



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Administración de  
Almacenamiento en Servidores  
Mainframe bajo el Sistema  
Operativo z/OS**

**INFORME DE ACTIVIDADES PROFESIONALES**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Luis Eduardo Revelo Domínguez

**ASESOR DE INFORME**

M.A. Miguel Eduardo González Cárdenas



Ciudad Universitaria, Cd. Mx., 2016



## CONTENIDO

I. Prefacio .....	1
II. Introducción.....	2
III. Objetivo .....	3
IV. Marco teórico .....	3
1. ¿Qué es un mainframe? .....	3
2. Cargas de trabajo en un mainframe .....	5
3. Sistema Operativo z/OS .....	6
4. Recursos de hardware usados por z/OS .....	7
5. El rol del programador de sistema frente al administrador del sistema. ....	8
V. Descripción de la empresa.....	9
1. Perfil de las empresas que usan mainframe .....	9
2. La empresa en la que se enfoca este informe .....	10
3. Organigrama.....	11
VI. Descripción del puesto de trabajo.....	12
1. Objetivos.....	12
2. Retos .....	13
3. Resultados esperados .....	14
VII. Participación en proyecto:Outsourcing de Almacenamiento Central .....	15
1. Antecedentes.....	15
2. Requerimientos del proyecto .....	15
3. Metodología empleada .....	17
4. Alcance del proyecto .....	18
5. Perfil del especialista .....	18
6. Recursos de almacenamiento administrados .....	20
6.1 Dispositivos de almacenamiento.....	20
6.2 Recursos de almacenamiento.....	24
7. Desarrollo del proyecto.....	26
7.1 Atención a requerimientos.....	26
7.2 Administración de los recursos almacenamiento .....	42
7.3 Atención a incidentes .....	46

8. Aportaciones profesionales al proyecto. ....	52
8.1 Administración, monitoreo y atención a requerimientos.....	52
8.2 Incidentes.....	56
9. Análisis e interpretación de los resultados de trabajo. ....	58
9.1 Atención a requerimientos.....	58
9.2 Administración de los recursos de almacenamiento .....	59
9.3 Atención a Incidentes .....	66
VIII. Participación en Proyecto: Migración Storagetek-VTCS a IBM-TS7700.....	67
1. Antecedentes.....	67
2. Requerimientos del proyecto .....	68
3. Metodología empleada .....	68
4. Alcance del proyecto .....	68
5. Desarrollo del proyecto.....	69
5.1 Estado inicial del proyecto.....	69
5.2 Descripción de la infraestructura.....	70
5.3 Descripción del problema.....	71
6. Aportaciones al proyecto .....	74
6.1 Despachador de jobs .....	74
6.2 Reportes de status de migración.....	75
7. Resultados obtenidos .....	76
IX. Conclusiones .....	78
Apéndice A. Ejemplo de JCL para crear un data set secuencial. ....	79
Apéndice B. Ejemplo de JCL para crear un data set particionado .....	82
Apéndice C. Ejemplo de JCL para crear base de GDG .....	83
Apéndice D. Ejemplo de JCL para crear un archivo VSAM.....	85
Apéndice E. Ejemplo de asignación de espacio para una aplicación.....	87
Apéndice F. Ejemplo de modificación de retención de data sets. ....	97
Apéndice G. Ejemplo de definición de recursos de almacenamiento para una aplicación .....	100
X. Bibliografía.....	108

## I. PREFACIO

La experiencia laboral que explicaré en este informe la desarrollé participando en dos proyectos para dos empresas que brindan servicios de TI. El primer proyecto fue un outsourcing para administrar datos en servidores mainframe bajo el sistema operativo z/OS y el segundo proyecto consistió en un proyecto de migración tecnológica entre sistemas de almacenamiento igualmente acoplados a la misma plataforma. En apego al cumplimiento de las políticas establecidas bajo contrato por estas empresas, he mantenido en anonimato el nombre de las mismas así como el nombre del cliente al cuál se le brindaron los servicios.

## II. INTRODUCCIÓN

La administración de los recursos de almacenamiento de datos es una actividad importante para garantizar el uso eficiente, confiable y continuo de la infraestructura de TI en un centro de procesamiento de datos. Las siglas *TI* o *IT* (en español y en inglés respectivamente) hacen referencia a "Tecnologías de la Información".

Por lo general, durante la operación del negocio, las interrupciones en el servicio que ocurren por una falla en la infraestructura de TI (hardware, software) o una mala administración (fallo humano) suelen significar pérdidas económicas para una organización. Por tanto, el contar con el personal debidamente preparado es crucial para poder atender los distintos requerimientos y prevenir o dar solución a incidentes que se presenten de forma oportuna.

En el siguiente informe describiré a grandes rasgos la tecnología mainframe, las características principales de las empresas que hacen uso del mainframe como recurso computacional para administrar sus datos críticos y principalmente hablaré de mi participación como especialista de TI enfocado a la administración de datos en esta plataforma.

Al hablar de mi participación, describiré los objetivos y metas que tuve que cumplir, las expectativas que la organización tuvo para mi puesto, las competencias profesionales que debí emplear para poder enfrentar las diferentes situaciones y el conocimiento técnico específico que debí dominar para completar las actividades a realizar.

### III. OBJETIVO

Mi objetivo como especialista de TI es garantizar la disponibilidad de los recursos tecnológicos para que una organización pueda llevar a cabo sus operaciones.

Siendo administrador de almacenamiento, debí cerciorarme de que los recursos estuvieran disponibles en cualquier momento para que los procesos que llevasen a cabo las operaciones de la organización se ejecutaran sin contratiempos. Para cumplir con este objetivo, realicé las siguientes actividades: monitorización de recursos, mantenimiento de los mismos, elaboración de reportes, soporte técnico durante incidentes y atención de requerimientos.

### IV. MARCO TEÓRICO

#### 1. ¿QUÉ ES UN MAINFRAME?

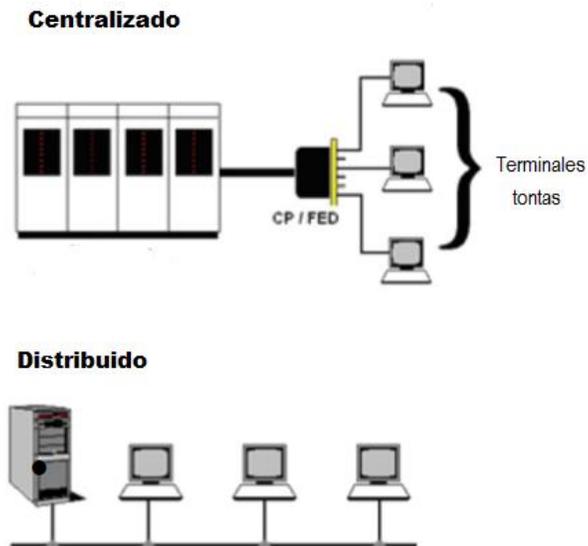
Debido a que los avances tecnológicos han hecho que el poder de procesamiento de las computadoras actuales más pequeñas sean comparables a las de las computadoras más grandes de hace varios años, el término mainframe no puede estar relacionado con una capacidad de procesamiento específico. Por tanto un mainframe se puede describir como "lo que las empresas usan para albergar bases de datos comerciales, servidores de transacciones y aplicaciones que requieren un mayor grado de seguridad y disponibilidad que el encontrado en computadoras de menor escala" <sup>1</sup>. Un mainframe es una computadora de propósito general que se caracteriza por el empleo de un gran número de dispositivos de E/S y por mantener una ingeniería interna redundante que resulta en una alta confiabilidad y seguridad.

En su contraparte, los de rango medio, existen servidores destinados a distintas tareas: servidores de transacciones, servidores de base de datos, servidores de email y los servidores web. Un gran conjunto de estos servidores se conocen como granjas de servidores o *server farms*.

---

<sup>1</sup>Ebbers,M.,Kettner, J., (2011), *Introduction to the new mainframe: z/OS Basics*, 3a Edición, Redbooks. International Technical Support Organization, U.S.A. ,Publicación No. SG24-6366-02, ISBN 0738435341,pag 10.

Un mainframe es un repositorio de datos central, relacionado con los usuarios a través de dispositivos menos poderosos como workstations o terminales. La existencia de un mainframe, por tanto, implica una forma de cómputo centralizado en oposición a la forma de cómputo distribuido en el cuál se usa un número masivo de ordenadores organizados en *clusters* o *grids* e interconectados por una red de comunicaciones para llevar a cabo una tarea informática.



Fuera del sector empresarial y gubernamental estas computadoras permanecen invisibles al público en general, el sector académico e inclusive a muchos profesionales de TI. Actualmente IBM tiene un dominio hegemónico en el mercado mundial del mainframe, probablemente Fujitsu sea el único competidor serio únicamente en Japón.



Fujitsu "Global Server" GS21 2600

Los fabricantes de sistemas computacionales y los profesionales de TI comúnmente emplean el término "plataforma" para referirse al hardware y al software que están asociados con una arquitectura particular. Por ejemplo, un

mainframe y su sistema operativo son considerados una plataforma. UNIX sobre la arquitectura RISC (*reduced instruction set computer*) son considerados una plataforma independientemente de cuál sea el tipo de computadora involucrada. Las computadoras personales pueden ser vistas como diferentes plataformas dependiendo del sistema operativo que estén ejecutando.

IBM fabrica y comercializa mainframes denominándolos actualmente como IBM z Systems, y hacen uso de procesadores cuya arquitectura se denomina z/Architecture la cual no es compatible con las arquitecturas x86 ni x64. Sin embargo existen programas que permiten emular la arquitectura de un mainframe en una computadora con arquitectura x86/x64 posibilitando el poder ejecutar software de mainframe en un PC.



IBM z13 de la serie z Systems, presentado en enero de 2015

## 2. CARGAS DE TRABAJO EN UN MAINFRAME

Dos tipos de cargas de trabajo se pueden distinguir en un mainframe:

**Procesamiento batch:** también conocido como procesamiento por lotes, se define como aquella actividad computacional que no requiere interacción con el usuario. Las características típicas de este procesamiento son:

- Gran cantidad de datos de entrada que son procesados y/o almacenados, y como consecuencia la gran salida de datos producidos.
- La inmediatez del tiempo de respuesta usualmente no es un requisito, sin embargo los procesos batch comúnmente deben completarse dentro de una ventana (periodo en el cual la actividad online sea baja) para cumplir con los niveles de servicio.

- Un solo procesamiento batch puede llevar a cabo la ejecución de cientos de trabajos (*jobs*) en una secuencia preestablecida.

Ejemplos de este procesamiento:

- Respaldos de base de datos y archivos críticos.
- Generación de reportes estadísticos del negocio.
- Reportes de estados de cuenta mensuales para ser enviados a los clientes de un banco.

**Procesamiento de transacción en línea** (*Online transaction process/OLTP*): Es un proceso que ocurre interactivamente con el usuario. Esta forma de procesamiento es la que es más familiar a la mayoría de la gente. Las características típicas de este procesamiento:

- Pequeña entrada de datos, pocos datos almacenados y procesados, y pocos datos de salida.
- Tiempo de respuesta inmediato, usualmente menos de un segundo.
- Muchos usuarios involucrados en un gran número de transacciones.

Entre los ejemplos podemos encontrar:

- Los cajeros ATM para realización de depósitos, retiros, transferencias etc.
- Pagos en supermercados con tarjetas de debito o crédito.
- Compra de mercancía por internet.

### 3. SISTEMA OPERATIVO z/OS

El sistema operativo z/OS es el principal de varios sistemas operativos que pueden correr sobre el mainframe. Ha sido el resultado de la evolución de los primeros sistemas operativos que se crearon para el System 360, computadora que data de los años 60's. Históricamente pasó de ser un sistema que sólo podía procesar una sola tarea a la vez hasta manejar miles de programas e interactuar con miles de usuarios de forma concurrente. z/OS se ejecuta en un procesador y reside en la memoria del procesador durante su ejecución.

Las características de este sistema operativo son:

- Diseñado para cómputo intensivo de E/S.
- Diseñado para garantizar la integridad de datos sin importar qué tantos usuarios concurrentes tenga.

- Puede procesar una gran cantidad de trabajos batch concurrentes, con balanceo de carga automático.
- Permite incorporar seguridad a aplicaciones, recursos y perfiles de usuarios.
- Puede correr aplicaciones de misión crítica de forma segura.
- Puede ser controlado desde una o más terminales de operador.

z/OS emplea el término de *memoria virtual* en donde residen los usuarios, los programas y el sistema. La memoria virtual está conformada por la memoria central, la cual también se conoce como memoria real, y la memoria auxiliar que reside en disco. La memoria central se divide en *frames*, la memoria auxiliar en *slots* y la memoria virtual en páginas, donde cada uno de estos elementos tiene un tamaño de 4 Kb.

z/OS maneja el concepto de *address space* similar a un proceso en UNIX. Ambos usan identificadores y permiten multiprogramación. Por un lado en UNIX un hilo de procesamiento se conoce como *thread* y en z/OS se conoce como TCB (*task control block*).

Este sistema operativo siempre está en continuo desarrollo desde sus antecesores y mantiene la compatibilidad hacia atrás de sus desarrollos.

Este software no se puede comprar, se alquila mensualmente.

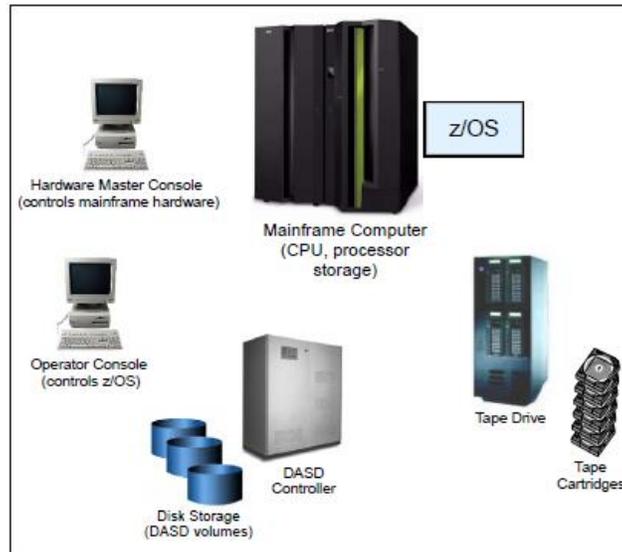
#### 4. RECURSOS DE HARDWARE USADOS POR Z/OS

El hardware del mainframe consiste en procesadores y una multitud de dispositivos periféricos como controladores de disco (llamados Direct Access Storage Devices(DASD)), drives de cintas magnéticas (Tape drive) y varios tipos de consolas de usuario. Las cintas y los discos son usados para funciones de sistema y por programas de usuario ejecutados por z/OS.

Existen dos tipos de consola:

**Hardware Management Console (HMC)** la cual controla el hardware del mainframe. Desde esta se controlan operaciones como la instalación y reinicio del sistema operativo, la definición de particiones lógicas para la CPU (LPAR's) y asignación de memoria y procesadores para cada partición lógica de forma dinámica.

**Consola de operador.** Controla las operaciones del sistema operativo z/OS.



## 5. EL ROL DEL PROGRAMADOR DE SISTEMA FRENTE AL ADMINISTRADOR DEL SISTEMA.

Los sistemas mainframe son diseñados para ser usados por un gran número de personas. La mayoría de los que interactúan con el mainframe son usuarios, personas que usan las aplicaciones hospedadas en el sistema. Sin embargo, debido al número de usuarios, la sofisticación del software, la concurrencia de aplicaciones corriendo en el sistema y la complejidad del software del sistema que soporta a los usuarios y las aplicaciones, una variedad de roles de trabajo son necesarios para operar y soportar el sistema. Los roles de TI que surgen son:

**Programadores de sistema.** (*system programmers*). Persona que instala, adecua y mantiene el sistema.

**Administradores de sistema** (ejemplos: DBA, almacenamiento, redes y comunicaciones, seguridad y performance). Persona que asegura y mantiene los datos críticos del negocio que residen en el mainframe.

**Programadores y diseñadores de aplicaciones.** Principalmente se van a encontrar programadores en COBOL, debido a que como sus siglas lo sugieren (*common business oriented language*), este lenguaje está enfocado al desarrollo de aplicaciones financieras y de negocios.

**Operadores de sistema.** Las personas que monitorizan y controlan la operación del hardware y software del mainframe.

**Administración, análisis y control de la producción.** Personal que asegura que las cargas de trabajo de tipo batch corran sin error o retrasos.

La distinción entre programador y administrador de sistema varía dependiendo del centro de datos. En organizaciones pequeñas, una persona puede ser llamada para efectuar varios roles y los términos pueden intercambiarse.

En grandes organizaciones de TI con múltiples departamentos, las actividades suelen ser mejor definidas. Los administradores del sistema realizan mayormente tareas del día a día relacionadas al mantenimiento del dato crítico del negocio que reside en el mainframe, mientras que el programador del sistema se enfoca en mantener el sistema en sí.

## V. DESCRIPCIÓN DE LA EMPRESA

### 1. PERFIL DE LAS EMPRESAS QUE USAN MAINFRAME

Las empresas que usan el mainframe como repositorio principal de datos, son aquellas que requieren el manejo de gran cantidad de datos críticos, así mismo requieren la mayoría de las veces acceso a estos datos en cualquier momento definiéndose, en consecuencia, un servicio de 24/7.

Otra característica de las empresas que usan mainframe es su antigüedad. Frecuentemente, grandes organizaciones que iniciaron sus operaciones en los años 70's u 80's siguen usando mainframes como medio fiable para administrar su información. Las empresas que llevan bastante tiempo usando mainframe se resisten a cambiar a soluciones de cómputo distribuido debido a que sustituir el procesamiento de un solo mainframe puede llegar a implicar la adquisición de cientos de servidores de computo distribuido, aunado a la migración de la plataforma que conlleva recompilar mucho código para migrar aplicaciones y mover una cantidad enorme de datos ya que estos no pueden permanecer en los dispositivos de almacenamiento utilizados por el mainframe debido a su incompatibilidad con los sistemas de archivos usados por los sistemas operativos como Linux, UNIX o Windows. En vez de sustituirlo, el mainframe comúnmente es

adherido como un componente más de un esquema de computo distribuido en el cuál se aprovechan mejor todas las capacidades de cada tecnología.

Los sectores donde actualmente se suelen utilizar mainframes son:

- Sector financiero y banca
- Aseguradoras
- Dependencias del gobierno

## 2. LA EMPRESA EN LA QUE SE ENFOCA ESTE INFORME

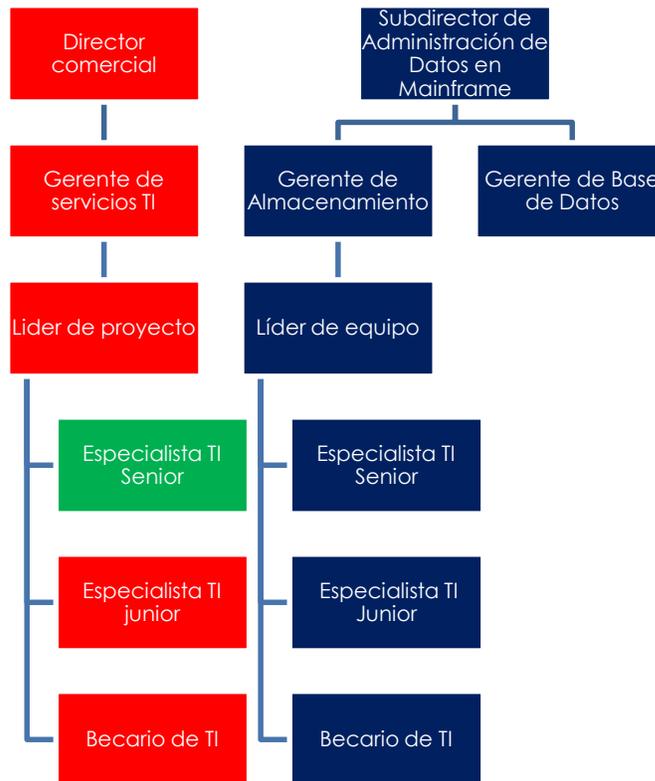
La empresa en la que se enfoca este informe pertenece al sector de la banca. Este banco realiza operaciones batch con muchos datos durante la noche y en el día se enfoca a las transacciones online, donde las operaciones se realizan desde cajeros automáticos, sucursales, servicios de la web y aplicaciones móviles. Las operaciones no se detienen durante las 24 horas del día y solo existen interrupciones en el servicio en caso de que existan cambios mayores en el sistema, que son planificados para realizarse en días y horarios con menos carga e impacto. El que una falla en el sistema o en la administración detenga las operaciones del banco es una de las desventajas que se observan del esquema de cómputo centralizado. En cuanto a su diseño, el mainframe ha sido manufacturado de manera que su proveedor garantiza la seguridad y la confiabilidad del sistema casi al 100% y de alguna manera las empresas que aún conservan estos servidores avalan esta aseveración, pero los incidentes que surjan como consecuencia de un error humano en la administración siempre están latentes.

Este banco delega la administración de su infraestructura y sus datos a terceros, con el fin de enfocarse principalmente a su negocio. Los terceros son empresas que brindan, por lo general, servicios de outsourcing y consultoría TI. Este tipo de tercerización de actividades no siempre sucede en todas las empresas que usan mainframe, a veces es la propia empresa la que lleva a cabo la operación del negocio y la administración de la infraestructura y datos.

El banco, que en este caso será también conocido como *el cliente*, solicita a las empresas que proveen servicios TI que le administren sus datos que residen en su mainframe y establecen un contrato en el cual el proveedor se compromete a cumplir con los acuerdos de niveles de servicio y con los tiempos estimados para realizar los cambios especiales o mantenimientos que surjan como proyectos.

### 3. ORGANIGRAMA

Para los proyectos de administración de almacenamiento generalmente se manejó el siguiente esquema de trabajo:



Los elementos en rojo representan el personal asociado a las empresas que brindan servicios TI y los elementos en azul representan el personal que está asociado con el cliente. Como puede observarse, el cliente también posee personal de TI trabajando directamente para él, esto con el fin de tener un contrapeso a nivel técnico que vea por los intereses del cliente y su número es mucho menor que los especialistas TI externos. Destaca el elemento en verde por ser el puesto que desempeñé en estos proyectos.

Los elementos de un lado y del otro están organizados de manera que cada elemento interactúa mayormente con los del mismo nivel. Así pues, el Director Comercial del proveedor de servicios TI busca oportunidades de negocio con el Subdirector del banco, el Gerente de servicios IT aborda las necesidades del cliente a través del Gerente de Almacenamiento y así mismo se encarga de ver la viabilidad de un determinado proyecto. El líder de proyecto realiza actividades de administración de los recursos humanos a su cargo al igual que el líder de equipo, sólo que su diferencia radica en que el líder de proyecto sólo existe durante la

consecución del proyecto y el líder de equipo tiene un puesto permanente. El gerente de base de datos es un elemento que no tiene que ver con los proyectos de Almacenamiento pero que simplemente sirve para dar a entender que el Director de Administración de Datos tiene a su cargo varias áreas de TI.

El especialista de TI *senior*, a diferencia del *junior*, es aquel recurso que cuenta con la experiencia suficiente para poder afrontar y tomar decisiones en las diferentes situaciones que se presenten durante la administración de los recursos tecnológicos.

## VI. DESCRIPCIÓN DEL PUESTO DE TRABAJO

### 1. OBJETIVOS

Mi objetivo como administrador de almacenamiento en z/OS fue ser el especialista de TI encargado de administrar la información del cliente cerciorándome de que los recursos de almacenamiento, vistos desde el enfoque del software, estuvieran siempre disponibles para la operación de la organización. Las principales actividades que desempeñé fueron:

- ✓ Monitorización de los recursos de almacenamiento.
- ✓ Mantenimiento de los recursos de almacenamiento.
- ✓ Atención a requerimientos para definir o modificar recursos de almacenamiento.
- ✓ Dar soporte a incidentes relacionados al software de almacenamiento. En caso de que algún incidente tuviera su origen en un problema de hardware, sería mi labor escalar el problema con el área de soporte de la infraestructura o con el proveedor del hardware.
- ✓ Apoyar a los system programmers para la realización de cambios en la infraestructura de almacenamiento en mainframe.
- ✓ Análisis y planeación de la capacidad de los recursos de almacenamiento.
- ✓ Administración de datos: migración de datos entre dispositivos de diferente performance y respaldo de información empleando y administrando las herramientas que hicieron posible la automatización de estas actividades.
- ✓ Particularmente mi puesto requería disponibilidad de tiempo para poder rotar en los distintos horarios dentro de un servicio de tipo 24/7.

## 2. RETOS

Mis retos en el puesto fueron variando en función de las asignaciones particulares.

El soporte a incidentes presentó el mayor reto para mí debido al amplio margen de situaciones posibles que se podrían presentar, la inmediatez y la exigencia que la atención de los incidentes acarrearán consigo. En consecuencia el trabajo bajo presión fue el día a día durante la atención a incidentes.

Otro reto durante la atención de incidentes fue la delegación y deslinde de responsabilidades para las situaciones en las que el área de Almacenamiento no debía tener responsabilidad. Debido a que el concepto de almacenamiento está presente en todo proceso que se ejecuta en un mainframe, a menudo suele ser la primera consideración de falla en la ejecución del mismo. La forma en el que z/OS está implementado para el hospedaje de archivos en disco hace que sea muy común recibir mensajes como "el proceso canceló por falta de espacio" o "no hay espacio suficiente para crear el archivo", pero no siempre estas aseveraciones implican una responsabilidad del área de almacenamiento, a veces es responsabilidad del usuario definir correctamente los parámetros de espacio para que el sistema pudiera asignar el espacio suficiente para un determinado trabajo. Por tanto era un reto y una responsabilidad mía poder demostrar por qué algún incidente en un proceso dado no entraba dentro del alcance de nuestra administración. Considero entonces, que en un mainframe la mayoría de especialistas TI, independientemente de su área, deberían tener conceptos básicos de almacenamiento en z/OS.

En proyectos que supusieron una migración o mantenimiento, el cumplimiento de las actividades dentro de los tiempos acordados de finalización fue la meta a cumplir pero el reto estaba en mi capacidad para automatizar las tareas repetitivas, garantizando en todo momento la integridad de los datos y el uso eficiente de CPU y memoria.

La CPU y memoria son recursos cuyo uso es preferido para los procesos que realizan operaciones relacionadas con el negocio. Las actividades que tienen que ver con la administración, monitoreo, o mantenimiento del sistema, normalmente deben abstenerse de hacer uso indiscriminado de CPU y memoria. Se debe considerar que IBM cobra cada mes a sus clientes por el uso de CPU de sus mainframes y a su vez cobra un excedente por sobrepasar el uso establecido.

### 3. RESULTADOS ESPERADOS

Tanto el cliente como el proveedor de servicios TI esperaban de mí que mi desempeño en la atención de requerimientos e incidentes cumplieran con los acuerdos de niveles de servicio. Lo más importante fue que no se perdiera la información crítica del negocio, pero a su vez el cliente necesitaba que las interrupciones al servicio causadas por los recursos de almacenamiento se presentaran lo menos posible.

Cuando se planificara la realización de actividades de mantenimiento y migraciones tecnológicas, se esperaba de mí que las actividades las cumpliera en los tiempos acordados. Así mismo, debí ser capaz de informar con antelación de una posible prórroga a causa de otros factores que quedaran fuera del alcance de nuestra administración.

Al cumplir con estos resultados, yo lograría afianzar mi posición en el área y, en consecuencia, en la empresa; obtendría la confianza del cliente y muy probablemente sería recomendado para llevar a cabo otros proyectos con este u otros clientes. A su vez, mi participación contribuiría para que se llevara a efecto la continuidad en los servicios ofrecidos al cliente por parte de las empresas de servicios TI para las que habría laborado.

## VII. PARTICIPACIÓN EN PROYECTO: OUTSOURCING DE ALMACENAMIENTO CENTRAL

El *outsourcing* o subcontratación es la gestión de una actividad empresarial llevada a cabo por un proveedor externo.

El cliente requería que un proveedor de servicios TI realizara la administración del uso del espacio de sus dispositivos de almacenamiento en mainframe bajo el sistema operativo z/OS. Para ello sería necesario formar un equipo de especialistas IT encargados de atender requerimientos, dar solución a incidentes, realizar reportes y análisis; todo dentro de un servicio 24/7.

### 1. ANTECEDENTES

Como resultado de experiencias previas, el cliente se percató que llevar a cabo directamente la gestión de toda su infraestructura de TI le había salido contraproducente debido a, entre otras cosas, los altos costos que acarrea el preparar al personal para el manejo de la infraestructura, las pérdidas de los recursos invertidos por la rotación del personal y la falta de experiencia en un área que no tiene que relación con su negocio. A su vez existían proyectos que por su naturaleza y duración, involucraban la contratación de personal de forma temporal. Esto originaba que la gestión por parte del cliente fuera más propensa a caer en fallos.

De esta manera el cliente optó por licitar y delegar la administración de varias áreas técnicas, incluyendo el área de "Almacenamiento Central", entre varias empresas especializadas en el ramo de servicios IT.

### 2. REQUERIMIENTOS DEL PROYECTO

El cliente y el proveedor determinaron las actividades que debieron ser realizadas por el equipo de especialistas durante el transcurso del proyecto. De manera global, estas son las actividades que se debieron realizar:

**Atención de requerimientos** cumpliendo con el nivel de servicio acordado. Para determinar el nivel de servicio, a partir de todos los requerimientos generados en un mes, se contarían aquellos que se atendieran dentro del tiempo establecido y que cumplieran con las características dadas por los usuarios. De esta forma el porcentaje obtenido sería equivalente al nivel de servicio. El acuerdo de nivel de servicio establecido para la atención de requerimientos fue de 90%.

**Administración de los recursos almacenamiento** es el control de los recursos de almacenamiento haciendo uso del software que viene incluido en la distribución de z/OS y software de terceros (no IBM) que fueron adquiridos por el cliente para complementar la administración del almacenamiento. De acuerdo al tipo de actividad a realizar, el cliente estableció un protocolo a seguir, el cual se detalla a continuación:

- Aquellas actividades que implicaran modificar la capacidad de algún recurso de almacenamiento de forma temporal y que no pusieran en riesgo la actividad del negocio, que conoceremos también como *la producción*, no iban a requerir autorización, únicamente notificación por escrito al nivel superior con el compromiso de restablecer el recurso utilizado una vez que la eventualidad hubiese finalizado.
- Aquellas actividades que implicaran modificar la capacidad de algún recurso de almacenamiento de forma permanente y que no pusieran en riesgo la producción, iban a requerir autorización oral al nivel superior y notificación por escrito.
- Aquellas actividades que implicaran modificar la capacidad de algún recurso de almacenamiento de forma ya sea temporal o permanente, pero que por su naturaleza requirieran atención inmediata, deberían ir autorizadas mediante escrito por parte de los dos niveles superiores.
- Aquellas actividades que implicaran modificar la capacidad de algún recurso de almacenamiento de forma ya sea temporal o permanente, pero que por su naturaleza no requirieran atención inmediata, deberían ser autorizadas por una solicitud de cambio. Las solicitudes de cambio fueron gestionadas mediante paquetes de software que facilitarían el control de los recursos a modificar, las fechas propuestas y las aprobaciones que fueron necesarias para llevar a efecto esos cambios.

**Atención a incidentes.** A cualquier hora del día, durante los 365 días del año al menos un especialista debía dar soporte a los incidentes que se presentaran en la producción. Dependiendo de la severidad del incidente, se estableció un nivel de servicio que vino a ser determinado por el tiempo de atención y la imputabilidad de

los incidentes al área de Almacenamiento. Se admitió como cumplimiento de los niveles de servicio a la atención en tiempo de 90% de incidentes ocurridos en un mes y que de todos ellos sólo el 5% fueran imputables al área de Almacenamiento con severidad baja. Tener en el recuento mensual al menos un incidente de severidad alta adjudicado al área sería entendido como un incumplimiento en el nivel de servicio.

**Acuerdos de niveles de servicio.** El objetivo de estos acuerdos fue procurar mantener el servicio con la mayor calidad posible. El no cumplir con estos acuerdos haría acreedor al proveedor de servicios IT a amonestaciones, multas o en casos más severos, a la terminación del contrato y/o acciones legales contra quien resultara responsable de los daños ocasionados al cliente.

### 3. METODOLOGÍA EMPLEADA

Las actividades establecidas para llevar a efecto la administración de la infraestructura de almacenamiento en mainframe se encuentran apoyadas en los conceptos y mejores prácticas establecidas en la Biblioteca de Infraestructura de Tecnologías de la Información o mejor conocida como ITIL (*Information Technology Infrastructure Library*). Parte de estos conceptos se vieron aplicados en las siguientes prácticas:

- Utilización de estándares para el control, operación y administración de los recursos.
- Revisión y reestructuración de procesos de manera que existiera una mejora continua.
- Generación de documentación sobre todas las actividades que se llevaron a cabo, de manera que hubiese una transmisión de conocimiento.

Varias de las actividades acordadas con el cliente las encontramos en los libros de ITIL:

- Gestión de incidentes
- Gestión de problemas
- Gestión de cambios
- Gestión nivel de servicio
- Gestión de la capacidad
- Gestión de la disponibilidad

#### 4. ALCANCE DEL PROYECTO

Nuestra administración dentro del equipo de almacenamiento se limitó a los recursos de software. La instalación y mantenimiento de hardware quedó a manos de proveedores (manufactureros) y, en su defecto, de los system programmers. En caso de que detectáramos a través de los mensajes del sistema una falla en el hardware de almacenamiento, sería nuestra labor comunicar de la situación al área de soporte adecuada para llevar a cabo su mantenimiento o remplazo.

La memoria virtual (memoria central + memoria auxiliar) no entró en la administración del área de Almacenamiento. Esa actividad quedó a cargo de los system programmers que dieron soporte directo al z/OS.

El contenido de los archivos a administrar fue de carácter confidencial y solo teníamos permitido realizar operaciones tanto de creación, modificación de características, reubicación y borrado de estos siguiendo las políticas establecidas por el cliente. En ningún caso tuvimos permitida la edición del contenido de un archivo que no fuera perteneciente a nuestros usuarios o al equipo de Almacenamiento. Aún así cualquier intento por violar estas políticas eran controladas por el sistema de seguridad de IBM para z/OS llamado RACF (*Resource Access Control Facility*).

#### 5. PERFIL DEL ESPECIALISTA

El perfil que debí cubrir para poder trabajar en este proyecto contemplaba las siguientes aptitudes:

- ✓ Trabajo bajo presión.
- ✓ Disponibilidad de tiempo y horario.
- ✓ Capacidad de análisis.
- ✓ Sensibilidad sobre el dato crítico que se iba a administrar.
- ✓ Fue deseable la capacidad de automatizar tareas rutinarias.

El conocimiento técnico específico bajo z/OS que debí dominar fue:

**TSO** (*Time Sharing Option*) es un ambiente que permite la comunicación e interacción con el sistema operativo haciendo uso compartido del CPU a través de todos los usuarios.

**ISPF** (*Interactive System Productivity Facility*) es una interfaz de usuario que permite acceder a varias facilidades del sistema operativo de forma amigable bajo TSO. Cuenta con un editor de texto y a menudo es usado como interfaz de programas de aplicación.

**JCL** (*Job Control Language*). Es un lenguaje para redactar instrucciones de ejecución de un programa. Con las instrucciones o sentencias se le indica al sistema operativo qué tareas debe realizar, la secuencia en que deben realizarse y las entradas y salidas que debe usar. Se deberán conocer las principales utilerías para manejo de datos, ordenamiento de registros y formateo de discos y cintas.

**SDSF** (*System Search and Display Facility*) Controla el procesamiento de jobs (detener, cancelar, reanudar, purgar). Sirve también para monitorizar procesos y visualizar su salida. Permite ejecutar comandos de sistema y visualizar el log del sistema.

**DFSMS** (*Data Facility Storage Management Subsystem*). Comprende un conjunto de productos para la administración de almacenamiento en z/OS. DFSMS es un ambiente operativo que ayuda a automatizar y centralizar el manejo del almacenamiento con base a las políticas que la instalación del cliente define para la disponibilidad, rendimiento, espacio y seguridad de los datos. DFSMS comprende los siguientes elementos:

- DFSMSdfp. Elemento base de z/OS
- DFSMSdss. Elemento opcional de z/OS y que el cliente adquirió para complementar la gestión.
- DFSMSrmm. Elemento opcional de z/OS y que el cliente adquirió para complementar la gestión.
- DFSMShsm. Elemento opcional de z/OS y que el cliente adquirió para complementar la gestión.
- DFSMStvs. Elemento opcional de z/OS y que el cliente no adquirió.

**DFSMSdfp.** Este elemento realiza las funciones básicas necesarias para la gestión del almacenamiento; entre ellas se puede destacar el manejo de los recursos usando las políticas que el administrador determina, gestión de montaje de cintas y gestión de los dispositivos de almacenamiento como son: controladores de disco, bibliotecas de cintas automáticas o servidores de cintas virtuales.

**DFSMShsm.** Este elemento administra el espacio en disco. Realiza tareas de respaldo, reubicación de datos a cinta y eliminación de datos de acuerdo a las políticas establecidas en DFSMSdfp por el administrador.

**DFSMSrmm.** Gestiona la información contenida en dispositivos removibles como pueden ser cintas y dispositivos ópticos.

**REXX.** Lenguaje de programación estructurada. Es una herramienta que viene integrada con z/OS. Su enfoque está en el manejo de cadenas de caracteres. La usaría principalmente como herramienta para extraer datos del sistema y elaborar reportes.

**Software SUN/StorageTek.** Para la solución de biblioteca de cintas automática de la marca StorageTek se hizo uso de un software en z/OS que controlaba los montajes de cintas y las políticas para el uso de estas. Para poder hacer uso adecuado de este sistema yo necesitaría conocer:

- Los comandos para dar soporte en caso de problemas en montajes.
- Uso de procedimientos para obtener reportes sobre el estatus y contenido de las cintas.
- Comandos de mantenimiento de cintas.

## 6. RECURSOS DE ALMACENAMIENTO ADMINISTRADOS

Los recursos de almacenamiento son entidades que el sistema operativo z/OS reconoce como tal y por tanto sólo pueden ser vistos desde una forma conceptual, no física.

Por otro lado, los dispositivos de almacenamiento sí pueden ser vistos tanto de forma tangible como conceptual. Los dispositivos virtuales son vistos por z/OS como si fueran reales. Actualmente se generan dispositivos con tecnología más avanzada que emulan la tecnología que usaban los mainframe de hace 20 años.

### 6.1 DISPOSITIVOS DE ALMACENAMIENTO

Reiterando, nuestra administración no contemplaba el realizar instalaciones ni modificaciones directas sobre los dispositivos de almacenamiento; en cambio, sí estaba a nuestro cargo el gestionar su uso y mantenimiento apoyándonos en el software de gestión de almacenamiento instalado en z/OS. A continuación listo los tipos de dispositivos de almacenamiento para mainframe que se utilizaron en el

centro de datos del cliente:

**DASD** (*Direct Access Storage Device*) es una tecnología que hoy en día es emulada con servidores de arreglos de discos RAID. Un par de ejemplos de esto son el servidor *IBM System Storage DS8880* y el *Virtual Storage Platform (VSP)* de Hitachi. Los proveedores de controladores de discos más reconocidos para mainframe son: IBM, EMC<sup>2</sup> y Hitachi.



VSP de Hitachi



IBM System Storage DS8880

**Cartucho de cinta Magnética** (*Tape cartridges*). Las cintas magnéticas pueden ser usadas tanto de forma emulada como virtual. La diferencia radica en que el dispositivo que realiza la emulación es de la misma naturaleza pero con una tecnología, por lo general, más avanzada o distinta; y la virtualización es el uso de una tecnología de naturaleza distinta para simular la función de otro dispositivo. Un ejemplo de emulación es el uso de nuevas tecnologías de almacenamiento en disco para emular el funcionamiento de DASD; un ejemplo en la virtualización, son las cintas virtuales se encuentran realmente almacenadas en disco y este dispositivo es el que simula su comportamiento.



Tape cartridges y tape drive usados en mainframe

**Bibliotecas de cintas automáticas** (*Automated Tape Libraries*). Son dispositivos que están formados por módulos que tienen celdas donde son ubicados los cartuchos de cinta. Hay un espacio donde hay drives para colocar los cartuchos que son solicitados por el mainframe y un área donde se gestiona el ingreso y salida de cartuchos. El sistema es manipulado por brazos que se van trasladando a lo largo de la biblioteca para localizar los cartuchos solicitados y colocarlos en los drives para las operaciones de E/S. Un ejemplo de este sistema es el sistema Storagetek SL8500 Modular Library System. Es común referirse a estos sistemas simplemente usando el término *library*.



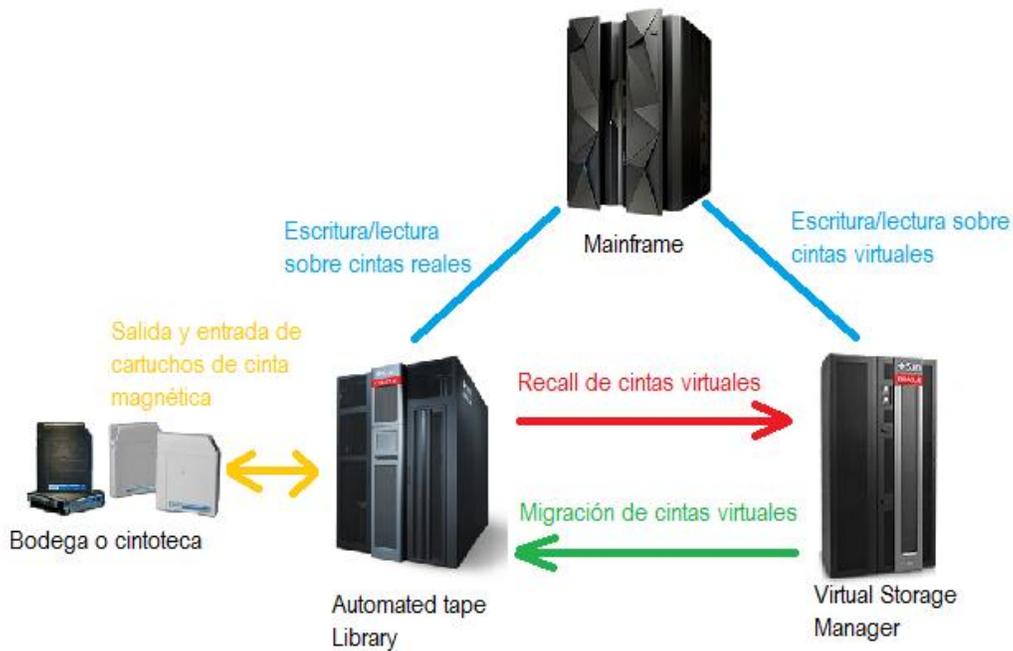
**Storagetek SL8500 Modular Library System**

**Bibliotecas de cintas virtuales** (*Virtual Tape System o VTS*). Son servidores basados en tecnología de disco magnético que simulan el funcionamiento de una biblioteca de cintas automática. Tienen la ventaja de tener menor tiempo de respuesta para las operaciones de E/S y alta disponibilidad debido a que todos los volúmenes se encuentran en virtualmente en el sistema.



**Storagetek Virtual Storage Manager, propiedad de Sun/Oracle**

El cliente adquirió una solución mixta de Storagetek usando un Virtual Tape System y una Automated Tape Library. Las cintas virtuales por ser de mayor rendimiento de E/S fueron usadas para recibir los datos que generaban los procesos en z/OS. Después, el sistema virtualizador se encargaba de migrar el dato menos actualizado hacia las cintas físicas encontradas en las celdas de la biblioteca.



**Esquema del sistema de almacenamiento virtual implementado en el CPD del cliente**

## 6.2 RECURSOS DE ALMACENAMIENTO

Desde la perspectiva de z/OS, se describirán los recursos de almacenamiento que debieron ser gestionados por el equipo de almacenamiento:

**Volúmenes DASD.** Existen varios tipos de DASD reconocidos por z/OS, sin embargo en la instalación del cliente se usó el DASD tipo 3390 en sus cuatro modelos: 3390-3, 3390-9, 3390-27 y 3390-54 con capacidades de 3, 9, 27 y 54 GB respectivamente. La variabilidad de modelos reside en que un modelo de menor capacidad tiene menos posibilidad de tener muchos archivos y por tanto su rendimiento aumenta al tener menos accesos concurrentes; en situaciones donde el rendimiento del dispositivo no es crítico, se usarían discos de mayor tamaño. En estos dispositivos se almacenan todos los datos activos tales como los archivos del sistema operativo, bases de datos, programas y todos los archivos de usuario que están disponibles para su uso inmediato.

**Volúmenes de cinta.** (*Tape Volumes*) Al igual que en DASD existen varios tipos de volúmenes de cinta reconocidos por z/OS sin embargo en esta instalación se emplearon los siguientes tipos: 3490 y 3590. La diferencia entre estos tipos radica en su capacidad y el formato de escritura, el 3590 tiene mayor capacidad de almacenamiento que un 3490.

**Data sets.** Un data set en z/OS es el equivalente a un archivo en otros sistemas operativos, sin embargo la documentación históricamente prefiere el uso del término *data set* en vez de archivo (*file* en inglés). A diferencia de los archivos de windows o Unix, los datos no están estructurados en flujos de bytes, salvo en el caso de módulos o programas (objetos); se estructuran en registros lógicos y bloques determinados por su organización. El sistema operativo a su vez incluye métodos de acceso para cada tipo de organización de data set. Existen tres tipos principales de data sets:

- **Secuenciales:** Son elementos de datos que son almacenados secuencialmente. Por ejemplo, para recuperar el décimo elemento, el sistema debe pasar por los primeros nueve.
- **Particionados (PDS):** Consiste en un directorio y sus miembros o *members*. El directorio conserva la dirección de cada member y así hace posible a los programas o al sistema operativo acceder al member directamente. Sin embargo, cada member consiste de registros secuencialmente almacenados. Los data sets particionados son comúnmente conocidos como *bibliotecas*. Los programas son almacenados como members de PDS.

- **VSAM (Virtual Storage Access Method):** En un data set de tipo VSAM los registros son elementos que son almacenados con información de control de manera que el sistema pueda recuperar un elemento sin pasar por todos los elementos precedentes a este.

**Storage groups.** Los *storage groups* son estructuras que sirven para agrupar discos o cintas y almacenar un tipo de dato particular en estos.

**Tape Libraries.** z/OS no distingue entre bibliotecas de cintas virtuales y físicas.

**Data sets de control (CDS).** Son bases de datos jerárquicas que contienen el registro de los archivos, dispositivos y configuraciones manejados por un sistema. Los registros se consultan a través de comandos y utilerías. En este caso sólo se daría mantenimiento a los archivos de control de los productos para la administración del almacenamiento. Normalmente, un archivo de control es de tipo VSAM, para incrementar la velocidad de acceso a los registros.

**Catálogos.** Son data sets de tipo VSAM que guardan el registro de los data sets y los volúmenes/dispositivos donde se encuentran almacenados estos. Con los catálogos se puede localizar un data set entre cientos de dispositivos de almacenamiento (cintas y discos) con solo indicar su nombre o parte de este. Son elementos críticos del sistema y deben ser monitoreados para evitar que se lleguen a saturar. Una consecuencia del uso de catálogos es que no pueden existir dos data sets con el mismo nombre que estén catalogados. En el caso de que existan dos data sets con el mismo nombre en diferentes volúmenes, entonces solo un data set se mantendrá catalogado mientras que el otro se mantendrá sin catalogar y no será posible localizarlo usando únicamente su nombre, se deberá indicar también el volumen y el tipo de dispositivo donde se encuentre.

**VTOC.** (*Volume table of contents*) es un data set que describe el contenido de un volumen DASD en el cual reside. El VTOC ubica todos data sets en ese volumen. Está compuesto de bloques de control de data set (DSCBs). Los DSCBs describen el tipo y la ubicación de los data sets en ese volumen, los registros no usados del VTOC y el espacio no ocupado en el volumen.

**Constructs.** Los constructs son objetos de SMS (storage management subsystem) que guardan las características físicas, de performance y de gestión que tienen los data sets administrados por SMS.

**Grupos de datos generacionales GDG.** Con esta estructura, se pueden actualizar sucesivamente generaciones de datos relacionados. Cada data set en el

grupo se llama generación y cada generación puede o no tener las mismas características que las otras. Las ventajas de esta estructura son:

- Todos los data sets en el grupo pueden ser referenciados por el mismo nombre.
- El sistema operativo es capaz de mantener las generaciones en orden cronológico.
- Las generaciones obsoletas o viejas pueden ser automáticamente borradas por el sistema operativo.

## 7. DESARROLLO DEL PROYECTO

### 7.1 ATENCIÓN A REQUERIMIENTOS

#### JUSTIFICACIÓN

Los datos generados y los recursos empleados por las aplicaciones que corren en ambientes productivos tienen relación directa con el negocio del cliente, por lo tanto se consideran información sensible y no pueden ser manipuladas por los mismos usuarios que las desarrollan. Así mismo es necesario llevar un control de los recursos que se generan en estos ambientes para evitar un uso descontrolado y no planificado que afecte el desempeño de otras aplicaciones del negocio.

Por otro lado, no existe restricción para que los usuarios generen información propia y hagan uso de los recursos de almacenamiento, sin embargo los recursos asignados para datos de usuario son más limitados que los asignados a las aplicaciones y el ingreso a ambientes productivos generalmente está reservado a los usuarios de áreas de soporte o mantenimiento. Los desarrolladores de aplicaciones a menudo solo tienen acceso permitido a ambientes de desarrollo.

Por consiguiente, si una aplicación es instalada o actualizada en un ambiente productivo y esta requiere de recursos de almacenamiento o modificaciones a los mismos antes de ejecutarse, entonces el área de almacenamiento es quien debe crear o modificar los recursos que la aplicación requiera.

## CATÁLOGO DE SERVICIOS

El cliente estableció el siguiente catálogo de servicios para la atención a los requerimientos de almacenamiento en mainframe:

- ✓ Creación de data sets secuenciales
- ✓ Creación de data sets particionados
- ✓ Creación de grupos de datos generacionales
- ✓ Creación de data sets VSAM
- ✓ Asignación de espacio para aplicaciones
- ✓ Modificación de retención de data sets
- ✓ Modificación de políticas de respaldo
- ✓ Definición de recursos para nueva aplicación

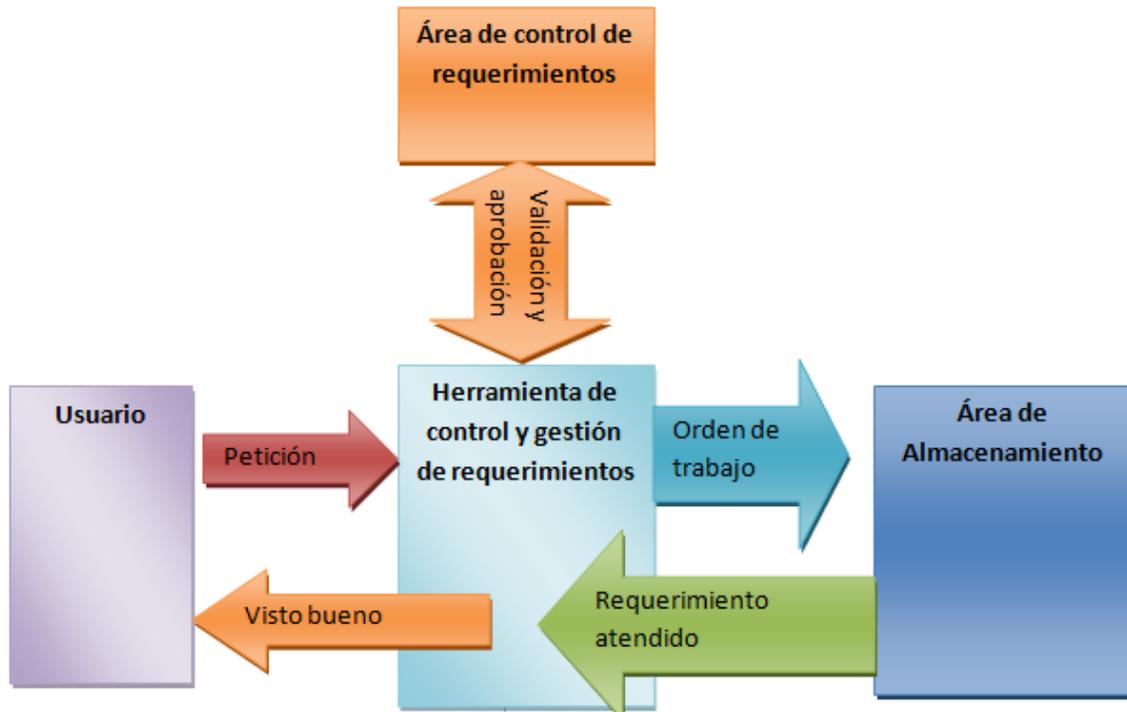
Todos aquellos requerimientos que no estuviesen incluidos en el catálogo serían considerados como peticiones especiales, en cuyo caso requerirían la solicitud de un cambio y, dependiendo las circunstancias y naturaleza de la petición, el área se reservaría el derecho de atenderlos.

Todos los data sets que hubieron de crearse se generarían vacíos, o sea sin datos. El usuario o aplicación sería el encargado de ingresar información en el data set. En ningún momento el área de almacenamiento podría realizar tareas de edición o visualización del contenido de los data sets de aplicación.

La realización de respaldos fue una tarea que se llevó a cabo de forma automática, apegándose a las políticas del cliente, por lo que era responsabilidad del usuario u aplicación realizar sus propios respaldos cuando fueran requeridos de forma especial o manual.

## ESQUEMA DE TRABAJO

El esquema de trabajo empleado para la atención de requerimientos fue siguiente:



El usuario realizaba la petición de un requerimiento a través de una herramienta gestora de requerimientos, posteriormente el área de control de requerimientos validaba que los datos aportados por el usuario fueran congruentes y solicitaba los vistos buenos de los responsables involucrados. Una vez que ya había sido aprobada la petición se generaba una orden de trabajo para el área de Almacenamiento, misma que se debía atender en los tiempos estipulados por el cliente. Cuando se completaba el requerimiento, el especialista que atendía el requerimiento se encargaba de cerrar la orden de trabajo y la herramienta gestora solicitaba a su vez el visto bueno del usuario para dar por atendido y cerrado el requerimiento.

#### APEGO A LOS ESTÁNDARES ESTABLECIDOS

Con el fin de poder utilizar y reutilizar los recursos de almacenamiento de forma eficiente, el cliente elaboró una serie de estándares a seguir con el fin de facilitar el control de estos aspectos a través del sistema. Estos estándares se basaron principalmente en la forma de nombrar un data set para determinar la aplicación a la que pertenecería, los usuarios o grupos de usuarios que podrían acceder a este, el storage group donde se almacenaría y las políticas de retención o respaldos que aplicarían en su caso.

Cuando el usuario realizaba una petición para la creación de un data set, uno de los requisitos fue que el nombre del mismo cumpliera con los estándares establecidos por el cliente.

Para poder conocer los estándares primero se debieron tener presentes las reglas de nombrado de data sets en z/OS. A continuación se describen estas reglas:

- La longitud máxima del nombre de un data set es de 44 caracteres dividido en segmentos de tamaño variable entre 1 y 8 caracteres alfanuméricos los cuales van separados por puntos. Estos segmentos se llaman calificadores.
- Un calificador no puede empezar con un carácter numérico.
- Los calificadores no pueden contener caracteres especiales excepto los siguientes: @,\$,#, y Ñ(mayúscula).
- El nombre del data set no puede comenzar ni finalizar con punto
- No puede haber dos puntos contiguos

Un ejemplo del uso correcto de estas reglas es el siguiente nombre:

ABC.ABC.ABC123

Breve descripción de las políticas del cliente

El cliente estableció el siguiente estándar para los data sets de aplicación:

- El primer calificador, el cual es el de más alto nivel, debería hacer referencia al nombre del sistema y el tipo de negocio al que estaría destinado.
- El segundo calificador representaría las siglas asociadas al grupo de usuarios, área o aplicación.
- El tercer calificador variaba dependiendo la retención que el usuario quisiera darle al data set. Del mismo modo, este calificador llevaría intrínsecas las políticas de respaldo y migración.
- A partir del cuarto calificador la nomenclatura era libre.

Los estándares para los data sets de usuario fueron los siguientes:

- ✓ El primer calificador del data set era el identificador del usuario. Por ejemplo si el usuario es IBMUSER, los data sets aparecerían como:

IBMUSER.ACBD.PAGOS

## IBMUSER.GASTOS.F55

- ✓ El usuario sólo tendría autoridad para crear data sets de usuario y los de las aplicaciones para los cuales su área o grupo de trabajo tendría los permisos requeridos.

La atención de requerimientos sólo contemplaba la creación de data sets de aplicación y no de usuario, por lo que era tarea del mismo usuario crear sus propios data sets.

Los data sets del sistema y programas producto tienen un estándar que viene por defecto del fabricante y en muchos casos no es posible modificarlo. En los casos en los que el cliente tuviera posibilidad de elegir el estándar, este sería distinto al utilizado para los usuarios y aplicaciones, con el fin de separar su administración.

## CREACIÓN DE DATA SETS SECUENCIALES

Para llevar a cabo la creación de un data set secuencial, yo necesitaba conocer y verificar la siguiente información que vendría incluida en la orden de trabajo:

✓ Imagen de z/OS o Partición Lógica (LPAR) donde se realizaría el trabajo.
✓ Nombre del data set apegado a los estándares establecidos por el cliente.
✓ Longitud de registro, la cual se contabiliza en bytes.
✓ Formato del registro, el cual podía ser fijo o variable. Cuando era fijo todos los registros del data set tendrían la misma longitud. Cuando el formato era variable entonces el tamaño de cada registro podía variar no excediendo el valor de la longitud de registro.
✓ Si el data set requería un formato que no fuera el <i>básico</i> debía indicarlo el solicitante, en caso contrario yo usaría el formato básico. Los formatos alternativos disponibles son: <i>Extended</i> y <i>Large</i> .
✓ El tipo de dispositivo donde se almacenaría el data set. Si era en cinta ya no requería más información. Si el data set se debía alojar en disco, el solicitante debía especificarme el tamaño de las extensiones de espacio que ocuparía el data set a crear.
✓ De faltar especificar alguno de los puntos anteriores mi labor era solicitar la información faltante o en su defecto rechazar el requerimiento.

Una extensión o *extent* (inglés) es un segmento de espacio en disco que se va a reservar para el ingreso de datos al data set. Cuando el data set hace uso de todo el extent entonces, dependiendo de los parámetros indicados, el sistema le asigna otro extent del disco para continuar ingresando más datos. El extent primario es el

que se asigna cuando se crea el data set; el extent secundario se genera cuando se llena el primer extent del data set y aún sigue haciendo falta ingresar más datos al data set. El espacio que tienen definido los extents primario y secundarios se puede especificar en términos pistas o cilindros.

Una pista o track en un dispositivo DASD 3390 es equivalente a 56664 bytes. Un cilindro es equivalente a 15 tracks.

Un data set secuencial básico tiene las siguientes limitaciones para el uso de los recursos en disco:

<b>Extents X volumen</b>	<b>Volúmenes</b>	<b>Tracks por volumen</b>
16	59	65535

Un data set con formato extendido tiene las siguientes limitaciones para el uso de recursos en disco:

<b>Extents X volumen</b>	<b>Volúmenes</b>	<b>Tracks por volumen</b>
123	59	Sin límite

Un data set con formato *large* tiene las siguientes limitaciones para el uso de recursos en disco:

<b>Extents X volumen</b>	<b>Volúmenes</b>	<b>Tracks por volumen</b>
16	59	Sin límite

En cinta, todos los data sets, sin importar el tipo, son tratados como secuenciales, esto es debido a las limitaciones inherentes a estos dispositivos. En cinta no se hace uso de extents y el límite máximo de volúmenes a usar es de 255.

Una vez que ya tenía listos todos los parámetros necesarios proseguía a iniciar una sesión en el sistema destino para crear el data set. Existen tres formas de crear un data set: por comandos de TSO, por paneles de ISPF (los cuales invocan a TSO) o vía batch por medio de JCL. Nadie del equipo podía generar data sets para otras aplicaciones desde procedimientos en primer plano, como TSO o ISPF, la única forma era vía batch especificando en el JCL un usuario virtual el cual era el único con las facultades para tratar con data sets y/o recursos fuera del alcance de nuestros usuarios. Configurado a través de RACF, el usuario virtual sólo podía ser usado por los integrantes del equipo de Almacenamiento. Dejo un ejemplo en el "Apéndice A. Ejemplo de JCL para crear un data set secuencial". El JCL lo

mandaba a ejecutar como trabajo o *job* el cual enviaba las instrucciones al sistema operativo para que fuera creado el data set.

Un job se ejecuta sin requerir intervención del usuario. La tarjeta de job está compuesta por una sección de identificación y a partir de ahí se describen las instrucciones agrupadas en pasos. Un paso es un componente de la tarjeta de job que realiza una tarea específica y que contiene las especificaciones de un programa y los archivos de entrada y salida que va a procesar y las instrucciones que le indicarán al programa cómo procesar esos datos de entrada para obtener la salida deseada. Al mandar a ejecutar el JCL, el job se ejecuta en segundo plano permitiendo al usuario seguir trabajando mientras la tarea se lleva a cabo. Para crear data sets secuenciales o particionados no se requiere que un programa realice una tarea, sin embargo es requisito de la tarjeta de job especificar un programa para ejecutar un paso, por lo que es común usar un programa de tipo DUMMY llamado IEFBR14.

#### CREACIÓN DE DATA SETS PARTICIONADOS

Un data set particionado es una estructura que sólo existe en DASD. Está dividido en miembros secuencialmente organizados. Funciona de manera similar a un directorio en UNIX. Cada miembro tiene nombre único de 1 a 8 caracteres. El máximo de miembros que pueden caber en el data set dependerá de las características con las que se haya creado el data set. Son indispensables para almacenar programas y módulos, y son muy útiles para almacenar códigos de programas, rutinas, JCL's y configuraciones del sistema. Su principal ventaja radica en que sin necesitar buscar en todo el data set uno puede recuperar cualquier dato de un solo miembro después de que el data set ha sido abierto. Existen data sets de este tipo con formato estándar y formato extendido.

Los data sets particionados con formato estándar tienen las siguientes limitaciones:

<b>Extents X volumen</b>	<b>Volúmenes</b>	<b>Tracks por volumen</b>
16	1	65535

Se debe definir el número de bloques de directorio que tendrá el data set del cual dependerá el número de miembros que podrá llegar a tener.

Los data sets particionados con formato extendido tienen las siguientes limitaciones:

Extents X volumen	Volúmenes	Tracks por volumen
123	1	Sin límite

No requiere especificarse el número de bloques de directorio, ya que va generándolos dinámicamente.

Para atender una solicitud de generación de data sets particionados era mi labor validar:

✓ LPAR donde se realizaría el trabajo.
✓ Nombre del data set (el cliente tenía estándares específicos para el nombrado de data sets particionados)
✓ Formato de registro.
✓ Longitud de registro.
✓ Formato del data set. En caso de ser estándar el solicitante debía especificar el número de bloques de directorio.
✓ Especificaciones de consumo de espacio

Dejo un ejemplo en el "Apéndice B. Ejemplo de JCL para crear un data set particionado."

#### CREACIÓN DE GRUPOS DE DATOS GENERACIONALES (GDG)

Los grupos de datos generacionales (GDG) constan de dos elementos: la base y las generaciones. La base es un registro que se almacena en un catálogo y esta tiene los parámetros que determinan el comportamiento del grupo. Las generaciones pueden ser data sets secuenciales o particionados, pero el cliente tiene restringido su uso a únicamente data sets secuenciales. Para generar las bases de GDG debí usar el programa IDCAMS.

Las generaciones de un GDG pueden ser almacenadas tanto en disco como en cinta.

Las políticas establecidas por el cliente determinaban que el comportamiento de los archivos GDG debía seguir las siguientes reglas:

- Las generaciones se iban a crear hasta llegar al número máximo de generaciones indicado por la base del GDG
- A partir de ahí, cuando fuera creada una nueva generación, la más antigua sería borrada de manera que la estructura tendría un comportamiento de cola.

Los datos que debían ser proporcionados para que yo pudiera atender una solicitud de creación de GDG's fueron:

✓ LPAR donde se realizaría el trabajo.
✓ Nombre de la base del GDG (Los nombres de las generaciones las crearía el sistema tomando como referencia el nombre de la base)
✓ Número máximo de generaciones.
✓ Si se requería que se generara la primera versión se continuaba con los siguientes puntos.
✓ Tipo de dispositivo: DASD o cinta.

Dejo a manera de ejemplo el "*Apéndice C. Ejemplo de JCL para crear base de GDG*"

#### CREACIÓN DE DATA SETS DE TIPO VSAM

Los data set VSAM son ocupados principalmente para almacenar datos que requieren acceso muy rápido y uso compartido a nivel registro sin comprometer la integridad de la información. Son usados principalmente como estructuras para albergar bases de datos jerárquicas y relacionales.

Los data sets de tipo VSAM en general pueden hacer muchos extents en disco, y los hay en formato estándar y extendido. El formato estándar tiene como principal limitante que no puede ser más grande de 4GB, el formato extendido tiene como límite la relación de esos 4GB multiplicados por una característica similar a la longitud de registro en los data sets secuenciales, llamada intervalo de control.

Para crear un VSAM también se usa el programa IDCAMS.

Los datos que debían ser proporcionados en la solicitud para que yo pudiera atender una solicitud de creación de VSAM son:

✓ LPAR donde se realizaría el trabajo.
✓ Nombre del data set.
✓ Atributos lógicos específicos del VSAM.
✓ Parámetros de consumo de espacio.

Los archivos VSAM emplean muchos parámetros de configuración que tienen impacto directo en su desempeño al accederlos, por tanto no podíamos trivializarlos en caso de que hubiera omisiones por parte del solicitante. Lo más

frecuente era pedirle al solicitante que nos facilitara el nombre de un data set modelo o un paso de JCL con los parámetros ya codificados al criterio del solicitante. Dejo un ejemplo de creación de VSAM en el "Apéndice D. Ejemplo de JCL para crear un archivo VSAM"

#### ASIGNACIÓN DE ESPACIO PARA APLICACIONES

Las aplicaciones consumen espacio en dispositivos DASD y cinta. Los estándares del banco establecían que no era necesario hacer requerimientos para consumir espacio en cinta; estos recursos se irían asignando conforme se fuesen requiriendo como resultado del análisis de la planeación de la capacidad.

Cuando una aplicación consumía espacio en disco y esta tenía asignada un storage group solo para ella, entonces era responsabilidad del usuario la planeación de su capacidad y la generación de los requerimientos de espacio correspondientes.

Si la aplicación compartía un storage group con varias aplicaciones más, entonces la asignación del espacio era parte de las responsabilidades del área de almacenamiento. La cantidad de espacio a asignar era determinada a partir de las actividades relacionadas con la planeación de la capacidad. Sin embargo, si el usuario tenía previsto realizar un consumo extraordinario era su responsabilidad hacer la solicitud de espacio correspondiente para cubrir este consumo.

Los requisitos para solicitar espacio fueron:

✓ LPAR donde se realizaría la asignación.
✓ Nombre la aplicación o máscara asociada a la aplicación.
✓ Cantidad de espacio solicitado en términos de bytes (MB,GB,TB).
✓ En caso de que la aplicación hiciera uso de dos o más storage groups, debería especificar para cuál de ellos es la solicitud.

Una máscara es una forma genérica de referirse a un conjunto de nombres de data set. Este método se apoya en el uso de los caracteres especiales \* y % para indicar valores genéricos. Su interpretación es la siguiente:

- La nomenclatura debe respetar las reglas de nombrado de data sets anteriormente expuesta.
- El caracter \* enmascara una cadena de caracteres de longitud 0 hasta 8. La cadena no puede contener el caracter punto (.)

- El carácter % enmascara solamente un carácter. El carácter oculto no puede ser punto.
- Un caso especial es el uso de dos asteriscos \*\*. Estos indican que el contenido que esconden no tiene límite de longitud, considera los puntos y puede variar de 0 a 44 caracteres siendo este último valor el límite para nombrar un data set. Cuando aparece un punto antes de los dos asteriscos (.\*\*) se puede ignorar para resolver uno de los casos de nombrado como la cadena de caracteres que le antecede.
- Se pueden concatenar hasta 8 caracteres % siempre y cuando se respete la longitud máxima por calificador. Por ejemplo, son errores los siguientes casos: AABBBB%, D12345%, %12345678.
- No pueden concatenarse más de dos asteriscos (\*\*, \*\*\*)

Suponga el siguiente nombre de data set:

BANCO.SOPORTE.REPORTE.F070215

Las siguientes son máscaras que satisfacen la resolución de su nombre:

- \*\*
- BANCO.\*\*
- BANCO.\*.SOPORTE.\*.F% % % % %
- B% % CO.\*\*.F070215
- BANCO.SOPORTE.REPORTE.F070215.\*\*

Es un error común esperar que el sistema considere el nombre de data set usando la siguiente máscara:

BANCO.SOPORTE.REPORTE.F070215.\*

Para atender la solicitud de espacio, mi primera labor era verificar que las máscaras efectivamente fueran dirigidas a los storage group para los que se solicitaba el espacio. Esto lo hacía consultando un miembro de un data set particionado que contiene las asociaciones de los storage group con las máscaras de data sets. Los miembros de esta biblioteca se llaman Automatic Class Selection Routines (ACS).

Una vez que se habían verificado las rutinas ACS, revisaba la disponibilidad de dispositivos DASD; y dependiendo del rendimiento que requiriera la aplicación se asignarían volúmenes de un tamaño particular.

Antes de asignar los volúmenes los debía formatear para que el sistema operativo los pudiera detectar. Se da por entendido que los discos ya estaban conectados e identificados por el sistema (esta tarea la realizaba el system programmer) pero faltaba que los discos estuvieran listos para poder trabajar con el sistema operativo. El formateo lo realizaba con un programa llamado ICKDSF y este se encargaba de poner la etiqueta y fabricar el VTOC del volumen con base en los parámetros que le indicaba, entre ellos su tamaño y posicionamiento en el volumen. El etiquetado de volúmenes estaba estandarizado por el cliente y tomaba en cuenta el nombre de la aplicación, el nombre del storage group o el tipo de información que almacenaría.

Para asignar los volúmenes a un storage group existe una interfaz llamada ISMF (Integrated Storage Management Facility) que puede ser invocada desde ISPF.

A manera de ejemplo dejo un procedimiento para asignar espacio a una aplicación en el "*Apéndice E. Ejemplo de asignación de espacio para una aplicación.*"

#### MODIFICACIÓN DE RETENCIÓN DE DATA SETS.

La retención de los data sets fue regulada por los estándares establecidos por el cliente. Para modificar la retención de un data set de forma rápida, bastaba con renombrarlo acorde al estándar para cambiar su retención. Sin embargo el cliente había tenido en consideración que existían aplicaciones que por diversas causas, a veces requerían alargar de forma temporal la retención de algunos data sets, sin modificar el nombre de estos, ya que de lo contrario podría ser necesario recompilar código, lo cual sería menos factible tratándose de situaciones eventuales.

Por ello se había establecido la modificación de retención de data sets como una actividad más dentro del catálogo de servicios de la atención a requerimientos de almacenamiento central.

Los requisitos que pedíamos para poder atender una solicitud de modificación de retención de data sets fueron los siguientes:

✓ LPAR donde se realizaría la modificación.
✓ Máscara o lista de data sets a modificar.
✓ Especificaciones de la nueva retención: Duración en disco, tiempo en cinta, tiempo que debía pasar antes de ser eliminado.
✓ Periodo en que la retención se mantendría vigente hasta que la retención debiera volver a los parámetros originales.

Para modificar la retención de un data set o máscara debía modificar un elemento de SMS asociado con los parámetros de retención de un grupo de data sets llamado *Management Class*. Una management class es un objeto conocido como *construct* que tiene definidas las políticas de retención y de respaldo de data sets. Pueden existir tantas management class como políticas se requieran. Las máscaras o data sets son asociadas a un management class a través de las rutinas ACS.

El procedimiento que seguía consistía en:

1. Localizar una management class que cumpliera con los parámetros indicados en la solicitud; si no existía, la debía crear.
2. Modificar la asociación de las máscaras o lista de data sets en la rutinas ACS, removiéndolas de la management class original a la nueva que sería usada de forma temporal.
3. Activar todos estos cambios en el sistema.
4. Mantener el registro de la actividad y llevar el control de estos cambios para poder llevar a cabo el regreso a la configuración original.

Dejo un procedimiento a manera de ejemplo para esta actividad en el "*Apéndice F. Ejemplo de modificación de retención de data sets* ."

#### MODIFICACIÓN DE POLÍTICAS DE RESPALDO.

Las políticas de respaldo vienen presentes también en la management class por lo que los requisitos para atender la solicitud y el procedimiento eran los mismos.

#### DEFINICIÓN DE RECURSOS PARA NUEVA APLICACIÓN

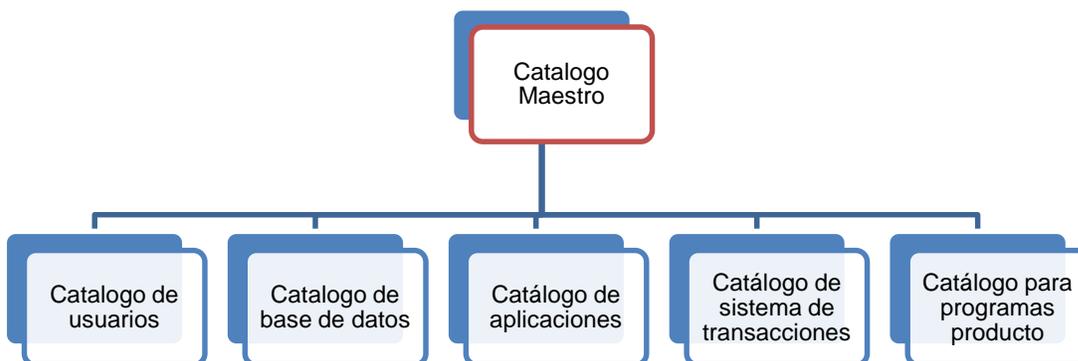
Para poder funcionar, una aplicación necesita poder usar los siguientes recursos de almacenamiento:

- ✓ Dispositivos de disco o DASD
- ✓ Dispositivos de cinta (opcional)
- ✓ Entradas de catálogo
- ✓ Tener perfiles de seguridad definidos para poder generar sus data sets con los permisos requeridos. (actividad realizada por el área de seguridad)

Las políticas del cliente establecían que para poder usar los dispositivos de almacenamiento en DASD, las máscaras de los data sets asociadas a una aplicación debían estar asociadas con los storage group y constructs que le correspondieran, apegándose a los estándares determinados por el cliente. De esta forma la información se almacenaría de manera organizada y sería administrada de acuerdo a las políticas establecidas.

Para poder usar entradas de catálogo, una aplicación debe tener permisos para hacer uso de un catálogo. En cualquier ambiente bajo z/OS es una práctica obligatoria designar catálogos para almacenar entradas de un tipo específico de aplicación. El objetivo de esta práctica, además de mejorar el rendimiento de la búsqueda, es disminuir el impacto en caso de que un catálogo se llegase a dañar. Por ejemplo, si todos los nombres de data set del todo el ambiente se llegasen a almacenar en un solo catálogo, en caso de que el catálogo se dañase, ningún data set podría ser accesible, esto es, cualquier usuario que quisiera iniciar sesión en el sistema no podría hacerlo, y ningún proceso nuevo podría correr en él. Recuperar un sistema en ese estado sería una tarea laboriosa y difícil. Aunado a lo anterior mientras se estuviese realizando la recuperación del mismo, no habría sistema, por lo que toda la operación del negocio que tuviese lugar ahí estaría detenida. Por ello es importante usar distintos catálogos para cada tipo de dato, separando por ejemplo, las entradas de base de datos de las entradas de data sets de sistema y a su vez estos de los datos de usuarios y aplicaciones.

El cliente, atendiendo a las sugerencias y mejores prácticas que le llegaron del fabricante (IBM), empleó el siguiente esquema de catálogos para dar cabida a todos los tipos de datos a manejar.



Este esquema así mismo no se limitaba a que hubiera un sólo catálogo por tipo de información. Existirían aplicaciones que por su tendencia a generar un gran número de entradas de catálogo requerirían de dos o más catálogos para mantener sus búsquedas lo más eficientes posibles y disminuir el riesgo de saturación de los mismos.

El catálogo principal al que accede el sistema para localizar información se llama catálogo maestro y todos aquellos catálogos que deriven de este se llamarán catálogos de usuario. En el catálogo maestro se deberán encontrar todas las entradas de los data sets que son necesarios para que el sistema pueda arrancar, así mismo se deberán encontrar almacenadas las ligas hacia los otros catálogos a través de unas entradas llamadas *alias*. Un alias es un prefijo que puede constar de uno o hasta 4 calificadores que al coincidir con la máscara o el nombre aportado por el usuario, hacen que la búsqueda sea redirigida desde el catálogo maestro hacia el catálogo de usuario al que apunta ese alias y se complete la localización y el acceso al dato.

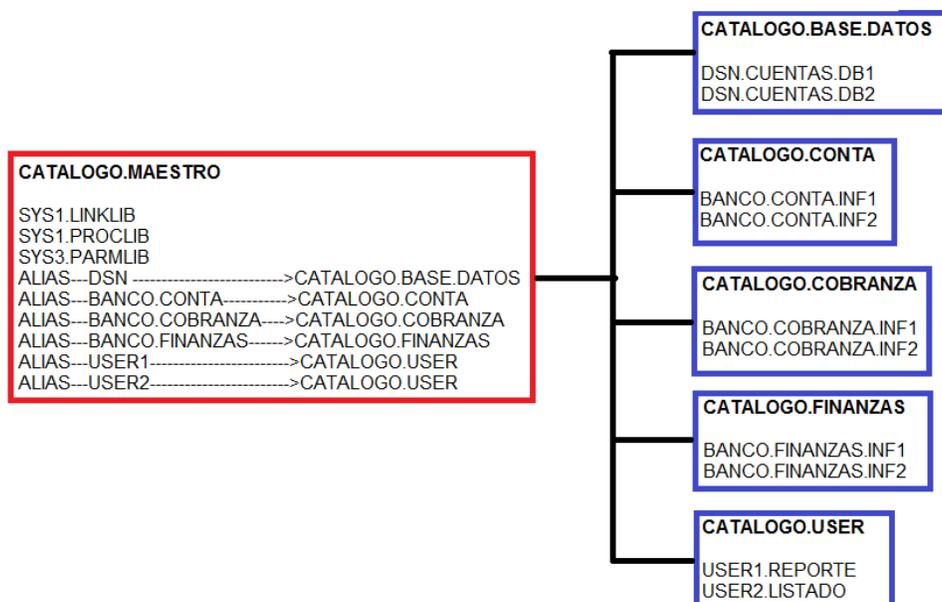
Una manera de ejemplificar este proceder es planteando la siguiente situación:

Los data sets de sistema cumplen con la máscara SYS1.\*\* y SYS3.\*\*. Los data sets de base de datos tienen la siguiente máscara DSN.\*\*. Los data sets de aplicaciones tienen las siguientes máscaras BANCO.CONTA.\*\*, BANCO.COBRANZA.\*\*, BANCO.FINANZAS.\*\*. Y los usuarios usan data sets que comienzan con su identificador.

Ahora supongamos el siguiente listado de entradas de catálogo que deben ser ubicadas donde corresponda:

- SYS1.LINKLIB
- SYS1.PROCLIB
- SYS3.PARMLIB
- DSN.CUENTAS.DB1
- DSN.CUENTAS.DB2
- BANCO.CONTA.INF1
- BANCO.CONTA.INF2
- BANCO.COBRANZA.INF1
- BANCO.COBRANZA.INF2
- BANCO.FINANZAS.INF1
- BANCO.FINANZAS.INF2
- USER1.REPORTE
- USER2.LISTADO

Siguiendo el esquema planteado por el cliente, el diseño del sistema de catálogos sería el siguiente:



Para poder generar data sets en cinta, también debí configurar las rutinas ACS de manera que se pudiera hacer uso de estos dispositivos. El elemento de SMS que tiene que ver con esto se llama *storage class*. Una *storage class* al igual que la *management class* es un objeto conocido como *construct*. Desde este se pueden especificar las características del dispositivo que deben usar los distintos data sets; es un requisito indispensable para poder asignar data sets hacia *storage groups*.

La retención de los data sets en cinta no se determina por la *management class* de SMS, para eso existe un software llamado *DFSMSrmm* (*removable media manager*) el cual va expirando volúmenes de cinta conforme van expirando los data sets que contienen. Los elementos donde se especifican estas políticas en este software se llaman *Vital Record Specifications* (VRS's). Los VRS se configuran especificando una máscara y la retención deberán tener todos los data sets que coincidan con ella antes de ser expirados. *DFSMSrmm* controla cuáles volúmenes de cintas pueden ser usados y cuáles no, dependiendo si están expirados o no.

En resumen, estas son las actividades que en mayor o menor medida yo debía realizar para definir los recursos de almacenamiento para una nueva aplicación:

- ✓ Asignar espacio en DASD (storage group)

- ✓ Definir un alias en catálogo, si no existía el catalogo adecuado para el alias, creaba uno nuevo y asociaba el alias a este.
- ✓ Definir las máscaras de la aplicación en las rutinas ACS.
- ✓ Definir los VRS con las políticas de retención para la aplicación en DFSMSrmm.
- ✓ Validarla disponibilidad de volúmenes en el storage group de cintas donde la aplicación almacenaría sus datos y asignar más volúmenes según se requiriera.

Para poder atender una solicitud de definición de recursos para una nueva aplicación, se debía especificar en la solicitud lo siguiente:

✓ LPAR donde se realizarían las definiciones.
✓ Máscaras a utilizar por la aplicación
✓ Espacio que se esperaba consumiera la aplicación. Dependiendo la situación podía requerirse que se hiciera una solicitud extra para asignación de espacio.
✓ Especificaciones especiales sobre las definiciones (storage group, políticas, etc.)

Dejo un ejemplo de este procedimiento en el "*Apéndice G. Ejemplo de definición de recursos de almacenamiento para una aplicación.*"

## 7.2 ADMINISTRACIÓN DE LOS RECURSOS ALMACENAMIENTO

Las actividades de administración de los recursos de almacenamiento básicamente eran aquellas que surgían como consecuencia del monitoreo, mantenimiento, del reemplazo de hardware y de la planeación de la capacidad.

Las actividades de monitoreo en las que me desempeñé fueron:

- ✓ Disponibilidad de espacio en Storage Group de disco.
- ✓ Disponibilidad de cintas en Storage Group de cintas.
- ✓ Porcentaje de uso de catálogos.
- ✓ Porcentaje de uso de archivos de control (SMS, DFSMSrmm, DFSMShsm)
- ✓ Funcionamiento correcto de los productos de DFSMS (DFSMSrmm, DFSMShsm)
- ✓ Porcentaje de uso de archivos VSAM de aplicaciones. Aunque no se les daba mantenimiento se le debía notificar al usuario para que realizara el soporte correspondiente.

- ✓ Porcentaje de uso de data sets particionados de aplicaciones. Era la misma política que con los VSAM.

El monitoreo derivó en actividades de soporte y mantenimiento que tuve aplicar. Estos son algunos ejemplos de esas actividades:

- ✓ Asignación de espacio a SG's de disco.
- ✓ Asignación de volúmenes a SG's de cintas.
- ✓ Reorganización o crecimiento de catálogos.
- ✓ Reorganización o crecimiento de archivos de control.
- ✓ Atención a incidentes con productos de almacenamiento, configuración de parámetros y puesta a punto.
- ✓ Reportes de porcentaje de uso en archivos VSAM.
- ✓ Reportes de porcentaje de uso en data sets particionados.

Cuando existía un reemplazo de hardware, las actividades de administración en las que yo apoyaba al área de infraestructura fueron:

- ✓ Formateo de los volúmenes asociados con el nuevo hardware
- ✓ Asociación de los nuevos volúmenes a los storage group.
- ✓ Migración de los datos de los volúmenes viejos a los nuevos
- ✓ Liberación de los volúmenes viejos (desasociarlos de los storage group, formateo para destrucción de datos, etc)

La planeación de la capacidad y detección de problemas la podía llevar a cabo gracias a la obtención de datos sobre el comportamiento de los recursos de forma periódica. De este modo a partir del dato histórico generaba gráficos y reportes que ayudaban a predecir cuántos recursos se iban a consumir en un futuro o si había un comportamiento anómalo de estos. Cuando tenía a la mano esta información, tenía posibilidad de intervenir para anticiparme a los eventos que pudiesen causar una interrupción en el servicio.

Todas las herramientas de SMS ofrecen métodos para obtener reportes de estatus sobre el consumo de data sets y de dispositivos. Al almacenar estos reportes de forma histórica se estaba posibilitando la realización de un análisis de consumo y a su vez una planeación de la capacidad. Es aquí donde la programación cobraba sentido para realizar análisis, formatear datos, generar resúmenes y automatizar tareas rutinarias.

Si bien existe mucho software de análisis y de gestión en muchos rubros, incluyendo el de almacenamiento, ese software es caro y sigue un esquema de licenciamiento igual que el software de IBM en mainframe, donde para mantener activo el software y el soporte que hay detrás de este, el cliente debía estar pagando una licencia ya sea mensual o anual. En la mayoría de los casos el cliente evitaba su uso a menos que fuera sumamente necesario. Por ello quedaba en nuestras manos como especialistas aprovechar de la mejor manera los recursos con los contábamos.

#### USO DE LENGUAJE REXX PARA GENERAR REPORTES Y AUTOMATIZAR TAREAS REPETITIVAS.

REXX es un lenguaje estructurado que se enfoca al manejo de cadenas de caracteres. Es muy útil para procesar registros y darles formato porque provee muchas funciones para manejo de estos. Además el lenguaje puede interactuar con otros subsistemas de manera que desde REXX se pueden ejecutar comandos de sistema o de otros subsistemas y recolectar las salidas y mensajes de estos.

Saber programar no era una aptitud que me habían solicitado para desempeñar mis labores en el área de almacenamiento, sin embargo, al aprovechar los beneficios que conlleva el poder automatizar procedimientos y generar reportes con análisis resumidos entonces fui consciente de que podía marcar una gran diferencia en cuanto a mi productividad con respecto al resto de mis compañeros de equipo.

#### GESTIÓN AUTOMÁTICA DEL ESPACIO CON DFSMSHSM

DFSMSHsm o mejor conocido como HSM (hierarchical storage management) es un software que provee facilidades para administrar los dispositivos de almacenamiento. HSM asegura la disponibilidad de espacio en los volúmenes DASD para que puedan ser almacenados nuevos data sets y extender los que ya existen. HSM también se asegura de que los respaldos de los data sets se mantengan disponibles en caso de que los data sets se dañen o se pierdan. HSM libera al administrador del manejo manual de almacenamiento para realizar labores de administración.

HSM utiliza las definiciones de las management class asociadas a cada data set para llevar a cabo su migración a dispositivos de menor rendimiento, su respaldo y finalmente su eliminación.

Mi labor como administrador del HSM era configurar este software para que llevase a cabo las tareas automáticas de manera que no interfiriera con las actividades de la producción en su consumo de CPU, la disponibilidad de espacio y de datos para procesos batch, etc.

Existen tres tareas principales que realiza el HSM de forma automática:

- Migración de data sets.
- Respaldo de data sets o volúmenes DASD.
- Eliminación de data sets.

Las tareas descritas también podían realizarse a través del HSM de forma manual. Las siguientes tareas sólo pueden realizarse de forma manual a través del HSM:

- Recall de data sets, que es una tarea inversa a la migración.
- Recuperación de data sets a partir de los respaldos generados.
- Reciclado de cintas para consolidar muchos volúmenes con poco uso en pocos volúmenes con mucho uso.

Durante las actividades de migración, el HSM trabaja considerando tres niveles de almacenamiento. El nivel 0 está compuesto por los discos DASD que albergan todos los datos activos, que están disponibles de forma inmediata si un proceso los solicita. El nivel 1 o ML1 (migration level 1) está conformado por dispositivos DASD ya sea de igual o menor rendimiento que los volúmenes del nivel 0, pero los data sets que sean migrados a este nivel quedarán inactivos. El nivel 2 o ML2 (migration level 2) puede ser conformado por cintas o por volúmenes DASD, pero es muy poco común el uso de DASD debido a que los data sets migrados sólo pueden ser multi-volumen en cinta. Para poder hacer uso de data sets inactivos en ML1 y ML2, estos se deben regresar al nivel 0 mediante *recall*.

El recall se puede invocar sin necesidad de dar un comando, basta que cualquier proceso o usuario autorizado intente hacer uso del data set para que el HSM comience a desmigrarlo.

ADMINISTRACIÓN DE DATA SETS CON DFSMSDSS.

DFSMSDSS es un software que ofrece las siguientes funciones:

- Movimiento y replicación de datos.
- Manejo de espacio.
- Respaldo y recuperación de datos.
- Conversión de data sets y volúmenes a otros formatos.
- Copia concurrente

El programa con el que se invocan estas funciones se llama ADRDSSU y su uso es únicamente vía batch.

Con este software me apoyé para mover eficientemente múltiples data sets de viejos volúmenes de DASD a nuevos volúmenes de DASD y, además me ayudaba a reducir la fragmentación de volúmenes.

La copia concurrente nos permitía copiar data sets mientras estos estaban siendo usados.

#### SOFTWARE DE STORAGETEK PARA ADMINISTRAR CINTAS REALES Y VIRTUALES.

Por medio del software de storagetek podía realizar actividades de mantenimiento sobre cintas físicas y virtuales.

Este software realiza las siguientes funciones de manera automática:

- Montajes de cintas reales y virtuales
- Migración de cintas virtuales a cintas reales

Las funciones que podía utilizar como administrador:

- ✓ Obtener reportes de cintas dentro de la biblioteca de cintas físicas.
- ✓ Obtener reportes de cintas usadas y cintas disponibles, tanto físicas como virtuales.
- ✓ Obtener el estado de una cinta o grupo de cintas físicas o virtuales.
- ✓ Correr procesos de validación y corrección de errores sobre cintas físicas y virtuales.

### 7.3 ATENCIÓN A INCIDENTES

Los incidentes básicamente se pueden agrupar en dos tipos:

- Pertenecientes al área de almacenamiento
- No pertenecientes al área de almacenamiento

Para entender la razón de por qué los incidentes que no pertenecían al área llegaban para ser atendidos, debemos revisar el siguiente esquema:

#### ESQUEMA DE TRABAJO



El centro de atención telefónica (CAT) se encargaba de recibir los llamados por incidentes que pudieran venir de los usuarios, el centro de mando o del operador del sistema, si bien, era posible que también las áreas de soporte lo contactasen, era poco frecuente.

El operador del sistema revisaba el correcto funcionamiento de z/OS, en caso que el sistema le mandara mensajes o se mantuviera a la espera de la intervención del mismo por una falla en algún software, entonces este buscaba contactar con el área de soporte al que consideraba responsable a través del CAT.

El Centro de Mando fue aquella área que se encargaba de vigilar que los procesos del cliente se ejecutaran sin contratiempos. Dependiendo de la característica de cada incidente que se presentaba, el Centro de Mando contactaba al CAT buscando a alguna área de soporte que considerara pertinente.

Las áreas de soporte podían ser cualquiera de las siguientes: sistema operativo, base de datos, comunicaciones, almacenamiento, seguridad, hardware, entre otras.

#### PROCEDIMIENTO

1. El operador, usuario o centro de mando detectaban un fallo en un proceso o en el sistema.
2. Se entablaba comunicación con el CAT para localizar a las áreas de soporte.
3. El CAT localizaba a las áreas de soporte requeridas y establecía una conferencia entre las áreas afectadas y los soportes.
4. El área afectada describía la situación al soporte.
5. El soporte determinaba si la situación quedaba dentro de su alcance.
6. En caso que la situación no quedara en el alcance del área de soporte requerida, esta debía exponer sus razones y sugerir el soporte de otra área. Se repetía el procedimiento desde el punto 2.

Cuando la situación estaba dentro del alcance del área de soporte, se realizaban las actividades para resolución del incidente, y se notificaba a las áreas afectadas para que reanudaran sus procesos.

El procedimiento parecía dar por entendido que el afectado debía conocer la naturaleza del incidente para poder determinar cuál área de soporte era la adecuada para atender el incidente. Esto permitía que llegasen a Almacenamiento Central casos que no pertenecían al área.

Mi proceder en la atención de un incidente para cumplir con los niveles de servicio contemplaba las siguientes actividades:

- ✓ Documentar el incidente de forma que se diera a conocer la severidad, la fecha y hora en que aconteció y la descripción resumida del mismo y la solución que fue aplicada en cada caso.
- ✓ Dar solución al incidente, de acuerdo al nivel de severidad, dentro de los tiempos establecidos. Se establecieron 4 niveles de severidad de acuerdo a las afectaciones hacia el negocio:
  - **Severidad 0.** Afectación a las actividades de un solo usuario. Tiempo de atención: 1 hora. Ejemplos: Fallos de inicio de sesión. Pérdidas de archivos de usuario.

- **Severidad 1.** Afectación a una aplicación secundaria. No impacta al negocio. Tiempo de atención: 15 min Ejemplos: Cancelación de proceso, pérdida de un archivo de aplicación.
  - **Severidad 2.** Afectación a una aplicación primaria. Afecta de forma parcial al negocio. Tiempo de atención: Inmediato. Ejemplos: Cancelación de procesos de ruta crítica, catálogo dañado, saturación de storage group.
  - **Severidad 3.** Afectación a todo el negocio. Tiempo de atención: Inmediato. Ejemplos: Degradación del sistema operativo. Daño de un catálogo importante o catálogo maestro.
- ✓ Notificar a los niveles superiores de responsabilidad cuando se presentaran incidentes de severidad 2 o 3, y además su atención se encuentre dentro del alcance del área.

INCIDENTES PERTENECIENTES AL ÁREA DE ALMACENAMIENTO

La experiencia en el área me ha permitido conocer una variedad de incidentes que entraban en el alcance del soporte de almacenamiento:

Incidente	Severidad	Descripción del impacto	Descripción procedimiento para solucionarlo
Saturación de archivo de control de backup del HSM	1	Los data sets de aplicación pueden quedarse sin ser respaldados o recuperados por HSM en caso de requerirse. La producción se mantiene en funcionamiento.	Cambio emergente para asignación de ventana para reorganizar el data set, de manera que el impacto sea mínimo.
Daño de archivo de control de backup del HSM.	1	Durante el proceso de recuperación, los usuarios y aplicaciones no pueden respaldar ni recuperar data sets bajo HSM.	Cambio emergente para asignación de ventana para recuperar el data set si hay respaldo del archivo de control, de manera que el impacto sea mínimo.
Saturación de archivo de control de migración del HSM	2	Los procesos batch pueden ser impactados de manera indirecta se puede provocar la saturación de storage groups.	Detención de procesos batch. Reorganización del archivo de control.
Daño de archivo de control de migración del HSM.	2	Los procesos batch pueden ser impactados de manera directa se puede provocar la saturación de storage groups y la imposibilidad de hacer recall de data sets migrados. Muchas cadenas batch corren el riesgo de tener que ser	Detención de procesos batch. Recuperación del archivo de control si hay respaldo. En caso de no haber respaldo el impacto puede ser crítico.

		reprocesadas.	
Saturación de catálogo	<b>2-3</b>	Los procesos que generen entradas dirigidas hacia ese catálogo van a fallar. Pueden ser tantos cancelados como el espectro de entradas que puedan ser dirigidas hacia el catálogo. El servicio está en riesgo.	Detención de mallas de procesos batch afectadas. Recomendable detener todo el batch. Reorganización o crecimiento del catálogo.
Daño de catálogo	<b>3</b>	Los procesos que generen entradas dirigidas hacia ese catálogo van a fallar. Pueden ser tantos cancelados como el espectro de entradas que puedan ser dirigidas hacia el catálogo. El servicio está en riesgo y probablemente se deba reprocesar cadenas batch. Si el daño es en el catálogo maestro el impacto puede involucrar a todo el sistema como tal.	Detención de mallas de procesos batch afectadas. Muy recomendable detener todo el batch. Recuperación del catálogo.
Saturación de storage groups	<b>2</b>	Afectación a los procesos que escriban datos hacia ese storage group específico.	Migración de data sets o asignación de espacio extra de forma temporal
Perdida de data set por error de configuración.	<b>2</b>	Ejemplos de esto: Storage groups que no debían ser habilitados para migración o asignación errónea de management class a grupos de archivos. El impacto puede ser provocado a una cadena o malla batch.	Retorno de los cambios o modificaciones realizadas. En caso de no contar con respaldos se puede llegar a necesitar reprocesar cadenas batch.
Consumo excesivo de CPU (HSM,RMM)	<b>3</b>	Retraso en cadenas batch, posible degradación de servicios linea.	El área de infraestructura debiera apoyar en la configuración de las metas de carga de trabajo para que el sistema dinámicamente diera prioridad a procesos críticos. Sin embargo el HSM y RMM son sistemas de uso crítico. De su desempeño depende la velocidad en la que recuperen o migren datos los procesos aplicativos. Una opción para bajar su consumo es configurar la reducción de número de tareas concurrentes.
Daño lógico en volumen DASD o cinta	<b>1-2</b>	Dependiendo la criticidad del dato puede impactar a un área sensible del negocio.	Búsqueda y recuperación de respaldos . Reprocesamiento de cadena batch.
Consumo excesivo	<b>3</b>	Retraso en cadenas batch,	Stop y Reinicio de tareas

de memoria		posible degradación de servicios línea por la paginación. Si el excedente llega al máximo puede provocar la detención de todo el sistema.	HSM o RMM. Cancelación de procesos con uso excesivo de memoria. Puede llegar a ser necesario reiniciar el sistema (IPL). El área de infraestructura puede prevenir esto realizando una configuración correcta de parámetros de sistema.
Saturación de VTOC	<b>1</b>	El disco ya no puede almacenar más data sets aunque tenga espacio suficiente.	Ver si es posible crecer la VTOC, si no entonces se debe vaciar el disco y reformatearlo con los parámetros adecuados para que pueda usar todo el espacio disponible.
Sistema en espera de dispositivo	<b>0-3</b>	Puede ser tan grande o tan pequeño dependiendo el número de dispositivos solicitados y el usuario que lo pida.	Normalmente se cancela el requerimiento por dispositivo cuando el que lo solicita es un usuario. Si el sistema es el que lo pide y si además pide un gran número de dispositivos se puede llegar hasta el IPL.
Error de edición sobre data sets comprimidos	<b>0-1</b>	Dependiendo la criticidad del proceso y el tiempo de atención. Un proceso intenta escribir sobre un data set comprimido, el cual no permite actualizaciones.	Se procede a descomprimir el data set y se modifica la regla en la rutinas ACS en la sección de DATACLAS
Montajes pendientes en dispositivos.	<b>0-2</b>	Puede ser tan grande o tan pequeño dependiendo el número de dispositivos solicitados y el usuario que lo pida. Puede llegar a impactar al negocio pero no al sistema.	Se solicita el mantenimiento del hardware afectado o si el problema es en el software se pueden reiniciar servicios y/o drives de cinta. También puede provocar este estado algún proceso atrapando un dispositivo el cual puede ser cancelado.

INCIDENTES NO PERTENECIENTES AL ÁREA DE ALMACENAMIENTO

Los siguientes, son incidentes de almacenamiento cuya atención quedaba fuera nuestra responsabilidad:

Incidente	Severidad	Descripción del impacto	Descripción procedimiento para solucionarlo
-----------	-----------	-------------------------	---

Contención de data sets	<b>0-1</b>	Un proceso espera a que otro deje de usar un data set. El proceso o procesos se mantienen a la espera provocando retraso en su ejecución.	Esperar a que el proceso que tiene atrapado el data set, finalice su trabajo. O bien cancelar el /los procesos que provocan la contención.
Falla de proceso por configuración de archivos de trabajo en área sort.	<b>0</b>	Impacta sólo en el proceso o cadena batch.	Indicarle al usuario cómo configurar las áreas de trabajo de su proceso.
Falla de espacio por mala definición de JCL	<b>0-1</b>	Proceso cancela ya que ha llegado a su máximo número de volúmenes en disco. Impacta sólo en el usuario o cadena batch.	Indicar al usuario cómo corregir la definición de espacio en el JCL.
Borrado de archivos de acuerdo a las políticas	<b>0</b>	El usuario se queja de que su información ha sido borrada y quiere saber por qué ha ocurrido tal evento.	Se le indica al usuario/aplicativo que el data set fue eliminado de forma automática de acuerdo a las políticas establecidas en la management class.
Recuperación de respaldos	<b>0</b>	El usuario/aplicación piden apoyo para recuperar un data set desde respaldo ya que su proceso cancela por falta de archivo.	El usuario debe ser capaz de recuperar sus respaldos. Sin embargo a veces se hacen excepciones debido al nivel de escalamiento.
Daño físico en dispositivos DASD o cinta	<b>1</b>	Puede haber afectación a alguna cadena batch.	Se escala con el soporte de hardware del proveedor de los dispositivos.
Fallos en procesos (no HSM y no RMM) por uso excesivo de memoria.	<b>3</b>	Retraso en cadenas batch, posible degradación de servicios línea.	El área de infraestructura debiera apoyar en la configuración de las metas de carga de trabajo para que el sistema dinámicamente diera prioridad a procesos críticos. En el momento se deberán cancelar o dormir los procesos que hagan uso excesivo de memoria, de lo contrario se podría llegar al IPL.

## 8. APORTACIONES PROFESIONALES AL PROYECTO.

### 8.1 ADMINISTRACIÓN, MONITOREO Y ATENCIÓN A REQUERIMIENTOS

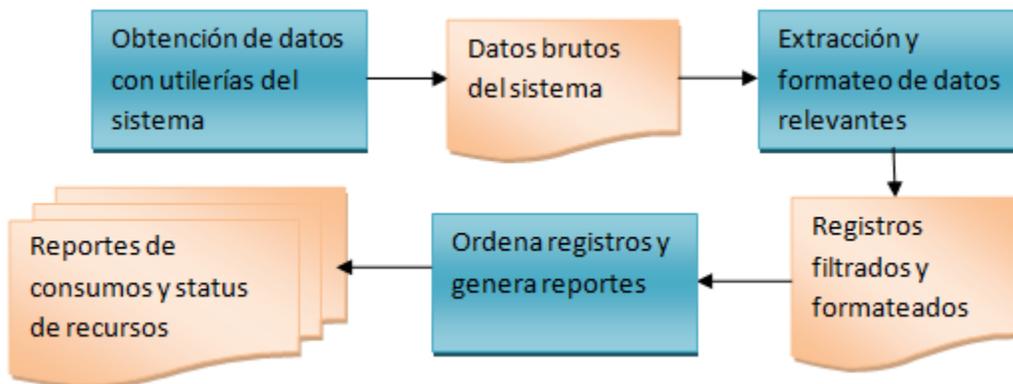
Durante mi labor dentro del proyecto pude percatarme de varias áreas de oportunidad que me posibilitarían proponer e implementar mejoras que ayudarían a que las actividades se ejecutaran de forma eficiente.

Los siguientes son los aspectos que encontré podían ser sujetos de mejora:

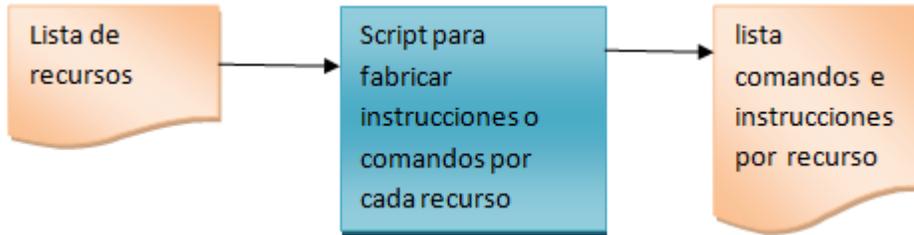
- Reportes con datos incompletos o imprecisos.
- Ejecución de tareas rutinarias, pero que requerían trabajar con gran cantidad de elementos, obligaban a dedicar mucho tiempo a codificar gran cantidad de sentencias o comandos de forma reiterada.
- No existía monitor de recursos en tiempo real.
- La mayor parte del monitoreo se llevaba a cabo de forma manual (obtención de de datos con base a reportes y ejecutando comandos).
- Al ser manual, el monitoreo estaba sujeto a fallos u omisiones.

Para atender esta problemática implementé las siguientes soluciones:

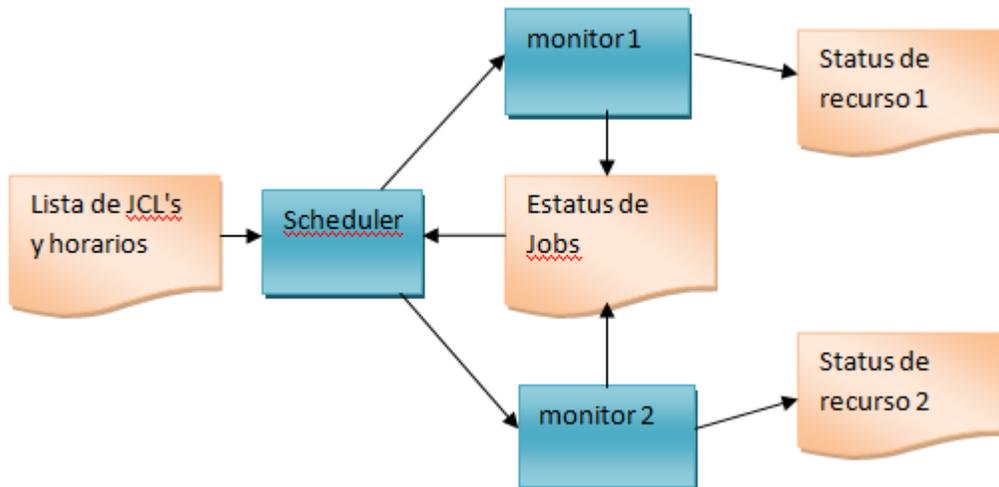
1. Reportes con resúmenes y status de todos los recursos que faltasen de monitorizar. Para llevar a cabo esta tarea obtuve datos brutos que se extraen a partir de las utilerías del sistema y posteriormente con un script se manipulaban de manera que se generasen reportes con datos inteligibles y resumidos.



2. Para poder codificar instrucciones de manera eficiente durante actividades que involucraran muchos elementos, implementé scripts que permitieron generar instrucciones de código para miles de elementos en un tiempo mucho más corto que cuando se codificaban a mano. Esto permitió, por ejemplo, reducir considerablemente el tiempo para el formateo de dispositivos DASD, el obtener atributos para miles data sets fabricando y ejecutando comandos individuales de forma inmediata. El diseño se basaba en simplemente agrupar los elementos en una lista y con el script se iba tomando cada elemento y se fabricaban las instrucciones respectivas y estas se almacenaban en un data set para su posterior ejecución.



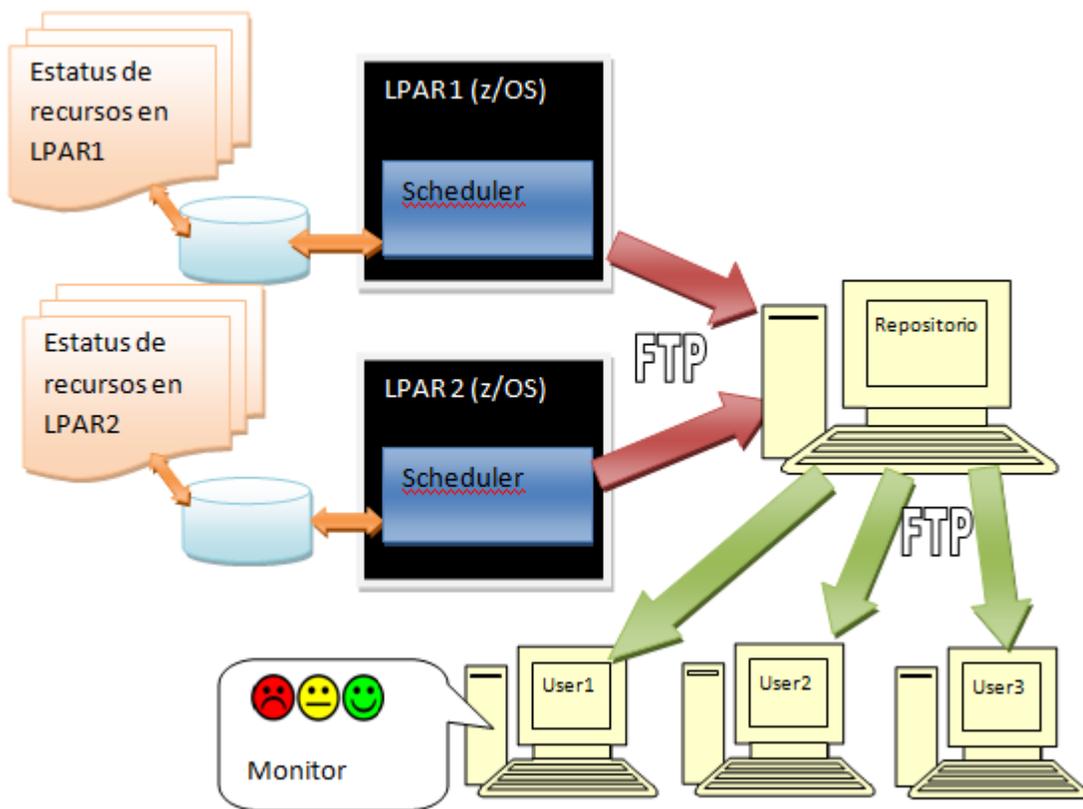
3. Implementé un sistema *scheluder* para ejecutar procesos de forma automática de manera que se pudieran ejecutar de forma continua procesos de monitoreo y reportes en diversos horarios sin necesidad de que interviniera una persona en su ejecución. Esto representó una gran ventaja, ya que no fue necesario realizar trámites para calendarizar procesos en el *scheluder* oficial del cliente (mismo que estaba diseñado para planificar principalmente procesos de las aplicaciones que tienen que ver con el negocio) y a su vez se podían planificar más procesos, cuya ejecución manual y continua sería una labor que requeriría la dedicación de un especialista de tiempo completo.



El esquema muestra la forma de funcionar el scheduler. Cuando ya se tenían listos los JCL de los jobs que iban a ejecutarse periódicamente, entonces se anotaban en la lista de JCL's para que el scheduler determinara la ubicación de los mismos y el momento en que debían ejecutarse. Así mismo se anotaba el estatus inicial de los jobs para que el scheduler pudiera determinar si se podía ejecutar un proceso o no. Si un proceso se mantenía en estatus de ejecución y era momento de volverlo a ejecutar, el scheduler conociendo el estatus actual del job podía evitar que se volviera a ejecutar el job para no encolar procesos; lo mismo podía pasar si el scheduler detectaba que un proceso canceló o terminó con error y de esta manera se evitaba que se volvieran a

ejecutar procesos que con un problema hasta que no fueran atendidos. Al proceso se le efectuaba el mantenimiento respectivo y cuando ya estaba listo para volverse a ejecutar simplemente se entraba al data set con el status de los jobs y se actualizaba el correspondiente al proceso para que a su momento el scheduler lo volviera a considerar.

4. Elaboré un monitor desde el sistema operativo windows que obtenía datos de las diversas particiones lógicas de z/OS a través del protocolo FTP y los presentaba a manera de semáforo en cada una de las PC de los especialistas, de forma que con sólo un vistazo a este monitor se podía conocer el status general de varios sistemas a la vez. Lo desarrollé en lenguaje C con uso de threads o hilos de manera que el monitor no dejaba de mostrar datos mientras recolectaba la información del repositorio. Este sistema se complementaba con el scheduler en z/OS de manera que los datos presentados se mantenían actualizados casi en tiempo real.



El esquema observado indica la forma de trabajo del sistema monitor. Por un lado tenemos a las particiones lógicas (LPAR) corriendo su propio scheduler que va ejecutando periódicamente procesos que obtienen datos sobre el estado del sistema. Estos

estados son recolectados y enviados vía FTP por un proceso que igualmente está planificado en el scheduler. Los datos que son enviados llegan a un repositorio el cual es visible únicamente desde la red interna del cliente. El repositorio entonces es accesible por cualquier computadora que se encuentre en la red del cliente facilitando a cualquier usuario conocer el estado de los recursos de almacenamiento. Sin embargo para poder usar esta facilidad se debe tener permiso de acceso a la computadora repositorio.

El usuario que tuviera los permisos de acceso ahora tendría que tener instalado una interfaz que permitiera leer los datos del repositorio e interpretarlos de manera que este generase un aviso en cuanto un recurso presentase un problema o alerta. La interfaz fue igualmente desarrollada por mí en lenguaje C y los avisos que generaba eran de tipo audiovisual, de manera que el administrador no requiriera estar al pendiente del monitor hasta que oyera el aviso.

## 8.2 INCIDENTES

Respecto a los incidentes, mis conocimientos técnicos me permitieron destacarme al ayudar al cliente a dar solución a las siguientes situaciones:

- ✓ Recuperación de datos por catálogos dañados con afectación a producción.  
Durante las actividades administrativas se aplicó un cambio, el cual consistió en separar un catálogo grande en dos de tamaño más pequeño con el fin de atender un alerta de saturación. El proceso de mover entradas entre catálogos implica mover también apuntadores y referencias que se deben actualizar una vez que cada entrada ha sido movida. Para realizar esta actividad se usó una herramienta que no era de IBM. El sistema operativo necesitaba una actualización que no estaba contemplada en este cambio. Durante el proceso, el catálogo original se dañó y no se pudieron terminar de mover todas las entradas. De las 40,000 entradas que podían visualizarse en este catálogo, ahora sólo se podían ver 4000 por lo que miles de procesos fallarían por no poder localizar sus data sets. Se recuperó un respaldo del catálogo, almacenándolo en otro disco y renombrándolo. Las entradas ya se veían y se reanudó la producción, pero muchos procesos fallaban por error cuando debían borrar data sets de trabajo. Eran muchos los jobs con esa situación, la cual tuvo su causa raíz en el renombrado del catálogo. Se observó que cuando se reanudaban los procesos cancelados estos ya no fallaban, pero generaban un retraso que al acumularse en conjunto impactaban en gran medida a la producción. Mi

solución fue ejecutar un movimiento de data sets entre volúmenes DASD con la utilería ADRDSSU la cual actualizaba la situación del catálogo renombrado en los atributos de los data sets evitando que mandaran error en los procesos. En promedio, nos notificaban alrededor de 30 procesos cancelados por hora y el cliente ya no podía soportar esta situación, se le dio un límite al proveedor para acabar con esta situación antes de que se diera por terminado el contrato . El proceso de reubicación de datos con ADRDSSU iba lento y descubrí que era por el movimiento de data sets muy grandes. Obtuve un estadístico de todos los data sets a mover y los data sets más grandes sólo representaban el 5% de todas las entradas, por lo que configuré el ADRDSSU para que sólo moviera los data sets más pequeños. Cuando finalicé de mover los data sets, lo cual no me demoró más de 30 minutos, la tasa de procesos cancelados disminuyó de 30 a 1 por hora. De esta situación mi posición como especialista senior fue consolidada en el área.

- ✓ Investigación y deslinde de responsabilidad por datos eliminados o dañados que afectaron la ejecución de procesos críticos. En una ocasión hubo un proceso sensible que usaba *sort* en el cuál un data set que se usaba como entrada y como salida del mismo programa aparecía sin más de la mitad de sus registros a la salida. Proceso terminaba sin mandar ningún error. Es una práctica ortodoxa utilizar un data set con nombre distinto para la salida que el de la entrada pero es muy común ver el mismo data set de entrada que de salida en los programas SORT. Pero el cliente no se explicaba por qué fallaba el proceso siendo que otros procesos con la misma situación no presentaban este fallo. Así que dejaba toda la responsabilidad de la pérdida de datos al área de Almacenamiento Central la cuál sería acreedora a multas y a revisión del contrato si no resolviera la situación o determinara la causa de la misma. Una manera de resolver el caso al parecer estaba en no usar el mismo data set de entrada que el de salida, pero el cliente no estaba dispuesto a modificar el JCL a no ser que le explicáramos el por qué ocurría esta situación. Comparé los data sets usados de la misma forma en otros procesos con respecto al proceso fallido y observé que el data set usado para este en particular era multivolumen y los de los otros procesos no lo eran. Revisé el manual del programa SORT y este explicaba que en el caso de trabajar con archivos multivolumen que fueran a la vez entrada y salida sólo se podía procesar el primer volumen. Con esto como evidencia, la perdida de datos se explicaba porque el SORT sólo procesó el primer volumen ignorando los demás y al escribir la salida en el mismo data set aparentemente desaparecieron el resto de registros contenidos en los demás volúmenes. Demostrada la causa del fallo, el cliente modificó su JCL y reanudo su proceso terminando

satisfactoriamente y sin pérdida de datos. En ese momento me convertí en especialista senior.

## 9. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS DE TRABAJO.

### 9.1 ATENCIÓN A REQUERIMIENTOS

Durante la primera etapa de la atención de requerimientos estos eran los resultados arrojados en promedio mes con mes:

Número de solicitudes en el mes	100
Atención de requerimientos en tiempo.	70
Atención de requerimientos fuera de tiempo.	30
Requerimientos cerrados y aprobados por el solicitante.	80
Requerimientos atendidos pero regresados por el solicitante.	20
Nivel de servicio	%70

Debido a que los requerimientos regresados por el solicitante debían ser revisados y atendidos nuevamente para poder cerrarlos, esto provocaba un retraso en su atención, lo que a su vez impactaba en el conteo de requerimientos atendidos en tiempo. El nivel de servicio resultante en el mes no cumplía con la meta acordada. Como equipo debíamos reforzar los procedimientos y el seguimiento a las solicitudes.

Cuando añadimos al servicio el uso de procedimientos y rutinas, pudimos apreciar un mayor grado de control y automatización de las tareas empleadas para atender las diversas solicitudes. Como resultado, durante los meses posteriores se fue avanzando hasta llegar reportar los siguientes resultados en los últimos meses:

Número de solicitudes en el mes	100
Atención de requerimientos en tiempo.	98
Atención de requerimientos fuera de tiempo.	2
Requerimientos cerrados y aprobados por el solicitante.	95

Requerimientos atendidos pero regresados por el solicitante.	5
Nivel de servicio	%98

Inclusive, se atendían fuera de tiempo un menor número de requerimientos que los que eran rechazados por el solicitante. Esto fue posible ya que cuando recibíamos la notificación de "regreso", inmediatamente contactábamos al solicitante para aclarar las causas por las que este regresaba el requerimiento y una vez que teníamos despejadas aquellas dudas podíamos atender y cerrar el requerimiento el mismo día de haber sido regresado.

Pero todavía nos anticipamos más. Usábamos el mismo procedimiento para no dar por sentado ningún parámetro que no tuviéramos claro. De esta forma si teníamos dudas en las especificaciones de un requerimiento, contactábamos con el solicitante para corroborar datos y recabar la información faltante de manera que fuimos más eficaces en la atención de requerimientos.

- En resumen:
  - ✓ Se disminuyó el tiempo invertido para atender requerimientos.
  - ✓ Se atendió un mayor número de requerimientos por día.
  - ✓ Se tuvo control para evitar fallas humanas gracias al empleo de plantillas y scripts que codificaban miles de instrucciones con un margen de error muy bajo.

## 9.2 ADMINISTRACIÓN DE LOS RECURSOS DE ALMACENAMIENTO

Durante mi incursión en la administración de recursos de almacenamiento teníamos en nuestra agenda de trabajo el analizar y dar solución a los siguientes aspectos:

1. Frecuentes cancelaciones de procesos por data sets que llegan a su máximo número de volúmenes.
2. Saturación Continua de storage groups durante fin de semanas, quincenas y fin de mes.
3. Fallas en el HSM al migrar archivos de nivel 0 a nivel 1.
4. HSM sin poder cumplir su ciclo de migración en la ventana establecida.
5. Mensajes de alerta de saturación en archivos de control.
6. Alto nivel de uso de catálogos.

La forma en la que aborde y resolví respectivamente cada problema la describo a continuación :

#### CANCELACIONES POR MÁXIMO NÚMERO DE VOLÚMENES

- El número de cancelaciones registradas eran de aproximadamente 40 al mes.
- Primero debía determinar si el origen del problema de espacio en el data set se podía solucionar ajustando los parámetros de espacio en el JCL.
- Segundo, debía determinar la naturaleza de los data sets usados. Esto es, verificar si eran usados como archivos de trabajo, archivos actualizables o como reportes finales. De ser usados como reportes finales los podría considerar candidatos para compresión. La compresión de data sets tiene el inconveniente de que no permite la edición o actualización del contenido del data set, por lo que tenía que asegurar que el data set no fuese utilizado para edición.
- Después determinaba las capacidades de los volúmenes usados por el data set generado por el proceso. Si las dimensiones resultaban muy pequeñas se podría hacer una sustitución de volúmenes DASD reemplazando los de menor capacidad por los de mayor capacidad. Un data set podría crecer en sus dimensiones usando volúmenes más grandes y también usando el formato extendido o el formato large.
- Para aplicar la compresión me ubicaba en las rutinas ACS y modificaba el construct *data class* asociado a la máscara del data set de manera que cada vez que se generaran los data sets que cumplieran con esa máscara el sistema los compactara a la par.
- Haciendo uso de la compresión, volúmenes más grandes y cambios de formato de data set en cualquier caso, se observó una reducción considerable de procesos cancelados, pasando de ser 40 hasta ser solo 8 procesos aproximadamente al mes.

#### SATURACIÓN CONTINUA DE STORAGE GROUPS

Asignar espacio extra a un storage group era la última opción que quedaba después de analizar y aplicar diversas opciones para recuperar espacio. Estas opciones abarcaban las siguientes actividades que debía efectuar:

- Eliminación de data sets descatalogados.
- Compactación de data sets.

- Afinación del HSM de manera que todos los data sets que fueran candidatos para migrarse fueran migrados durante la ventana establecida para tal propósito.

Finalmente obtuve graficas de consumo en los storage groups para medir los resultados de las actividades realizadas.

Estos fueron los resultados obtenidos para estas actividades:

Número de storage groups detectados con falta de espacio.	6
Número de storage groups liberados únicamente mediante las actividades de recuperación de espacio.	2
Número de storage groups solventados tanto con actividades de recuperación como con asignación de espacio.	4
Porcentaje de recuperación de espacio con las actividades de recuperación.	20%

Debido a la alta tasa de rotación de información en algunos storage groups y al crecimiento natural de las aplicaciones, esta actividad la debíamos realizar periódicamente.

#### FALLAS EN MIGRACIÓN DE NIVEL 0 A NIVEL 1

La revisión periódica de los logs de migración mostraba una aparición frecuente de mensajes de error al migrar data sets del nivel 0 a nivel 1.

La explicación del código de error decía que los data sets no se podían migrar porque no cabían en ninguno de los volúmenes del nivel 1. Los data sets migrados a nivel 1 no podían ser multivolumen, por lo que si un data set en nivel 0 ocupaba más 65535 tracks entonces sería muy probable que hubiera falla al intentar migrarse al nivel 1. Por otro lado, no existe esta restricción en el nivel de migración 2, permitiendo usar hasta 255 cintas para almacenar un solo data set.

Vale la pena aclarar que el nivel de migración 1 está pensado para almacenar data sets de uso frecuente, de tamaño pequeño y de expiración a corto plazo. El nivel de migración 2 está pensado para almacenar data sets históricos y de expiración a

largo plazo. La management class es el objeto donde se indica si un grupo de data sets van a ser migrados a nivel1 o a nivel 2 en primera instancia.

La solución temporal que implementé fue generar un script en REXX que revisara los logs de migración del HSM y que tomara los nombres de los data sets que no habían podido ser migrados al nivel 1. Después, con esos datos, el mismo script construía un JCL para migrar esos data sets directamente a nivel 2. El JCL contenía los comandos de HSM para la migración de cada data set.

La solución permanente consistió en habilitar una *exit* que los desarrolladores del HSM nos facilitaron para que los data sets que excedieran un tamaño determinado fueran migrados a nivel 2 sin pasar por el nivel 1.

Una exit es una subrutina invocada por un paquete de software para un evento predefinido en la ejecución del mismo. El cliente de este software puede sustituir las rutinas por defecto por aquellas propias de manera que se tenga una funcionalidad personalizada. Generalmente estas rutinas vienen codificadas en lenguaje ensamblador, y el usuario puede ser capaz adaptar los valores de las variables de acuerdo a su conveniencia antes de compilarlas.

#### HSM SIN PODER CUMPLIR SU CICLO DE MIGRACIÓN DURANTE LA VENTANA ESTABLECIDA

Durante el monitoreo de funciones fuimos hallando que el HSM no culminaba su ciclo de migración con más frecuencia. Mientras el consumo de espacio en los storage group iba aumentando conforme crecían las aplicaciones, el HSM seguía migrando la misma cantidad de espacio en promedio.

A continuación se muestra una tabla con el rendimiento del HSM respecto a la ventana de migración en una LPAR específica durante el periodo anterior al detectado con alertas:

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Inicio de ventana	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs
Fin de ventana real	19:05hrs	21:16hrs	20:44hrs	21:31hrs	21:20hrs	21:12hrs	20:33hrs
Fin de ventana programado	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs
Cantidad de espacio migrado	16TB	21TB	19TB	20TB	21TB	20TB	18TB
Status de la tarea de migración	OK	OK	OK	OK	OK	OK	OK

Cuando se detecta el problema:

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Inicio de ventana	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs
Fin de ventana real	19:44hrs	22:20hrs	22:33hrs	23:00hrs	23:00hrs	23:00hrs	21:20hrs
Fin de ventana programado	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs
Cantidad de espacio migrado	19TB	21TB	19TB	23TB	23TB	23TB	22TB
Status de la tarea de migración	OK	OK	OK	NOK	NOK	NOK	OK

Como solución decidí aumentar el número de tareas paralelas de migración. Cada tarea realiza la migración de los data sets de un volumen específico y a su vez dedica un drive de cinta para tal efecto. Aumentar el número de tareas trae consigo usar mas drives de cinta. La decisión de aumentar tareas va acompañada de la idea de no afectar a la producción en la disponibilidad de drives de cinta para los procesos y también de no hacer mas uso de recursos de CPU y memoria que los permitidos para esta actividad.

Revisando la infraestructura con la que se contaba decidimos pasar de 10 tareas paralelas a 15 tareas paralelas. El comando con el que se le indica al HSM a cuántas tareas aumentar la migración es el siguiente:

```
HSEND SETSYS MAXMIGRATIONTASKS(15)
```

Con este cambio aplicado el rendimiento posterior del HSM fue el siguiente:

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Inicio de ventana	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs	13:00hrs
Fin de ventana real	18:34hrs	20:24hrs	20:11hrs	22:07hrs	22:02hrs	19:10hrs	18:40hrs
Fin de ventana programado	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs	23:00hrs
Cantidad de espacio migrado	17TB	21TB	21TB	29TB	30TB	23TB	22TB
Status de la tarea de migración	OK	OK	OK	OK	OK	OK	OK

De esta forma se estabilizó la recuperación de espacio en los storage groups.

#### ALERTAS DE SATURACIÓN DE ARCHIVOS DE CONTROL Y DE CATÁLOGOS.

Cuando se presentaba un alerta de catálogo o de archivo de control entonces se planificaba el mantenimiento en la herramienta de control de cambios. Estos recursos requieren que no exista actividad sobre ellos antes de aplicar su mantenimiento porque si se llegasen a corromper o a destruir traerían consigo la pérdida de muchos data sets. Durante mi experiencia en el proyecto sólo tuve un problema al aplicar estos mantenimientos con impacto en la producción. Aproximadamente, en total realicé 20 mantenimientos a catálogos y archivos de control.

#### RESULTADOS SOBRE EL USO DE MONITORES

- La utilización de procesos monitores dio como resultado:
  - ✓ Crecimiento de recursos a monitorizar.
 Anteriormente solo se monitorizaban los siguientes recursos de forma manual:

Utilización de storage groups cada 2 horas
Utilización de data sets de control una vez al día
Utilización de catálogos una vez a la semana
Disponibilidad de cintas una vez al día.
Estatus de funciones de HSM un par de veces al día

Después de la implementación y automatización de monitores, el catálogo se servicio aumentó:

Utilización y tendencia de storage groups cada 5 minutos
Utilización de data sets de control cada hora
Utilización de catálogos cada día
Número de alias usados por catálogo
Disponibilidad de espacio en disco para catálogos
Utilización de espacio y extents en archivos VSAM y Particionados cada día
Utilización de directorios en data sets particionados cada día
Disponibilidad de cintas cada hora
Utilización de VTOCs de volúmenes DASD
Estatus de funciones HSM cada media hora

- ✓ Mayor control de los recursos de almacenamiento.  
Este punto tiene relación directa con el uso monitores. A continuación muestro una tabla con los elementos sobre los que se tenía control antes y después de la implantación de los mismos:

Antes	Después
Storage Groups	Storage groups
Disponibilidad de cintas	Disponibilidad de cintas
Utilización de catálogos	Utilización de catálogos
Utilización de data sets de control	Utilización de data sets de control
Estatus de funciones HSM	Número de alias por catálogo
	Disponibilidad de espacio en disco para catálogos
	Utilización de espacio en VSAMs y Data sets particionados
	Utilización de directorios de PDS
	Utilización de VTOCs
	Estatus de funciones HSM

- ✓ Disminución de la probabilidad de incidentes gracias a los mantenimientos preventivos.  
En el apartado de incidentes de este mismo tópico muestro los resultados que los mantenimientos preventivos tuvieron sobre el número de incidentes adjudicados al área.
- ✓ Disminución de tiempo y recursos humanos para llevar a cabo tareas de monitoreo.  
Anteriormente cada recurso dedicaba al menos 2 horas de su horario laboral en realizar el monitoreo y reporte del estatus de los recursos de almacenamiento en todas las LPAR. Con la implantación de monitores únicamente fue necesario disponer de menos de un minuto en visualizar el estatus de todos los recursos de almacenamiento.

- ✓ Incremento de disponibilidad tiempo para llevar a cabo análisis de problemas y planeación de la capacidad.

### 9.3 ATENCIÓN A INCIDENTES

La siguiente tabla muestra las estadísticas de incidentes atendidos de forma mensual antes y después de mejorar la administración y el monitoreo de recursos de almacenamiento:

<b>Tópico</b>	<b>Antes</b>	<b>Después</b>
Número de incidentes promedio	120	81
Número de incidentes sin responsabilidad para el área de Almacenamiento Central	86	77
Número de incidentes con responsabilidad para el área de Almacenamiento Central	34	4
Número de incidentes atendidos dentro de los tiempos acordados en el nivel de servicio.	110	80
Porcentaje de incidentes imputados al área de Almacenamiento Central	28%	4.5%
Cumplimiento de los acuerdos de nivel de servicio	<b>NO</b>	<b>SI</b>

Las actividades preventivas y administrativas ayudaron a disminuir considerablemente el número de incidentes adjudicados al área facilitando el cumplir con los acuerdos de niveles de servicio.

## VIII. PARTICIPACIÓN EN PROYECTO: MIGRACIÓN STORAGETEK-VTCS A IBM-TS7700.

### 1. ANTECEDENTES

El cliente determinó cambiar de proveedor de solución de cintas por cuestiones prácticas y de negocios que finalmente tienen que ver con aspectos económicos. En el ámbito práctico quiso dejar atrás la tecnología de cartucho de cinta para optar de lleno a la tecnología de virtualización de cintas. Storagetek ya contaba con un sistema de virtualización de cintas, sin embargo el cliente no estaba contento con el soporte ni la calidad de los sistemas desarrollados por este proveedor después de haber sido absorbida por Sun y Oracle años atrás. Era el momento de revisar otras opciones. Así que el cliente optó por IBM como el proveedor de los virtualizadores de cintas requeridos.

IBM se encargó de proveer de los servidores que fungirían como virtualizadores de cintas y una empresa dedicada a servicios de TI fue designada para llevar a cabo la migración de los datos entre ambas soluciones.

El problema surgió debido a que después de 4 meses los avances en la migración de datos no habían sido los esperados, por lo que la finalización del proyecto se estimaba que no se cumpliera en los tiempos acordados. Esto, desde luego provocaría pérdidas económicas para el cliente, quien esperaba que la migración de datos se llevara a cabo en los plazos antes que se renovara el contrato por la renta de los equipos de Storagetek. La administración de cintas físicas también representaría un costo adicional, ya que varias de estas debían viajar por avión y camión a un centro de datos alterno que el cliente implementó en caso de desastres.

El inconveniente mayor para el prestador de servicios fue que, de no cumplirse los tiempos acordados para la migración de datos, se habría hecho acreedor a una penalización económica, por lo que era primordial contar con el personal adecuado y suficiente para llegar a cumplir las metas disminuyendo al mínimo los costos por contratación de personal.

## 2. REQUERIMIENTOS DEL PROYECTO

- ✓ Migración de datos históricos de equipos de Storagetek a equipos de IBM
- ✓ Finalización del proyecto en los tiempos acordados
- ✓ Garantizar la integridad de la información después de haber sido migrada.
- ✓ Se debía realizar la migración de manera paralela a la operación normal del negocio del cliente sin que hubiese afectación.
- ✓ El único momento de intermitencia tolerado sería aquel que durante la migración de un data set, este quedase no disponible hasta que hubiese finalizado su migración.
- ✓ Debido a que las dimensiones de los volúmenes tanto reales como virtuales de Storagetek variaban con respecto a los de IBM y también a causa de la fragmentación de los volúmenes por a la expiración de datos, se requería que los data sets a migrar fueran almacenados en el menor número de volúmenes posibles.

## 3. METODOLOGÍA EMPLEADA

Se siguió empleando la metodología ITIL como base para llevar a cabo las actividades de migración tecnológica.

## 4. ALCANCE DEL PROYECTO

Se trabajaría bajo el sistema operativo z/OS. La migración incluía los datos almacenados en cintas físicas y virtuales de Storagetek.

Las actividades dentro del alcance incluían:

- Soporte a nivel de software siempre que estuviese relacionado con la migración y fueran problemas de configuración. En caso de que el problema fuera de depuración de software el soporte lo brindaría el proveedor del producto para migrar datos.
- Seguimiento y control de mantenimientos y soporte de hardware.
- Elaboración de informes de avance y recolección de evidencias de procesos ejecutados.

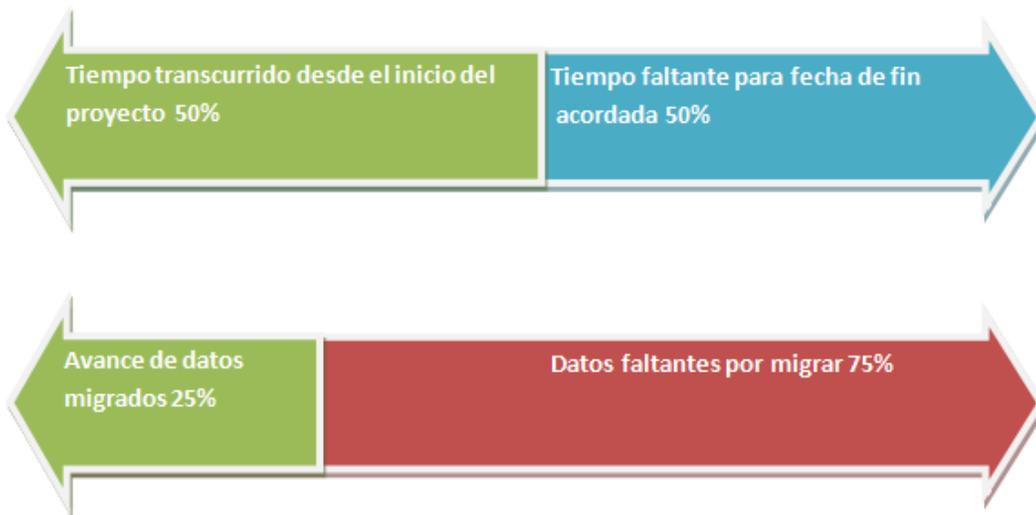
Las actividades que no estaban dentro del alcance:

- Soporte y mantenimiento de hardware (cintas, virtualizador, biblioteca de cintas)

## 5. DESARROLLO DEL PROYECTO

### 5.1 ESTADO INICIAL DEL PROYECTO

Cuando tuve participación en este proyecto el estatus del mismo era el siguiente:



La siguiente tabla muestra los datos aproximados obtenidos al inicio del proyecto:

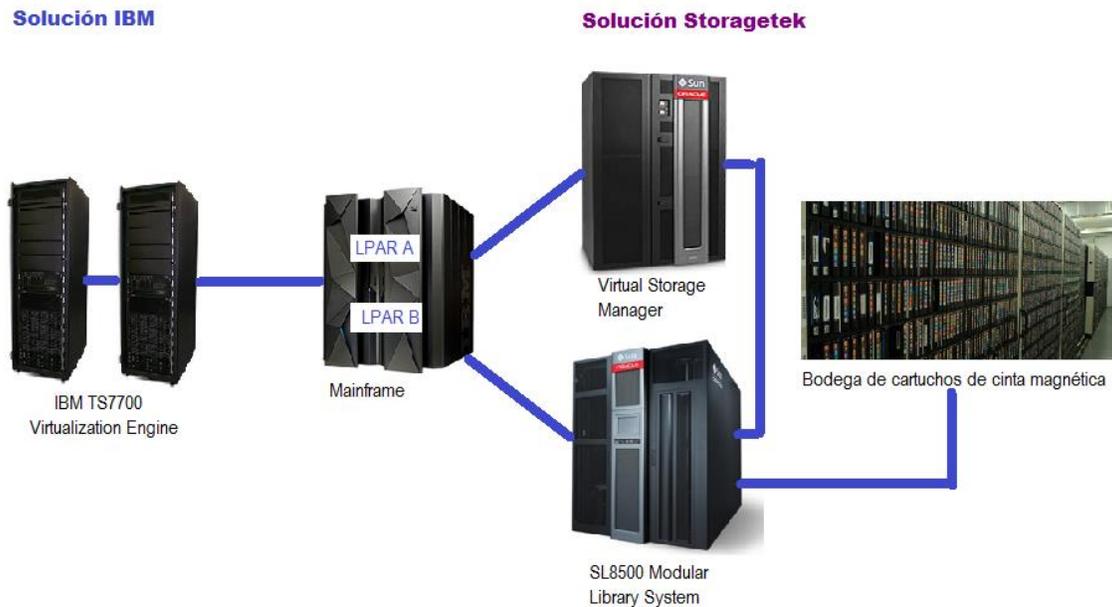
Cantidad de bytes a migrar.	<b>1 Petabyte</b>
Número de volúmenes de cintas virtuales a migrar	<b>2 millones</b>
Número de volúmenes de cintas reales a migrar	<b>100 mil</b>
Numero de Particiones Lógicas (LPAR) de z/OS involucradas	<b>5</b>
Drives de cintas virtuales por LPAR	<b>10</b>
Drives de cintas reales por LPAR	<b>2</b>
Drives para recuperar cintas virtuales migradas a cinta real	<b>10</b>
Capacidad de la biblioteca de cintas.	<b>10000 cintas</b>

A partir de los datos obtenidos se determinó que faltaban por migrar 750 TB.

Para poder llevar a efecto la migración se debían ejecutar miles de jobs continuamente y gestionar de forma adecuada el ingreso y salida de cintas a la biblioteca ya que esta no podía albergar todas las cintas físicas a la vez. Hay que tomar en cuenta que la operación del banco no se detenía y necesitaba usar cintas físicas también.

La estrategia era simple, ejecutar el mayor número de jobs posibles durante todo el día para garantizar el máximo avance. Sin embargo esto no sería una tarea sencilla.

## 5.2 DESCRIPCIÓN DE LA INFRAESTRUCTURA

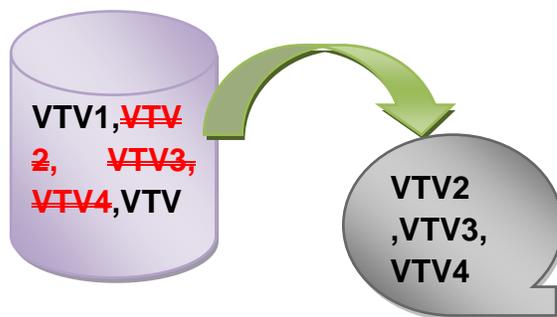


Los LPAR tenían drives para conectarse tanto con el virtualizador TS7700, la biblioteca de cintas y el manejador de cintas virtuales. Los LPARs compartían la infraestructura de almacenamiento pero esto no quiere decir que desde una sola partición pudiese llevarse a cabo toda la migración, esto se debe a que los enlaces hacia los data sets son dependientes del entorno donde se generan. Si un data set se moviera de dispositivo sin haberlo hecho desde el entorno donde se generó entonces el enlace no se actualizaría y el sistema perdería la ubicación del data set.

Cuando se crea un data set en cinta o en disco se debe definir una entrada en el catálogo correspondiente para que su localización sea inmediata. Además, cuando

el data set es almacenado en cinta, el manejador DFSMSrmm crea un registro donde va a llevar a cabo el control de su expiración y sus montajes. Todo esto se lleva a cabo desde la partición donde el archivo es creado, por lo tanto no puede moverse desde otra partición porque deben actualizarse estos elementos y cada catálogo e instancia de DFSMSrmm pertenece a una sola imagen de z/OS.

El manejador cintas virtuales VSM tenía una capacidad de almacenamiento limitada, por lo que cuando se empezaba a saturar, comenzaba a migrar volúmenes de cintas virtuales o VTV (virtual tape volume) hacia volúmenes de cintas físicas llamados MVC (migration tape volume) los cuales pueden almacenar cientos de VTV's.



La biblioteca entonces podía manejar dos tipos de volúmenes físicos: los MVC y los cartuchos de uso *nativo*. Cabe aclarar que cuando se habla de uso nativo se refiere a emplear un dispositivo de la manera en que fue concebida originalmente. Físicamente los dos tipos de volumen son lo mismo pero de forma lógica, los datos que almacena el cartucho nativo los agrupa en data sets y los datos que almacena un MVC los agrupa en VTV's.

Por su parte el TS7700 de IBM sólo maneja volúmenes virtuales y si se llegase a saturar no habría más espacio para almacenar más datos. Sin embargo este sistema cuenta con algoritmos de compresión de datos y la capacidad adquirida por el cliente fue suficiente para poder almacenar todos los datos provenientes de la solución de Storagetek.

### 5.3 DESCRIPCIÓN DEL PROBLEMA

El problema más importante estaba en que al ritmo en que se avanzaba no sería posible terminar a tiempo para la fecha de entrega; se necesitaba migrar más rápido.



Debido al límite de drives que podían usarse en cada LPAR sólo se podían correr 10 procesos para migrar VTVs a la vez y 2 procesos para migrar cintas de uso nativo. Por lo que en general sólo podían correrse 12 procesos de manera paralela por LPAR. Tomando en cuenta que se debían migrar 5 LPAR's entonces se podían ejecutar hasta 60 procesos a la vez.

Estar al pendiente de la finalización de cada proceso para poder ejecutar otro proceso parecía una tarea imposible y hubiera requerido de un especialista trabajando día y noche para llevar a cabo la gestión de los jobs.

Una solución planteada por los especialistas antes de mi arribo al proyecto era la siguiente:

#### ENCOLAMIENTO DE JOBS

Tomando en cuenta que el despachador de jobs no permite la ejecución de dos jobs con el mismo nombre, se podían nombrar todos los jobs de manera que hubiesen 10 nombres distintos y los que queden duplicados se podían encolar

para que el despachador de jobs los fuera ejecutando conforme fueran finalizando cada uno de ellos. El esquema de esta estrategia:



La ventaja de esta estrategia es que no requería que el especialista estuviera de lleno vigilando el progreso de los jobs para mandar a ejecutar nuevos jobs, bastaba con mandar un grupo de jobs suficientes a ejecución para que se quedasen ejecutando durante su ausencia (en la noche por ejemplo).

Son varias las desventajas que se podían encontrar en esta estrategia:

- El espacio destinado al encolamiento de jobs era limitado, por lo que saturarlo con una inmensa lista de jobs en espera podía provocar degradación en el sistema. El cliente no estaba dispuesto a sacrificar la salud del sistema por esta práctica, por lo que dio la indicación de que no podía haber más de 300 procesos encolados.
- Debido a que esta actividad convivía con procesos de la producción, en caso de que llegase a haber una contención de procesos por requerir los mismos volúmenes que se estuviesen migrando, el operador tendría autorización para cancelar todos los procesos activos que considerara estuviesen realizando la afectación. Esto provocaba que cuando el especialista realizara una validación de los procesos que se ejecutaron satisfactoriamente de los que no, resultara una tarea complicada. No se sabía desde donde reanudar.
- Los procesos se agrupan en colas pero no todas las colas avanzaban a la misma velocidad, inclusive podía haber colas tan lentas que podían llegar a pasar horas y no avanzaban:

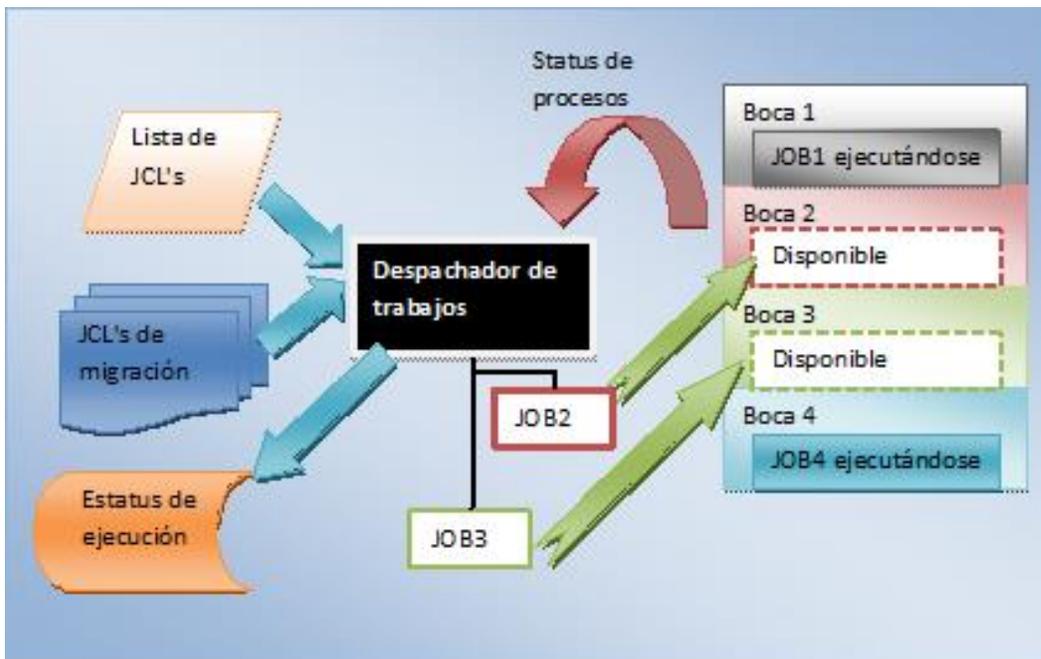


## 6. APORTACIONES AL PROYECTO

### 6.1 DESPACHADOR DE JOBS

Sabiendo que el área de oportunidad radicaba en la estrategia para ejecutar los jobs, entonces planteé el desarrollo de un despachador que permitiera aprovechar las colas disponibles de ejecución.

Partiendo del diseño del scheduler desarrollado en el proyecto de outsourcing de almacenamiento, implementé el despachador de jobs siguiendo este esquema:



El despachador funcionaba localizando todos los JCL's que habían sido generados previamente por un producto de IBM para migrar datos y almacenaba una lista de los JCL en la memoria para llevar el control de su ejecución. Posteriormente el despachador validaba que hubiera disponibilidad de nombres de job para ejecutar; esto lo realizaba cotejando la lista de nombres de job posibles con aquellos que estaban en ejecución. Si no encontraba un job en ejecución con el nombre de uno de la lista de nombres de job posibles entonces el despachador editaba el nombre de job en el código de JCL y lo envía a ejecución.

Este esquema presentaba varias ventajas con respecto a la estrategia de encolamiento de jobs.

- Mantenía la ejecución constante de trabajos sin intervención manual.
- No utiliza encolamiento, nombraba dinámicamente cada job conforme se fueran liberando las bocas, y sólo hasta ese momento los mandaba a ejecución.
- Garantizaba que siempre se mantuvieran en ejecución el número máximo de procesos en paralelo.
- En caso de que se cancelara el despachador, este guardaba en disco el último job mandado a ejecutar y este valor lo tomaba como referencia cuando reanudara su ejecución.
- Se podía mandar a ejecutar toda la lista de jobs generados sin reparar en el uso de espacio de la cola de jobs en espera, ya que no se enviaba el código de JCL al despachador de jobs del sistema sino únicamente se mandaba la lista al despachador propio y se mandaba únicamente el código de los jobs que se ejecutarían al momento.

## 6.2 REPORTES DE STATUS DE MIGRACIÓN

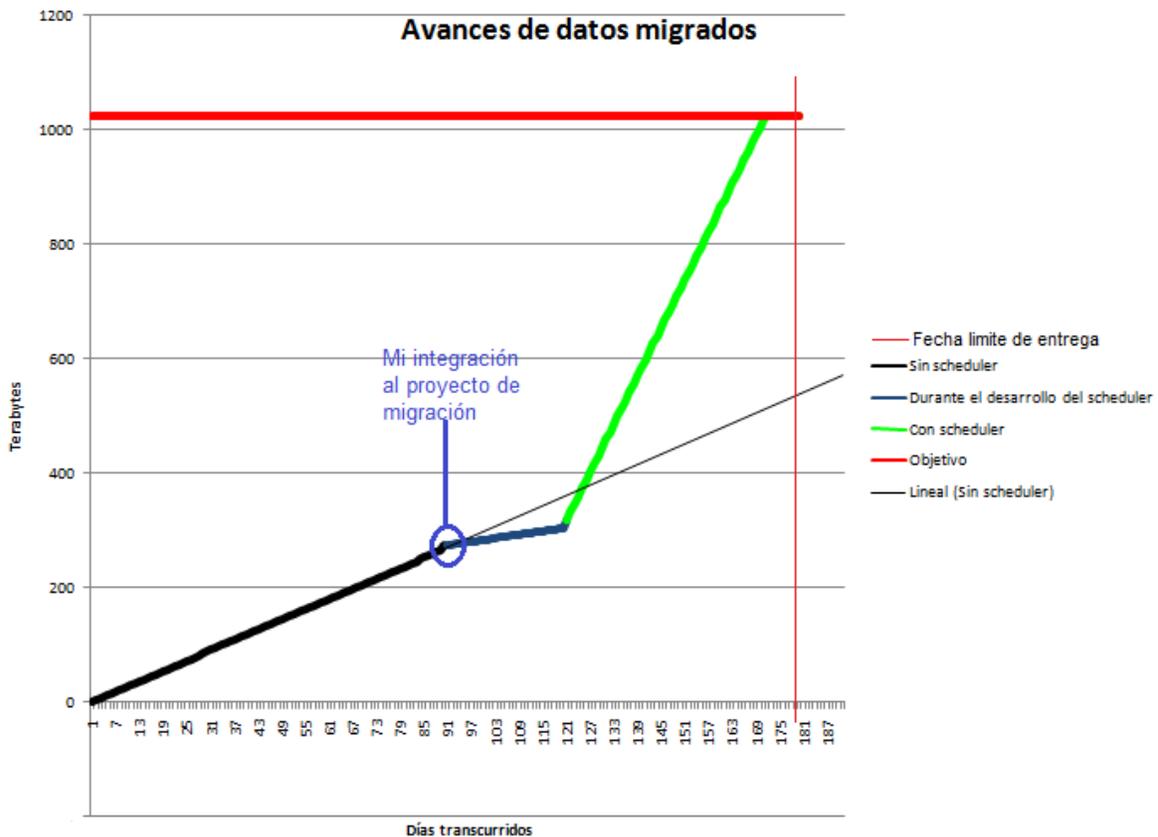
Los reportes fueron mejorados para mostrar la mayor cantidad información posible de cada data set a migrar, de forma que se pudiera llevar el detalle de aquellos que faltasen por migrar, y si alguno no pudo migrarse ya sea por que presentó error su manejo, está dañado o duplicado.

Así mismo se elaboró un reporte con el resumen de los datos que habían sido migrados y los que faltaban por migrar de manera que se pudieran realizar estadísticas y llevar el control del progreso de la migración.

Para generar estos reportes sin necesidad de seguir un largo procedimiento, implemente varios scripts de REXX. Esto facilitó obtener los reportes con sólo ejecutar los scripts cada vez que se requiriera la información.

## 7. RESULTADOS OBTENIDOS

La siguiente gráfica muestra el avance obtenido antes y después de implementar las automatizaciones para la ejecución de jobs de migración:



De haber seguido con el mismo procedimiento que se estaba usando cuando me incorporé al proyecto habríamos llegado a migrar alrededor 550 TB, antes de que sobrepasáramos la fecha límite de entrega y apenas habríamos conseguido migrar la mitad de datos.

Etapa	Tasa de migración de datos
Migración sin scheduler	3 TB/día
Durante el desarrollo del scheduler	1 TB/día
Migración con scheduler	14 TB/día

Durante el desarrollo del scheduler la tasa de migración de datos disminuyó considerablemente debido al tiempo dedicado a la primera actividad. Esto representaba un riesgo de mayor retraso el cuál tuve que asumir basándome en los datos obtenidos de las pruebas individuales con procesos de migración seleccionados de manera aleatoria y también en el hecho de que conocía a detalle la infraestructura y podía predecir algunos comportamientos de esta.

De esta manera, finalmente mantuvimos el cumplimiento de las fechas para la finalización de la migración de datos de Storagetek a IBM.

Se aseguró en todo momento la integridad de la información a migrar. El cliente pudo corroborar este dato en la operación del día a día. Los datos históricos que se pedían en los dispositivos de Storagetek por los procesos productivos, ahora eran leídos desde la solución de IBM sin ningún problema.

Además de cumplir con la migración de datos en tiempo, realicé de manera indirecta una depuración de datos inválidos, corruptos o dañados con lo que hubo una liberación de espacio en beneficio del cliente.

<b>Contenido de datos en Storagetek</b>	<b>Contenido de datos en el IBM TS7700</b>
Espacio de data sets vigentes =1PB	Espacio de data sets vigentes =1PB
Espacio de data sets descatalogados= 500 TB	Espacio de data sets descatalogados= 1.5 TB
Espacio de data sets dañados, corruptos o inválidos = 100 TB	Espacio con data sets dañados, corruptos o inválidos = 0 TB

De esta tabla se concluye que se depuraron 600 TB de datos no válidos durante el paso de Storagetek al IBM TS7700.

## IX. CONCLUSIONES

Durante mi participación de los proyectos de "Outsourcing de Almacenamiento Central" y de "Migración Storagetek VTCS a IBM TS7700", desarrollé y afiancé el conocimiento necesario para poder desempeñarme como especialista IT de almacenamiento en mainframe.

Mis capacidades brindaron tanto al cliente como al proveedor de servicios la percepción de seguridad y control sobre los recursos que administraba.

Implementé automatizaciones que permitieron al cliente obtener reportes y resúmenes de los recursos de almacenamiento de manera pronta y sencilla, además ayudaron a que los requerimientos se atendieran de una forma más rápida cumpliendo con los acuerdos de niveles de servicio. Mis aportaciones también ayudaron a realizar tareas rutinarias o laboriosas con pocos pasos y permitieron monitorear y tener el control de los recursos de una forma eficiente y económica en recursos humanos.

Así mismo, con mis conocimientos técnicos de almacenamiento en z/OS tuve un importante aporte para colaborar en la atención de incidentes de considerable severidad para darles solución de manera satisfactoria.

De esta manera he llegado a obtener el reconocimiento de tanto del cliente como de las empresas para las que he laborado, manteniéndome como un referente del área y un elemento recomendable para proyectos de características similares.

## APÉNDICE A. EJEMPLO DE JCL PARA CREAR UN DATA SET SECUENCIAL.

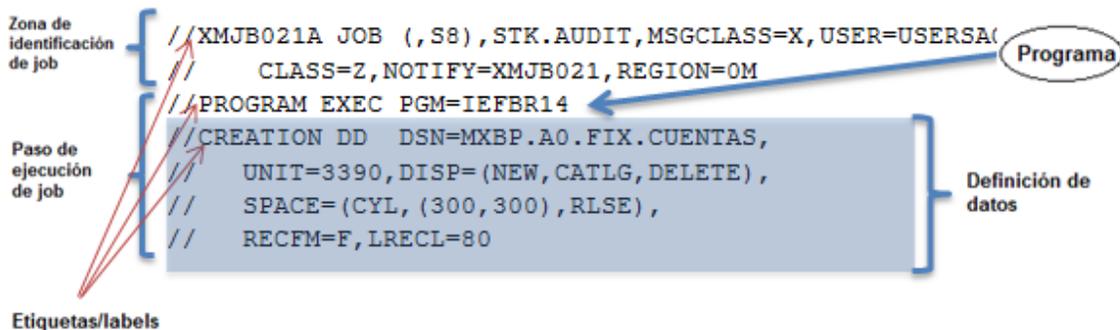
En este ejemplo nos llega una solicitud para crear un data set secuencial:

✓ Imagen de z/OS o Partición Lógica (LPAR) donde se realizaría el trabajo.	ADCD
✓ Nombre del data set apegado a los estándares establecidos por el cliente.	MXBP.A0.FIX.CUENTAS
✓ Longitud de registro, la cual se contabiliza en bytes.	80
✓ Formato del registro, el cual podía ser fijo o variable. Cuando era fijo todos los registros del data set tendrían la misma longitud. Cuando el formato era variable entonces el tamaño de cada registro podía variar no excediendo el valor de la longitud de registro.	Fijo
✓ Si el data set requería un formato que no fuera el <i>básico</i> debía indicarlo el solicitante, en caso contrario yo usaría el formato básico. Los formatos alternativos disponibles son: <i>Extended</i> y <i>Large</i> .	Básico
✓ El tipo de dispositivo donde se almacenaría el data set. Si era en cinta ya no requería más información. Si el data set se debía alojar en disco, el solicitante debía especificarme el tamaño de las extensiones de espacio que ocuparía el data set a crear.	Disco, 300 cilindros de espacio primario y 300 cilindros de espacio secundario.

El JCL para crear el data set queda de la siguiente manera:

```
//XMJB021A JOB (,S8),STK.AUDIT,MSGCLASS=X,USER=USERSAG,
// CLASS=Z,NOTIFY=XMJB021,REGION=0M
//PROGRAM EXEC PGM=IEFBR14
//CREATION DD DSN=MXBP.A0.FIX.CUENTAS,
// UNIT=3390,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(300,300),RLSE)
```

Para describir para qué sirve cada sección del job, muestro la siguiente imagen:



Los *labels* o etiquetas identifican a cada elemento del código de JCL. En este JCL existen tres tipos de funciones diferentes:

`//XMJB021A JOB` es usado para asignarle nombre al job y especificar los parámetros de identificación del job. `USER=USERSAG` es una reasignación de usuario que en representación de nuestro usuario va a realizar el trabajo, tiene mayores privilegios para crear o modificar o eliminar data sets de otras aplicaciones. De esta manera se condiciona a que solamente vía batch se puedan hacer estas actividades.

`//PROGRAM EXEC` es usado para identificar la ejecución de un paso. Siempre debe especificarse un programa o el llamado a un procedimiento desde esta sentencia. Un procedimiento es otro fragmento de código de JCL que se invoca en el momento de ejecución de un job.

`//CREATION DD` Es un identificador de definición de datos. Desde ahí se define los siguiente:

**DSN:** Nombre del data set

**UNIT:** Tipo de dispositivo. Por ejemplo, para disco se usa 3390 y para cinta 3490 o 3590. También es frecuente ver el uso de esotéricos los cuales apuntan a un rango de direcciones de un mismo tipo dispositivo.

**DISP:** Abreviación de *DISPOSITION*. Sirve para hacerle saber al sistema el estado en el que se encontrará el data set en el momento que sea referenciado. El primer campo puede tener los siguientes 4 valores: NEW, OLD, SHR y MOD.

- NEW - El data set va a ser creado en el momento que sea referenciado. Durante la ejecución del paso únicamente este proceso podrá hacer uso de este.

- OLD - El data set ya deberá existir en el momento de ser referenciado, así mismo será usado de forma exclusiva durante la ejecución del paso.
- SHR - El data set ya deberá existir en el momento de ser referenciado, así mismo podrá ser compartido por otros procesos durante la ejecución del paso.
- MOD - El data set ya deberá existir en el momento de ser referenciado, así mismo será usado de forma exclusiva durante la ejecución del paso. El apuntador de operación de E/S va a ser colocado después del último registro del data set en vez del primer registro como en las otras disposiciones.

El segundo campo en la disposición se refiere a las acciones a realizar en caso de terminar el paso sin error. El tercer campo en las disposición se refiere a las acciones a realizar en caso de terminar el paso con error. En el ejemplo *DISP=(NEW,CATLG,DELETE)* el sistema va a verificar que el data set no exista antes de generarlo y si el paso termina bien entonces lo registrará en un catálogo de usuario y en caso de finalizar con error entonces lo borrará.

**SPACE:** Cantidad de espacio a reservar en el dispositivo. Aplica únicamente a dispositivos DASD, se puede expresar en términos de tracks, cilindros o bloques. Se expresa en tamaño de extents. El parámetro RLSE es la abreviatura de *release* y significa que el sistema va a liberar el espacio no usado en el momento que finalice el paso.

**RECFM:** Formato de registro, que este caso es fijo "F".

**LRECL:** Longitud de registro.

El programa IEFBR14 no realiza ninguna función y no requiere especificación de entradas o salidas. Se usa para cumplir el prerequisite de especificar un programa en el paso de ejecución.

Para una descripción detallada de todos los parámetros y sentencias que puede tener un JCL favor de remitirse al manual *z/OS MVS JCL Reference* referido en la bibliografía.

## APÉNDICE B. EJEMPLO DE JCL PARA CREAR UN DATA SET PARTICIONADO

Este es un ejemplo de solicitud para crear un data set particionado:

✓ LPAR donde se realizaría el trabajo.	ADCD
✓ Nombre del data set (el cliente tenía estándares específicos para el nombrado de data sets particionados)	MXBP.A0.PO.JCLLIB
✓ Formato de registro.	Fijo
✓ Longitud de registro.	80
✓ Formato del data set. En caso de ser estándar el solicitante debía especificar el número de bloques de directorio.	Estándar con 40 bloques de directorio.
✓ Especificaciones de consumo de espacio	300 cilindros de espacio primario y 300 cilindros de espacio secundario.

```
//XMJB021A JOB (,S8),STK.AUDIT,MSGCLASS=X,USER=USERSAG,
// CLASS=Z,NOTIFY=XMJB021,REGION=0M
//PROGRAM EXEC PGM=IEFBR14
//CREATION DD DSN=MXBP.A0.MON.JCLLIB,
// UNIT=3390,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(300,300,40),RLSE),
// RECFM=F,LRECL=80,DSORG=PO
```

**DSORG:** (*Dataset organization*). Donde "PO" significa *partitioned organization*.

Los bloques de directorio son especificados en el parámetro SPACE en un tercer campo de la especificación de tamaño de extents.

## APÉNDICE C. EJEMPLO DE JCL PARA CREAR BASE DE GDG

La siguiente es una tabla con los datos de la solicitud para crear una base de GDG:

✓ LPAR donde se realizaría el trabajo.	ADCD
✓ Nombre de la base del GDG (Los nombres de las generaciones las crearía el sistema tomando como referencia el nombre de la base)	MXBP.A0.GDG.REPORTS
✓ Número máximo de generaciones.	5
✓ Si se requería que se generara la primera versión se continuaba con los siguientes puntos.	Sin primera versión
✓ Tipo de dispositivo: DASD o cinta.	No aplica

Para generar una base de GDG se debe usar el programa IDCAMS en el JCL:

```
//XMJB021A JOB (,S8),STK.AUDIT,MSGCLASS=X,USER=USERSAG,
// CLASS=Z,NOTIFY=XMJB021,REGION=0M
//PROGRAM EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEF GDG(NAME(MXBP.A0.GDG.REPORTS) LIMIT(5) SCR NEMP)
```

Con este JCL le especificamos al sistema que queremos generar un GDG con límite de 5 generaciones.

**SCR:** (*Scratch*). Significa que el sistema eliminara cada generación que quede fuera del rol del GDG.

**NEMP:** (*noempty*). Significa que una vez que se alcance el límite de generaciones, la base no ponga fuera de rol todas las generaciones sino únicamente la generación más vieja.

**//SYSPRINT.** Definición de datos que usa IDCAMS como salida para los mensajes que el programa genere. Es frecuente especificar SYSOUT para mandar los mensajes a la salida del sistema pero también es posible direccionar esta salida a un data set.

**//SYSIN.** Definición que sirve para especificar la entrada de comandos y parámetros al programa.

La descripción de cada uno de los comandos, parámetros, sintaxis reconocidos en IDCAMS vienen detallados en el manual *z/OS DFSMS Access Method for Catalogs* referido en la bibliografía.

## APÉNDICE D. EJEMPLO DE JCL PARA CREAR UN ARCHIVO VSAM

El siguiente es un ejemplo para crear un data set de tipo VSAM

✓ LPAR donde se realizaría el trabajo.	ADCD
✓ Nombre del data set.	MXBP.A0.VSAM.CUENTAS
✓ Atributos lógicos específicos del VSAM.	Especificados en el paso del JCL
✓ Parámetros de consumo de espacio.	100 tracks de espacio primario y 50 tracks de espacio secundario

Para crear un archivo VSAM también se usa el programa IDCAMS.

```
//XMJB021A JOB (,S8),STK.AUDIT,MSGCLASS=X,USER=USERSAG,
// CLASS=Z,NOTIFY=XMJB021,REGION=0M
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER (NAME(MXBP.A0.VSAM.CUENTAS) -
        INDEXED -
        RECSZ(80 80) -
        TRACKS(100,50) -
        KEYS(5 0) -
        CISZ(4096) -
        FREESPACE(3 3) )
        DATA (NAME(MXBP.A0.VSAM.CUENTAS.DATA) ) -
        INDEX (NAME(MXBP.A0.VSAM.CUENTAS.INDEX) )
/*
```

El sistema creará un VSAM de tipo KSDS (*key sequenced data set*) el cual es referido como un clúster que se conforma de dos componentes:

**INDEX:** Contiene la lista de los valores de las llaves para los registros en el clúster con apuntadores a los registros correspondientes en el componente DATA.

**DATA:** Contiene el dato actual. Cada registro en el componente de DATA contiene un campo de llave con el mismo número de caracteres y ocurre en la misma posición relativa de cada registro.











```

VOLUME  UNIT      SYSTEM= 1 2 3                                STORGRP NAME
USR001  0AB0              + + +                                USER
USR002  0AB1              + + +                                USER
USR003  0AB2              + + +                                USER
USR004  0AB3              + + +                                USER
USR005  0AB4              + + +                                USER
***** LEGEND *****
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
> THE VOLSER IN UCB IS DIFFERENT FROM THE VOLSER IN CONFIGURATION

```

11. Se determina que de acuerdo al estándar el siguiente volumen debe llamarse USR006.

12. La elección de la unidad dependerá del modelo del disco. Tomando en cuenta que sólo se requieren 2GB de espacio, entonces se elige una dirección de disco modelo 3 ya que su capacidad ronda los 2.83 GB y es el modelo de disco más pequeño .

13. Se determina que la unidad A0BF esta acoplada a un disco modelo 3. Por lo que se debe preparar el volumen para ser asignado en el storage group. Esto implica darle formato para que el sistema operativo reconozca y pueda trabajar con el volumen. La unidad en este momento no puede ponerse en línea por esta causa. La evidencia de esto la muestro después de dar el comando de sistema /V A0BF,ONLINE

```

RESPONSE=ADCD   IEF503I UNIT 0ABF I/O ERROR

```

14. Para dar formato al volumen se emplea un JCL con el programa ICKDSF

```

//INITDASD JOB ('000,LRD'),'LERD',MSGCLASS=X,MSGLEVEL=(1,1),
//          CLASS=A,NOTIFY=&SYSUID
//INIT      EXEC PGM=ICKDSF,PARM='NOREPLYU'
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
          INIT NOVALIDATE VERIFY(*NONE*) VOLID(USR006) -
          UNIT(0ABF) VTOC(1,0,75) INDEX(6,0,15) SG

```

Con este JCL al sistema se le va a indicar que formateé el volumen en la unidad 0ABF asignándole la etiqueta USR006 y verificando que no tenga etiqueta anterior. Así mismo vienen especificados los parámetros para ubicar el VTOC en el disco y la especificación de que se requiere que el SMS lo reconozca para ser asignado a un storage group.

15. Una vez que ya está formateado el volumen se verifica que el sistema operativo lo detecte poniendo la unidad en línea.







Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY  
> THE VOLSER IN UCB IS DIFFERENT FROM THE VOLSER IN CONFIGURATION  
SYSTEM 1 = ADCD            SYSTEM 2 = DUZA            SYSTEM 3 = ROCCAPL

0020

. . . . .

## APÉNDICE F. EJEMPLO DE MODIFICACIÓN DE RETENCIÓN DE DATA SETS.

✓ LPAR donde se realizaría la modificación.	ADCD
✓ Máscara o lista de data sets a modificar.	El único data set para modificar su retención es CLBP.A0.FIX.REPORTE
✓ Especificaciones de la nueva retención: Duración en disco, tiempo en cinta, tiempo que debía pasar antes de ser eliminado.	El data set deberá ser eliminado después de un mes de no usarse. Actualmente tiene retención de 7 días antes de ser eliminado.
Periodo en que la retención se mantendría vigente hasta que la retención debiera volver a los parámetros originales.	Únicamente para este nombre de data set y por el periodo que dure la nueva retención.

La management class contiene la información sobre la retención del archivo que el HSM utilizará para migrarlo o eliminarlo llegado el momento. Sin management class asociada, el HSM no puede determinar si el data set debe ser manipulado por lo que lo ignora. Si el HSM no se encontrara activo tampoco habría migración ni eliminación de automática de archivos con cualquier management class asociada.

Para saber cuál es la management class asociada al data set CLBP.A0.FIX.REPORTE se ejecuta el siguiente comando:

```
LISTCAT ENT(CLBP.A0.FIX.REPORTE) ALL
```

La respuesta obtenida fue la siguiente:

```
NONVSAM ----- CLBP.A0.FIX.REPORTE
      IN-CAT --- CATALOG.SYSPBBV.CHLPPGR12
      HISTORY
DATASET-OWNER----- (NULL)      CREATION-----2016.124
      RELEASE-----2          EXPIRATION-----0000.000
      ACCOUNT-INFO----- (NULL)
      SMSDATA
      STORAGECLASS ---SCPRDGRL      MANAGEMENTCLASS-MCFIXGRL
      DATACLASS  -----DCFIXGRL      LBACKUP ---0000.000.0000
      VOLUMES
      VOLSER-----XM3463          DEVTYPE-----X'3010200F'      FSEQN-----
-----0
      ASSOCIATIONS----- (NULL)
      ATTRIBUTES
***
```



MCBKUPDB	NOLIMIT	NOLIMIT	NOLIMIT	NO	9999
MCBRSDIA	NOLIMIT	3	0	YES_IMMED	2
MCBRSEM	NOLIMIT	6	0	YES_IMMED	2
MCBRSVFY	NOLIMIT	4	0	YES_IMMED	2

Del listado se localiza la management class que cumpla con las especificaciones del solicitante:

```

Panel List Utilities Scroll Help
-----
                                MANAGEMENT CLASS LIST
Command ==>                                Scroll ==> HALF
                                           Entries 1-1 of 1
                                           Data Columns 3-7 of 40

CDS Name : ACTIVE

Enter Line Operators below:

      LINE      MGMTCLAS EXPIRE      EXPIRE      RET      PARTIAL      PRIMARY
      OPERATOR  NAME      NON-USAGE  DATE/DAYS  LIMIT  RELEASE  DAYS
---- (1) ----  -- (2) ---  --- (3) ---  --- (4) ----  -- (5) --  ---- (6) ----  --- (7) --
      MCMES01          31      NOLIMIT  NOLIMIT  YES          4
-----  -----  -----  BOTTOM OF DATA  -----  -----  -----

```

Para modificar la management class de MCFIXGRL a MCMES01 se usa el comando ALTER de IDCAMS en el siguiente JCL:

```

//XMJB021A JOB (,S8),STK.AUDIT,MSGCLASS=X,USER=USERSAG,
// CLASS=Z,NOTIFY=XMJB021,REGION=0M
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        ALTER CLBP.A0.FIX.REPORTE -
        MGMTCLAS (MCMES01)
/*

```

Se envía a ejecución el job y cuando finalice, el data set CLBP.A0.FIX.REPORTE asimilará la nueva política que respetará el HSM durante su gestión de espacio.

## APÉNDICE G. EJEMPLO DE DEFINICIÓN DE RECURSOS DE ALMACENAMIENTO PARA UNA APLICACIÓN

Nos llega la siguiente solicitud para definir una nueva aplicación:

✓ LPAR donde se realizarían las definiciones.	BANK
✓ Máscaras a utilizar por la aplicación	BANK.FIN.TMP.** BANK.FIN.PERM.**
✓ Espacio que se espera consuma la aplicación.	30 GB en disco. No se van a generar datos directamente en cinta.
✓ Especificaciones especiales sobre las definiciones (storage group, políticas, etc.)	10 GB a storage group SGFINTMP y 20 GB al storage group SGFINPRM. De acuerdo a las políticas del cliente los archivos TMP deben durar un día y al día siguiente ser eliminados. Los archivos PERM deben permanecer en disco 2 días y después ser migrados al nivel 2 con una retención permanente. Los archivos PERM deben contar siempre con un respaldo.

Para atender el requerimiento para esta nueva aplicación se debe realizar lo siguiente:

- Definir los storage groups SGFINTMP y SGFINPRM que albergarán los datos de la aplicación FIN.
- Asignar el espacio (discos) necesario para cubrir el requerimiento de espacio por storage group. Ver "*Apéndice E. Ejemplo de asignación de espacio para una aplicación.*"
- Definir las storage class y las management class que el sistema utilizará para elegir los dispositivos y administrar las retenciones y respaldos de los data sets de la aplicación.
- Definir un catálogo de usuario con su respectivo alias destinado a guardar todas las entradas generadas por la aplicación.
- Definir las reglas de asignación de constructs para máscaras de data sets en las rutinas ACS del SMS.













Command ==>

To Perform ACS Translation, Specify:

```
SCDS Name . . . . . 'SYS1.SCDS'
                                     (1 to 44 Character Data Set Name)

ACS Source Data Set . . 'SYS1.SMS.CNTL'
                                     (1 to 44 Character Data Set Name)

ACS Source Member . . . STORCLAS (1 to 8 characters)

Listing Data Set . . . IBMUSER.PRINT
                                     (1 to 44 Character Data Set Name)
```

Use ENTER to Perform ACS Translation;

Lo mismo se debe hacer para cargar la management class y el storage group.

Para activar la regla se ingresa a la opción 8 del menú principal del ISMF, y se especifica la CDS Name que tiene cargadas nuestras modificaciones para ser activadas en el sistema.

```
. . . . .

Panel Utilities Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                                     CDS APPLICATION SELECTION
```

Command ==>

To Perform Control Data Set Operations, Specify:

```
CDS Name . . SYS1.SCDS
                                     (1 to 44 Character Data Set Name or 'Active')
```

Select one of the following Options:

- 5 1. Display - Display the Base Configuration
- 2. Define - Define the Base Configuration
- 3. Alter - Alter the Base Configuration
- 4. Validate - Validate the SCDS
- 5. Activate - Activate the CDS
- 6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
- 7. Cache Update - Define/Alter/Delete CF Cache Sets
- 8. Lock Display - Display CF Lock Structure Names for all CF Lock Sets
- 9. Lock Update - Define/Alter/Delete CF Lock Sets

If CACHE Display is chosen, Enter CF Cache Set Name . . \*

If LOCK Display is chosen, Enter CF Lock Set Name . . . \*

(1 to 8 character CF cache set name or \* for all)

Use ENTER to Perform Selection;

## X. BIBLIOGRAFÍA

Ebbers,M.,Kettner, J., (2011), *Introduction to the new mainframe: z/OS Basics*, 3a Edición, Redbooks. International Technical Support Organization, U.S.A. ,Publicación No. SG24-6366-02, ISBN 0738435341

Rogers, P. (2011),*ABCs of z/OS System Programming: Volumen 1*, 3a Edición, Redbooks. International Technical Support Organization, U.S.A, Publicación No. SG24-6981-02, ISBN 0738435791

Rogers, P. (2011),*ABCs of z/OS System Programming: Volumen 3*, 3a Edición, Redbooks. International Technical Support Organization, Publicacion No. SG24-6983-03, ISBN 0738434094

*z/OS V1R10.0 DFSMS Introduction*, (2008), 4a Edición, International Business Machines Corporation, United States of America, Publicación No. SC26-7397-03

*z/OS Managing Catalogs*,(2008),7a Edición, International Business Machines Corporation, United States of America, Publicación No. SC26-7409-06.

*z/OS DFSMS Access Method for Catalogs*, 9a Edición, International Business Machines Corporation, United States of America, Publicación No. SC26-7394-08

*z/OS DFSMSHsm Storage Administration Guide*,(2009),10a Edición, International Business Machines Corporation, United States of America, Publicación No. SC35-0421-09.

*z/OS DFSMSdss Storage Administration Guide*,(2009),11a Edición, International Business Machines Corporation, United States of America, Publicación No. SC35-0423-11.

*z/OS DFSMSrmm Guide and Reference*,(2008),9a Edición, International Business Machines Corporation, United States of America, Publicación No. SC26-7404-08.

*TSO/E Users Guide*, (2008) 5a Edición, International Business Machines Corporation, United States of America, Publicación No. SA22-7794-04.

*z/OS MVS JCL Reference*, (2008), 13a Edición, International Business Machines Corporation, United States of America, Publicación No. SA22-7597-12

*TSO/E REXX Reference*, (2011),1a Edición, International Business Machines Corporation, United States of America, Publicación No. SA22-7790-10.

*Virtual Tape Control System Software, Administrator's Guide*, (2010) SUN microsystems- StorageTek, United States of America.