

ARQUITECTURA DE PRUEBAS PARA AÑO 2000

RESUMEN:

Se define el marco teórico que debe seguirse en el diseño y realización de pruebas Año 2000. Se define el problema, los elementos a probar, el uso de simuladores, la normalización de los casos de prueba, el entorno de pruebas y los ciclos de pruebas, concluyéndose con recomendaciones de ámbito general. El papel del Auditor en las pruebas Año 2000 se analiza como elemento fundamental de garantía.

Se explican las pautas que deben seguirse en la creación de un entorno de pruebas orientado al año 2000. Se estudian los aspectos relevantes sobre la necesidad del propio entorno de pruebas 2000, su aislamiento con los entornos habituales de producción y pruebas, la secuencia de actividades para proceder al cambio del reloj desde la fecha presente al año 2000 y su vuelta atrás.

Se analizan los riesgos en las pruebas con el reloj en el año 2000, así como la manera de reducirlos mediante la división de la tarea en unidades más pequeñas, la utilización de simuladores y otras herramientas de prueba, y las reglas a seguir para evitar los problemas "culturales" asociados a la inusual tarea de trabajar con entornos informáticos funcionando en fechas futuras, con los riesgos de "contaminación" de los entornos de producción y pruebas habituales que conlleva.

AUTORES:

Ignacio Boixo Pérez-Holanda.
Victoria Irala San José.
Carmen Sierra Blanco.

BIOGRAFÍAS:

Ignacio Boixo Pérez-Holanda. Licenciado e Ingeniero en Informática. Miembro de la Asociación de Ingenieros en Informática de Madrid. Actualmente participa en YES Corps, ONG Internacional de informáticos voluntarios para ayuda altruista sobre Año 2000 (www.iy2kcc.org). e-mail: boixo@bde.es

Victoria Irala San José. Licenciada en Matemáticas. Responsable del Área Técnica del Departamento de Sistemas de Información y Telecomunicaciones del Hospital Clínico de San Carlos de Madrid. e-mail: viral@insalud.es

Carmen Sierra Blanco. Licenciada en Informática. Responsable del Área de Soporte del Departamento de Sistemas de Información y Telecomunicaciones del Hospital Clínico de San Carlos de Madrid. e-mail: csierra@insalud.es

Madrid, 14 Setiembre 1.998.
Actualizado en Junio 1999.

ARQUITECTURA DE PRUEBAS PARA AÑO 2000

DEFINICIÓN DEL PROBLEMA

El diseño de una arquitectura de pruebas para el año 2000 ha de tener siempre presente que nos encontramos básicamente ante un problema cultural. Si bien los ordenadores, y especialmente los programas, podrían haber sido diseñados para trabajar indistintamente con fechas pasadas y futuras, y para suponer que el tiempo pueda moverse en cualquier sentido (del pasado al futuro o viceversa), de hecho no es así.

En nuestra cultura, el tiempo sólo tiene un sentido (del pasado al futuro), sentido que no es reversible. Simplemente, al intentar construir una frase que indique que algo que ha sucedido en el futuro modifica el pasado, y por tanto el presente, se precisa forzar los tiempos de conjugación de los verbos, pues la gramática al uso no está preparada para expresar estos "bucles temporales".

Por este motivo, no se debe esperar que los informáticos hayan programado los sistemas de manera que puedan tratar fechas futuras, posteriores a las del día, de una manera siempre correcta. Por ejemplo, si la fecha de creación de un fichero es posterior al reloj del ordenador que lo trata, existe un serio riesgo de que el sistema operativo determine su situación como errónea. Sin embargo, leer un fichero que fue creado en el pasado es una situación absolutamente normal. Como regla fundamental, ha de evitarse toda situación no controlada al tratar eventos con fecha futura: EL TIEMPO NUNCA RETROCEDE.

Por tanto, para evitar la creación de "bucles temporales", donde ordenadores con fecha presente reciban datos con fechas futuras, el entorno de pruebas 2000 debe tratarse como un "agujero negro", que puede recibir información desde el presente, pero ningún dato fecha-dependiente debe "volver" sin control desde el futuro.

ELEMENTOS A PROBAR

Una arquitectura de pruebas en el año 2000 debe estar diseñada para probar todos los componentes informáticos. En aras a la simplicidad, puede utilizarse una aproximación por niveles, que pueden resumirse en sistemas (hardware y software de base), aplicaciones y datos.

Los sistemas de base se pueden probar, en gran medida, independientemente de las aplicaciones y de los datos. Una vez superada esta prueba, se tendrán garantías de que los errores que aparezcan corresponderán probablemente a las aplicaciones, simplificando así su determinación.

Mediante esta aproximación por niveles, las pruebas pueden simplificarse partiendo del supuesto que cada nivel admite cierta independencia de los otros. A su vez, dentro de cada nivel, pueden identificarse componentes (ej.: cada aplicación o conjunto de aplicaciones aislable del resto) que se puedan probar aisladamente unos de otros, de manera que se simplifique la determinación de problemas, y se evite tener que realizar pruebas exhaustivas de todos los componentes ante un cambio acaecido en uno de ellos.

USO DE SIMULADORES

Gracias al aislamiento entre sistemas y aplicaciones/datos es posible el uso de simuladores. Un simulador intercepta las peticiones de fecha que la aplicación hace al sistema, proporcionando, en lugar de la fecha presente, la fecha que se desee. Esto permite probar independientemente cada aplicación en un sistema con su reloj marcando la fecha presente.

Se parte del supuesto de que un componente toma la fecha del exterior de un número limitado de maneras, de manera que interceptando todas ellas, se puede simular la fecha deseada, y entonces comprobar que los resultados son correctos.

Para realizar la simulación, se precisan de unas entradas con la fecha apropiada, bien como ficheros o como tecleo manual, y de un "simulador de fechas" que intercepte las peticiones de fecha al entorno y provea la adecuada. Las actualizaciones y ficheros de salida deben estar controlados, para evitar interferencias con otras tareas.

El simulador de fechas será tanto más completo cuantas más peticiones de fechas al entorno pueda interceptar, por lo que es un pre-requisito inventariar las diferentes maneras efectivamente utilizadas de tomar fechas del entorno.

El simulador es especialmente útil cuando se cumplen estas tres condiciones:

- 1º El componente funciona independientemente (ej.: un programa batch clásico). Si no fuera así (ej.: un módulo de en programa), hay que estudiar el conjunto que lo integre.
- 2º Es determinable como toma la fecha el componente. (ej.: se dispone del código fuente de un programa). Si no fuera así (ejemplo: no se dispone del fuente), existe el riesgo de que el simulador no intercepte todas las peticiones de fecha.
- 3º Las salidas del componente son independientes de a otras tareas (ej.: el componente tiene sus propios ficheros de E/S). Si no fuera así (ej.: alteración de fechas en ficheros de uso común), debe irse a pruebas conjuntas, o aislar/borrar las salidas.

A efectos prácticos, la mayor utilidad del simulador de fechas será para aplicaciones, especialmente aquellas de desarrollo propio, ya que son más fácilmente inventariables sus entradas y aislables del entorno donde se ejecutan. En el caso del software de base, la utilidad del simulador es mucho más limitada, debido a la dificultad de aislar cada componente de los demás, y de saber exactamente como toma y transfiere las fechas.

Las aplicaciones y los datos normalmente se prueban conjuntamente, aplicando habitualmente pruebas de regresión, para comprobar que el mismo conjunto de datos procesado en dos fechas distintas ofrece resultados equivalentes.

Para preparar los datos en fecha futura, pueden utilizarse herramientas de "envejecimiento de datos", que toman un conjunto de datos y le cambian las fechas que se especifiquen. Con ello, a partir de un conjunto de datos original, se obtiene el mismo conjunto de datos "envejecido" para proceder a las pruebas de regresión.

Son también muy útiles las herramientas de captura y comparación de resultados. Partiendo de la base que una aplicación toma sus datos de ciertos ficheros y/o del usuario, se procede a interceptar los datos que la aplicación lee y que el usuario teclea, así como los resultados que en forma de fichero o de información visual la aplicación presenta. Capturados los datos de entrada/salida en la fecha presente y en la futura, la herramienta los compara y pone de manifiesto las diferencias, simplificando el análisis y comprobación de resultados.

NORMALIZACIÓN DE CASOS BÁSICOS DE PRUEBA

Existen al menos tres períodos críticos en donde el "efecto 2000" tiene mayor riesgo de aparición. Por tanto, para que un componente reciba la certificación debe ser probado, al menos, en los casos básicos.

El definir los casos básicos también permite homogeneizar las pruebas a realizar en los componentes, y posibilita la ejecución de ciclos de prueba, donde comprobar las interacciones entre componentes. Estos ciclos de prueba pueden repetirse periódicamente, y así disponer de una planificación de pruebas de año 2000 que enmarque las actividades a realizar. El ciclo básico de pruebas debiera cubrir: 31/12/1999:04/01/2000 28/02/2000:02/03/2000
31/12/2000:01/01/2001 (opcional)

CASO 1999 COMO MARCA

Se ha venido utilizando el año 99 como marca de situación. Por ejemplo, en muchas ocasiones una fecha de expiración de 99/365 significa que el fichero nunca expira. Como esa fecha se va a alcanzar, hay que comprobar que su "metasignificado" como marca no afecte a la operativa.

CASO COMIENZO AÑO 2000

Al principio del año 2000 se tiene el máximo riesgo de errores en fechas. Para facilitar la identificación de fallos (y poder celebrar la Noche Vieja), todos los sistemas se debieran parar el Viernes 31/12/1999 y reanudar a partir del sábado 1/1/2000, para comprobar su funcionamiento, y que todo esté dispuesto cuando las aplicaciones comiencen su funcionamiento con usuarios en real el Lunes 4/1/2000.

CASO 2000 BISIESTO

El año 2000 es múltiplo de 4 -bisiesto-, de 100 -no bisiesto- y también de 400 -bisiesto definitivamente-. Se debe comprobar la transición del Lunes 28/2/2000 al Martes 29/2/2000 y Miércoles 1/3/2000. OPCIONALMENTE, y sólo si hay implicaciones de negocio, también se debe comprobar la transición del Viernes 29/12/2000 al Martes 2/1/2001, y que se calcule correctamente que el año 2000 tiene 366 días.

No es posible "acelerar" el tiempo dentro de la prueba, por lo que una hora con el reloj en el año 2000 también dura sesenta auténticos minutos. En la ocurrencia de cada ciclo hay que planificar las horas concretas que van a ser simuladas, que dependerán de las pruebas a realizar (día completo, tareas puntuales, cierres, aperturas, etc) y de su duración prevista. La hora en la prueba 2000 no tiene que coincidir con la hora presente, pero, por simplicidad, la diferencia se recomienda que sea siempre múltiplo de hora (ejemplo: doce horas)

ENTORNO DE PRUEBAS PARA EL AÑO 2000

Se precisa construir un entorno de pruebas para el año 2000, independiente de los entornos normales de pruebas y producción. Esta independencia puede alcanzarse destinando un conjunto de hardware y software específico (conurrencia en el tiempo) o utilizando los entornos habituales de pruebas/producción en períodos de no-uso (serialización en el tiempo).

La recomendación es crear un entorno específico (conurrencia en el tiempo) para evitar riesgos de que las pruebas del año 2000 afecten al funcionamiento normal de la instalación informática. Además, se podrán realizar las pruebas en horario normal. El mayor inconveniente es el coste, pues normalmente será necesario comprar/asignar y montar el nuevo equipamiento.

Si se opta por utilizar el equipamiento normal de la instalación en períodos no laborables (serialización en el tiempo), se ahorran costes de creación del entorno de pruebas año 2000, pero las pruebas habrán de realizarse fuera del horario normal de funcionamiento, y se corre el riesgo de problemas en la restauración del entorno de manera que se compromete la disponibilidad para el siguiente período de funcionamiento en producción.

La solución ecléctica de utilizar el entorno de pruebas habitual para probar también el año 2000 presenta las ventajas de ahorro de costes, pero el inconveniente de que mientras se prueba con el reloj en el año 2000, no se pueden realizar al mismo tiempo pruebas para el mantenimiento del entorno de producción, con lo que habría que optar entre modificar directamente producción sin pasar por el entorno de pruebas (lo que va en contra de las buenas prácticas) o demorar el mantenimiento de producción a la restauración del entorno de pruebas desde el 2000.

La arquitectura que se describe parte de un entorno año 2000 funcionando en fecha presente, con posible uso del simulador de fechas, estando preparado para que durante ciertos períodos funcione con el reloj en el año 2000.

El entorno 2000 va actualizando su reloj en la fecha presente y, en períodos predefinidos, se pone el reloj con fecha 2000, se realizan las pruebas, y se hace regresar el reloj a la fecha real. Este ciclo se repite las veces necesarias.

Para evitar la aparición de "bucles temporales", donde eventos con fecha futura puedan afectar al presente, "contaminándolo", las modificaciones y preparación de las pruebas se harán con el reloj del entorno 2000 situado en el presente.

Mientras el entorno 2000 funcione con su reloj en el año 2000, se realizarán las pruebas, y se podrá enviar información desde los entornos habituales en pruebas y producción al entorno 2000, pero sólo se podrá recibir la información desde el entorno 2000 que sea no fecha-dependiente. Terminada la prueba con el reloj en el año 2000, se restaurará el entorno 2000 a la última situación que tuviera en el presente, eliminándose todo lo modificado durante la prueba con reloj en el 2000.

A efectos conceptuales, el entorno 2000 ha de ser un nodo o conjunto de nodos independientes, con su propio software y hardware. Debe disponer de núcleo de sistema operativo, Base de Datos, CPU, memoria, periferia de discos/dispositivos y demás elementos, de modo funcionalmente dedicado y absolutamente independiente a los entornos producción y pruebas habituales.

Pueden alcanzarse ahorros compartiendo elementos, utilizando una partición del ordenador existente en vez de una máquina dedicada, compartiendo la red de comunicaciones, ciertos servicios de Base de Datos, etc. En cada componente, se puede estudiar como conseguir ahorros en la creación del entorno 2000 mediante la compartición de componentes, pero siempre manteniendo la independencia funcional entre el entorno 2000 con reloj en el año 2000 y el resto de la instalación, por lo que, a efectos funcionales, la forma concreta de implementar el entorno 2000 ha de ser irrelevante.

CICLO PRESENTE - AÑO 2000

FUNCIONAMIENTO FECHA REAL (PRESENTE)

El entorno 2000 dispondrá de los mismos servicios e interoperatividad que cualquier otro entorno, con las peculiaridades que le sean propias. Mientras esté funcionando con fecha presente, se modifican los programas, se preparan juegos de prueba, se actualiza el software de base y se utiliza el simulador de fechas para probar componentes.

TRANSICIÓN HACIA 2000

- Cerrar accesos de red de comunicaciones.
- Obtener copia de seguridad del núcleo operativo básico y del resto de ficheros.
- Desactivar fechas de expiración
 - Identificaciones y contraseñas de usuarios, ficheros y Bases de Datos
 - licencias de productos, planificadores de tareas, copias y migración de datos, autorización y protección de recursos, etc.
- Rearrancar con el reloj en el año 2000.
- Abrir accesos de comunicaciones con SENTIDO ÚNICO HACIA ENTORNO 2000.

FUNCIONAMIENTO AÑO 2000

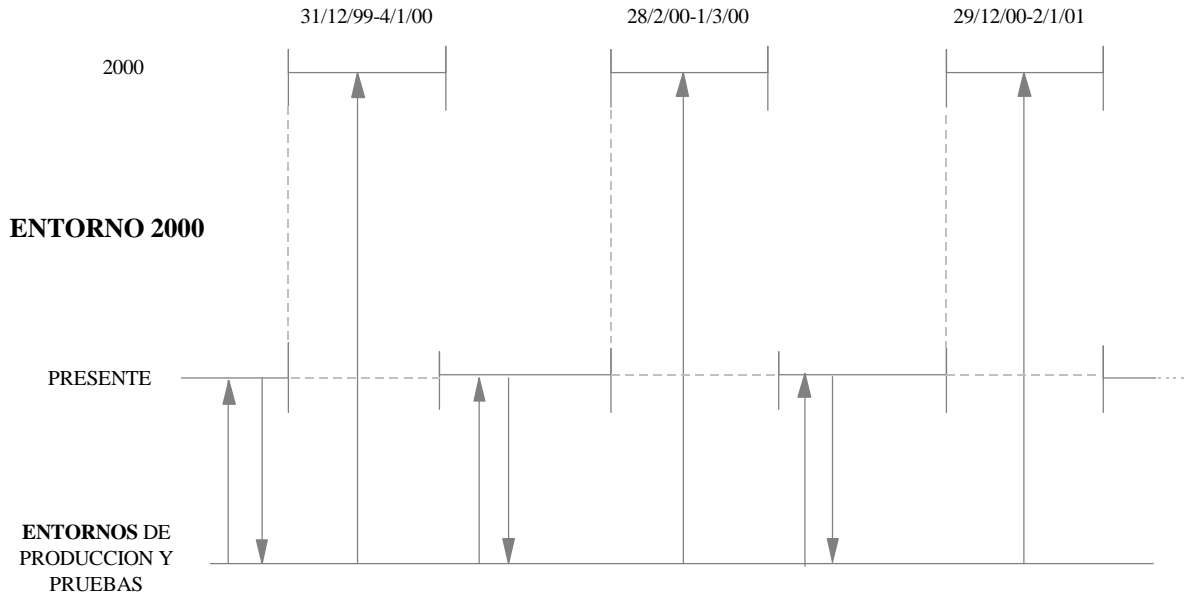
- Modo "agujero negro". La información puede entrar, pero no puede salir, para evitar "contaminar" los entornos con fecha presente.
- Realización de pruebas, con entorno 2000 básicamente aislado.
- Salvar los resultados que se precise en soporte no fecha-dependiente (papel, cinta con etiqueta controlada, transmisión "ad hoc" de ficheros y Bases de Datos, etc.).
- Simulador de fechas -en principio- inactivado, para no complicar las pruebas a realizar. En casos concretos pudiera activarse, previo estudio.

VUELTA AL PRESENTE

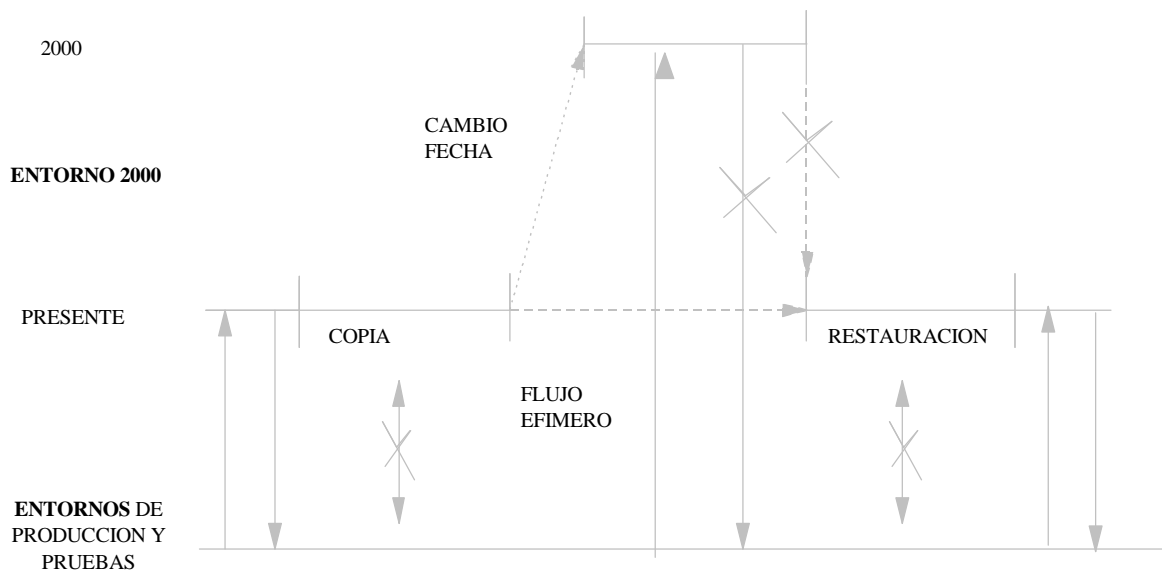
- Cerrar accesos de red de comunicaciones
- Rearrancar en fecha presente con la copia del núcleo operativo básico.
- Borrar todos los ficheros con fecha posterior al presente.
- Restaurar desde la copia los ficheros que falten.
- Reactivar (si fuera preciso) fechas de expiración (usuarios, licencias, ficheros, planificadores, etc.).

CICLOS DE PRUEBAS DEL AÑO 2000 (Gráficos)

CICLOS DE PRUEBAS AÑO 2000



CICLO PRESENTE - AÑO 2000 (DESGLOSE)



RECOMENDACIONES

- 1º EL TIEMPO NO RETROCEDE. NO RESTAURAR A UNA FECHA ANTERIOR.
- 2º CERTIFICAR "HACIA ARRIBA", DESDE COMPONENTES INDIVIDUALES HACIA SISTEMAS COMPLETOS.
- 3º NORMALIZACIÓN DE CICLOS BÁSICOS DE PRUEBA
31/12/1999:04/01/2000 28/02/2000:02/03/2000 29/12/2000:02/01/2001
(opcional)
- 4º PLANIFICAR PARA LAS PRUEBAS AL MENOS LA MITAD DEL COSTE TOTAL
- 5º CONSTRUIR UN ENTORNO DE PRUEBAS ESPECÍFICO AÑO 2000
- 6º AISLAR EL ENTORNO CON RELOJ EN AÑO 2000 PARA EVITAR "CONTAMINAR" ENTORNOS DE PRODUCCIÓN/PRUEBAS FUNCIONANDO EN FECHA PRESENTE
- 7º UN CONTROL DE CALIDAD COMPROBARÁ EL GRADO DE FIABILIDAD ALCANZADO Y LOS RIESGOS REMANENTES.

BIBLIOGRAFÍA

- | | |
|--|--|
| Global 2000 Co-ordinating Group. | www.global2k.com/home |
| Contingency Planning in Year 2000. | www.ffiec.gov/y2k/contplan |
| Centro de Servicios de Información 2000/Euro. | www.map.es/csi/2000 |
| Efecto 2000. Rafael Estrella (PSOE). | www.ctv.es/users/estrella/2000gov |
| L'efecte del canvi de mil.lenni. | www.cambrescat.es/e2000/p-01 |
| The Year 2000 and 2-Digit Dates. | www.software.ibm.com/year2000/resource |
| MITRE Corporation (Fuerzas Aéreas U.S.A.). | www.mitre.com/technology/y2k |
| Year 2000 Journal. Revista dedicada al "año 2000". | www.y2kjournal.com |
| Informáticos Voluntarios Año 2000 (YES Corp) | www.iy2kcc.org |
| Directorios especializados en el "efecto 2000". | |
| www.y2ktool.com | www.y2kinfo.com |
| www.year2000.com | www.it2000.com |
| www.y2klinks.com | |
| www.y2ksavers.com | |
| Información de fabricantes | |
| www.ibm.com/year2000 | www.sun.com/y2000/index |
| www.hp.com/gsy/year2000/cure | www.software.ibm.com/year2000 |

www.microsoft.com/technet/topics/year2k

AUDITORÍA DEL PROYECTO AÑO 2000

El objetivo de la auditoría en estos proyectos se centra en comprobar que los sistemas y aplicaciones hayan sido adecuadamente comprobados en su adecuación al año 2000. Las consideraciones mas importantes son:

1ª La funcionalidad de los sistemas no ha de variar.

En el extremo, si un programa tiene un error en fechas anteriores al 2000, ese mismo error debe existir en fechas posteriores. No se debe admitir ninguna modificación funcional so pretexto de adecuación al 2000; si es necesario, se abrirá un proyecto independiente. La razón es que la depuración de errores crece exponencialmente en función del número de cambios introducidos. Es mucho mas seguro y sencillo hacer (y probar) modificaciones funcionales, y POSTERIORMENTE comprobaciones año 2000, que mezclar ambas actividades en el mismo proceso.

2º Los cambios funcionales han de ser probados de nuevo en relación al año 2000.

Si se modifica un sistema probado, nada garantiza que se siga manteniendo su adecuación al 2000. Se deben repetir las pruebas. Bien es cierto que pueden ser mas sencillas y rápidas, ya que sólo habrá de probarse, en relación al Año 2000, lo modificado.

3º Ha de establecerse una moratoria alrededor de la transición al 2000.

En algún momento hay que detener el ciclo modificación-comprobación, con el fin de estabilizar sistemas y aplicaciones en la transición al 2000. Como la transición es instantánea, indefectible y simultánea, a la medianoche del 31/12/1999, se recomienda una moratoria mínima del 1 Octubre 1999 al 31 de Enero del 2000.

4º Las pruebas Año 2000 han de ser auditables, y auditadas ex-ante y ex-post.

Como la transición al 2000 es un proyecto del que no se cuenta con experiencia previa, se corre el riesgo de grandes divergencias en su comprobación. Diferentes equipos pueden tener muy diferentes visiones del alcance de las comprobaciones, profundidad de las pruebas y garantías a obtener. Existen riesgos de sobre-comprobación, que poco va a aportar, y, sobre todo, de infra-comprobación, donde no se alcanzan las garantías mínimas.

Las pruebas Año 2000 han de quedar perfectamente documentadas, para permitir su repetición en caso de modificaciones funcionales posteriores, para permitir su auditoría antes del 2000, y para contar con pruebas fehacientes en caso de incidencias y/o litigios posteriores al 2000.

5º El Proyecto 2000 abarca Informática y Áreas de negocio.

Una definición efectiva del Año 2000 es "Adaptación de Informática con repercusión en el Negocio", contraponiéndose al Euro que sería "Adaptación de negocio con repercusión en Informática". Ambos proyectos son imprescindibles para la continuidad de las operaciones, aunque en ninguno de ellos se obtiene ninguna nueva funcionalidad inmediata.

Al no aparecer beneficios tangibles inmediatos, se corre el riesgo de que el "día a día" reste recursos y prioridad, no alcanzándose las metas mínimas para asegurar una transición sin sobresaltos. La máxima "mas vale prevenir que curar" tiene aquí un ejemplo arquetípico. La labor del auditor es fundamental para el análisis de riegos y alcanzar el punto óptimo de equilibrio.

6º "El todo es mas que la suma de las partes"

Esta visión "holística" de los sistemas de información ha de tenerse en cuenta en el diseño de las pruebas y de los planes de contingencia. No sólo hay que hacer pruebas individuales, sino también pruebas de conjunto donde se prueben las interrelaciones, no siempre explícitas, entre los diversos componentes.

Hay que controlar las pruebas de componentes individuales, para evitar redundancias que poco aportan (ejemplo: probar exhaustivamente un sistema operativo no es misión de una empresa comercial), y enfatizar las pruebas de conjunto, que son las que realmente aseguran una transición "suave" al Año 2000. Estas pruebas de conjunto son únicas para cada sistema de información, porque cada sistema de información es único. Es aquí donde hay que poner el mayor énfasis, precisamente porque las pruebas de conjunto nunca van a poder alcanzar el techo teórico.

No es viable hacer que una empresa, así como sus clientes, proveedores y otras entidades con las que se tengan flujos de información, dediquen varios días a probar su funcionamiento con todos los relojes en el año 2000. Hay que alcanzar un compromiso entre factibilidad y costes, compromiso que ha de ser auditado, y cuyos riegos remanentes han de ser explicitados.

7º Plan de contingencias.

No hay un "corpus teórico" ni experiencia previa que permitan predecir, a priori, que fallos se van efectivamente a producir en la transición al 2000. Las Áreas de negocio, con ayuda de Informática, han de tener preparados planes de contingencia ante los diversos fallos potenciales. Estos planes de contingencia han de ser auditados y probados.

Además de los fallos conceptualmente triviales (servicios básicos: electricidad, agua, teléfono), han de revisarse fundamentalmente potenciales fallos internos en los sistemas de información, de duración indeterminada. Estos fallos no hace falta que sean catastróficos para tener consecuencias graves. Una seria acumulación de fallos simples puede saturar los servicios de atención al usuario y de mantenimiento, provocando consecuencias que, en circunstancias normales, no existirían. Ha de informarse a los usuarios sobre las pautas de comportamiento durante la transición al 2000, para evitar potenciales crisis de "saturación" autogeneradas.

8º Debe existir dos Equipos de Proyecto Año 2000: Directivo y Técnico.

Involucrar a la Dirección es imprescindible para llevar a cabo este proyecto. Los riesgos que se corren son muy importantes, y es la Dirección quien debe asignar recursos y prioridades, así como asumir los riesgos remanentes. El coste de adaptación al 2000 ha sido sistemáticamente minusvalorado, y sólo la Dirección puede tomar decisiones en este ámbito.

Técnicamente se precisa un punto focal que recopile y disemine la información relevante Año 2000, oriente a los informáticos participantes, y establezca los niveles de garantía técnica requeridos. Se debe explicar a los informáticos participantes su función en la consecución de un nivel adecuado de adaptación al 2000. Comúnmente se adolece de formación en temas año 2000, y esta es un área de hay que impulsar decididamente, para poder mantener los niveles de rendimiento habituales en el personal técnico.

9º Debe existir una metodología Año 2000

Algunas premisas básicas han de ser compartidas por todos los participantes. De esta metodología se deducirán planes detallados, fechas límite, prioridades, etc.

No todas las aplicaciones tienen la misma importancia. En la metodología se debe incidir cuales son las funciones críticas de negocio, y cuales tienen prioridades mas bajas. El caso de aplicaciones no prioritarias que directamente no se prueben (falta de tiempo/recursos) ha de estar contemplado. La orientación de como documentar pruebas, para que sean repetibles y auditables, debe ser seguida por los participantes.

10º Entorno de pruebas.

La experiencia acumulada indica que un entorno de pruebas, dedicado a comprobaciones Año 2000, es una parte prácticamente imprescindible del proyecto. El entorno de pruebas Año 2000 permite ahorrar muchos esfuerzos a la hora de comprobar las aplicaciones y sistemas, tanto en pruebas individuales como pruebas de conjunto. Permite repetir fácilmente las pruebas que sean necesarias. Puede utilizarse en horario normal de trabajo. Facilita el tener una referencia clara de los temas técnicos, y un punto común donde articular las tareas de detalle.

Hacer pruebas en producción no sólo es mucho mas caro a la larga, sino que se incrementan riesgos de fallos en las operaciones de negocio después de realizadas las pruebas. El proteger la producción si en los mismos equipos se realizan pruebas año 2000, aunque sea en momentos diferentes, ha resultado una tarea de gran complejidad.