casa.

Una vez que el sistema está definido, que está probado, entramos en la segunda etapa.

#### ♦ **Segunda etapa: "Realización del sistema"**

Acá hay que adquirir el hardware y el software necesarios, la capacitación necesaria que se requiere, tener la definición de los sistemas hasta su puesta a punto; hay que instalar las cosas físicamente. Para poder hacer toda esta tarea en general, es conveniente crear un comité de coordinación, éste tiene el propósito de hacer el seguimiento del plan director, perdón del plan de realización que incluye el plan director. El comité de realización debe estar compuesto por el gerente o el representante máximo del área de sistemas y por lo menos 2 gerentes de las áreas que van a ser más impactadas por los sistemas. Es conveniente llevar registros, no por burocracia sino para poder hacer balances después, de lo que ocurrió. Hay proyectos que no se hacen en dos días o en seis meses, y la gente pierde la memoria. inclusive reunirse periódicamente o por lo menos reunirse a firmar el acta de que no hay novedades, para darle continuidad a eso. Reunirse al menos dos veces por mes o cuando sea necesario. Tiene que tener cierta capacidad de decisión, por ejemplo: si se atrasa una tarea en el programa, hay que decidir a nivel de ese comité, si se atrasa todo el programa o se hacen horas extras o se busca más personal o se busca otro camino.

El hacer un proyecto informático de estas características, trae problemas específicos de diferentes campos técnicos, no solamente del campo del desarrollo del sistema. Como administrador de un proyecto informático, uno tiene que conocer qué cosas están vinculadas a otros campos técnicos y hay que atender. A través de los cambios tecnológicos voy a hacer cambios en la organización, nadie va a hacer un pase de un producto a otro para que me dé la misma información. Uno aprovecha un cambio en las necesidades para hacer cambios tecnológicos. Cuando yo quiero hacer cambios, normalmente debo ver como debo acomodar ahora, estos cambios y su relación con el conjunto de cosas, cuál va a ser su ubicación en relación en la nueva organización en función de estos cambios, cuáles van a ser los nuevos roles, cuáles son los roles que se transforman. ¿esto qué significa? que yo voy a definir una nueva organización, porque no voy a poder cambiar funcionalidades importantes sino cambio la organización. Cuando uno quiere cambiar la organización o la organización presente que algo cambia, existe nueva distribución de responsabilidades, ¿qué significa nueva distribución de responsabilidades en blanco y negro?: nuevo organigrama, nuevos puestos de trabajo, puestos de trabajo que se funden con otros. Hay distribución de los espacios o cambios de circuito, aparecen nuevos manuales de función.

Bueno, ese es el mayor problema que hay para cuando uno está haciendo esto en una organización. ¿Cómo atacar o minimizar estos problemas?: primero, que exista un representante informático que en el período de concesión trabaja junto, cada aplicación debe tener uno, varias aplicaciones pueden tener el mismo, lo que además fija otros aspectos del proyecto, todos los aspectos del proyecto. Este representante, debe estar afectado con prioridad por lo menos, mientras se esté desarrollando el sistema. ¿qué tiene como función?: coordinar los trabajos preparatorios de los usuarios, normalmente hay preparar nuevas codificaciones, preparar carpetas. Vamos a suponer que estamos en el registro civil y queremos codificar los libros, hay que ver, hay que organizar a los usuarios cómo van a entregar los libros a la gente que va a hacer escaneo. Aprobar todos los diseños de interfaces, todo lo que sea pantallas, papeles, lo que sea; aprobar los controles y los procesos a hacerse por computadora; fijar la frecuencia o performance que requiere (si se quiere tiempo de respuesta menor a 1 segundo); fijar los circuitos internos que va a tener la información.

Cuando la aplicación ya está en marcha es necesario que por cada aplicación, haya un representante de *exportación*, también del usuario. Éste sería como una mesa de ayuda de primer nivel, o sea, vamos a suponer en una empresa chica, qué es lo que hace: es el que hace los procesos de reenganche, es el que hace los procesos de backup, además de estar trabajando en el área; conoce operaciones especiales, por ejemplo: si yo estoy trabajando en la base de datos y se me corta la luz por cualquier motivo, voy a tener un programa machetero para que me modifique los archivos; el que se pone en contacto con la gente de sistemas cuando el problema lo supera. El no tener esta persona, hace que la gente de sistemas se dedique en lugar de a programar sistemas, a resolver problemas de usuarios.

Entonces, qué otro problema debemos resolver, además de la elección de los representantes?: el otro problema que es muy delicado es el de la información y la formación de la organización. ¿qué significa esto?: que la gente debe saber lo que ocurre, eso significa primero. O sea, que hay que hacer reuniones para explicarle a la gente cuál es el alcance que tiene el proyecto, cuál es el objetivo que se logra alcanzar, explicarle a los niveles directivos (tal vez en reuniones separadas porque los intereses pueden ser distintos) qué objetivo se pretende cumplir, qué pasos hay que seguir necesariamente para llegar a ese objetivo. Si nosotros vamos a decir, por ejemplo: “vamos a mejorar los stocks”, porqué?: porque vamos a hacer modelos de rotación de stock para tener stock óptimo. Eso que por ahí se dijo de stock 0, es un verso porque o yo tengo el stock o lo tiene el otro; entonces hay que tener un stock mínimo, creo que el mejor ejemplo de esto es la farmacia, siempre tienen stock único y al otro día lo reponen; y a veces en farmacias grandes, reponen dos veces por día, al fin del mediodía y al fin de la tarde, entonces tienen stock mínimo pero con reposición permanente. Pero alguien lo está teniendo al stock, que es la droguería; sino, si la droguería trabajara con el laboratorio con stock mínimo, estaríamos todos perdidos.

¿Qué les quiero decir con esto?: para cualquier sistema informático, que quiera pasar algoritmo de optimización de stock, necesita tener información de consumo de por lo menos 2 años; si yo no tengo el stock histórico guardado tal cual lo quiero para el modelo, voy a estar dos años como mínimo para poder explotar esto.

¿Cuáles son los umbrales de rentabilidad?, ¿adónde esperamos encontrar los beneficios del sistema?, ¿cómo se va a rediseñar la nueva organización?, ¿quiénes van a participar?: la mucha confianza, necesita gente de la propia organización incluida en el equipo de desarrollo como es el representante informático que hablábamos, porque es un referente válido para el resto de los usuarios. Y en los niveles ejecutantes, ¿qué ventaja operativa va a tener como trabajo?, es decir, qué ventajas le va a dar aportarles funciones el nuevo sistema. ¿qué tareas preparatorias hay que tener para alcanzar esas etapas?: si yo por ejemplo a una empresa de venta por catálogo, le digo: “bueno, pero ahora ya las cartas van a venir por orden de cómo deben entrar en el correo, por código postal porque tengo domicilio estructurado”, le estoy diciendo las ventajas que voy a tener porque si al correo le doy el domicilio estructurado, le cobra más barato.

Hay dos mitos que hay que combatir:

- “Que la computación es igual a desempleo”: la computación no genera por sí desempleo, lo que yo sí he notado, es que por un lado disminuye el crecimiento vegetativo en empleados tipo de puesto, sobre todo en puestos que antes eran de hacer trabajo rutinario; y por el otro, crecen el nº de puestos en lo que es el soporte tecnológico de la empresa, no solo en el ámbito informático, sino gente que sepa analizar la información. En administración me refiero, en automatización no, en automatización es drástica la reducción de personal. Si por ejemplo, el personal de compañía telefónica EEUU, hubiese crecido proporcionalmente a la cantidad de líneas, hoy tendrían más empleados que la población activa de EEUU. Es decir, no creció en relación a la cantidad de líneas que administraban. Hoy la compañía de teléfono de cable tiene aprox. 1 empleado cada 300 líneas (promedio mundial), y en las que se están manejando por celulares, necesitan 1 empleado cada 7000 líneas. Eso en la administración es válido.
- “Que computación es igual a inteligencia”: en esto acá hay que hacer una distinción, hay dos tipos frente a la tecnología de humanos: el tipo que dice yo no sé nada así que hacémele vos, y el tipo que se cree que conoce todo y se imagina que un servidor de una empresa es lo mismo que la PC que tiene en la casa y dice es más grande así que puedo hacer más cosas y no es así. Por ejemplo: si uno quiere hacer puchero para 1 no es lo mismo que hacerlo para mil, hay que tener un equipo, todo sincronizado, etc., es totalmente diferente. Otro tema que está vinculado con el que se cree que sabe todo, es que porque trabajamos en computación, somos naturalmente inteligentes (aunque esto ahora ha disminuido porque cada vez hay más gente que trabaja en la rama) y un trabajador profesional como cualquier otro, que nos equivocamos como cualquier otro, es más que el método de trabajo de por prueba y error, no tenemos forma de comprobarlo, ni siquiera tenemos la seguridad de un ingeniero que hace el cálculo de una viga en una obra y después si el que hizo la obra puso lo que el dijo, salió bien.

Otro tema a tener en cuenta en esta etapa, es solucionar los siguientes problemas:

- la organización durante el desarrollo del proyecto son pasos y cambios: porque yo tengo que mover instalaciones, tengo que correr espacios, tengo que hacer tareas en paralelo. Lo que uno no sabe por ahí medir, cuando está definiendo algo, es el impacto que está teniendo, a veces uno lo minimiza y después tiene consecuencias graves en el momento de la implementación (ejemplo: la empresa donde tenían que cambiar de nº de proveedor a cuít, eran más de 10.000 carpetas y no podían ser empleados externos).
- dar un plan de prioridades para la generación de los archivos básicos y para las aplicaciones. O sea, no necesariamente yo para arrancar tengo que tener el conjunto de las aplicaciones o todas las funcionalidades de las aplicaciones. Y junto con esto, hay que considerar cuál va a ser el plan de migración. En sistemas grandes es muy costoso determinar el plan de migración. Por ejemplo, cuando la gestión anterior del gobierno, minimizó el problema de implementación del sistema de rentas acá en Entre Ríos, fue un fracaso total; minimizaron el problema de la migración y todavía están pagando las consecuencias, o minimizaron el problema de performance.
- Uno cuando define, tiene que definir una coherencia dentro de los materiales. Es decir, ustedes no pueden decidir trabajar con sistemas que no tienen soporte. Si definen una matriz donde ponen

procesador, sistema operativo, motor de base de datos, desarrollador procesador de base de datos, y si quieren trabajar con base de datos documentales lo ponen, si quieren trabajar con Datawarehouse lo ponen. No cierra todo con todo, no todos los productos datawarehouse andan en todas las máquinas ni van con todos los sistemas operativos ni son generados por todos los desarrolladores. Entonces, esa matriz es importante cuanto más compleja es la intensidad del producto, es muy importante para tener coherencia con los materiales. Además se da un problema muy grande en esto que no se detecta fácilmente, que se llama en la máquina: "niveles de ingeniería", tanto de hardware como de software, porque un nivel de ingeniería solamente es utilizable en un nivel de tecnología (para arriba se entiende) de hardware; pero a veces hay niveles que no son soportados. Vamos a suponer yo hoy puedo tener, hay, existen, sistemas de backup que yo no puedo poner en una máquina por más que sea Intel o Risc del año 95/96, no lo va a soportar por el nivel de ingeniería de la máquina; y no se puede actualizar ese nivel de ingeniería porque la máquina por ser barata, no son de niveles de ingeniería regrabables digamos. Hay versiones de S.O. que no soportan máquinas de hace 4-5 años, no por la capacidad sino por diferencias entre ingeniería del hardware e ingeniería del software. No se olviden que entre el hardware y el software, está el firmware, la lógica fina que le llaman, que es lo que vincula estas dos cosas. Entonces, qué les quiero decir con esto, como cuando uno decide un sistema informático decide tenerlo por ver unos años para poder utilizarlo, se le puede hacer dificultoso ampliarlo en el futuro, porque el negocio de informática de este tipo, pasa que el cliente cambia la máquina lo más rápido posible y el negocio de proveedor de la empresa es poder utilizar su inversión el mayor tiempo posible porque sino está todos los años invirtiendo en tecnología, siempre invirtiendo nunca está ganando. Entonces qué pasa, uno dice: "esta máquina está prevista aquel horizonte". Todo esto es muy importante tenerlo documentado porque de acuerdo a qué sale en cuatro años, empiezan los lamentos.

- La recolección de los datos: qué significa esto?: los datos se generan en un punto que es distinto al punto donde tienen que procesarlo normalmente. Una maestra de Victoria trabaja en victoria, y ahí en la escuela esa se enteraron que faltó, no se enteraron en el ministerio. Entonces, dónde recojo el dato, ¿en la escuela? O lo mando por papel a consejo de educación para que ellos lo procesen. Y el problema de la distancia en la captura de datos, es un problema serio. Por ejemplo: a nivel de todo lo que es medición domiciliaria (gas, luz), lo que trataron las empresas, lo más rápidamente que llegue desde el dato que tengo yo en el medidor a la factura. Entonces grandes consumidores, lo primero que le pusieron fue un PC conectada por red de MODEM con uzina. Entonces, los tipos chupan los medidores esta noche y ya mañana está saliendo la factura. Con esto se están ahorrando un promedio de 45-50 facturas por día, ganaron plata. Antes demoraban entre 35 y 40 días desde que tomaban el medidor hasta que facturaban. Usan la informática para hacer negocio, no dijeron cuánto me cuestan los medidores, sino cuánto ganamos poniendo esos medidores. Otra cosa que hicieron, computadoras de palma en otros lugares más baratos, donde se tiene el consumo anterior, se le coloca consumo, lo obliga a pasar por cada medidor, lo tiene controlado o sea que le fija recorrido, le marca la hora que marca el medidor para saber que no se quedó en el villar como hacían antes y después llenaban el consumo con el consumo promedio que venía. Y lo tipos llegan a la oficina de energía, chupan los datos y ya pueden facturar por lo menos ese sector y tienen el seguimiento del empleado. Con esto quiero decirles que en tecnología el límite lo da el costo, todo es posible.
- Un problema que se daba, era el problema con el modelo computacional, esto se da en organizaciones grandes con distribución geográfica, no se da en organizaciones chicas. En las organizaciones chicas suele ser centralizado, es el mismo el que programa, procesa, releva, opera, etc. Pero nosotros tenemos recursos informáticos: planeamiento, datos, desarrollos, proceso y existían 3 modelos: uno es un modelo fácil, centralizado, donde al ser centralizado en un 100%, está todo en un solo punto. Otro esquema es el descentralizado, éste es un modelo claro también, cada uno se hace responsable de la informática, yo puedo decir: "la informática la descentralizo a nivel de repartición", que en realidad yo a esto lo llamo modelo desparramado. Por último, está el modelo distribuido, que no es un modelo puro, es un modelo de compromiso: qué distribuyo, cuánto distribuyo. Por ejemplo: en Entre Ríos, igual que Santa Fe, el desarrollo lo tiene bastante distribuido, por ejemplo catastro tiene su gente, salud tiene su gente de desarrollo de sistemas. Y otro porcentaje lo tiene centralizado que es lo que conocemos normalmente como la dirección de informática creo que le llaman acá. Yo puedo distribuir datos, decir: "estos datos son críticos y lo tengo en forma central, y estos datos no son críticos porque no se utilizan en forma compartida, entonces lo tengo en forma descentralizada, en forma distribuida". Si alguna vez quiero esos datos, lo que hago es chuparlos y manejarlos con alguna planilla de cálculo, lo que sea. Proceso es lo mismo; pero ojo, proceso distribuido no significa tener máquinas desparramadas, proceso distribuido significa que yo proceso en varios procesadores, Internet por ejemplo, Internet es un proceso distribuido, yo tengo un solo logon, es decir, me presento una sola vez frente a Internet y sin salir de Internet puedo trabajar con muchos procesadores distintos de diferentes lugares. O sea, si yo no tengo las cosas, si todas las máquinas que tengo distribuidas, no las tengo unidas por algún fuerte lazo de comunicación, para poder hacer aplicaciones entre las diferentes máquinas, no tengo proceso distribuido, tengo máquinas

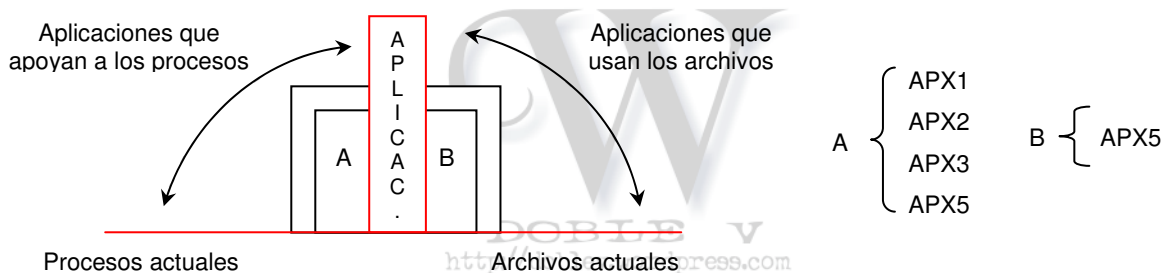
desparramadas. Normalmente, en ciertas organizaciones, para que el otro se entere lo que yo tengo, lo dificulto porque no le puedo decir que no quiero darle la información porque tengo obligación de dársela, pero se la dificulto, que me la pida a mí. Y planeamiento, el planeamiento en general se hace distribuido, yo no le puedo decir al gerente de ventas qué cosas son prioritarias para él, pero yo puedo hacer un planeamiento centralizado, en todo lo que es infraestructura: mesa de ayuda, manejo del correo electrónico, documentación interna de la empresa que hace a la normativa de la empresa; eso es común para todos, entonces fijo los sistemas que son comunes para todos y los administro en forma centralizada, el sistema de personal es un sistema común para todos. Entonces, esto es un modelo de compromiso, hoy con Internet, con las applets, el proceso vuelve a entrar a que el déficit sea 100% centralizado, tener una máquina tipo terminal boba con Windows como les decía hace un rato. El planeamiento normalmente distribuido; los datos tienden a ser centralizados porque son mejor administrados los sistemas de seguridad, mantenimiento de backup; y el desarrollo tampoco tiende a ser centralizado.

**CLASE PRACTICA DEL 30/04/01**

**Fase 1 – Evaluación y planeamiento**

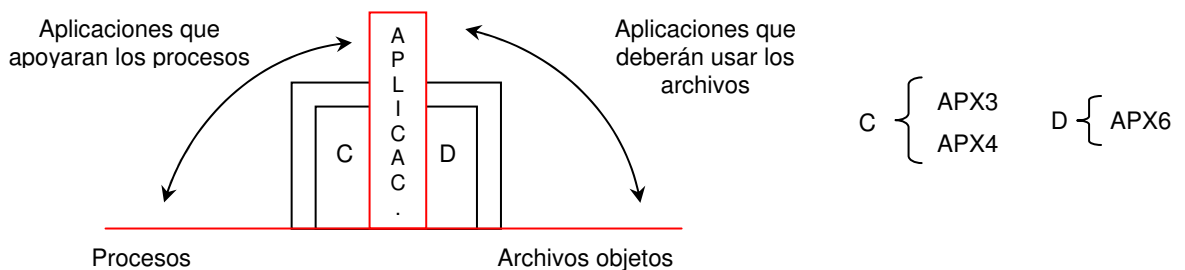
Evaluación de la fase 0. Sus etapas son:

1. Presentación: se vuelca toda la información recabada y se detallan los problemas y las dificultades encontradas en cada área.
2. Construcción de la matriz actual: en ella hay que colocar la información de los procesos, aplicaciones actuales de la empresa y los archivos actuales. Evaluación del sistema que posee la empresa (en caso de que posea uno).
3. Análisis de la situación actual: detalle de la situación crítica. Archivos con datos redundantes. Plantear la



crítica de que los datos sean redundantes, en cualquier tipo de sistema, manual o computarizado.

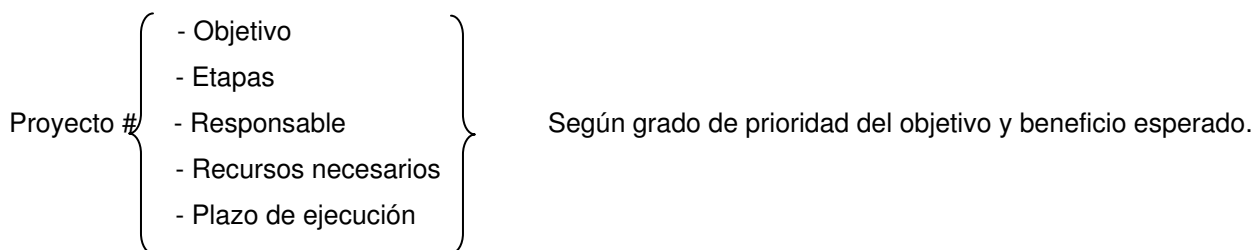
4. Construcción de la matriz objetivo: posee los mismos procesos que la actual, cambia los archivos, plantea nuevas aplicaciones.



APX6

<b>Fuente de Información:</b>	
<b>Fecha:</b>	
<b>Referencia</b>	<b>Área/Proceso</b>

5. Recomendaciones: el grupo de trabajo plantea las estrategias a llevar a cabo.
6. Plan de acción: definir como llevar a cabo la transformación dada en las recomendaciones. Para eso se definen varios proyectos, en los cuales hay que detallar:



Aquí concluye el APX del trabajo en grupo. Se presenta a la gerencia para que tome la decisión de aceptar o no el análisis.

## CLASE TEÓRICA DEL 07/05/01

La última clase estábamos viendo la serie de problemas que se presentan para desarrollar problemas en informática, el último que vimos fue: ¿qué organización se iba a dar a los servicios de informática?: vimos modelo centralizado, descentralizado y modelo distribuido.

Habíamos dicho que no solamente es importante dónde están las máquinas sino también, la planificación, los datos y el desarrollo del sistema. No solamente organización.

Otro problema que hay que resolver es la relación y los llamados a licitación, o sea, los pedidos formales de propuesta. Siempre que uno no compre algo estándar, (una impresora epon, una PC Pentium) o sea, algo que salga de lo estándar y que además tengo impacto en la organización; hay que poner en blanco y negro lo que se quiere. Algunas organizaciones, sobre todo las organizaciones más grandes, tanto grandes empresas como el estado, tienen normas de cómo se debe pedir. En las organizaciones más chicas, se tiende a un tome y daca, entonces no están tan habituados a redactar lo que se quiere.

Porqué debemos redactar lo que se quiere, por dos cosas: para que todos los proveedores tengan exactamente la misma información; y segundo para eliminar estos ( ) porque esto está escrito entonces el proveedor sabe lo que queremos y va a saber qué hacer.

En general existen dos formas de encarar la cosa (cómo redactar): una es decir cuál es el problema y que los posibles proveedores presenten soluciones. Eso es muy práctico a nivel de empresas privadas porque si la solución no te importa por más que sea barata vos la tirás; dar algunas pautas de lo que se quiere pero no dar arquitectura, salvo que se tenga muy en claro la parte de la arquitectura. O sea que le damos creatividad a los proveedores para que ellos presenten soluciones. En el estado se hace muy difícil hacer esto, porque usted quería eso porqué no me lo dijo, en cambio en una empresa privada no me gustó lo que presentaste y chau.

La otra es definir con la mayor precisión posible lo que se quiere, con el riesgo de que uno tenga un solo proveedor después, porque todos saben que las máquinas son parecidas pero no son iguales. Cuando uno quiere definir cómo quiere la relación de un motor de base de datos con un sistema operativo o define el conjunto vacío o está definiendo uno en particular, se puede decir en general cómo quiere la relación pero no en particular. Entonces, dejo un margen de antigüedad para poder elegir, pero acota los posibles proveedores.

Lo que no debe faltar en un pliego citatorio, es la presentación de la organización:

- quienes son;
- qué son;
- cómo están distribuidos geográficamente, si tienen sucursales o si no;
- qué actividad realizan;
- cómo es la situación informática de la organización;
- qué se quiere realizar: describir las aplicaciones no a nivel de carpeta de sistemas sino decir cuál es el problema general de la aplicación;
- qué cosas indicativamente tendrían que tener las bases de datos: indicativamente no físicamente por ejemplo: datos de la familia, se entiende que la familia son por lo menos primer

- grado si son de segundo o tercer grado ya es otra relación. O domicilio, no hace falta decir el domicilio estructurado: casa, barrio, n<sup>o</sup>, etc.;
- volumen de la información;
  - actividad de la información por períodos de tiempo;
  - horas de presencia en línea de la aplicación: no es lo mismo tener la aplicación corriendo las 24hs. del día los 7 días de la semana que tenerla andando en horario de comercio.
  - Los picos de procesamiento que tengan: estos picos son por un lado cantidad de transacciones en una hora pico, o sea, cuántas transacciones voy a atender en una hora pico. O sea, yo diseño un sistema para que el tiempo de respuesta, el tiempo de performance que se requiere sea menor a 3 segundos. Después el usuario me dice: "pero esto es muy lento", y qué culpa tengo yo, si al comienzo se cumplen con la performance de 50 facturas, si después aparecen 80 facturas para la empresa mejor pero no es que el sistema ande mal;
  - qué horizonte de planeamiento se requiere al sistema: es decir, de acuerdo al volumen y a los picos, yo quiero tener sistemas sin tocar por 4 años por ejemplo. Acuérdense que los sistemas no son exactamente modulares por definición, tienen una capacidad de creación en el origen, pero después puede pasar por la evolución de la tecnología y deje de fabricarse esa máquina y después no encuentren cosas para ampliar. Me pasó, tuve que recurrir al mercado secundario de Miami para ampliar un Workstation, y en Argentina no había y era IBM, IBM no tenía más repuesto para nada, entonces como en EEUU hay un buen mercado secundario justamente para el canibalismo, es la única forma que tuve de poder conseguir repuesto. Nunca se va a cubrir en un banco, no hay con qué darle a las dos últimas horas del último día hábil del año, donde cierra todo, todo el mundo está apurado por cerrar sus transacciones porque después se van de vacaciones, porque pagan impuestos, porque cobran, porque tienen balances; no hay con qué darle a eso, tenés que quedarte tranquilito. Pero sí, los bancos cómo actúan: (vieron que los bancos ahora tienen fila única) arrancan con una caja o varias de acuerdo al volumen, y a medida que la gente va llegando, hay gente que va dejando las tareas atrás para pasar al frente, entonces se habilitan más cajas, el tema es que tengas espacio. El último día hábil de la semana también es un pico, el último día hábil del mes también es un pico. Esto es distinto a lo que pasa en las secciones de alumado, donde la cantidad de personas que atiende es estable, entonces, o no hacen nada o están locos por una cola que tiende a infinito que son ustedes.
  - Los picos de impresión: son importantes en tareas de tipo batch, o sea, yo quiero hacer una emisión de impuestos que son 3 o 4 boletitas de 3 o 4 pulgadas cada una, en qué tiempo quiero que se haga, eso lo define su sistema de impresión.
  - Indicaciones generales sobre cómo quiero que trabaje la gente: porque una cosa es que quiera un puesto de trabajo por persona y otra cosa es que diga los puestos de trabajo se comparten hasta tantas personas salvo tal, tal y tal que tienen un puesto de trabajo particular. Hoy está muy de moda en las empresas medianas, el workflow: la automatización del flujo de trabajo, pero las empresa que tienen workflow, tienen una pc por empleado, no se consigue de otra manera; entonces, puede haber objetivos que si no se dan cuenta lo que significa tecnológicamente, no se pueden cumplir por la inversión que tienen que hacer.

Una vez que esto está escrito, las empresas pueden pedir aclaraciones, normalmente no se pone la aclaración hasta tantos días antes se van a empezar a analizar las propuestas económicas. Esas aclaraciones son contestadas por escrito y son circularizadas a todos los que accedieron al pliego, para que todos tengan la misma información (por ejemplo en la licitación que estamos haciendo que yo les dije, las empresas pueden pedir aclaraciones hasta 20 días antes y nosotros le podemos contestar hasta 15 días antes).

Lo que conviene hacer siempre, son dos sobres: un sobre que se proyecte lo técnico y otro sobre cerrado con la oferta económica. ¿Porqué esto?: para que no nos veamos asociados al momento de analizar las propuestas por un precio, entonces dicen "el mejor es éste".

Hay dos formas de evaluar la propuesta:

- una es pasa o no pasa, si no pasa se descarta y no se abre el sobre económico, se lo devuelve entonces no sé si era caro o barato, no me importa, justamente para no dejarme influenciar.
- La otra es calificar, hacer un check list y decir esto vale hasta tantos puntos, esto vale hasta tantos puntos. Y pueden decir "el que saca 0 en alguno de los rubros queda afuera"; pueden decir también que además de no tener 0 en ningún rubro, tiene que sumar por lo menos el 70%. Y la otra cosa que está vinculado a la clasificación, que también se puede hacer, es darle un valor a la oferta técnica y

otro valor a la oferta económica, por ejemplo: una oferta que tenga 100 puntos contra una oferta que tenga 70 y una diferencia de precio del 2%, no se justifica darle al más bajo, entonces uno hace un mix. En el pliego licitatorio esto debe figurar.

Esto acá, todo lo que se ofrezca en el período de negociación (cuando uno ya eligió un proveedor y está tratando con él cara a cara y empieza a negociar cosas), por ejemplo mayor capacitación, vamos a suponer que el proveedor es el instituto, yo sé que le puedo pedir más capacitación porque tiene toda una estructura de capacitación montada: tiene aulas, tiene docentes; lo que no le puedo pedir al instituto como proveedor, es que me dé más máquinas, porque eso la va a tener que salir a comprar y la terceriza, eso le va a costar. Y puede pasar al revés, yo le puedo sacar ventajas de software a IBM pero no le puedo pedir capacitación porque le va a salir muy caro la persona al empleado. Lo importante de esto, es que debe plasmarse en un contrato, si no se plasma en un contrato todo lo asociado, negociado, acordado, fijado y que parte del contrato sean especificaciones técnicas o sea el pliego licitatorio; después se define por el código de comercio, que no es exactamente lo mismo que tenemos nosotros. Ustedes tienen que pedir para lo que es hardware por lo menos 3 años de garantía, para lo que es software 1 año por lo menos que incluya los servicios técnicos (upgrade, update, etc.). Es muy importante también colocar en el contrato, cuánto va a ser el costo a futuro, el costo a futuro es cuánto como máximo voy a pagar las futuras actualizaciones los próximos 3-4 años. Yo les voy a contar una anécdota (triste pero sirve), estábamos por contratar un sistema con una empresa chica y a mí me quedó la duda de qué pasaba si la empresa desaparecía, entonces le hicimos firmar un contrato por el proveedor de ellos de que subrogaban en caso de que la empresa desapareciera. Un día los dueños de esta empresa fueron a hacer un trabajo a otro lado y a la vuelta por no chocarse una bicicleta se chocaron un camión, fallecieron los dos y la empresa se disolvió. Cuando tuvimos problemas, llamamos al proveedor y no tuvo ningún problema, al año nos querían cobrar una cuota de mantenimiento más cara pero yo les mostré el contrato y se tuvieron que quedar en el molde. Por eso hay que escribir el contrato y que lo firme un abogado porque cuando viene el juicio, el que va a litigio es el abogado no somos nosotros, por eso todas las empresas tienen un asesoramiento jurídico por algún lado, el abogado puede poner cláusulas en el contrato que sirven después en el alegato de su defensa. Hay algo que ya me lo sé de memoria que es algo de la esencialidad de todas las cláusulas para que no venga después que hay cosas que son importantes y cosas que no son importantes y como no son importantes le podemos dar otro tratamiento. Entonces, al darle esencialidad a todas las cláusulas, el cumplimiento de una implica el cumplimiento de todas, yo le puedo ejecutar el contrato. En informática es muy difícil que un juez pueda dirimir qué cosas son esenciales o no y esa es a estrategia de los abogados.

Lo importante es comprar productos estándar, lo más estándar posible. No necesariamente el estándar es el mejor producto, lo que pasa es que tiene idea de más permanencia en el mercado.

Un tema muy importante a fijar es *quién es mi proveedor*. Yo tengo que ver: que cumpla con lo que me está especificando en la propuesta, quién es el proveedor comercialmente hablando y qué soporte me ofrece acá donde yo estoy físicamente. Y les voy a decir porqué, ustedes leen la revista "fortun", anualmente aparecen las 7 empresas de computación más grandes del mundo por facturación, y hay empresas que tienen productos conocidos que todos los años están perdiendo un poquito de plata, no están ganando. Entonces, qué garantía de continuidad tengo de esa empresa, es más, si alguien viene y me dice yo te regalo runtime si vos desarrollas con mi motor, no digo ¡qué bueno!, "digo qué malo, ésta se va a fundir porque sino porqué te regala cosas?". Y lo peor que me puede pasar es que elija un entorno de desarrollo y un motor de base de datos, en algo que va a desaparecer en un par de años, porque migrar cuesta carísimo, y emigrar solamente por emigrar no porque además voy a hacer cosas nuevas. Yo mi opinión particular de hoy, es que en base de datos van a quedar 3: SQL Server, DB2 de IBM y Oracle. Son las 3 bases de datos que más están creciendo en cantidad de monto de venta digamos, son las que tienen las bases más instaladas. Son tres productos para distintas cosas, todos trabajan bases de datos pero no todos lo hacen igual. Fíjense una cosa curiosa, en Windows NT el mayor porcentaje de base de datos está dado en Oracle.

El pliego que les voy a traer en disquete (tiene más o menos 500 páginas), es un pliego que cumple con todos los requisitos del banco mundial para poder salir a la calle. Ellos lo que piden primero es que el pliego sea autoportante, o sea, que yo puedo cotizar desde Japón sin venir a la Argentina a ver cómo es el tema. El pliego tiene dos etapas:

- En la primer etapa del pliego uno dice: cumple con todas las especificaciones del banco, con las condiciones generales, y pone los términos de referencia, o sea, técnicamente qué se

quiere. Esto se distribuye a las empresas, mejor dicho, se llama a licitación pública. Aparecen un montón de empresas, las empresas hacen pruebas al software que tiene, se le piden aclaraciones. Fundamentalmente el objetivo del pliego es comprar un software aplicativo para transferencia tecnológica y las herramientas de desarrollo no las de producción. Las empresas presentan sus propuestas de cómo van resolver eso y nosotros, los que estamos en la comisión para el pliego: ponemos sobre el pliego original, todas las mejoras que a nosotros nos interesa que salieron por propuestas de los demás y dejamos un solo pliego definitivo y decimos: cotiza esto. No voy a hacer una propuesta ahora de vuelta, entonces todos cotizan la misma obra, hay un solo precio que se cotiza y todos tienen que hacer lo mismo, o sea, se cotiza sobre lo mismo. Entonces, lo que pasa en la primer etapa, porque tienen las condiciones empresarias reunidas, porque tienen las condiciones técnicas adecuadas, paso a la segunda etapa.

- En la segunda etapa, se pone precio nada más. Cosas interesante que tiene es que está exigido desde un plan de calidad siguiendo las directivas de las normas ISO, está pedido un plan de retirada, es decir, qué pasa si ustedes se van, qué hacemos nosotros, cuál es el plan de ustedes. O sea puede ser que se retiren comercialmente o puede ser que retiren el producto del mercado. Lo interesante es que antes de iniciar la primer etapa, nosotros hicimos una reunión con los posibles proveedores (con los posibles porque en una licitación pública puede venir cualquiera), le dimos una copia del pliego y le dijimos: "dígannos dentro de 15 días si pasan o no pasan, si no pasan porqué", es decir, si están en condiciones de cotizar eso o no. Y después, más interesante aún (toda la gente contestó que pasa por supuesto), fuimos después a una reunión privada, una para saber quiénes son porque no conocíamos (eran empresas de Bs. As.), HSBC tiene 600 profesionales consultores en Argentina, yo no sabía, eso me impacto pero después en la realidad Of the record, no se comprometían ellos ni no comprometíamos nosotros con las preguntas; pero se aclaran cosas que no se pueden aclarar por escrito.

Otro problema que tenemos que resolver, son las **condiciones de contexto urbano**. Ustedes quieren saber cuál es la primera causa por la cual quedan afuera los sistemas informáticos?: lo primero que falla es la energía eléctrica, es lo primero que falla. Ahora, ¿qué les quiero decir con esto?: cuando hablamos de servidores, estamos hablando de máquinas de misión, la más importante de toda la empresa, nosotros tenemos que asegurarnos: primero la finalización normal no es solamente estabilizar la corriente normal que voy a tener que hacerlo porque sino te puede dar cualquier cosa, sino asegurarte terminar las operaciones ante una contingencia en forma normal, que no me queden archivos abiertos, etc. Ustedes, no sé, pero deben saber que Unix cuando le dicen que grave en realidad no graba físicamente, salvo que se le den una orden pero se hace lentísima la máquina; para eso, tienen que poner un servicio de energía ininterrumpido.

Servicio de energía ininterrumpida, hay dos tipos de sistema:

- offline: el sistema offline, entra a funcionar cuando detecta que no viene energía por la red, normalmente reaccionan en 15 milisegundos, los aparatos del hogar deben tener capacitores que es esa energía que cuando uno apaga el aparato éste sigue andando de 300 milisegundos. La duración normal para cerrar, es media hora como mínimo. Entonces, las empresas dicen bueno, con media hora está bien, pero si vos querés seguir trabajando, y ponele dos horas. ¿Cuál es la diferencia entre media hora y 2 horas?: es la batería que tiene el sistema. El costo es más cantidad de baterías. Hasta 2 horas, 2.5 horas no pasa nada, sino después los costos suben mucho.
- Online: como el nombre lo indica le da el sistema de energía domiciliario que es alterno, pasa por algo que lo vuelve continuo para poder alimentar la batería, y a la salida de la batería, la convierte en alternada de vuelta para que la utilice el equipo que tenemos.

El offline tiene la ventaja de que es mucho más barato, tanto de costo inicial como de costo de mantenimiento, porque la batería no trabaja en forma continua, sino que trabaja cuando se necesita. Tiene cómo desventaja es que no hay una alimentación pura de energía a la computadora, sino que solamente hay energía pura cuando trabaja el sistema UPS. Si yo a la electricidad la doy pura, me evito un montón de problemas: fatiga de los circuitos, errores esporádicos, lo que pasa es que hay que pagarla más cara. Les aclaro, en EEUU, no hay fábricas de UPS offline, todas son online y eso que la energía de ellos es mucho más pura que la nuestra. Me ha pasado, hace una semana una mujer me decía "siempre esta máquina me da error, no sé que tiene" y yo le dije que la cambie por otra y en el otro lugar no tenía error, el problema era la corriente eléctrica, por más que tenga estabilizador. El

estabilizador lo que hace es cuando viene mucha energía bajarla pero cuando viene poca ya pasó, es como el rayo.

La segunda causa por la cual las máquinas no andan, es porque no están en condiciones ambientales adecuadas, fundamentalmente me refiero a los servidores. Hay una vieja premisa que no la inventé yo, que dice que los metales se dilatan con el calor. Y los metales por ejemplo las cintas de backup, entran en ignición antes que el papel; eso pueden hacer poniendo un cassette de Luis Miguel en el patio en verano y van a ver que van a tener problemas con el cassette. Vos perdés el cassette, pero para poder perder un BIT se necesita mucho menos; es aconsejable que el papel de procesamiento y los archivos magnéticos que se va a trabajar ese día, esa consejable que estén en la misma sala que el servidor, para que tengan la misma temperatura, la misma humedad. En papeles de impresoras de alta velocidad (más de 50 páginas por minuto), cuando está muy seco se lo puede tragar el cursor, se lo quemar; y si está muy húmedo se empasta, entonces, tienen que estar a la misma temperatura sobre todo en zonas como la que vivimos nosotros, zonas de humedad. Las computadoras son máquinas de oficina y van a trabajar en ambientes de oficina, pero piensen que quién lo dice, el ambiente de oficina de los EEUU no son los ambientes de oficina de Argentina. Acá hay ambientes de oficina con 45° de sensación térmica, allá no van a encontrarlo nunca. Entonces, hay que tener cuidado con eso, lo segundo que hay que tener cuidado, es que son ambientes cerrados, sellados, no tienen ventanales que abren que entran aire, polvo. No se olviden que todo eso del polvo y la humedad en la máquina genera falsos contactos. Yo ví en Italia, un puesto de trabajo industrial, que yo tenía que poner la mano en le agua para operarla, y andaba al pelo; o sea, son unidades totalmente selladas, no es la PC de \$800 que compramos acá en la esquina. Entonces, vos ponés una PC de esas en un molino, y cada semana tenés que tirarla porque se te llena de polvo, en la planta de un molino porque si le haces una sala naval, cerrada, limpia; sala naval porque en un barco tenés que llevar cosas inflamables y nadie puede tirar un pucho, entonces tenés que hacer una sala aislada desconectada totalmente del exterior. Entonces, para máquinas chicas un aire acondicionado de pared alcanza; para máquinas grandes tiene que ser un sistema de acondicionamiento de aire, no solamente enfriamiento como hace el acondicionador de pared. Hubo unas máquinas que se llamaban 770 de Olivetti, que funcionaban solamente entre 1000-1500 metros de altura, en Sta. Fe no pudo andar nunca esa máquina, y vos la sacabas y la llevabas a Salta y en Salta andaba.

Otro tema importante a tener en cuenta es el piso, el mejor piso es el piso que están pisando (cerámico), porque es fácilmente limpiable. Hace unos años se puso de moda el cableado estructurado y el servidor tiene subsistemas conectados a él (impresora, lectora de cartiong, jumbos, sistema de disco separado), van a tener que poner cables, y si esos cables son muchos, van a tener que andar a los tropezones, entonces se recomienda un piso técnico. Un piso técnico, es un piso elevado, que uno tiene que decirle cuál es la cantidad de carga por metros cuadrados distribuida que quiere; nosotros en Santa Fe tenemos una máquina que lee directamente la bovina cruda, pesa más de 2000 k, esa máquina no la podés poner en un 4to. Piso porque se te viene abajo, la tenés que poner en planta baja. En ese caso pudimos poner esa máquina nomás en planta baja porque no podíamos hacer falso piso porque la distancia entre el piso y el techo era corta, sino teníamos que poner operadores de 1.50 mts más o menos. Yo creo que el ejemplo de lo que no debe ser un edificio de informática, es la dirección de informática de Santa Fe, porqué?: porque está en una peatonal, entonces tenés que acarrear cosas media cuadra; porque la integracion en informática es horizontal no vertical, porque no hacen cosas aisladas en cada piso, está todo vinculado; porque hubo que poner los servidores en un séptimo piso porque era en único que nos permitía hacer un falso piso. Para poder subir y bajar máquinas, hubo que hacer las ventanas que dan a la calle desmontables, y poner grúas para poderlas subir y bajar. El estado es un poco irracional, lo que pasaba en esa época, era que la dirección de informática funcionaba en lo que es hoy la sede del partido justicialista, y lo había sido también en la historia hasta el 55 y había un compromiso de devolverle el edificio. Eso por un lado, por otro lado, el banco de Santa fe (el que era), iba a tener un directorio unificado, entonces había que hacer un edificio para el directorio, la sede del banco central estaba en la misma cuadra. Mientras se estaba haciendo el edificio (que ahí sí se justifica un piso por director), se cambia de nuevo la política y dicen no, dos sedes una acá y otra en Rosario, entonces no tenía sentido ese edificio. Entonces, el estado, decide cedernos ese edificio para que nosotros nos podamos mudar, entonces tuvimos que readaptar ese edificio para podernos mudar, y sino la calle. Antes, en Francia, las computadoras se instalaban a la calle porque era un símbolo de Status que la empresa esté sistematizada; entonces, nosotros en Santa Fe, para no ser menos, también pusimo nuestra primer computadora con vista a la calle. En la década del 70', 3 veces tiraron un tiro ahí y en Francia pasaba exactamente lo mismo, las minorías árabes para boicotear

políticas antimigración del gobierno francés, le boicoteaba todos los centros de cómputos. Es decir, si vos tenés un sistema crítico para tu organización, no lo pongas ahí porque está muy expuesto.

Problemas específicos de la teleinformática: ahora tenemos diversos problemas, de diferentes áreas temáticas que tenemos que resolver, nosotros no somos especialistas en compras, no somos especialistas en aire acondicionado, no sabemos hacer obras, no nos enseñan a hacer un pliego licitatorio, no nos enseñan a dialogar con los usuarios. Entonces problemas específicos de la teleinformática, primero ¿qué es un sistema teleinformático?: son sistemas que son distribuidos geográficamente, sus operaciones están distribuidas geográficamente, o sea, los usuarios están fuera del edificio. Los sistemas teleinformáticos tienen algo que ver pero no es lo mismo que el sistema interactivo, pero una cosa es que yo tenga interactividad sin teleinformática y otra es que tenga interactividad con teleinformática. Entonces, ¿qué significa tener un sistema teleinformático?: introducir, recibir datos por algún medio de transporte de datos (aire, tierra, etc.). Más adelante, cuando veamos selección de software de comunicaciones, vamos a ver cómo puede ser los diferentes tipos de conectividad que podemos tener, desde el punto de vista de las aplicaciones, no de las máquinas. Un problema que se da, es que se requiere mayor nivel técnico en la concepción de estos sistemas. Este tipo de sistemas son más complejos, hay que fijar toda una estrategia para poder hacerlo. Cuando uno lo pone en marcha, trae aparejado una reorganización de los servicios del usuario, o sea, el usuario debe reorganizarse porque ahora es teleinformático, cambian las funciones.

Otro problema que hay, la gestión y la operación en la empresa, está integrada, por lo tanto al pararse la operación del sistema se para la gestión. ¿Qué les quiero decir con esto?: si Aerolíneas Argentinas para su sistemas, no puede hacer pasajes, porque está todo integrado. En una zapatería no tengo problema porque si se le para la PC, yo le puedo decir y bueno mira facturará manual; pero no puedo hacer lo mismo con organizaciones grandes, en algunas es imposible.

Y por último, algo que a lo mejor ustedes no llegaron a percibir porque lo tomaron muy natural, la teleinformática requiere equipamiento especial, que no es la clásica computadora, requiere hardware especial. Recién entrados los 80', las universidades empezaron a incorporar como tema específico de la computación, el tema de las telecomunicaciones.

¿Cuáles son los problemas específicos en los sistemas teleinformáticos?:

- uno es el que ya nombramos, la gestión está muy integrada, entonces al interrumpirse el sistema se interrumpe la gestión.
- El sistema está directamente conectado con los usuarios, tenés la posibilidad de rescate. En los sistemas interactivos, no tenés esa posibilidad o la tenés minimizada porque vos al usuario lo tenés alrededor tuyo. Por ahí lo rechaza, a nosotros por un error yo diría de visión, nosotros queríamos pasar del Office al StarOffice, queríamos implementarlo primero en la dirección de informática, aprenderlo, manejarlo, ver todos los caminos y después correrlo a otra oficina del ministerio, donde podamos tener poder y después extenderlo a toda la provincia. Y en ese momento llegaron intimaciones de Microsoft, entonces, como se asustaron, el ministro de hacienda sacó una resolución que opuso al Startoficce, contra la opinión nuestra. Era un simple cambio y había mucho rechazo. Cuando uno tiene al usuario local, el usuario aprueba la interface, pero cuando uno está que tiene usuarios en toda la provincia, ¿qué usuario aprueba la interface?. Ya en la ACM (fabricante de máquinas de computación), recomienda en las curriculas para enseñanza de Computer Science que se de psicología, para ver cómo es la reacción natural del hombre frente a una operación que tiene que hacer, cómo es transacción natural en el hombre. Yo les voy a dar un ejemplo: el programa de contabilidad más importante que había en Argentina es el de Berjerman, después cambio por el tango. Bejerman, un contador se sentaba de la universidad que fuera, y naturalmente trabajaba con bejerman, era intuitivo para el contador entenderlo porque bejerman es contador de estudio contable entonces hizo un sistema genérico para lo contable, y no uno para lo que él quiere. Tango es al revés (bueno ahora cambió mucho), si no había un informático, el contador no lo entendía, el informático le tenía que explicar al contador el sistema, era mucho más complejo para él entenderlo porque estaba hecho por informáticos y seguro que entorno a una empresa.
- Otro problema que existe en la concepción, es el tiempo de respuesta. Éste es el elemento de performance más importante, es un factor de calidad que debe pactarse. Esto no sirve, si uno no simula el estado, no sirve saber si va a llegar, ¿porqué?: muy fácil, las bases de datos relacionales, hacen la relación en memoria real, entonces vos tenés un lote de prueba de 200 unidades de información por ejemplo, lo probas en un equipo PC y tenés buena respuesta, vos estás trabajando en memoria real en realidad, cuando después si tenés 1 millón de casos en la realidad y lo llevas, no hay quién te lo dé. Esa es la razón por el cual los Unix se fueron a 64 bits, porque con 32 bits yo no

puedo direccionar más de 4GB. Y ahora las bases de datos llegan al nivel de terabytes, yahoo creo que baja 1TB por día de datos. El tiempo de respuesta está compuesto por dos tiempos en realidad: uno es el tiempo en las línea, el tiempo de tele digamos; y el otro es el tiempo de proceso, tiempo in. Uno tiene que analizar el tráfico que va a tener la línea, para saber el ancho de banda que necesita. Ustedes piensen que 1 byte tiene 11 bits transmitido, por la trama de cabecera y cola, eso es lo primero. Lo segundo que de ese tráfico que ustedes esperan no debe estar ocupado en promedio más de un 20%, para no tener encolamiento en la línea. Tercero, tienen que tener alguna herramienta de monitoreo de la línea para saber cuánto está transmitiendo realmente y cuánto está repitiendo mensaje. Para saber lo que está adentro yo tengo que saber desde que llegó el mensaje a la cpu mía hasta que devolví el mensaje, cuánto tiempo estuvo adentro. Ahí yo puedo tener diversos tiempos:

1) tiempo de espera, o sea, que estoy esperando para entrar o estoy *swap*, tiempo de espera, tiempo muerto;

2) es el tiempo de cpu, es el tiempo que estuve procesando realmente, estuve usando los 400MHZ;

3) es el tiempo que estuve haciendo el trámite de ir al disco y 4) la demanda de memoria no satisfecha que tuvo. Siempre el problema de performance lo van a encontrar en alguno de estos factores: falta de memoria, falta de acceso al disco (el disco está muy accedido entonces hay mucho encolamiento) o problema de cpu; cuando agrandan una cosa, se desequilibra por otro lado; pero no se olviden de una cosa: que la IO con milisegundos y memoria son nanosegundos, la diferencia es sideral. Entonces, también hay que tener herramientas para poder medir la performance del sistema.

- Es muy importante hacer para los servidores de este tipo de sistemas, lo que se denomina MTBM o tiempo promedio entre dos fallas, esto es una medida estándar junto con MTBR que es el tiempo promedio de reparación. Les comento los productos industriales cuando el origen es EEUU, tienen obligación de traer escrito el tiempo promedio entre dos fallas. Uno lleva el tiempo promedio a cero, en la medida que empieza a poner redundancia, por ejemplo: hoy existen redundancias en la memoria en el sentido de que pueden tener un BIT, o en el caso de alteración de un BIT, el banco de memoria puede seguir trabajando; hoy existen discos con tecnologías RAID, los que le permiten tener redundancia y que le recuperen un plano de archivo, estos trabajan en forma transparente. La cpu si no muere de entrada, puede durar mucho más que el tiempo de vida de la máquina, de todas maneras, por ejemplo el banco Galicia tiene un equipo gemelo operando a 150mts de donde está el centro de cómputos principal y en forma sincrónica, o sea, que se le quema el edificio y no pierde su información, que es lo más importante que tiene. O sea, que eso se puede minimizar, ¿para qué se usan estas dos medidas?: porque entonces uno le pone cláusulas, la primera que ponen en los contratos es: ¿cuál va a ser el tiempo promedio de atención, o sea, cuando va usted a empezar a atender?: entonces una cosa que se pide en la parte crítica, es que el tiempo de atención si se te cae la red, son dos horas. El otro es el tiempo promedio de reparación, si vos no cumplís te cobro una multa, multa que implica no pagarte los servicios que están vinculados con eso. Esto es lo que se llama la disponibilidad que queremos del sistema.
- Otro problema es vinculado al estudio del sistema, cómo va a funcionar. Si ustedes hacen una grilla, y ponen por un lado sistemas operativos, por otro lado procesadores (CISC, power cc, alpha), bases de datos, desarrolladores de aplicaciones, bases de datos documentales, data workhouse, van a hacer un montón de agujeros, no van a tener la intersección de todo en todos los puntos. Entonces, no pueden comprar cosas sueltas, tienen que conseguir el sistema que van a tener en su funcionamiento y en su relación, ¿qué es un sistema?: un conjunto de elementos en interacción, o sea, ustedes no pueden pretender trabajar en Visual Basic y trabajar con plataforma RISC porque no anda; y quieren trabajar con windows nt en algo que no sea Intel porque no anda. Tienen que buscar una coherencia porque no hay coherencia entre los materiales.
- Tienen que tener un horizonte de planeamiento, o sea, por cuánto tiempo quiero tener esta máquina, porque en un momento dado se van a dar cuenta de que no tienen más producción y no pueden ampliarlo. El problema vinculado con la realización además, es la formación de los usuarios, los usuarios están remotos.
- Otro problema, es la integración y la prueba en paralelo. El desarrollo del sistema termina cuando uno lo entrega, teóricamente tiene que andar. Ahora, es responsabilidad del usuario hacer la prueba en paralelo o no. Hay algunos autores que a raíz de esto dicen: "agoten el período de prueba", pero

también tal cual trabajamos en Argentina, hay que ser suicidas a veces de desplegar la aplicación en todos cuando es el corazón de mi negocio lo que está en juego.

Bueno, entonces, los problemas vinculados a la implementación tenemos:

- la prueba en paralelo.
- El seguimiento.
- Que los archivos que se requieren arrancar tengan la estructura de datos que nosotros pretendimos tener. A veces una cosa es a nivel de cuando uno piensa las cosas y otra cosa es a nivel de cuando uno las hace. Todo el mundo dice vamos a tener la dirección estructurado, pero vos en tu archivo no lo tenés estructurado y no hay forma de estructurarlo fácilmente. Si bien en la Argentina existen normas de cómo codificar nombres (nombres de calles, nombres de personas, etc.), no es fácil convertir un archivo que no está estructurado a uno que sí está estructurado. La tarea de depuración de datos es una tarea permanente, lo que les quiero decir, es que nunca deben perder los datos existentes. Y saben cuál es el problema de los sistemas heredados, que como los costos eran muy grandes, estaban totalmente desestructurado, un mismo campo podía significar dos cosas distintas dependiendo de cómo estaban relacionadas las variables. No se olviden que además, en archivos que no son bases de datos (que son la mayoría de los sistemas heredados) modificar un campo significaba modificar todos los programas que utilizaban ese archivo.
- Un problema a tener en cuenta en este tipo de sistema es que la red esté planteada físicamente. Si voy a poner un sistema teleinformático tengo que tener algún tele, algún sistema de comunicaciones instalado.
- El mantenimiento de hardware y software: cómo van a hacer, si van a tener un proveedor local o un proveedor provincial que a su vez va a subcontratar a proveedores locales o se van a entender con múltiples proveedores. La mesa de ayuda, tengo que tener una mesa de ayuda centralizada para saber lo que quiere el tipo del otro lado porque poner un tipo en cada lugar para que haga de mesa de ayuda me genera un costo terrible. Por eso hoy se busca mucho la tecnología web, una intranet y terminales flacas, puestos de trabajo flacos para tener la menor cantidad de problemas en el cliente, la mayor cantidad e problemas tenerlos en el servidor donde estoy yo prestando servicio.
- Otro grupo de problemas está vinculado a la transmisión. Mientras estoy en un sistema que estoy migrando a otro sistema, ¿cuáles son los problemas que tengo?: primero hay que hacer un plan de estabilización y consolidación de lo nuevo, no hacer cambios inmediatamente sobre lo nuevo porque después nunca sé de donde viene el origen de la causa del problema. O sea, no hacer perfeccionamiento de los programas actuales, hacer una estabilización de lo que se tiene.
- Un problema que se da, es sobre cómo hago la exportación de los sistemas que todavía no pasé al nuevo sistemas (aplicaciones, me refiero). Todo el mundo le dice a ustedes que Cobol está obsoleto, sin embargo en el mundo, el lenguaje más utilizado realmente es el cobol. Si vos escribís las líneas que hay escritas en EEUU, las ponés a hacer líneas por pulgadas, la lista llega a luna y vuelve, tiene 600.000 Km. dividido por 243 por 6, y van a ver cuántas líneas. Y eso hay que mantenerlo; yo entiendo que si decimos que migramos aplicaciones, voy a cambiar el entorno de desarrollo, entiendo que alguien pueda pensar eso. Entonces no va a comprar cobol en la nueva máquina, pero entonces tiene que seguir conviviendo hasta tanto migre todo, en Cobol; eso puede tener un costo caro porque tiene que mantener dos sistemas. Es tanto a nivel de operación como a nivel de servicio técnico, a nivel de conocimiento. Entonces, lo mejor es que si yo puedo migrar lo viejo a la nueva máquina, había casos que era imposible, había máquinas que estaban en AS400 que hay un montón en el mundo, trabajaron en origen RPG que es un lenguaje que no es universal, no es como el cobol que lo tenés en cualquier máquina, en cualquier sistema operativo. Entonces, tenían que seguir trabajando en rpg, tenían que recodificar todos los programas eso demora un tiempo. Todos lo sistemas de producción, es decir, que requerían análisis matemático, estaban desarrollados a través de cálculo numérico en Fortran, y no existía el C; ¿entonces qué hacían mientras tanto rescribían todo en C?: la gente tenía que trabajar o con la otra máquina o aspirar una máquina afuera para correr el sistema. Por ejemplo: el grupo Safra cuando decidió cambiar de tecnología, se quedó con una máquina en el país para hacer todo lo que era el proceso de planta *empañadora*.
- Después hay que tomar medidas para prepararse para el D, para el día del lanzamiento. Una cosa que muy poca gente hace, dice vamos a hacer, vamos a hacer pero no introduce las

nuevas técnicas de gestión. Y esto pasa por un problema vinculado a la propia universidad, toda la gente que tenemos más de 40 años, los profesores nuestros, no existía cuando ellos se formaron, el fenómeno de la informática, entonces no daban técnicas de gestión que si bien ya estaba planteada a nivel teoría, requería mucho cálculo, y muchas veces esos profesores eran preparados por profesores de la misma cultura, entonces venía por herencia ya. Entonces, hoy existen un montón de técnicas ya sea utilizando métodos heurísticos, de inteligencia artificial, de tecnología del conocimiento, etc. Que permiten no darte la verdad pero sí darte camino para poder tener patrones, pero si el administrador no lo conoce cómo lo va a hacer. Entonces, hay que hacer toda una capacitación en ese tipo de cosas.

¿Qué se debe consolar con respecto a los problemas a resolver? los problemas a resolver son las aplicaciones. Yo no les voy a dar la metodología de hacer sistemas porque ya se aburririeron de verla en dos años, lo que sí les voy a dar es qué debe controlar el jefe de proyecto, el gerente de sistemas para cómo se están desarrollando las aplicaciones, que son puntos que ustedes conocen por eso se los doy directamente:

- Que cada aplicación tenga un único responsable, eso es problema de administración.
- Que se planifique la tarea, es decir, que exista un plan. Este plan va desde qué tareas se tienen que hacer y se van a iniciar, hasta cómo se van a migrar los archivos, cuál es la estrategia de la puesta en marcha sería eso. Por supuesto que si los analistas no hacen un plan, jamás va a poder controlarlo porque si no tenés contra qué medirlo. Este plan basta que esté hecho un diagrama de gant, de barras porque las tareas todos conocemos cuales son las tareas.
- Que exista firmado por el usuario un pliego de condiciones de los usuarios que es lo que nosotros llamamos comúnmente carpeta del sistema. Lo que tiene que firmar el usuario es el qué. Hay un error conceptual que no viene de este instituto sino de todas las carreras, que el manual de usuario es lo primero que hay que hacer porque es lo que quiere el usuario, no hacerlo último. Porque el usuario lo está viendo, lo está viendo todos los días y es el lenguaje que él entiende, si lo hacemos al final para qué lo quiere, después que puso en marcha el sistema, no lo usa más. El manual de usuario normalmente las metodologías lo dan cuando están viendo el diseño detallado, tienen que verlo al principio, es lo primero que hay que hacer porque eso es lo que quiere el usuario, esas pantallas. Además, el manual de usuarios se elabora mal en todos lados, el manual de usuarios debería ser un manual de procedimientos no un manual de qué resuelve la máquina, porque un procedimiento puede llevar varias pantallas, entonces no tenés un pantalleo un menú jerárquico. O sea, debe partir de la actividad, no de la pantalla.
- Que los programas se realicen con *comer framework*, el framework es el marco de trabajo, que no solamente es el desarrollador, el lenguaje sino la convención de nombres, que esté controlado lo que se escapa del desarrollador. ¿Qué les quiero decir con esto?: yo puedo trabajar con Power Builder para desarrollar, pero hay cosas que si no se hacen es antiperformance, por ejemplo rutinas de cálculo de interés, eso lo hago en c pero eso nomás hago en c. Uno tiene que controlar lo que normaliza y desnormaliza, eso es normalizar, saber cuándo se normaliza. Por ejemplo: hoy teléfono es una tabla, no es un campo del archivo base porque hoy tenés teléfono celular, tenés teléfono de primera línea, segunda línea, tercera línea, y más las empresas tienen conmutador, fax, etc. Entonces, la tabla es bueno para archivos chicos, pero si yo tengo que tener en un millón de jubilados buscar un teléfono referencial, permito guardar 3 y desnormalizo porque tengo atributos multivalor, pero es mucho más performance que crear una tabla. Los domicilios lo mismo, hoy las empresas no tienen un solo domicilio, entonces uno tiene que tener esos domicilios y qué actividad se realizan en esos domicilios, antes tenían uno solo.
- Que se estén construyendo los archivos básicos, esta tarea normalmente, si son grandes las aperturas de los archivos, se contratan a terceros porque son tareas puntuales por única vez y la organización no se puede ocupar de un pico digamos.
- El plan de pruebas. Las pruebas deben estar documentadas, ¿documentadas qué quiere decir?: que se puedan volver a hacer, que si yo pongo de nuevo esta prueba y me tiene que dar el mismo resultado que dice acá en el papel.
- El calendario sobre todo lo que es batch, en lo que es online, tiene que estar el tema de la cantidad de transacciones en hora pico que soporta.

- Lo último que es lo más importante, es el procedimiento de recepción, verificación y aceptación. Esto hay que tomarlo, por más que esté trabajando en forma interna, como si estuviera trabajando con una empresa privada. Independientemente del analista que me dice anda, todo anda bien, lo entregamos al usuario. El centro de desarrollo del sistema, debe buscar un procedimiento de recepción y aceptación, ¿qué significa esto?: verificar que de distintos puntos de vista, si esto está en condiciones de ponerlo en marcha el usuario, independientemente del usuario. Esto además, está fijado en las normas ISO 9000 ya a esta altura, que se llama revista de fin de fase y la auditoría. Quienes deben participar en este procedimiento: por parte del usuario debe participar gente que represente los 3 niveles de información que habíamos visto antes en el primer día de clase (nivel operacional, funcional y jerárquico), salvo que el sistema no dé información al nivel superior. En el nivel de sistema ustedes deben tener gente que tenga experiencia en desarrollo de sistemas, sobre todo si ha trabajado en el área usuario aunque no en esta aplicación. Y debe figurar personal técnico, ¿qué significa personal técnico?: desarrolladores, especialista en comunicaciones, en sistemas operativos, en bases de datos fundamentalmente, para que lo puedan ver desde todos los puntos de vista. Esta gente es la que va a decir si el sistema está en condiciones de ser operable. ¿Qué hace el agente de sistemas?: ve el continente adónde va a estar el tema aplicativo, cómo es el contenido, los esquemas de contingencia, los esquemas de seguridad, los esquemas de backup, qué uso de motor de base de datos, qué tiene como procedimiento almacenado, es decir, una serie de cosas que son propias del continente, de la administración del sistema.

## **CLASE TEÓRICA DEL 14/05/01**

Acá les dejo el pliego que les dije que iba a traer, ...después van a ver un pliego de referencia que es la parte técnica, y después va a ver las condiciones generales y las condiciones especiales del contrato. Este pliego tiene le calculan, aprox. 5 millones de dólares lo que va a costar el producto. En realidad la problemática del poder judicial es una problemática del juzgado, o sea, cada juzgado tiene su propia forma de actuar.

Nosotros estábamos viendo cuáles son los puntos en que el jefe del proyecto informático o el jefe del centro de datos, debe tener en cuenta con respecto a cómo controlar el desarrollo de las aplicaciones, o sea, de los problemas a resolver. Bueno ahora viene la implantación del sistema.

### **◆ Implantación del sistema**

Los objetivos son:

- controlar una buena calidad del sistema y su compatibilidad con condiciones reales.
- Efectuar la puesta en marcha del servicio.

Con esto qué les estoy diciendo, que la prueba en paralelo es una responsabilidad del usuario hacerla o no hacerla. Teóricamente, cuando nosotros entregamos un software, no debe tener errores, sin embargo no podemos asegurar que no tenga errores. Y lo que yo les dije el primer día de clase, que el 80% de los errores encontrados sobre sistemas implementados son problemas de las especificaciones, o sea, no son especificados. Y las especificaciones son aprobadas por el usuario, y el analista desarrolla en contra de las especificaciones, entonces tengo un alto riesgo de no hacer la prueba en paralelo, el usuario siempre nos dice las especificaciones generales pero nunca las particulares por su propia mentalidad normal.

La prueba en paralelo tiene el problema del costo, hay que trabajar dos veces. La implementación hay que hacerla parcialmente, no se puede golpear las manos y decir ahora no se juega más fútbol, es básquet porque van a seguir pateando la pelota. Entonces tenés que ver cómo se va comportando en condiciones reales, implementar de a poco porque el grupo de desarrollo tiene que estar en todos los lugares que se esté implementando.

¿Qué acciones técnicas se toman en esta etapa?: por ejemplo, si ustedes quieren perder el empleo, lo que tienen que hacer es un lunes implementar el sistema e irse de vacaciones, seguro que cuando vuelvan siguen de vacaciones. Lo mejor es implementarlo un viernes, porque el viernes todo el mundo está contento, todos de buen humor, además tenés sábado y domingo para arreglar cualquier problema.

¿Cuáles son los mayores problemas que se dan?: que los programas requieren pequeñas correcciones, es decir, uno terminó todo, las instalaciones eléctricas, todo pero seguro que quedó algún cortocircuito en algún lugar y hay que buscarlo. O es algo que se entendió mal. Nosotros estamos

implementando una nueva organización, hay cambio de funciones. Entonces, la gente no sabe bien cómo hacer hasta tanto no se acostumbren a la rutina, siguen tratando de hacer lo otro. Entonces, hay que incorporarlos a la rutina; además, es el momento más dramático en la vida de una aplicación, porque los problemas se conocen ahora. Aparecen un montón de pantallas, los manuales los hacemos mal.

Si sobrevivimos, lo que corresponde ahora es el balance. ¿Qué es el balance?: nosotros cuando teníamos el comité de seguimiento, que seguía el plan, iba contabilizando: los medios empleados, costos, los atrasos, etc. Entonces, ahora hay que apreciar lo realizado, es decir, qué es lo que nos imaginábamos que íbamos a hacer, qué es lo que hicimos realmente, en cuanto a tiempo, en cuánto a costo y porqué tuvimos esa desviación, para que en el próximo proyecto podamos trabajar mejor. Para hacer un balance yo tengo que contabilizar los medios empleados, no hay otra. Hace 30 años era la tendencia contabilizar para controlar al empleado, hoy eso se cambió totalmente, hoy la tendencia es contabilizar para mejor posicionarnos en el próximo presupuesto. No se controla al empleado, se controla el proceso, el empleado se controla de otra manera, inclusive hay formas de trabajar que al empleado lo controlan los mismos compañeros porque se trabaja en equipo, incluso en una de esas valorizaciones que dejé en el pliego era la experiencia previa que tenía el grupo de trabajo en trabajar junto, trabajar en equipo porque se conocen los códigos entre ellos. Entonces, con este balance se pueden evaluar los costos y los beneficios implícitos. Les comento algo, cuando estaba Paraná Mercantil (se acuerdan de Paraná Mercantil que estaba en calle Almafuerde), un día se decidió contratar un desarrollo de software externo, entonces, se sacó un aviso en el diario de Paraná y de Santa Fe. Se presentaron 3 grupos de desarrollo, 3 pequeñas empresitas de desarrollo de software, les dijimos cuál era el problema, les dijimos en qué consiste y les dijimos que vengan tal día. Entonces, una de las empresas decís que priorizaba el tiempo. Entonces, una empresa que iba a demorar 3 meses cotizaba por ejemplo \$80; otra empresa dijo que lo iba a hacer en 6 meses y lo cotizaba en \$100; y la otra lo iba a hacer en 4 meses y medio y lo cotizaba \$110. Entonces cuando entrevistamos a esa empresa que demoraba más tiempo, uno de los socios le dice si no podía bajar lo tiempos; la empresa le contestó que esos eran los tiempos que iba a poder cumplir. Pero otras empresas me dicen menor tiempo, entonces, el tipo desplegó la laptop y le dice: "mire, en este proyecto que es más o menos parecido, yo demoré tanto tiempo. Para esta empresa que es tal características y que es de construcciones yo demoré tanto tiempo, entonces yo sé cuanto tiempo voy a demorar para hacer esto. Entonces, yo no le puedo prometer que voy a demorar más o menos de 6 meses, yo me comprometo a hacérselo en 6 meses." Al final se quedaron con ese, porque los otros decían que iban a hacerlo nomás, total...

Bien, conclusión: ¿cuáles son las dificultades que se encuentran en este tipo de desarrollo?:

- Primero el aporte de dificultad técnica, o sea, no es fácil, no es una PC grande que tengo acá. Si yo estoy en un proceso de construcción permanente de prototipo digamos, porque no hay dos softwares aplicativos iguales. La capacidad de prueba no es la misma para todos.
- La constante renovación de la informática, siempre aparecen productos nuevos. No es solamente que cada vez tengo hacer un prototipo nuevo, sino que además con nuevas herramientas.
- La simbiosis creciente que existe entre la informática y la vida de la gente. Eso qué significa, que cada vez más las empresas están dependiendo de la informática. Entonces, para ellos es el corazón del tema. En la actividad informática hay mucho más fracaso de lo que se cree. Normalmente nos enteramos más de los fracasos en el sector público, porque son más conocidos; y los del sector privado, salvo por algún amigo, no te dicen nada porque la empresa queda mal, pierde imagen, seriedad, entonces no lo dicen.

Uno de los temas que yo hablé acá y que ahora voy a desarrollar: cómo hacer la justificación técnica de un producto, qué tiene que contener. ¿Cuáles son los factores económicos?:

- reducción de personal: qué quiero decir con esto?: que el personal que estaba haciendo una actividad pasa a hacer otra, pero esa actividad que estaba haciendo, como yo la automatizo, se reduce el costo.
- Reducción de bienes de consumo: porque puedo mejorar el stock. Ya no trabajo con una pila sino con valores racionales de stock.
- Reemplazo costo y equipo: es decir, hay cosas que se dan de baja. El costo de mantenimiento de los equipos cada vez es más barato, con costo de mantenimiento me refiero a electricidad que consumen y a energía. Esto pasa con todos los artefactos, los equipos de aire acondicionado de antes te consumían una casa de luz, ahora consumen muchísimo menos.

- Mejoras financieras: porque puedo tener un cashflow, es decir, presumo la plata que va a entrar y lo que voy a gastar y o anticipo créditos más baratos para que no me gire en descubierto o deposito para que me dé algún beneficio financiero. Hay estimadores indirectos porque son muy difíciles de medir, por ejemplo: cuando yo puedo armar un plan de producción inmediato, es una mejora lo que voy a hacer. Por ejemplo, si yo estoy en una fábrica de automóviles y yo digo: con el stock que tengo ¿qué me conviene fabricar?: despiece, productos semielaborados digamos motores semiarmados. Un estimador indirecto muy importante es cuánto es tener la información en tiempo y forma, cuánto gano con eso. En hardware, por lo general los costos se prevén que se amorticen en 3 años.

Todo esto tengo que compararlo con los costos, estos son los beneficios que obtengo, tengo que ver los costos: cuál es el costo del hardware, cuál es el costo de garantía o servicio técnico por 3 años, cuál es costo del software y cuál es el costo de asistencia durante los 3 años. Para que ustedes se den una idea, una persona que ingresa a un centro de cómputos, al cabo de 2 años, el 50% de su tiempo está haciendo mantenimiento, por cambios correctivos y perfectivos en su mayoría. ¿Cuál es el costo de la estructura de personal que voy a tener?: eso implica sueldos y costo de capacitación externa. En informática, la capacitación no es un derecho, algo que nos dá la empresa, es una necesidad. Se entiende que la gente hoy entra con una capacitación general de la informática, es decir, ustedes terminan acá y saben todo lo que es una base de datos pero no conocen todas las bases de datos; si la empresa decide trabajar con Sybase, tendrán que aprender Sybase, saben lo que es una base de datos relacional. ¿cuánto cuestan las instalaciones físicas?, ¿cuánto nos van a costar los suministros?. Suministros por ejemplo: soporte magnético, papel, uno puede hacer sistemas en la medida en que la PC está cada vez más cerca de ser tener una relación una PC un empleado, uno puede eliminar mucho papel, eso es costo. ¿cuánto cuestan los servicios auxiliares?: los servicios auxiliares como costo de energía eléctrica y aire acondicionado.

Para hacer la evaluación técnica, hace un análisis de los recursos humanos, si se puede trabajar con personal interno del propio centro de datos, si necesitan capacitación, si hay disponibilidad en mercado de personal técnico para lo que se requiere. Es decir, no es tan fácil encontrar a veces el personal técnico.

Hay que hacer un análisis comparativo con el sistema anterior, de las prestaciones que va a tener el nuevo y de la facilidad de uso. Hay que hacer, una evaluación real del proceso con cada equipo, o sea, para este tipo de procesos que equipo es mejor. Y después hacer un becmán, la técnica de becmán la van a ver después, ésta hay que hacerla con datos reales, por lo general se mide concurrencia (cómo es capaz de atender clientes simultáneamente) y escalabilidad (performance del sistema). Hay herramientas para simular la concurrencia pero por lo general son caras.

Una importancia es ver la base que tiene el usuario del sistema. Hay que analizar el servicio de hardware y software que ofrece, o sea, la capacidad que tiene de respuesta y la disponibilidad que tiene. La multa no es tanto para empobrecer al proveedor sino que la multa reconocida por el proveedor, hay un reconocimiento público de que no prestó eficientemente la atención por lo menos ese día. Y eso involucra a un tercero. Plazo de entrega compatible con expectativa, o sea que las cosas estén perfectamente cuando se las necesita de acuerdo al plan. Anticipar la entrega de cargos es anticipar los pagos. Entonces, lo que ponemos (yo le pongo), que el material haya llegado en los tantos días, la relación provisoria se halla cumplido. Hubo una época en que IBM para parar a la oposición que tenía en ese momento (Control Data), te vendía papeles, decía: "nosotros le hacemos la máquina a su pedido, dígame que máquina quiere"; y los ingenieros les decían que tenían que esperar porque esa máquina todavía no existía, esto le dio tiempo a sacar la 370, paró a la oposición pero esos clientes se quedaron sin cumplir sus expectativas de plazo de entrega.

Otro tema importante a ver es el horizonte de planeamiento, es decir, hasta cuándo con la capacidad de procesamiento que tiene esta máquina, yo puedo durar, y la carga de trabajo y el aumento de actividad. Y otro tema a tener en cuenta es el apoyo específico que ofrece.

Ahora vamos a empezar un tema que es la segunda parte de la materia que hablábamos, que es la elección del software.

Nosotros tenemos muchas clasificaciones de software, yo les voy a dar una que es muy sencilla:



¿Ustedes saben lo que es el hardware de la máquina?: es algo a lo que uno le pega cuando no anda el software.

¿Cuáles son las fuentes de adquisición del software?: al principio no existía el software, todo el software tenía que hacerlo el usuario. Esas son las primeras máquinas año 58', o sea, lo prendías, era una tecla de arranque donde arrancaba a ejecutar la instrucción que estaba en una posición predeterminada de la memoria y uno le ponía el tipo de programa que era en octal. El primer soporte que se creó fueron dos soportes simultáneos que se crearon de distintas empresas, ncr la cinta de papel perforada. Bueno, la primera conclusión que sacaron los fabricantes de hardware, es que si la cosa seguía así (como en los 60'), no iban a vender más máquinas porque no iban a tener gente que la programe (utilizando lenguajes de primera generación). Entonces, empezaron a hacerse los sistemas operativos con funciones que antes tenían que programarse y salieron los lenguajes de programación de segunda generación; y luego los de 3ra. Y cuarta generación. Cada vez se hizo por un lado más fácil poder acceder a desarrollar por computación, aparecieron las carreras de informática y se hizo popular la cosa.

Las herramientas que hoy existen para producir aplicaciones, son cada vez más sofisticadas desde el punto de vista del nivel técnico de la gente que la tiene que usar, pero te da mucho más facilidades para venderle servicios al usuario.

Los segundos grandes proveedores o los principales proveedores de software, es el software hecho por los fabricantes de hardware. La empresa que más vende (en miles de millones de dólares) es IBM no es Microsoft. Sacando el acuerdo corporativo que existe entre Microsoft e Intel para uno adelantarse los cambios que le va a hacer al otro, para que pueda adaptar su software, y prohibirle Microsoft que Intel fabrique software y prohibirle Intel que Microsoft fabrique software para otro tipo de procesadores (software de base me refiero, sistema operativo). Los fabricantes de hardware tienen la ventaja que conocen su hardware, o sea, conocen la relación entre el software operativo y el software de utilidad; esto es privado no lo conoce ninguno, por eso hay pocos virus (yo no conozco ninguno) en hardware de RISC por ejemplo, porque todo el que hace hardware de RISC, HP, IBM, tienen su propio sistema operativo, entonces aprovechan mejor los recursos y los desarrolladores dialogan con los hardware y ven cómo aprovechar mejor estos circuitos para hacer mejor el software, más performance la máquina. No se olviden que todos los virus se meten a través de un archivo, pero pasan a través del hardware para poder hacer daño (tengo que pasar y leer el disco para poder destruirlo). Entonces, esto acá está reservado, por eso es más seguro.

Esa fecha no la vana a perder porque es mucha plata. Fíjense, ustedes tiene SQL Server, pero SQL Server corre solo en máquinas Intel. En el caso de IBM, tiene DB2 que es una flor de base de datos y corre en todos los ambientes (Intel, RISC, HP). Yo creo que van a quedar 3 bases de datos: SQL Server, DB2 y Oracle; y no por razones de que sean malas sino por razones comerciales, las están ahogando a las otras empresas porque no quieren tener competencia. Entonces, vos tenés DB2 en una línea, Oracle en todo lo que es Server y SQL Server en todo lo que es chico, y entre ellas no compiten. Tiene una gran ventaja, donde ellos bases de datos corporativas a nivel mundial sobre problemas, inclusive, algunos clientes pueden acceder (si piden) directamente ellos a la base corporativa para hacer resolución de problemas. Entonces, cuando el sistema operativo está out, tengo toda mi empresa out, conectándome con esa online (todas las empresas grandes tienen ese servicio) o llamando a un 0800 o una llamada de cobro revertido, y los tipos entran a la base de datos si no podes directamente, y te dicen que parche le podes que hacer para poder seguir adelante. A mí me pasó con un fabricante de software (no con un fabricante de hardware), que en Santa Fe en una empresa compramos un equipo que era el primer NCR que tenía microchan inalambrer que era una tecnología de IBM, que era la tecnología del microcanal en PC, que después se abandonó eso pero ya se está volviendo de vuelta pero con otra, el mismo concepto del canal. Resulta que santa fe operaciones no tenía ningún Unix con

la configuración de esa máquina y teníamos problemas; entonces, los tipos se comunicaban con nosotros para hacernos la reparaciones, para decirnos que teníamos que hacer para poder andar.

Otro problema que hay con esto acá que porque no se dedican empresas por ejemplo Microsoft, a hacer sistemas operativos para otra firma, porque yo en las máquinas RISC tengo fireware, le cambio el fireware porque yo soy fabricante y si no hago convenio con él lo dejo afuera.

También tenemos software de los fabricantes de software; este es el caso de Microsoft, de Santa Cruz producciones, de Computer Asociator, Borland. El principal proveedor, por supuesto, es Microsoft, segundo está Computer Asociator que vende aproximadamente la mitad de lo que factura Microsoft, ellos se dedican fundamentalmente a software de servidores no a software de clientes por eso no se conocen. Los proveedores europeos tienen prácticamente copado como fabricantes de software, todo lo que es software de base. Los proveedores norteamericanos prácticamente le robaron a los europeos lo poco que tenían desarrollado. El software de aplicación que se desarrolla en Norte América no se puede exportar a Argentina por razones de folclore digamos, tenemos distintos usos y costumbres. Más, son negocios distintos fabricar el software de base que vender software aplicativo, no conozco (por lo menos en Argentina) casos de empresas que vendan software aplicativo y les haya ido bien con el software de base como es el caso de Oracle que es un muy buen motor de base de datos pero para software de base no va porque no están acostumbrados a hacer lo que el cliente quiere sino que dicen lo que hace el software. Los proveedores europeos, en lo que es software de base, se dedican a vender los nichos que pudieron quedar pendientes, por ejemplo: software de comunicaciones como es el caso de Laneta, pero en general se dedicaron mucho a lo que es el mercado común europeo. Yo estuve por ejemplo en la URSS, en Barcelona, en el año 90' y los tipos ya estaban trabajando para la unificación de la moneda (la moneda se unifica en le 2002), los tipos estaban posicionados 10 años antes. En ese momento la ciudad que más equipos por habitantes tenía era París, porqué?: porque ellos tenían un sistema minitel de la compañía de teléfono que te ofrecía junto con el teléfono un sistema de red que era de la propia compañía. La India por ejemplo, es un país que el año pasado exportó en software 8.000 millones de dólares, una cosa no tradicional, los hindúes tienen algunas ventajas: primero que todo hindú que haya pasado por la universidad tiene que conocer inglés; segundo son más mansos digamos, ellos comiendo todos los días ya están contentos. Después tenemos los elaboradores argentinos; acá se tienen representantes de temas norteamericanos y se elabora software aplicativo, el software de exportación es muy poco. Argentina tiene una balanza sumamente deficitaria en lo que es software, no exporta más de 30 millones de dólares al año, es muy poco mercado. Para que tengan una idea Uruguay está exportando 80 y tiene menos habitantes y profesionales, en vez de exportar mano de obra exportamos mentes de obras porque educamos profesionales que nos cuestan 100 mil dólares y después se van al exterior y vuelven a los 40 años cuando ya están viejos para que los cuidemos acá.

Una forma particular de adquirir software, es adquirirlo con el hardware incluido. Porqué les digo esto?: porque hoy lo que la empresa realmente es en software, al cabo de 3 años, lo único que le queda a la empresa es la experiencia de la gente y el software que realizó. Además, hoy te permiten que vos tengas una herramienta con un entorno de desarrollo y un entorno de producción, que no tienen que ser idénticos, sino que un entorno Windows por ejemplo con un desarrollador puedes simular un entorno mainframe, un entorno RISC. O sea, hoy la tecnología te permite que vos hagas lo que querés, una vez que tengas el software vayas y compres el hardware no al revés, que primero compras el hardware y después cuando tenés el software pasaron 1 año, 2 años y no te sirve.

Y la alternativa intermedia es comprar llave en mano, comprar el hardware y el software de base y aplicativo juntos, software aplicativo andando. A veces la llave en mano resulta una llave inglesa, pesada.

¿Cuál es el problema básico que hay con el desarrollo propio?: es que la gente que administra, no conoce técnicas de administración de grupos técnicos. Yo para poder administrar un grupo técnico tengo que conocer el *meclier* (cómo es sus características laborales, como es su trabajo, sus características psicológicas, etc.). Entonces, hay un concepto que todavía tienen en las escuelas de administración, y es que la administración es una sola, y es falso. No es lo mismo administrar una iglesia que un cabaret, otra cosa es ser usuario. Entonces, se necesita conocimiento especial de administración de grupos técnicos, no es llevar una contabilidad. ¿Porqué?: porque administrar ¿qué significa?: que se cumpla con los plazos, que se cumpla con el presupuesto, que se cumpla con la calidad, que se cumpla con el mantenimiento. Cualquiera va a saber controlar si vos la ajustaste a tu presupuesto y al plan, poniendo una raya en el almanaque te das cuenta. Pero lo que no puede saber si no conoce el *meclier*, es si se está cumpliendo con la calidad de ese proceso, o no está apretando mucho al programador, este dice esto no me dijeron nada no lo hago, y cumplir con las cosas

correctamente y documentadas. La variable de ajuste de los sistemas viene dada por la calidad del sistema. Por eso, las grandes empresas pueden tener un administrador de sistemas en un sector especial; el drama se da en las pequeñas empresas que le cuesta muy caro mantener esa estructura, por eso buscan grupos externos o hay una alta rotación del personal porque creen que se están engañando mutuamente y hay fueron donde se pierde la confianza en un grupo técnico.

Entonces, yo tengo dos alternativas si no quiero hacerlo por desarrollo propio: hacerlo hacer o comprarlo hecho. Comprarlo hecho, implica comprar un paquete, una caja. ¿qué tipos de paquetes conocemos?:

1. Paquetes paramétricos o enlatados que le llaman: aquí uno personaliza ese software a través de archivos que va llenando de parámetros, pero no ve el código, un enlatado es el word (uno le da parámetros y después el formatea pero uno no puede cambiar el código), tango gestión, power point. Hay dos tipos de enlatados en software aplicativo:
  - Enlatado estático o de parámetros estáticos donde todas las constantes del programa están afuera, por ejemplo: nombre de la empresa, nombre de las sucursales, tasa de impuestos, todo lo que es constante está afuera para poder cambiarlo.
  - De parámetros inteligentes: se llaman así porque tienen pseudo lenguaje que me permite poner como parámetros variables del programa. O sea, pueden acceder a tablas, pueden hacer suma de operandos, pueden hacer restas, pueden hacer bifurcaciones. Y hoy con las memorias baratas, en realidad esos parámetros, se cargan completo en memoria esos archivos, entonces vos estás trabajando como si estuvieras adentro del código. Por ejemplo: ERP, CRM, todos los motores de base de datos permiten trabajar con procedimientos almacenados. No es tan fácil este tipo de aplicaciones con parámetros inteligentes.

Ahora, no es la solución integral para las empresas usar ese tipo de productos, porque si todas las empresas tenemos la misma información en qué nos diferenciamos. Entonces, todo lo que es operacional bárbaro, lo resuelve muy bien, pero todo lo que yo quiero para resolver para el nivel gerencial, tengo que tener mis propios programadores si quiero diferenciarme. En EEUU por ejemplo, si vos entras a trabajar en una empresa por un contrato, te dicen que terminado se contrato sino se te renueva y vos te vas por propia voluntad no por voluntad de la empresa, vos por tantos años (entre 3 y 5 años), no puedes entrar a trabajar en otra empresa del ramo.

2. User exit o salida de usuario: estos son los sistemas que trabajan con API. Las API (Application Program Interface) cuando son estándar, se dice que el software está estandarizado, por ejemplo: se pusieron de acuerdo hace algunos años las empresas más importantes de software como son Unit International, OPS, Xopen, IEEE e hicieron las normas posit. Entonces, las API permiten que en las aplicaciones portables se puedan programar cosas vinculadas que limiten el uso de esas entradas. O sea, la API es independiente a la interface de usuario, es una herramienta que tiene el desarrollador para comunicarse con el sistema operativo.

Entonces, vos puedes tener Oracle, corre bajo Spec1170, pero Oracle es portable no lo que desarrollaste bajo Oracle. ¿Vos desarrollaste bajo Oracle?: si, sigue bajo Oracle. Ustedes van a decir, pero SQL es estándar, minga!. Primero, SQL en sí es bastante estándar pero no todos los SQL son iguales porque hay cosas que no están normalizadas en sql2, sql3, por ejemplo: los caracteres de control, los mensajes de error. Y segundo, todo lo que no sea SQL, todo lo que vos tenés como procedimiento almacenado, gatillo, administración, es particular de cada base de datos, entonces tenés que cambiarlo. Hay herramientas de migración de bases de datos de una a otra que dicen que son transparentes, pero son transparentes como el vidrio, si no sabes abrir la puerta por más que viste el vidrio chocas.

3. Paquetes modulares: es impartir en su conjunto en mi sistema para venderlo por módulos, cosa de que el costo inicial para mi cliente sea menor, por ejemplo: si el cliente no vende por tarjeta de crédito, no venderle el módulo tarjeta de crédito.
4. Paquetes pre-planeados: acá se adapta el código a las necesidades del cliente. O sea, se deja en el fuente muchos mensajes para que después pueda adaptarlo.
5. Sistemas a medida.
6. Sistemas experto: en un comienzo fue un proyecto japonés que les sirvió para conocer en qué estaban los norteamericanos. Después vamos a ver cómo funcionan estos sistemas

## CLASE TEÓRICA DEL 11/06/01

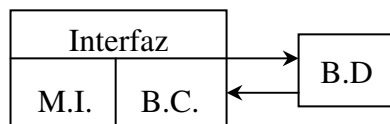
### Tipos de paquetes de software

Existen los paramétricos o enlatados, los User exit, los modulares, los adaptados al cliente, a medida y los sistemas expertos.

En general los sistemas expertos son parte de lo que se conoce como inteligencia artificial. La inteligencia artificial se define como un conjunto de técnicas que tratan de resolver los problemas vinculados con el proceso de información simbólica, utilizando para ello métodos heurísticos. Este es el conjunto de lo que en 1956 "Mclar" definió como inteligencia artificial. Es decir que este término ya tiene muchos años. El optimismo era que con la computadora se podía hacer inteligencia artificial. La computadora es indispensable.

En general, para lo que no están familiarizados con esta terminología, el enfoque heurístico, está en la naturaleza de la persona, es un enfoque descubridor, la persona decide a través de un sistema de descarte de opciones, a través de un sistema intuitivo de probabilidad. El enfoque algorítmico, que sería lo contrario al enfoque heurístico, es un enfoque que explora todo el espacio, es decir, no deja posibilidades o alternativas afuera. Si se hace una búsqueda dicotómica en una tabla, si el algoritmo está bien, o se llega a la verdad o a la mentira, a un sí o no, está o no está en la tabla. No se puede equivocar. En los sistemas heurísticos es posible equivocarse, como por ejemplo un sistema de diagnóstico médico, o sea el médico va descartando posibilidades en el diagnóstico, pero puede estar equivocado, no necesariamente todo va a ser cierto. Los tres tipos de aplicaciones más conocidos en inteligencia artificial son:

- **Redes neuronales:** las redes neuronales aprenden, como las neuronas. Va aprendiendo en función que va actuando.
- **Sistemas por casos:** sea una sencilla función de una base de datos por cargos. Por ejemplo para dar créditos en un banco en forma automática. Yo tengo 100 mil casos de créditos. Entonces digo que personas coinciden con esta solicitud, por los niveles de ingreso, por el grupo familiar, por la zona donde vive. Entonces tomando de esa base los casos que más se asemejan a este particular que estoy tratando de analizar, digo si hay más de un 70 % de aciertos, es decir la gente cumplió con el crédito y está con casos similares otorgo el crédito, por 80 por 90. Triplico la tasa de riesgo. Se analizan los casos que se presentaron parecidos a esos. Esto requiere que de acuerdo a la cantidad de casos va a ser más fuerte mi sistema de inteligencia artificial que si tengo menos. O sea que este sistema va aprendiendo en función de que voy poniendo casos.



- **Sistemas expertos:** Un sistema experto es tener las reglas que cumple el experto programadas. Las partes de un sistema experto son la interface, el motor de inferencia y base de conocimientos. Después tenemos la base de datos tal cual la conocemos nosotros, por ejemplo la base de créditos. La base de conocimientos es el dominio de aplicación del sistema, o sea si yo estoy en el sistema médico de diagnóstico general, la base de conocimientos serán todas las posibles topologías que puedan existir y su posible situación ... El motor de inferencia contiene las reglas que va a aplicar el sistema experto para que al cargar la base de conocimiento pueda llegar a una dirección. Se requiere para poder hacer un sistema experto como mínimo una carga inicial. No puedo hacer un sistema experto de diagnóstico médico sin tener conocimientos de....Se requiere tener un especialista en conocimientos, el ingeniero de conocimientos que pueda saber como puedo sacar unas reglas de decisión para hacer el motor de inferencia al instante. Se necesita tener una herramienta para poder programar. Y hacerlo. De acuerdo a la cantidad de reglas se clasifican los sistemas expertos, cuanto más reglas tienen más sólidos es el motor de inferencia. Si tenemos hasta 50/100 reglas tengo un prototipo de demostración, no se puede usar, está a nivel de prototipo para demostrar. Si tiene entre 200 y 500 es un prototipo de investigación. Si tiene entre 500 y 1000 reglas es un prototipo de campo. Si tiene entre 1000 y 1500 reglas y un usuario es en un prototipo de producción, y si tiene muchos usuarios es un sistema comercial.

Pasó el 50 y no se solucionó el problema de que las máquinas eran lentas. Pasó el 60 y seguía lo mismo. Pasó el 70 y en el 76/77 ya decían en la ASM va a pasar mucho tiempo hasta tanto los sistemas expertos puedan ser una realidad. Y en la década del 80 los japoneses sacaron en 1984/85 el computer quinta generación. Estaba la tercera generación de computadoras y no iban a pasar a la cuarta generación sino directamente a la quinta, iban a hacer un salto tecnológico. Hecho que alarmó muchísimo a los norteamericanos. Hubo toda una política de fomentar este proyecto de quinta generación, donde daban becas a latinoamericanos, europeos, africanos para ir a ver como estaban trabajando en quinta generación. De Argentina habrán ido unas 200 personas más o menos. No solo fueron del área informática sino también de medicina. Porque querían aplicarlo para la medicina. Y los norteamericanos se asustaron, entonces empezaron a mandar gente que estudiaran ahí y vieran que era lo que estaban haciendo los japoneses.

PROLOG (PROceso LOGico) es el primer lenguaje que los japoneses estaban desarrollando para interpretación... Los japoneses abandonaron el proyecto hará cinco años, pero ya estaba caído hace diez. Hay muchos lenguajes que permiten trabajar para armar sistemas expertos. El de moltok es el más conocido, el de prolog, el kee (ambiente de ingeniería-conocimiento), el LISP (1958).

Y acá se dio un pequeño problema, nadie se imaginó en la década del 80 que las máquinas iban a tener un descenso en su costo, nadie se imaginó que el costo por disco iba a bajar 150 mil veces en 10/15 años. Nadie se imaginó eso. Entonces el hecho de empezar a explorar mucha información se empezó a que el método algorítmico que podía hacer más cosas era el que se había pensado antes. En traducción por ejemplo, los productos comerciales que hay son todos algorítmicos, exploran todo el espacio. Productos que traducen no a nivel de palabra ya, sino a nivel de frase y hacen cierta relación entre las frases. IBM ofrece para sitios de Internet traducir del inglés al español y viceversa, al italiano, al francés, al coreano, al japonés, automáticamente. Porque sino uno agarra un sitio en inglés y por más que se sea un experto por ahí quedan algunas dudas, y es preferible verlo en español. Más si hay que poner plata. Y son algorítmicos. A pesar de que hay problemas de traducción se planteó ya, antes de que aparezca la palabra inteligencia artificial en la década del 40, ya se empezaron a hacer máquinas que traduzcan. La beep loop, la máquina que le ganó a Karparosv, era una máquina que era secuencial y algorítmica. Esta máquina podía ver entre los tiempos que tenía y los tiempos que lo dejaba a Karparosv pensar, podía ver hasta 50 jugadas para adelante, completas de todo el espacio. Una persona muy inteligente puede llegar a 25. Pero además de eso tenía el análisis de casos, tenía todas las partidas del mundo que pudieron juntar como base de datos para ver cual era la actitud que se había tomado en algunas partidas el campeón del mundo. Por eso cuando Karparosv hizo una jugada tonta para ver, como pasa con las máquinas heurísticas de juego, se fue al diablo porque lo mató el otro le salió con 4 torres por allá y lo liquidó. De todas maneras cuando los espacios son extensos de analizar, a los métodos heurísticos no hay con que darle. Si yo quiero traducir voz a voz, no tengo otra forma que pensar en métodos heurísticos. El traductor arranca cuando se trabó en la frase engancha la siguiente. Las reglas tienen que decir exactamente eso, cuando no encuentra como sigue la relación de la frase, tiene que esperar la siguiente y empezar de vuelta.

Tampoco hay posibilidades de explorar todo un espacio en lo que sea reconocimientos de formas. Esto sirve para hacer un piloto automático de automóvil. Yo le doy las calles donde quiero ir, la dirección, con un software adecuado que tenga mi vehículo y me lleva el vehículo. Tiene que ser heurístico. Este tipo de cosas se van investigando con sistemas expertos.

Otra orientación que hubo en los 80 se desarrollaron muchos sistemas expertos, por ejemplo en España para el área comercial, y que con el paso del tiempo se han ido enriqueciendo. Se utilizan para:

- ver posibilidades de riesgo en un crédito,
- ver la capacidad del contribuyente,
- determinar si una persona puede ser autónoma o tiene que ser empleado en dependencia, la situación laboral es distinta y la impositiva también.
- ver posibles sectores que podrían que tributar y no tributan.

### **Políticas de desarrollo de un paquete**

1. Adquisición de un producto en torno a un cliente: por ejemplo, me llaman de una casa comercial para que haga un sistema de sueldos. Entonces uno dice voy a tratar de hacerlo lo más amplio posible para que pueda entrar cualquier empresa de comercio. Entonces los nombres de las empresas los pongo como constantes en el programa, como un archivo, variables, etc. La ventaja con esto es que alguien te está financiando de arranque y te está pagando conocimiento

gratis porque te paga y vos aprendés. Las desventajas son, primero que puede tener un cejo muy particular vinculado a ese cliente y vos no te das cuenta porque no conoces el gremio y segundo que hay que entregar las cosas en un plazo, y como hay un plazo a veces se pierde calidad, y la calidad en este caso es el objetivo que hay que plantearse, porque no puedo decir esperame dos meses más y yo te lo entrego. Por ejemplo el paquete Tango (hace 10 años atrás) empezó a través del entorno cliente, y vos te dabas cuenta porque los controles no los entendías, tenía que venir un informático a resolver el tema para instalarlo.

2. Desarrollo interno: pensar un mercado vertical potencial independientemente de cualquier cliente. Por ejemplo Sistemas de Contabilidad Bejerman. Cualquier contador lo instalaba solo. La adquisición en desarrollo interno lo pueden hacer las casas de software, porque pueden aprovechar los tiempos muertos de sus empleados, los programadores. Como tienen que pagarle igual todos los meses y a veces no tiene para hacer cosas, en los tiempos muertos que están esperando la respuesta de un cliente, etc., se ponen a hacer ese tipo de desarrollo. El riesgo que se corre con esto es que directamente no se pueda vender el producto.

### **Política comercial del software**

Es la más entreverada y de variantes posible. Por ejemplo, compro una casa y la compro con todas sus edificaciones salvo que haya una cláusula de algún derecho real, por ejemplo que no pueda edificar más de tantos metros porque el vecino de al lado compró el espacio aéreo, hizo una torre y entonces para que yo no le quite respiración me compró el espacio aéreo. Es un derecho que el vecino tiene por toda la vida. Nadie puede construir más de tantos metros en ese espacio. Pero sino, normalmente lo que ocurre en estas cosas uno compra toda la edificación.

En el software hay muchas vías de comercialización. La más fácil es la de Microsoft y el producto de oficina. Te compras la caja, te la llevas a tu casa y la puedes usar hasta el año 3000. No hay ningún tipo de problema. En el nivel empresarial, no personal, Oracle, por ejemplo, comercializa su base de datos de la siguiente manera: usuario nominal, (eso incluye el ID, un número de usuarios en una lista de usuarios, no usuarios concurrentes), y si vos tenés 200 usuarios colgados tenés que pagar licencia por los 200 o por los megahertz que tenga la máquina, estos MHz si son máquinas RISC se multiplican por 1.5, el equivalente (300Mhz RISC=450MHz Intel). Y tenés el application Server por un lado que tiene un precio y tenés el motor de base de datos por el otro, que tiene un costo diferente.

Si vos tenés, por ejemplo, una máquina Intel con dos procesadores de 400 MHz cada uno, Oracle me obliga a comprar y a poner el application Server y el motor de base de datos. Como yo no puedo decir estos 400 MHz para una cosa y los otros para la otra, tengo que pagar 800 MHz por application Server y 800 MHz por motor. Ahora si tengo dos máquinas, pago 400 MHz por application Server que pongo en una máquina y 400 MHz por el motor que pongo en la otra máquina. Cuando las máquinas tienen código de CPU por hardware (son privilegiadas) ellos te dan el software propicio para correr en esa máquina, o sea chequea el número interno de CPU.

Otra forma de comercializar es por tiempo de uso. Todo lo que sea proceso a través de Internet. Yo te doy un hosting de aplicación y me pagas por el tiempo de uso del hosting y de los programas. Esto lo inventaron para, por ejemplo si yo tengo una máquina grande, y vos tenés una máquina con un procesador Informix y tenés todas las aplicaciones en Informix, yo quiero venderte pero vos no me puedes comprar para esa máquina grande lo que cuesta porque al principio tenés un coto enorme, entonces lo que dicen es vos instalas el motor en la máquina grande y yo cada vez que vos ingreses al motor yo llevo un contador adelante y cobro por eso. Como si fuera la luz, un medidor de uso del software. Cuando llega un momento que a vos te convenga pagarme la tarifa normal, la pagas. Para hacer aplicaciones por Internet también se está usando éste método, yo no puedo cobrarle a un tipo que se conecta 50 horas lo mismo que a aquél que se conecta dos horas. El mayor problema que hay hoy por Internet es que no hay forma de estipularle el cobro, siendo vos el prestador de servicio. En otros países si hay forma de estipular el cobro porque la compañía de teléfonos es la que te lo pasa.

Otra forma es alquiler: todas las empresas tienen un servicio de upgrade y de update, y de línea caliente (uno puede llamar las 24 horas del día por cualquier problema, los 7 días de la semana) y ese servicio, las actualizaciones o sea que te den las nuevas versiones del software, y que te den servicio telefónico de atención está costando entre el 15% y el 25% del valor del producto por año. ¿Que pasa si vos no lo haces?. Si no lo haces y estás en una empresa grande o en la administración pública y a vos se te cae el sistema operativo, se te cae la base de datos, tenés que llamar al proveedor, pedirle un presupuesto, el proveedor tiene que hacer un diagnóstico y te cobra por el diagnóstico, tenés que autorizar el cobro del diagnóstico, después te presupuestan en función, después tenés que autorizar y

estás dos meses, no puedes, entonces te obligan a tener este tipo de servicio. ¿Y quien lo tiene? Y todas las administraciones públicas del mundo y todas las grandes o medianas empresas. Los que no pueden decidir con el bolsillo, con la caja del bolsillo. Un banco no puede decidir con el bolsillo. Además a vos se te cae la base de datos o el sistema operativo, tenés el sistema down (no trabaja nadie).

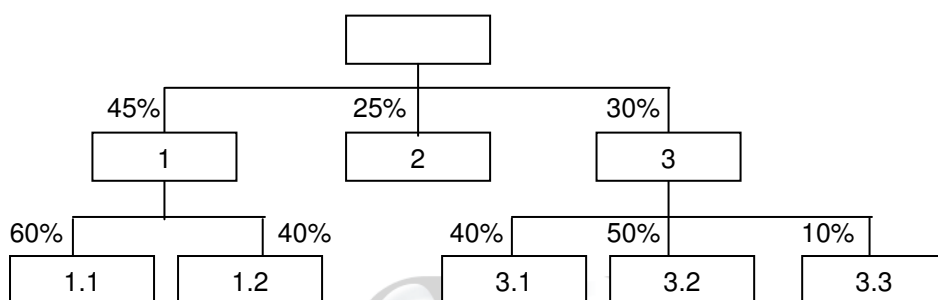
Otra forma de comercializar es por uso concurrente, no puede haber más de 15 usuarios conectados, o sea no pueden estar login más de tantos usuarios.

Registrar un software, ¿por qué hacerlo?. Simple, alguien nos roba la idea, la registra y después nos puede denunciar porque nosotros estamos en infracción.

**Criterios generales para la selección de un paquete de software**

Vamos a ver un método general, en el que los criterios son válidos para cualquier tipo de paquete. La evaluación se hace en tres fases:

1. Definir criterios de evaluación en forma top-down
2. Se aplican factores de peso a cada uno de los criterios. Los factores de ponderación son absolutamente arbitrarios, se analiza el paquete, se califican los criterios elementales de 1 a 10
3. Se evalúan los paquetes.



**Ejemplo**

Está vinculado a los porcentajes, pero no así a los criterios.

Criterio		Factor	Criterio		Factor
1	Desarrollo	15%	3	Resultados obtenidos	30%
1.1	Estructura	20%	3.1	Calidad	20%
1.2	Flexibilidad	30%	3.2	Cumplimiento de necesidades	50%
2	Operatividad	20%	3.3	Validez	30%
2.1	Flujo de información	30%	4	Adaptabilidad	15%
2.2	Controles	40%	4.1	Técnica	50%
2.2.1	Integridad de datos	40%	4.2	Económica	50%
2.2.2	Validación de datos	60%	5	Mantenimiento	15%
2.3	Rendimiento	30%	5.1	Actual	50%
			5.2	Futuro	50%
			6	Referencias	5%

Cada cosa de esto es a modo de título. Hay que hacer una encuesta y desarrollarlo. Cual es la estructura, como está hecho el software, cual es el lenguaje de programación, cuales son los métodos de análisis, muéstrame un ejemplo de un sistema, cómo esta hecho, cuáles son las normas de programación que tiene, cuál es el grado de flexibilidad que tiene, que cosas no puedo hacer y que puedo hacer. Explotarlo. Cuando decimos mantenimiento futuro cuantas veces hizo el dir, cuándo saco el producto, cada cuánto hace los cambios, o sea hay que tener todo un listado, no es decir mantenimiento.

El propósito de esto es no comprar un software solo por la señorita que te lo muestra y la interface. Hay que ver lo que hay detrás de la interface. El fabricante o vendedor de software trata de que vean la interface nada más, no carga mucho. O sea es rapidísimo pero tiene tres clientes, ¿que pasa si le pongo 10 mil como quiero yo?. Además llegamos a un número, o sea cada paquete que analicemos vinculado a esa compra va a tener un número de 1 a 10. Si bien los criterios son subjetivos,

el comité evaluador da notas como cuando uno toma un examen de historia, hasta donde uno puede saber si es exacto o no lo que te contestan.

La siguiente cosa es que uno puede entrar a hacer variaciones, por ejemplo si alguien saca 0 o menos de 4 en algún criterio queda fuera. Estaba bien hasta el año 1999, pero no anda para el año 2000, no sirve. Segunda cosa que puede hacer uno es decir bueno todo aquél que tenga menos de 7, por más que no tenga ningún aplazo tampoco pasa, o sea que por lo menos tengan el 70% de nota. Uno decide una cosa además, la evaluación técnica entra en un juego de binomio con la evaluación económica. La evaluación técnica puede valer el 50% de la cosa y la económica el otro 50%. O podría ser 80% y 20%. O todo aquel que sacó más de 7 y no fue aplazado puede cotizar y no me interesa la diferencia, me interesa el menor precio. Ya no entra a jugar lo económico en conjunto, sino lo más barato después. Todas son posibilidades. En el disquete hay un ejemplo de ficha de evaluación, de cómo se evalúa y que porcentaje hay para cada cosa.

Lo importante que es lo que hay que buscar que no se debe adaptar la organización al programa. El fabricante de software trata de venderte el software tal cual está y trata de convencerte de que es lo que vos necesitas. Y si vos querés cambiar la organización es otra cosa. Primero proponete el cambio de la organización y después busca el software. No pretendas si la organización anda bien así cambiarla por un problema del programa. Ojo que a veces uno puede adoptar una organización intentando empezar de cero a partir de un programa. Yo quiero poner un negocio de alquiler de video y voy y me fijo como funciona una casa de video y adquiero el software y me organizo así porque el software está bien organizado.

### **Ventajas y desventajas de los paquetes**

1. Costo y tiempo de implantación: se compra un paquete y ya se puede instalar. De lo contrario hay que hacer un análisis, un diseño, prueba, manuales, etc., y seguir con todos los pasos hasta tanto se pueda implementar. Lleva más tiempo.
2. Invertir tiempo en la búsqueda del paquete: hay que hacer las planillas de guía para evaluación. Hay que ver cuales son los posibles proveedores, hay que probarlo.
3. El paquete cubre únicamente parte de las necesidades: nunca vamos a encontrar un paquete que nos satisfaga mentalmente en el 100% de lo que quisiéramos. Siempre hay que considerar que siempre algo hay que agregar.
4. Existen dificultades de adaptación de los paquetes a una necesidad específica: tengo que modificar el paquete, por ejemplo Tango, la unidad de medida de la entrada es diferente a la de salida. Tango no permite tener distinta unidad de medida de entrada y de salida del producto. Si le pedís a Tango que te arregle el paquete y te cuesta 4 o 5 veces más caro que el paquete en sí. Eso no sería lo grave. El tema es cuando hay que hacer un cambio mandatario, a quien Tango le va a dar permiso primero, al caso que tiene el paquete general. Primero hace lo de Capital Federal, después va a llegar a Santa Fé. Si el paquete no te sirve como está, dejalo.
5. No existe un servicio de mantenimiento adecuado: si el mantenimiento está bajo mi administración, yo lo tengo en la empresa, yo los pongo a los programadores les doy un sándwich y les digo hasta que no terminen no se van, pero yo no puedo hacer eso con una empresa de terceros. Tengo que hablarlo, depende de la prioridad de trabajos que tengan ellos, ver los compromisos que tengan ellos. Tengo serios problemas.
6. Poca documentación: esto se debe a que en Argentina no hay quien revise la documentación. Entonces quien lo desarrollo y no documenta es obvio que para él todo es obvio. Entonces documenta poco. En EEUU en los centros de software existen los que se llaman revisores de la documentación, y son los más brutos de todos. Si lo entienden ellos lo entiende cualquiera.
7. La construcción del sistema no es eficiente a mis necesidades: por ejemplo en lenguaje Assembler sumar dos campos binarios ocupa 4 bytes, en lenguaje Cobol ocupa 2 Kbytes, contra 2 bytes, es considerable. ¿Por qué? Porque los tipos tienen una rutina, un subprograma que cualquier tipo de suma la resuelve por ella, pasándole los parámetros. Cuando uno hace un Query en una base de datos puede estar cargándole 4 o 5 millones de instrucciones de eco. Las instrucciones que se ejecutan a partir de ese Query pueden ser 4 o 5 millones. Cuando hago un sistema que trato de hacerlo lo más portable y lo más escalable posible evidentemente todo el problema va a estar en ..., por eso es más eficiente. Los norteamericanos no tienen problema,

ellos te dicen compra más máquina, pero claro se realimenta la economía de ellos pero no la nuestra. El esquema es distinto.

8. Los problemas aparecen en la implementación: es la contracara del ítem 1. Si no hice una buena selección de paquete me doy cuenta de que no me sirve cuando lo estoy implementando. O que tiene un montón de problemas. No lo puedo detectar antes, cuando lo seleccioné. Después que lo compré no tengo la instancia del relevamiento, firmar el procedimiento, las normas funcionales, el diseño detallado, las pruebas, no tengo esa instancia que tengo en el esquema clásico de desarrollo.
9. El crecimiento del producto puede estar limitado.
10. Bajo costo: hoy lo único que se desarrolla por terceros, por administración propia es el software aplicativo, que no es de propósito general.

### **Consecuencias de una mala elección de paquete (de querer elegir un paquete y no encontrar ninguno)**

1. Pérdida de tiempo en la búsqueda
2. Costo de adaptación altos.
3. Mantenimiento dificultoso o caro.
4. Dificultades en la implementación.
5. El producto puede tener un ciclo de vida corto.

Si no consigo el paquete que busco tengo dos alternativas (para el desarrollo de mi producto):

- a) Por propia administración.
- b) A través de terceros.

En ambos casos debo fijar criterios, bajo que condiciones se va a hacer ese desarrollo. Tenemos:

#### **Criterios de organización:**

1. Que estructura queremos del sistema y que flexibilidad pedimos en los resultados.
2. Nivel de documentación.
3. Guías estimativas para el cálculo del tiempo y del costo. Esto depende de mi incertidumbre. Si yo tengo mucha incertidumbre para un proyecto no puedo pedir que me lo coticen, porque no puedo cotizar la incertidumbre, uno no tiene idea del alcance. Si yo lo cotizo caro y el otro lo acepta me puedo estar enriqueciendo indebidamente, porque después no pudo justificar ese costo. Y si es al revés, lo cotizo menos, lo estoy favoreciendo. Entonces se hacen estas guías estimativas de tiempo y costos y se trabajan con costos abiertos, y después a fin de mes se aprecia lo realizado, se dice cuanto se trabajó, cuantas horas, se pone un valor y se cobra. Amerita mucha confianza.

#### **Criterios de implantación**

- a) Tiempo necesario para su puesta en estado operativo
- b) ¿Cuál es el plan de capacitación?
- c) ¿Cómo se va a realizar el mantenimiento?
- d) ¿Qué posibilidades de adaptación tiene?
- e) ¿Cómo es el plan de migración?

#### **Criterios para asegurar la calidad del producto**

1. Globalización: este fenómeno era el de bajar las barreras arancelarias y trajo problemas fundamentalmente a los norteamericanos, japoneses y alemanes porque desde el mercado asiático, tanto del cercano como del lejano oriente, le llegaban productos a bocha, en detrimento de sus propios productos. Entonces como no podían fijar aranceles, allá hay un dumping (trabajan distinto), fijaron una nueva barrera arancelaria que se llaman normas ISO de calidad. O sea yo compro tus productos si vos me cumplís con estas normas de producción. Por ejemplo los coreanos cuando montaban sus barcos los llevaban con las bodegas llenas de auto partes de autos, y lo que hacían era en el transcurso de llegar a Europa armaban los autos. Como en altamar no hay ningún tipo de policía y los marineros trabajan mucho cuando suben y cuando bajan del barco, no había problemas en poner más gente y ponerlos a armar los autos mientras la gente estaba trabajando. Esto hacía que los autos que eran de la misma marca coreana que se fabricaban en Alemania eran un 25% más

barato. Entonces los tipos fijaron como norma de calidad que el tren de armado de la línea de montaje debería tener tal nivel de vibración, que en el barco no se cumplía.

Por ejemplo la carne que tenga enfermedad animal y que no esté demostrado absolutamente que no puede en algún futuro afectar a la carne humana, no puede pasar. La aftosa hasta antes de que exista la ISO 9000 para el control de calidad de la carne, se vendían los animales con aftosa a Europa y a cualquier lado.

Y estas normas que fueron desarrolladas por estos tres países fueron rápidamente adaptadas y utilizadas para controlar la calidad de los productos.

¿Cómo puede ser que la misma droga de distintos laboratorios tenga dos precios distintos, diferencias por el 100%?. Es muy fácil. Una de las monodrogas que constituye el producto está homologada por las ISO 9000 y las otras son de la India, que no está homologadas por las ISO 9000. A Europa y EEUU no pueden entrar, pero acá sí. La diferencia de costos está en el proceso de producción.

Las normas ISO 9000 que constituyen un marco de trabajo para desarrollar, fueron adoptadas para el proceso de desarrollo del software. En general las importantes de destacar para el desarrollo del software son la ISO 8402, ISO 9000, ISO 9004.

**ISO 8402:** es el vocabulario de la gestión y aseguramiento de la calidad. El diccionario. Ponerse de acuerdo norteamericanos, japoneses y alemanes en los conceptos es muy importante. Para nosotros mano es desde la muñeca, para los judíos desde los codos.

**Las ISO 9000** son directrices para la utilización de las normas ISO 9000. La 9001 trata sobre calidad en el diseño, producción, instalación, y servicio post-venta. Esta es la que más se acerca al proceso de producción del software. Porque tenemos diseño, producción, instalación y servicio post-venta (mantenimiento). De todas maneras como estaba prevista para la producción de servicios en general, pero no para software en particular salió la 9000.3.

**ISO 9000.3:** son directivas para aplicar la ISO 9001 en el desarrollo, provisión y mantenimiento del software. La 9000.3 te dice como aplicar la 9001. La otra norma importante de la serie ISO 9000 para nuestro tema es la ISO 9004, que habla sobre reglas generales para la gestión de la calidad y elementos de un sistema de calidad, cómo tenés que organizar la calidad. No solo dice qué tenés que controlar sino como tenés que controlar, como debe trabajar el grupo de calidad.

En síntesis tenemos que tener la 9000, la 8402 por si tenemos alguna duda con alguna palabra, la 9001 y la 9000.3 para ver que podemos pedir, y la 9004 para organizar el grupo de calidad. Otras normas por ejemplo son: **ISO 9002:** calidad en la producción y en la instalación. Esto se presta más a la industria del armado. **ISO 9003:** calidad en las inspecciones y en los ensayos finales. Se aplica mucho en las industrias químicas. Después están las normas **ISO 10011.1/.2/.3**, que hablan sobre auditoria de los sistemas de calidad, cómo auditar los sistemas de calidad. Existen otras normas anteriores y posteriores a las de ISO, que hablan sobre el software. El equipo que trabajo mucho es el de la IEEE.

**IEEE 729:** estándar sobre terminología en ingeniería del software.

**IEEE 730:** estándar para asegurar calidad en el software.

**IEEE 830:** estándar sobre como definir aplicaciones.

**IEEE 983:** guía para asegurar el planeamiento de la calidad del software.

Hay una norma francesa Z67-130 que habla sobre recomendaciones para realizar un plan de calidad del software. Lo que hizo la ISO 9000 fue impulsar el tema de calidad que antes estaba, pero no estaba impulsado a nivel comercial.

### **Aseguramiento de la calidad**

Antes la comprobación de la calidad se hacía al final. En el desarrollo de un software decimos prueba del sistema. Lo hacemos al final. Lo que propone acá es asegurarse la calidad siguiendo todo el proceso de desarrollo del software, o sea haciendo control en cada etapa de desarrollo, y no en un punto final.

El problema de calidad es un problema de todos, no hay problemas que no le competan al usuario. Al usuario le compete la calidad porque el producto es para él. Es como decir el dulce de leche tiene caca y me lo comí yo, pero la culpa es de Sancor, y que hago con eso, fui yo el que se comió la caca no Sancor. Entonces es problema mío asegurarme la calidad. Es un problema de los colegas también. Si el Instituto se compromete y hace un fracaso también nos involucra a nosotros por más que no tengamos nada que ver. Pero ante la comunidad estamos vistos como el instituto.

Calidad es un problema de la comunidad. Porque si yo hago un software para rentas y esta hecho mal o es muy complejo hago trabajar de más o de menos o mal a los contadores, entonces también es un problema de ellos. No es un problema solamente del grupo que desarrollo el software.

No se miden las personas, se miden los procesos. Cuesta mucho tomar conciencia de que no se deben medir las personas. Tengo que medir los procesos para mejorarlos porque la única forma de saber si mejoro es por mediciones. No hay calidad sin registros.

Calidad del punto de vista subjetivo: satisfacción del cliente.

Calidad del punto de vista objetivo: cliente insatisfecho. Si vos encargas un software y tiene muy pocos defectos, que le vas a discutir que está mal hecho. Vos pediste una porquería embromate. Es como cuando vamos a comer, si vos pediste una sopa y decís esta fea porque querías comer asado, pero pediste sopa, embromate.

### Calidad del software

- a) Sujeción a especificación, tender a hacer lo que se pide, no lo que se imagina uno. O no hacer lo que se pide;
- b) Costos ajustados;
- c) Fecha cierta de entrega;
- d) Cantidad de errores encontrados por cada 1000 sentencias fuentes o eventos que generan sentencias fuentes entregadas al usuario. Yo desarrollo el software y yo te aseguro que bajo estas condiciones, yo te cumplo con las especificaciones, te lo entrego en plazo, no te voy a cobrar de más, pero tenés que admitirme que esto es bueno que tiene menos de tantos errores por cada 1000 sentencias fuentes. Porque errores va a tener. Yo no puedo asegurar que no tenga errores. Hay contratos de calidad que fijan el tiempo promedio entre dos errores no puede ser inferior a tanto, no pudo tener un montón de errores. El 80% de los errores sobre software entregado son fallas en las especificaciones.

### Forma de hacer calidad

Testeo e inspección, siempre estar probando en todas las etapas. A través de los sistemas de aseguramiento de calidad se pretende acortar en el software la diferencia de brecha de crecimiento que existe entre la industria del hardware y la del software. Hoy una PC es más confiable que un mainframe de hace 10 años. En costo la primera PC con disco rígido y monitor monocromo valía \$10000. Sin embargo la primer planilla de cálculos y el procesador de textos valía \$200. La diferencia con lo que tiene el Office y la diferencia de costo como se aminoró el costo del hardware en relación al software no es lo mismo. Y los errores siguen apareciendo en el software.

Para hacer calidad se requiere inversión. ¿Cuál es el costo de la no calidad?. Algunas empresas, sobre todo las norteamericanas que miden todo, hacen calidad porque el costo de no calidad es más alto. La calidad es un círculo de mejoramiento continuo. No es un cambio brusco de reingeniería. Calidad es tratar de ir por mejoramientos sucesivos.

Una estadística proporcionada por un desarrollador de la NASA decía: hicieron una encuesta a 163 empresas que compraron paquetes de software en los EEUU en 1991. El 47% de esos paquetes fue entregado y nunca utilizado satisfactoriamente (si pero no era lo que nosotros pensábamos). El 29% software pagado y nunca entregado. El 29% fue abandonado, el 3% fue utilizado después del año, y el 2% utilizado tal cual.

Se debe convenir entre desarrollador y cliente los factores de calidad. Que cosa yo tengo cumplir, debo fijar un número.

- ★ **Exactitud:** está vinculado a la performance de la exactitud. Yo quiero tener un tiempo de respuesta menor a tantos segundos para el 90% de los casos.
- ★ **Fiabilidad:** porcentaje de fallas. Es considerado bueno en el mundo tener dos errores cada 1000 sentencias fuentes o eventos que generen sentencias fuentes de código entregado. La NASA estaba en el año 1996 en 1.7.
- ★ **Eficacia:** ocupación de recursos, que recursos necesita para producción para que cumpla con la performance, hay que definirlo antes de poner en marcha el desarrollo. Martín decía que los desarrolladores y los generales son iguales no se fijan en costo ni en tiempo. Siempre es bueno contar con algo a priori y no a posteriori, yo le pongo mas memoria RAM a una máquina y voy a tener mejor tiempo de respuesta.
- ★ **Integridad:** coherencia en los datos. Mismo tratamiento con el mismo código.

- ★ **Testabilidad:** las pruebas puedan repetirse. Capacidad de poder repetir la prueba. Es muy importante porque yo puedo asegurar como productor del software que bajo determinadas condiciones de prueba hubo errores. Entonces cuales son las condiciones de prueba que usaste.
- ★ **Flexibilidad:** que exista un archivo de configuración. Una administración de la configuración.
- ★ **Mantenibilidad:** vinculado a la vida útil. Cuanto más mantenible sea el software más tiempo lo voy a poder tener en producción, sin tener que cambiarlo.
- ★ **Portabilidad**
- ★ **Reutilización**

Se calcula que hay que tener un responsable en calidad cada 7 personas. Por ejemplo una de las normas que se utilizan para las normas ISO 9000, es la IEEE 729, que es la norma sobre como hacer la prueba. *(Tener en cuenta el Libro de Meier El arte de hacer pruebas, que le gusta al profe).*

Hay certificación a ISO 9000. Cada país tiene un organismo que certifica a ISO 9000, acá en Argentina está IRAM. Instituto Argentino Racionalización de Materiales. Ellos tienen grupos de investigación, trabajan con grupos internacionales de IEEE fundamentalmente. Atrás de esos grupos hay promociones universitarias, consultores, etc. Y atrás de esos grupos hay empresas. Si uno quiere certificar ISO 9000 para la industria cementera, ellos ya tienen una consultora que está vinculada porque lo conocen, donde pueden certificar la elaboración de cemento.

Uno puede certificar el conjunto de la empresa, un departamento de la empresa o una cadena de producción. Una cadena de producción puede ser un software. Este software está homologado según la norma ISO 9000. Hay un organismo que lo certifica, esto es caro. Lo que se hace normalmente es trabajar con las normas ISO 9000, pero siempre tiene un organismo que lo certifique.

¿Por qué es importante hablar de flexibilidad en cuanto a gestión de la configuración?

La gestión de la configuración es un repositorio en el que están todos los referenciales del proyecto. Un referencial del proyecto es una PC instalada, un usuario, un documento, un programa, un DFD, todo lo que esté vinculado a mi proyecto, hardware y software, lo almacenamos en ese repositorio. Tener en ese repositorio trazabilidad. Es decir entrar por cualquier lado y tener todo vinculado a ese documento a esa PC particular, a ese usuario particular, a ese software en particular. Administro las versiones del software, saber cuando se inició, en qué fecha, quién lo programó, cuando le di Ok, cuando fue comunicado al usuario, cuándo fue puesto en producción, cuándo fue pedido un cambio, cuándo se realizó el cambio, hasta que fecha tiene vigencia la versión anterior. Es decir que tengamos una navegabilidad por la configuración en forma total. Eso permite montar un help desk, una ayuda de escritorio en forma remota. Yo sé que me están hablando y yo sé que usuario es, sé que perfil tiene autorizado para hacer, y yo puedo repetir el perfil en mi puesto de trabajo y puedo repetir el error o no a ver si se repite la situación que se le da. Con asistencia remota de problema puedo ganar muchísimo. Además me permite administrar los cambios de forma más fácil. Porque yo sé que cosas tengo que tocar y quien las tocó la última vez. Qué afecta, a quién impacta ese cambio. Voy a tener los componentes, sus atributos y la dependencia.

Se necesitan herramientas metodológicas para garantizar la calidad. El plan de calidad debe tener las siguientes tareas:

- ✓ Inspecciones a los productos: son reuniones que tienen el objeto de evaluar un elemento producido, por ejemplo un documento, un diseño, la programación de un módulo. Participan en esta reunión el responsable del objeto, gente del grupo de calidad, colegas, y personal de prueba. ¿Cómo funciona esto? Cada martes y jueves se hacen reuniones de calidad de dos horas para inspección. Entonces cada persona que tenga algo que quiera que se lo apruebe lo lleva y lo presenta. Por ejemplo un diseño de formulario.
- ✓ Reuniones periódicas: se hacen siempre, en forma periódica, los lunes, los jueves, cada 15 días. Participan todos los anteriores más el jefe del proyecto.
- ✓ Revista de fin de fase: son reuniones que tienen el propósito de decidir si se puede pasar a la siguiente fase. Además de los anteriores participa el cliente. Cualquiera que sea el método uno define etapas, espiral por ejemplo, al terminar cada etapa se hacen las reuniones.
- ✓ Auditorías: trabaja sobre la globalidad del proyecto. Se deben hacer obligatoriamente antes de entregar el producto y también si se tienen dudas sobre alguna parte del proyecto, entonces se revisa completo. Y además de todos los anteriores participan terceros, por ejemplo colegios de profesionales, universidades, etc.

Hay 4 formas de especificar requerimientos:

1. Lenguajes formales: lógica de primer orden, automata finito, lenguaje zeta, una forma matemática. No por usar lenguaje formal garantizo que la especificación de requerimientos esté completa, porque lo que no especifique por olvido no va a estar especificado, no me garantiza que hice un buen relevamiento porque trabajo en lenguaje formal. Y hay determinadas cosas que se hace muy dificultoso hacerlo en lenguaje formal. Por ejemplo todo lo que es lógica.
2. Métodos semiformales: métodos estructurados, métodos orientados a objetos.
3. Lenguajes normalizados: norma IEEE 830.
4. Lenguaje natural.

